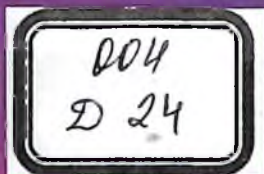


**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA  
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**



**DASTURLASHI III  
(O'quv qo'llanma)**

**FL**

**ADOBE® FLASH® PROFESSIONAL CS5**

Initializing Tools...

© 2007 Adobe Systems Incorporated. All Rights Reserved.  
The Adobe Flash Player is a trademark of Adobe Systems Incorporated.



**TOSHKENT - 2022**

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA  
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

## **DASTURLASH III**

(O'quv qo'llanma)

**TOSHKENT - 2022**

Ushbu o'quv qo'llanmadan foydalanuvchi Macromedia Flash muhitida ishlay olish; animatsiyalarni yarata olish; web-sahifalar uchun videoroliklar tayyorlay olish; interfaol va ovozli lavhalarni yarata olish; actionscript dasturlash tili orqali dasturlashni bilish; boshqariladigan animatsiyalarni yaratish; ilovalarni ishlatish bo'yicha ko'rsatmalar tayyorlash kabi bilimlarga ega bo'ladi. O'quv qo'llanma 5350200-Televizion texnologiyalar (Audiovizual texnologiyalar, telestudiya tizimlari va ilovalari) yo'nalishi bo'yicha ta'lim olayotgan talabalar uchun mo'ljallangan.

#### **Tuzuvchilar:**

O.N.Mirzayev - PhD, Raqamli texnologiyalar va sun'iy intellektni rivojlantirish ilmiy-tadqiqot instituti doktoranti, TATU dotsenti  
A.X.Aliqulov - TATU «Informatika asoslari» kafedrasida assistenti  
G.R.Mirzayeva - TATU «Informatika asoslari» kafedrasida assistenti

#### **Taqrizchilar:**

RTSIR ITI katta ilmiy xodimi t.f.n. S. S. Radjabov.  
TATU "Sun'iy intellekt" kafedrasida katta o'qituvchisi PhD. S. Mahmudjanov.

Mazkur o'quv qo'llanma Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universitetining Ilmiy-uslubiy kengashida tasdiqlangan (2022 yil \_\_\_\_\_dagi yig'ilishning \_\_\_\_\_sonli bayonnomasi).

# MUNDARIJA

<b>KIRISH</b> .....	6
<b>I-bob. FLASH MX MUHITIDA ISHLASH</b>	
1.1. Flash texnologiyasiga kirish. Flash dasturi bilan ishlash asoslari.....	7
Flash-texnologiyaning imkoniyatlari.....	7
Web-sahifalarni yaratishda rastri va vektorli grafikalarining o'rimi.....	9
Flash dasturi bilan ishlash asoslari.....	9
Flash dasturining asosiy oynasi ko'rinishi.....	10
1.2. Flashning uskunalar paneli. Flashning ish maydoni.....	11
Flash MX 8 interfeysi.....	11
Flash dasturining uskunalar paneli.....	14
1.3. Foydalanuvchi ishini tashkil qilish vositalari. Asboblar panelini o'zgartirish. Tezkor tugmalar birikmalarini sozlash.....	34
Asboblar panelini o'zgartirish.....	34
Yordamchi panellar.....	35
Tezkor tugmalar birikmalarini sozlash.....	42
1.4. Macromediya flashda animatsiya yaratish. Interfaollik.....	43
Animatsiya tushunchasi va turlari.....	44
Vaqt diagrammasi – Timeline.....	44
Qatlamlar bilan ishlash.....	48
Flash dasturida kadrma-kadr animatsiya .....	49
Hisoblangan animatsiya. Motion Tweening animatsiyasi.....	51
Traektoriya bo'ylab animatsiya. Yo'naltiruvchi qatlamlar.....	54
<b>II-bob. ACTIONSCRIPT DASTURLASH TILIDA ISHLASH</b>	
2.1. ActionScript dasturlash tilining imkoniyatlari va harakatlar paneli bilan ishlash haqida asosiy ma'lumotlar.....	56
ActionScript haqida umumiy ma'lumot.....	56
ActionScript terminologiyasi.....	57
ActionScript ob'ekt modeli.....	58
Skriptni bajarish konteksti.....	60
ActionScript paneli.....	62
Normal Mode va Expert Mode rejimi.....	65
2.2. ActionScriptda o'zgaruvchilar. O'zgaruvchilarni e'lon qilish va ularga qiymatlarni belgilash.....	72
ActionScript elementlari (sintaksis).....	72
ActionScript 3.0 - ob'ektga yo'naltirilgan dasturlash tili.....	76
ActionScript o'zgaruvchilari. Umumiy ma'lumot.....	77
ActionScript o'zgaruvchisini yaratish.....	78
ActionScriptdagi ma'lumotlar turlari haqida .....	78
Ma'lumotlar turini e'lon qilish.....	79
2.3. Trace() funksiyasi. Oddiy arifmetik operatorlar. Satrlar. Satrlarda bajariladigan amallar. Typeof() funksiyasi.....	81

ActionScript xatolarini tekshirish.....	81
trace() funksiyasi.....	82
Oddiy arifmetik operatorlar.....	84
Satrlar. Satrlarda bajariladigan amallar.....	86
typeOf() funksiyasi.....	91
<b>2.4. Flash dasturida matematik hisob-kitoblarni amalga oshirish.</b>	
<b>Matematik ob'ektning metodlari va xususiyatlari. ParseFloat funksiyasi...</b>	<b>93</b>
Number (Sonlar uchun) turi. ActionScriptdagi sonlar turlari.....	94
Sonlarni ifodalash xususiyatlari.....	94
Satrlarni sonlarga aylantirish. parseInt ( ) va parseFloat ( ) funksiyalari.....	97
Matematik funksiyalarni amalga oshirish xususiyatlari.....	99
Matematik funksiyalar va konstantalar.....	100
<b>2.5. Munosabat va tarmoqlanuvchi operatorlar.....</b>	<b>106</b>
Munosabat operatorlari.....	106
Tarmoqlanuvchi operatorlar. if shart operatori.....	106
else operatori va else if operatorlarining kombinatsiyasidan foydalanish.....	108
Mantiqiy operatorlar VA (&&), YOKI (  ).....	109
SWITCH-CASE tanlash operatori.....	110
<b>2.6. Takrorlash (takrorlanish) operatorlari. for, while va do while</b>	
<b>operatorlari.....</b>	<b>112</b>
Takrorlanishlarni qanday va nima uchun ishlatish kerak.....	112
Takrorlanish turlari.....	113
Takrorlanish shartlarini o'rnatish.....	115
Ichki o'rnatilgan takrorlanishlarni yaratish.....	117
Takrorlanishlarni to'xtatish.....	118
<b>2.7. Tugma hodisalarini boshqarish. O'tish operatori va kadr yorlig'i.</b>	
<b>stop() va play() funksiyalari.....</b>	<b>119</b>
Hodisalar bilan ishlovchilar. To'g'ri ishlov beruvchini tanlash.....	119
Sichqoncha va klaviatura hodisalari.....	120
Klip hodisasini qo'llash.....	124
Hodisalarni qayta ishlash usullari haqida.....	126
stop() va play() funksiyalari. O'tish operatori va kadr yorlig'i.....	129
<b>2.8. Rolik va videoroliklarning xususiyatlari. Matn maydonlarini yaratish.</b>	
<b>Roliklarda shartli if, if...else operatorlaridan foydalanish.....</b>	<b>132</b>
Rolik va videoroliklarning xususiyatlari.....	133
Kliplarni ko'paytirish, bo'sh kliplar yaratish.....	135
Matn maydonlari. Matn maydonlarini yaratish.....	138
Matn maydonlarini dasturiy yaratish.....	142
Roliklarda shartli if, if...else operatorlaridan foydalanish.....	143
<b>2.9. Dinamik maydonlarning asosiy xususiyatlari. Aylantirish funksiyasi...</b>	<b>149</b>
dynamic text Flash. Flash CS5da dinamik matn yaratish.....	149
Dinamik ActionScript matnini yaratish (dynamic text).....	152
Actionscript 3.0 yordamida dinamik matn yaratish.....	152
Flash CS5 va CS6da kiritish matn maydonini yaratish.....	153

TextFieldType.INPUT, TextFieldType.DYNAMIC TextField AS3 sinfining nusxalari.....	155
textInput AS3, change, link, scroll - TextField hodisalari.....	156
Matn hodisalari bilan ishlash.....	157
2.10. Klipli-keyslarga ishlov berish haqida umumiy ma'lumot.....	161
Dinamik kliplar nusxalarini yaratish.....	161
Film - belgisini bog'lash jarayoni.....	164
Bo'sh film belgisining yangi nusxasini dinamik ravishda yaratish.....	166
ActionScript bilan dinamik chizish.....	167
2.11. getTimer() funksiyasi. Key obyektining asosiy usullari.....	169
FLASHda vaqtni orqaga hisoblashdan foydalanish.....	169
Joriy vaqt va sanani aniqlash.....	173
Vaqtni ortga hisoblash. getTimer() funksiyasi.....	171
Klaviatura bilan ishlash. Key obyektning asosiy usullari.....	173
<b>GLOSSARIY.....</b>	<b>180</b>
<b>ADABIYOTLAR RO'YXATI.....</b>	<b>180</b>

## KIRISH

Mazkur o'quv qo'llanmadan foydalangan har bir foydalanuvchi Flash MX dasturi muhitida dasturlash, foydalanuvchi muhitini yaratish va undan foydalanish hamda kichik loyihalar yaratish imkoniyatlari to'g'risida ko'nikmalar hosil qiladi. Bu qo'llanma Flash MX dasturlash tili yordamida turli xil ilovalarni yaratishni to'liq qamrab olgan.

Ushbu o'quv qo'llanmadan foydalanuvchi Macromedia Flash muhitida ishlay olish; animatsiyalarni yarata olish; videorolik va web-sahifalarni yarata olish; interfaol va ovozli lavhalarni yarata olish; actionscript dasturlash tili orqali dasturlashni bilish; boshqariladigan animatsiyalarni yaratish; videorolik, web – sahifalarni yaratishni; ilovalarni ishlatish bo'yicha ko'rsatmalar tayyorlash kabi bilimlarga ega bo'ladi.

O'quv qo'llanma 5350200-Televizion texnologiyalar (Audiovizual texnologiyalari, Telestudiya tizimlari va ilovalari) ta'lim yo'nalishida tahsil olayotgan talabalar hamda mazkur sohaga aloqador professor-o'qituvchilar va mustaqil o'rganuvchilar uchun mo'ljallangan.

## INTRODUCTION

Each user who uses this tutorial will develop skills in programming, creating and using a user environment, and creating small projects in a Flash MX environment. This tutorial covers all aspects of creating various applications using the Flash MX programming language.

Be able to work in Macromedia Flash using this tutorial; be able to create animations; be able to create videos and web pages; be able to create interactive and soundtracks; knowledge of programming through actionscript programming language; create managed animations; creating videos, web pages; knowledge of how to use applications.

The textbook is intended for students studying in the field of education 5350200-Television technologies (Audiovisual technologies, TV studio systems and applications), as well as professors and independent researchers in this field.

## ВВЕДЕНИЕ

Каждый пользователь, использующий данное учебное пособие, приобретает навыки программирования в среде Flash MX, создания и использования пользовательской среды, а также возможности создания небольших проектов. Это учебное пособие полностью охватывает создание различных приложений с использованием языка программирования Flash MX.

С помощью этого учебного пособия пользователь сможет работать в среде Macromedia Flash; создавать анимацию; создавать видеоролики и веб-страницы; создавать интерактивные и голосовые кадры; знать программирование с помощью языка программирования actionscript; создавать управляемые анимации; создавать видеоролики, веб – страницы; готовить инструкции по использованию приложений.

Учебник предназначен для студентов, обучающихся по направлению образования 5350200-Телевизионные технологии (Аудиовизуальные технологии, телестудийные системы и приложения), а также преподавателей и независимых исследователей в данной области.

### 1.1. Flash texnologiyasiga kirish. Flash dasturi bilan ishlash asoslari

#### Reja:

1. Flash-texnologiyasining imkoniyatlari.
2. Web-sahifalarni yaratishda rastrlı va vektorlı grafikalarning o'rni.
3. Flash dasturi bilan ishlash asoslari.
4. Flash dasturining asosiy oynasi ko'rinishi.
5. Nazorat savollari.

**Kalit so'zlar:** *Flash, web, Macromedia Flash, ActionScript, SWF, FLA, tweened-animation, motion animation, rolik, kadr, rastrlı, vektorlı, grafika.*

#### 1. Flash-texnologiyasining imkoniyatlari.

**Flash** texnologiyasiga - **ShockWave Flash (SWF)** formatlı vektorlı grafikadan foydalanishga asoslangan texnologiyadir. Bu format eng samaralı grafik formatlardan bo'lmasada, **SWF** formatı foydalanuvchılarga grafik imkoniyatlari cheklanmagan grafiklar bilan ishlovchi vositalar va natijani Web - brauzerlarda, keraklı muxarrirlarda foydalanish imkoniyatlarini beradi. **Flash** texnologiyasining imkoniyatlaridan yana biri - bu uning moslashuvchanlıgidır, ya'ni bu format barcha platformalarda (MacOS tizimli Macintosh kompyuterlari yoki Windows tizimli kompyuterlarıda) ishlatilishi mumkin. Yana bir qulay imkoniyati uning yordamıda yaratilgan tasvirlar nafaqat animatsiyalı bo'lishi, balki interfaol elementlar va tovush bilan boyitilishi hamda dasturlash orqalı boshqarilishi mumkin.

**Flash** texnologiyasining moslashuvchanlik va interfaol multimedıya dasturlar yaratish imkoniyati ko'pchilik Web-dizaynerlar o'rtasidagi bahslarga sabab bo'lib, uni mashhurligini oshishiga imkoniyat berdi. Shuning uchun bu texnologıyaning yaratilishi bilan bir vaqtda **Macromedia** kompaniyasi tomonidan ikki asosiy web-brauzerlari, **Internet Explorer** va **Netscape Communicator**lar uchun elementlar **Plug-In** yaratildi. Bu esa, o'z navbatıda Flash texnologiyasini Internetda yana ham keng tarqalishiga olib keldi. Natijada ushbu web-brauzerlar yaratuvchılari **swf** formatını o'z dasturlarını asosiy formatlar bazasiga qabul qildi. Bunday usulni boshqa yirik dasturiy ta'minot yaratuvchılar (masalan, Adobe firmasi) ham qo'llay boshladı. **Macromedia** kompaniyasi **swf** formatını juda oddiy va qulay uskunalar bilan ta'minlaganligi bu formatdan ko'p talabgorlarning foydalanishiga olib keldi. Shuni aytish kerakki, hozirgi vaqtda ushbu uskunalarını bir qancha to'liq to'plamlari ham mavjud. Ushbu uskunalarını bir turi **Macromedia Director Shockwave Studio** - multimedıya taqdimotlarını yaratish, **Macromedia FreeHand** va **Macromedia Fireworks** - grafik tasvirlar muharriri, **Macromedia Authorware** va **Macromedia CourseBuilder** - interfaol o'rgatuvchi kurslarını yaratish muharriri va boshqalarını misol qilib olish mumkin. Web-sahifa yaratuvchılar orasıda eng ko'p ishlatiladigani bu **Macromedia Flash** dasturidir, chunki ushbu dastur ixtiyoriy Web-sahifaga mashhurlik olib keluvchi banner va animatsiya, interfaol lavhalar yaratish imkonini beradi. Balki shuning uchun **swf** formatını oddiy qilib Flash deb atalishi odat tusiga kirib qolgan.

Flash texnologiyalar tarkibining elementlari:



- vektorli grafika;
- animatsiyani bir qancha usullarda ishlashi;
- interfeysda interfaol elementlarini yaratish;
- sinxron ovoz qo`shish;
- HTML formati va boshqa internetda foydalaniladigan barcha formatlarga o`tkazishni ta'minlash;
- mustaqil platformali;
- Flash-roliklarni avto rejimda ham, Web - brauzer yordamida ham ko`rish imkoniyati mavjud;
- vizual uskunlari mavjudligi Flash-rolik yaratuvchilarini ko`plab murakkab amallardan xalos etadi, shuningdek Flash-texnologiyalarning texnik aspektlarini o`rganishini talab etmaydi.

Hozirgi vaqtda veb-sayt ishlab chiquvchilari Flash texnologiyasidan tobora ko'proq foydalanmoqda. Flash texnologiyasi navigatsiya elementlarini, animatsion logotiplarni, to'liq ovozli multfilmlarni va hatto turli interaktiv elementlarga ega butun saytlarni yaratishga imkon beradi. Flash ishlatadigan vektor grafikasi tufayli animatsiyalar kichik hajmga ega va shuning uchun tez yuklanadi va brauzer oynasi o'lchamiga o'zgartiriladi.

Flash texnologiyasi veb-dizaynerlarning grafiklarni tayyorlash vositalariga bo'lgan talablariga to'liq javob beradi, vektorli grafikalar va animatsiyalarni yaratish dasturini taqdim etadi - Macromedia Flash. Ushbu dastur tomonidan ishlab chiqarilgan animatsiyalar filmlar deb ataladi. Bundan tashqari, dasturning animatsiya imkoniyatlari faqat animatsiya bilan cheklanmaydi - siz hamma narsani, shu jumladan navigatsiya va menyu elementlarini ham jonlantirishingiz mumkin.

Flash faqat Internet uchun animatsiya yaratish bilan cheklanmaydi. Shuningdek, filmlarni xotira qurilmalari yoki elektron pochta orqali tarqatish uchun mustaqil FlashPlayerni yaratishingiz va foydalanishingiz mumkin. Filmlarni QuickTime (MOV) yoki Windows AVI kabi boshqa formatlarga eksport qilishingiz mumkin. Lekin Macromedia Flash dasturining asosiy maqsadi interaktiv veb-saytlarni ishlab chiqishdir.

Ehtimol, Flash-texnologiyasining asosiy afzalligi deb atash ham mumkin va bu dasturni odiy grafik muharriridan animatsiya yaratish uchun kuchli vositaga aylantiradi - bu ActionScript tili bo'lib, uning yordamida filmlarning barcha interaktiv elementlari ishlaydi. Flash MX 2004 ning zamonaviy versiyasida ActionScript skript tili rivojlangan dasturlash muhitiga aylandi, bu dasturchiga Flash-filmini to'liq nazorat qilish, tugmalar, kliplar, menyular, havolalar, aylantirish panellarini "ishlash", hisob-kitoblarni amalga oshirish, tasavvur qiladigan har qanday ma'lumotni ko'rsatish. Macromedia Flash bilan ishlaydigan dasturchining imkoniyatlari faqat uning tasavvuri va foydalaniladigan kompyuterning kuchi bilan chegaralanadi.

Animatsiyalangan Flash videolar ko'pincha saytning asosiy sahifasini ochishdan oldin ekranlar sifatida ishlatiladi - deyarli har qanday zamonaviy teleko'rsatuvlarning ekрани bilan bir xil, taqdimotlar sifatida - "kompyuter" teleko'rsatuvi kabi narsa, saytning ko'ngilochar elementi sifatida. , masalan, "mavzuda" jonlantirilgan aforizm yoki "tirik" illyustratsiyalar sifatida.

Har bir yangi versiyada Macromedia Flashga yangi funksiyalar, bu dasturning imkoniyatlarini kengaytiradi va uni o'rganish hamda foydalanishni osonlashtiradi.

## 2. Web-sahifalarni yaratishda rastri va vektorli grafikalarining o'rnini

Hozirgi vaqtda Web-sahifalarni yaratishda birinchi o'rinlardan birini **rastri grafika** egallaydi. Rastri formatlardan GIF (**Graphics Interchange Format** - ma'lumotlar almashuvi uchun grafik format), JPEG (**Join Photographic Experts Group** – tasvir bo'yicha mutaxassislar birlashgan guruhi) va PNG (**Portable Network Graphics**- ko'chirma grafik format) va boshqa formatlarni keltirish mumkin. Rastri grafikani ishlatishda tasvir nuqtalar majmuasi (piksellar – inglizcha pixels) dan iborat bo'ladi. Bu nuqtalar bir - biri bilan bog'liq bo'lmaganligi uchun ushbu nuqtalarni har birga rangi va koordinatasi berilishi kerak. Oddiy holda, agar ikki xil rangli tasvir ishlatilsa (masalan, oq-qora), u holda har bir pikselni ta'riflash uchun bitta ikkili razryad (0- qora, 1- oq) ta'riflash etarli bo'ladi. 256 - rangli rasm uchun har bir pikselga bunday razryadlardan 8 ta kerak bo'ladi ( $256=2^8$ ). Juda ham murakkab fotorealistik rangli tasvirlar 1 pikselga 24 razryad talab qiladi. Natijada rastri tasvirli fayllar o'z ichida tasvirni rang chuqurligi o'sgani sari oshib boradi. Rastri tasvirlarni yana bir kamchiligi shundan iboratki, tasvir sifati piksel o'z ichiga bog'liq, u esa o'z navbatida monitorni imkoniyati bilan belgilanadi. Shuning uchun bir xil rasm turli monitorlarda har xil ko'rinishga ega bo'lishi mumkin. Rastri tasvir o'z ichiga o'zgartirish juda ham murakkab ishdir. Chunki bunday tasvirni kattalashtirish piksellari sonini o'sishiga olib keladi. Kompyuter grafikasi sohasidagi mutaxassislar tomonidan juda murakkab rastri tasvirlar piksellari «ko'chirish» yoki «o'chirish (agar tasvirni kichraytirish kerak bo'lsa)» algoritmlari ishlab chiqilgan, lekin ular doim ham ushbu masalani oqilona bajara olmaydi.

Web-sahifalarni yaratishda birinchi o'rinlardan birini **vektorli grafika** ham egallaydi. Bu tasvirni rasmdagi joylashuvi matematik formulalar bilan berilgan egri chiziqlar majmuasi yordamida namoyish etish usulidir. Masalan, istalgan doirani tasvirlash uchun uch-to'rt raqam kerak bo'ladi: radius, markaz koordinatalari va chiziq qalinligi. Shuning uchun, vektorli grafika rastri grafikaga nisbatan bir qancha afzalliklarga ega:

- vektorli tasvirlarni belgilovchi matematik formulalar kompyuter xotirasida rastri tasvir piksellariga qaraganda kamroq joy egallaydi;
- tasvir (yoki uning ayrim qismlarini) sifatini yuqotmasdan chegaralanmagan kattalashtirish imkoniyati mavjudligi;
- tasvirni bir platformadan ikkinchisiga ko'chirishning qulayligi.

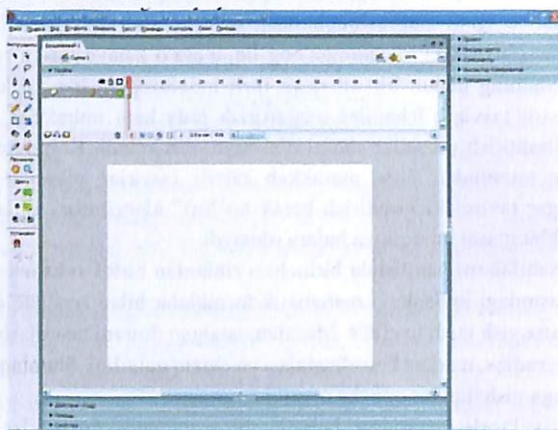
Albatta, vektorli tasvirlarni o'z kamchiliklari ham mavjud. Masalan, fotorealistik tasvirni vektorli formatda namoyish qilish murakkabroq. Flash yaratuvchilari bunga echimni tonishgan. Flash yordamida Web – sahifalar tuzishda siz nafaqat vektorli balki rastri tasvirlarni ishlatishingiz ham mumkin.

## 3. Flash dasturi bilan ishlash asoslari

**Flash** dasturida ishlashni o'rganish uchun avvalo uning interfeysi bilan tanishish lozim. Agar Macromedia firmasi tomonidan yaratilgan biror-bir dasturiy ta'minotlar (masalan, Dreamweaver muxarriri) bilan tanish bo'lsangiz Flash interfeysini o'zlashtirish unchalik murakkab bo'lmaydi. Agar Macromedia tomonidan yaratilgan uskunalarni o'zlashtirishni aynan Flash dasturidan boshlasangiz, u holda uning menyulari, uskunalari va ob'ekt xususiyatlar paneli va boshqalarning ko'rinishi Windowsning amaliy dasturlaridan farq

qiladi. Masalan, aksariyat Windowsdagi amaliy dasturlarda (ofis majmuasi) uskunar paneli tarkibini boshqarish buyruqlari “View” menyusiga kiradi. Flash dasturida esa bu buyruqlar “Window” menyusida joylashtirilgan. Tahrirlanayotgan obyekt xususiyatlar panelining ko`rinishi ham Flash dasturiga xosdir. Lekin bu turdagi yangiliklar unchalik ham qiyinlik tug`dirmaydi.

Muharrirning foydalanuvchi interfeysini tashkil etilishi umumiy holda Windows dasturlar kabi standart ko`rinishga egadir, oynaning yuqori qismida sarlavha, menyular to`plami, asosiy buyruqlar joylashtirilgan uskunar paneli va boshqalar. Muharrirda yangi vaqt diagrammaning ko`rinishi mavjud. Muxarrirni ilk bor ishga tushirganda asosiy oyna ustidan qo`shimcha muloqot oynasi naydo bo`ladi. Bu oyna sizga ish boshlashni o`ziningiz xohishingiz bo`yicha sozlash imkonini beradi. Bu muloqot oynasida 3 ta **Open a Recent Item** (oxirgi flash faylni ochish), **Create NEW** (yangi fayl yaratish), **Create form Template** (shablondan foydalanib fayl yaratish) bo`limlari bor. Bu muloqot oynasidan keraklisini tanlab olish mumkin. Aksiryat holatlarda **Create NEW -> Flash document** bandi tanlanadi va ekranga **Flash** dasturining asosiy oynasi hosil bo`ladi.



1.3.1-rasm. Flash dasturining asosiy interfeysi

#### 4. Flash dasturining asosiy oynasi ko`rinishi

**Flash** dasturining asosiy oynasi menyular, uskunarlar, obekt xususiyatlar va qo`shimchalar, ishchi maydon kabi 4 ta asosiy sohalardan iborat. **Flash** dasturining menyusi 10 bo`limdan iborat bo`lib, ular quyidagi jadvalga keltirilgan vazifalarni bajaradi.

1.4.1-jadval. Flash dasturining menyu bo`limlarining vazifalari

T/r	Nomi	Vazifasi
1	Файл	Yangi Flash hujjat yaratish, ochish, yopish, xotiraga olish usullari, tasvirlarni import va eskport qilish usullari, publikatsiya, sahifaning xususiyatlari kabi ishlarni amalga oshiradi.
2	Правка	Flash hujjatdagi tahrirlash ishlarini amalga oshirish (bekor qilish, qaytarish, nusxalash, qirqish, qo`yish usullari, qidirish usullari va boshqalar )

3	<b>Вид</b>	Flash hujjatning ko'rishini o'zgartirish amallarini bajaradi (masalan, o'lchamini kattalashtirish yoki kichiklashtirish, chegaralash va boshqalar)
4	<b>Вставить</b>	Flash hujjatga qo'yiladigan obyektlar qo'yish (yangi belgi – rolik, tugma, tasvir va boshqalari)
5	<b>Изменить</b>	Flash hujjatdagi obyektlarni tahrirlash, o'zgartirish amallarini bajaradi.
6	<b>Текст</b>	Flash hujjatdagi matnlarning yozuvi, o'lchami, stili va boshqa amallarni bajarish uchun mo'ljallangan.
7	<b>Команды</b>	Flash hujjat uchun turli buyruqlarni bajartirish va natijalar olish uchun mo'ljallangan.
8	<b>Контроль</b>	Flash hujjatni boshqarish amallari uchun mo'ljallangan.
9	<b>Окно</b>	Flash interfeysining ko'rinishi ustida amallarni bajarish uchun mo'ljallangan.
10	<b>Помощь</b>	Flash hujjat bo'yicha yordam tizimi.

## 5. Nazorat savollari



1. Qaysi kompaniya Flash texnologiyasini yaratuvchisi hisoblanadi?
2. Flash-texnologiyalarni yaratishga nima imkon beradi?
3. Flash texnologiyasi qayerda ishlatiladi?
4. Macromedia Flash dasturining imkoniyatlari?
5. Flash texnologiyasining asosiy afzalligi nimada?
6. Flash texnologiyasi vektor grafiklardan qaysi formatda foydalaniladi?
7. Flash texnologiyasi yordamida qanday animatsiya turlarini yaratish mumkin?
8. Flash texnologiyalar tarkibining elementlari nimalardan iborat?

### 1.2. Flashning uskunalar paneli. Flashning ish maydoni.

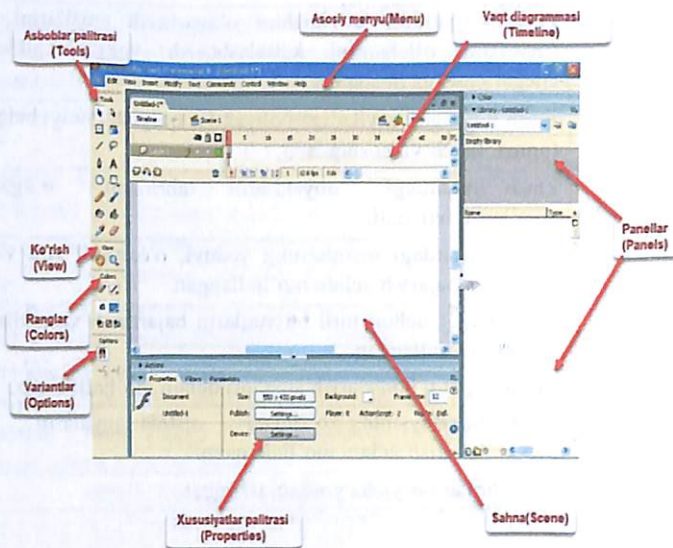
#### Reja:

1. Flash MX 8 interfeysi.
2. Flash dasturining uskunalar paneli.
3. Nazorat savollari.

**Kalit so'zlar:** *interfeys, menyu paneli, asboblardan paneli, vaqt jadvali paneli, ish maydoni, xususiyatlar paneli, o'zgartirgichlar, belgi, chiziqlar, to'ldirish, rang, kontur, sayl.*

#### 1. Flash MX 8 interfeysi

Interfeysning umumiy ko'rinishi rasmda ko'rsatilgan:



2.1.1-rasm. Flash MX 8 interfeysining umumiy ko'rinishi.

### Asosiy menyu

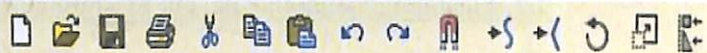
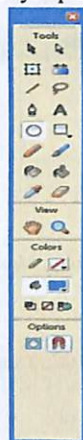
Flash dasturining ko'k rangli sarlavha satri ostida dasturni boshqarish va Flash film yaratish uchun buyruqlar to'plami bo'lgan Asosiy menyu satri joylashgan:

File Edit View Insert Modify Text Commands Control Window Help

2.1.2-rasm. Asosiy menyu satri.

Buyruqni tanlash uchun uning nomi ustida chap tugmasini bosib va keyin paydo bo'lgan pastki Menyuda kerakli buyruq nomini bosib.

Foydalanishni osonlashtirish uchun asosiy buyruqlar panelida eng ko'p ishlatiladigan buyruqlar ko'rsatiladi, bu panelni ochish uchun **Window->Toolbars->Main:**



2.1.3-rasm. Eng ko'p qo'llaniladigan buyruqlar.

Asosiy menyu va asosiy buyruqlar paneliga qo'shimcha ravishda interfeys quyidagilarni o'z ichiga oladi:

- Tools asboblari paneli,
- TimeLine vaqt diagrammasi,
- turli xil mulkni tahrirlash panellari,
- Flash filmining asosiy dizayni amalga oshiriladigan ish maydoni

Yuqori o'ng qismda siz ish maydonining o'lchovini o'rnatishingiz mumkin. Pastki qismida siz mulk panelini ko'rsatishingiz yoki yashirishingiz mumkin.

2.1.4-rasm. Asboblari paneli.

## Asboblar paneli (Window->Tools bosh menyusu)

Asboblar paneli Flash obyektларini chizish va tahrirlash uchun ishlatiladi. U bir nechta bo'limlarni o'z ichiga oladi:

- **Tools** - chizish asboblar paneli
- **View** - sahnani ko'rish (ko'paytirish, kamaytirish) va harakatlantirish uchun asboblar paneli
- **Colors** - kontur va to'ldirish uchun rang o'zgartirish paneli
- **Options** - Asboblar panelida tanlangan asbobning o'zgartirish paneli

## Vaqt shkalasi (Window->TimeLine asosiy menyusu)



2.1.5-rasm. Vaqt shkalasi

Bu erda biz ramkalar va qatlamlar bilan ishlaymiz ( 1-4 piktogramma tugmalari ) va filmni tahrirlash rejimini boshqaramiz ( 4-11 piktogramma tugmalari ) Vaqt shkalasi belgisi tugmalari 1. Qatlam qo'shiish.

2. Yo'naltiruvchi qatlamini qo'shish.
3. Qatlamlarni jildga birlashtirish.
4. Qatlamni o'chirish.
5. Ramkani markazga qo'yning.
6. Oldingi ramkalarni ko'rsatish (yarim shaffof).
7. Xuddi shu, lekin faqat oldingi ramkalarining konturlari.
8. Bir vaqtning o'zida bir nechta ramkalarni tahrirlash.
9. Ko'rish yoqilgan paytda oldingi kadrlarni ko'rsatish sozlamalari.
10. Ko'rsatish - qatlam tarkibini yashirish.
11. O'chirish - qatlamda tahrirlashni yoqish.
12. Qatlamdagi ob'ektlarning faqat konturlarini ko'rsating.

Vaqt shkalasining ushbu qismi quyidagilarga xizmat qiladi:



1. Turli xil sahnalar o'rtasida almashish.
2. Turli belgilar o'rtasida almashish uchun (bu nima - biz keyinroq ko'rib chiqamiz).
3. Masshtabdagi o'zgarishlar.

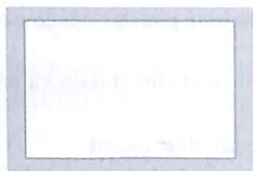
## Yordamchi panellar

Ob'ektlar, grafikalar, belgilar qo'shish yoki o'zgartirish uchun xizmat qiladi.



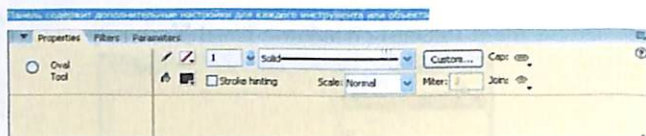
2.1.6-rasm. Ish maydoni.

Aynan shu sohada ishlaymiz. Belgilar va ob'ektlarni chizish, yozish, yaratish.



2.1.7-rasm. Xususiyatlar paneli

Panelda har bir asbob yoki ob'ekt uchun qo'shimcha sozlamalar mavjud




2.1.8-rasm. Qo'shimcha sozlamalar

## 2. Flash dasturining uskunalar paneli


### Chiziq(Line) vositasi. Foydalanishga misol

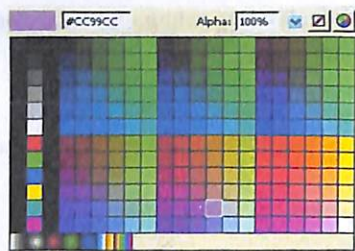
Flash-ni o'rganish eng yaxshi aniq misollar bilan amalga oshiriladi, shuning uchun yangi faylni oching. Bu har qanday usulda amalga oshirilishi mumkin:

- Asosiy menyuda **Fayl (Fayl) - New (Yangi) - General - Flash Document** yorlig'ini ketma-ket tanlang.
- **Ctrl + N** tezkor tugmalarini bosib (klaviaturadagi Boshqarish tugmachasini bosib ushlab turganda, tegishli harf tugmachasini bosib)

-  panelida asosiy buyruqlar.

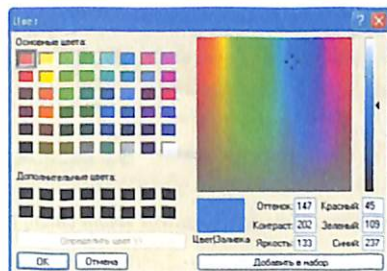
Asboblard panelida sichqonchani chap tugmasi bilan bosish orqali **Line (Line)** asbobini tanlang.

Xuddi shu joyda, rangni o'rnating (Colors bo'limida):  chiziq rangini o'zgartiruvchi sichqonchani chap tugmachasini bosib, paydo bo'lgan palitrada istalgan rangni tanlang.



2.2.1-rasm. Ranglar palitrasi

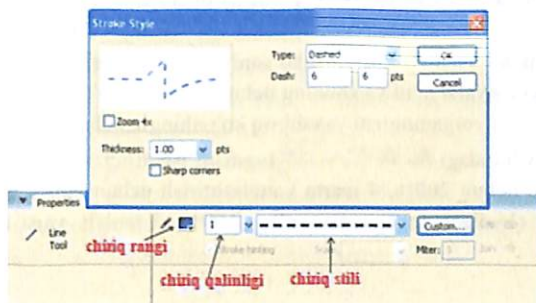
Agar sizga kattaroq ranglar diapazoni kerak bo'lsa, **Палитра** tugmasini bosib va ranglarning to'liq diapazonining yangi dialog oynasida rangni tanlang yoki uning raqamli ko'rinishini belgilang.



2.2.2-rasm. Ranglar palitrasining raqamli ko'rinishi

Boshqa qator parametrlarini tanlash uchun Asosiy menyudan **Window - Properties** buyruqlari yoki **Ctrl + F3** tugmalar birikmasini tanlash orqali ochilishi mumkin bo'lgan **Properties** panelidan (ekranning pastki qismida) foydalaning.

**Stroke Style** modifikatori qattiq, singan, qirrali chiziqlarni tanlash imkonini beradi. Buni amalga oshirish uchun **Properties** panelidagi ikkinchi tahrirlanadigan maydonning pastga o'q tugmasini bosing.



2.2.3-rasm. Stroke Style modifikatori

**Custom** tugmasini bosganingizda, **Stroke Style** dialog oynasi paydo bo'ladi, unda shakl maydonlarida turli xil variantlarni tanlab, siz yangi turdagi chiziq yaratishingiz mumkin. Agar siz **OK** tugmasini bossangiz, yangi tur chiziq turlari ro'yxatiga qo'shiladi. E'tibor bering, **Stroke Style** panelining tarkibi chiziq turini tanlashga qarab o'zgaradi.

**Chiziq qalinligi** modifikatori chiziq qalinligini o'zgartirishga imkon beradi. Buni amalga oshirish uchun ushbu modifikatordagi pastga o'q tugmasini bosing, chiziq qalinligini o'zgartirishingiz mumkin bo'lgan tutqichni harakatga keltiruvchi slayder paydo bo'ladi. Ikkinchi usul - **Chiziq qalinligi** maydonida raqamli parametrlarni oddiygina qo'lda o'rnatish.

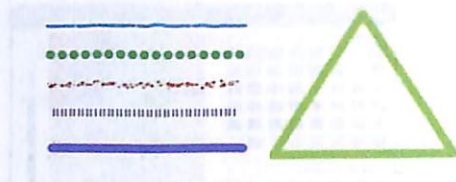
**Chiziq rangini** nafaqat asboblari panelida, balki Xususiyatlar panelida ham o'zgartirish mumkin. Rang modifikatoriga sichqonchani bosilganda ranglar palitrasini paydo bo'ladi, unda 216 ta veb-xavfsiz ranglardan biri yoki gradient tanlangan.

Agar sizga ko'proq ranglar kerak bo'lsa, tugmani bosganingizda ochiladigan to'liq ranglar palitrasidan foydalaning.

**Eslatma:** agar siz **Shift** tugmachasini bosgan holda chiziqlar chizsangiz, siz 45 daraja burchak ostida qat'iy gorizontal, vertikal va to'g'ri chiziq'larga ega bo'lasiz.


**Vazifa:** Materialni tuzatish uchun chiziq parametrlarini o'rnatish va o'zboshimchalik bilan kompozitsiyani chizish. Turli xil asbob o'zgartirgichlar bilan tajriba o'tkazing









2.2.4-rasm. Materialni tuzatish uchun chiziq parametrlarini o'rnatish. ✓

### Eslatma


Chiziq uchlarini bir-biriga to'g'ri kelishi uchun (agar kerak bo'lsa), ob'ektlarni tarmoqqa avtomatik yopishtirishni yoqing (Snap to Objects):  asosiy buyruqlar panelidagi tugmani bosib yoki View (Ko'rish) - ni tanlang -> **Snapping** -> **Snap to Objects** (obyektlarni avtomatik ravishda tarmoqqa ulash).



To'g'ri chiziq muvaffaqiyatsiz chizilgan bo'lsa, asosiy buyruqlar panelidagi  **Undo** (Bekor qilish) tugmasidan foydalaning. Oxirgi operatsiya bekor qilinadi.

Agar siz ushbu tugmani yana bir marta bossangiz, oldingi operatsiya bekor qilinadi va hokazo. Flash-da bekor qilish (Bekor qilish) operatsiyasi bir nechta, mumkin bo'lgan bekor qilishlar soni asosiy menyuda o'rnatiladi: **Edit** (Edit) - **Preferences** (Preferences) - **Undo Levels** (mumkin bo'lgan bekor qilishlar soni). Sukut bo'yicha - 100. Esda qolishi kerak bo'lgan maksimal operatsiyalar soni va shuning uchun mumkin bo'lgan bekor qilish - 300.


Nimani chizayotganingizni yaxshiroq ko'rishingiz uchun nozik tafsilotlarni chizish uchun masshtab maydonidagi   97%  tugmani ishlatib. Tugmani bosish orqali siz 2 marta kattalashtirish uchun 200%, 4 marta kattalashtirish uchun 400% va hokazolarni tanlashingiz mumkin. Va chizilgan tugagandan so'ng, 100% ni tanlab, yana rasmning haqiqiy hajmiga qayting.


### Oval (Oval Tool) uskunasi.

**Tools** (Asboblar) panelida sichqonchani chap tugmasi bilan bosish orqali  **Oval** (Oval Tool) asbobini tanlang.

Xuddi shu joyda asboblar panelidagi **Tools** (Asboblar) konturning rangini o'rnatib: kontur rangini o'zgartiruvchi ustiga sichqonchani chap tugmachasini bosib va paydo bo'lgan palitrada istalgan rangni tanlang . Agar sizga kengroq ranglar diapazoni kerak bo'lsa, Palitra  tugmasini bosib va ranglarning to'liq diapazoni yangi menyusida rangni tanlang yoki uning raqamli tasvirini belgilang.

Konturning rangi va boshqa parametrlarini tanlash uchun siz chiziq asbobni qanday tanlaganimizga o'xshash **Properties** (**Xususiyatlar**) panelidan foydalanishingiz mumkin.

**Tools** (**Asboblar**) panelidagi ovalning ichki qismini to'ldirishni tanlash uchun to'ldirish rangi o'zgartirgichni sichqonchani chap tugmasi bilan bosib va paydo bo'lgan  palitrada istalgan rangni tanlang.

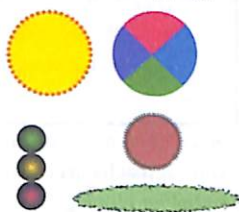
Agar sizga kengroq ranglar diapazoni kerak bo'lsa, Palitra  tugmasini bosib va ranglarning to'liq diapazoni yangi menyusida rangni tanlang yoki uning raqamli tasvirini belgilang.

✓ E'tibor bering: oval va konturning ichki qismini to'ldirish uchun siz nafaqat rangni, balki gradientni ham tanlashingiz mumkin.

Kontur parametrlari va ovalning ichki qismi o'rnatilgandan so'ng, sichqonchanning chap tugmasi yordamida ish joyiga (oq varaq) bir nechta ovallarni torting, buning uchun:



- sichqonchanning chap tugmasidagi ish joyini bosib va uni qo'ymasdan, sichqonchani istalgan yo'nalishga torting,

- sichqonchanning chap tugmasini qo'yib yuboring. (Agar siz Shift tugmachasini bosgan holda xuddi shunday qilsangiz, siz mukammal doiralarga ega bo'lasiz).





2.2.5-rasm. Oval (Oval Tool) uskunasiidan foydalanish.

Bir qatlamda chizilganligi sababli, biz quyidagilarni e'tiborga olishimiz kerak: avvalroq chizilgan oval yoki doira quyida joylashgan bo'ladi, ya'ni. barcha yangi ovallar va doiralar, xuddi ilgari chizilganlarga o'rnatilgan.

**Muhim!!**   Doira shaklidagi o'zgartiruvchiga e'tibor bering. Modifikator yoqilgan bo'lsa, aylana/kvadrat chizilganda kontur va to'lg'azish birlashtiriladi va chizilgan doirani ish maydoni bo'ylab sudrab o'tkazganingizda, to'ldirish va kontur birga harakatlanadi, agar modifikator qo'yib yuborilgan holda obyekt chizsangiz, to'ldirish va kontur alohida tortiladi. Ammo hamma narsani birgalikda o'tkazish qobiliyati doira / kvadrat markazida ikki marta bosish orqali mumkin. Ob'ekt panjara bilan qoplangan bo'ladi. Bu haqda keyinroq. Dastlabki bosqichda, agar u yoqilgan bo'lsa, modifikatorni bosishni tavsiya qilaman.

### To'rtburchak (Rectangle) uskunasi.

Yangi fayl yarating (avval tavsiflanganidek)

- Asboblar panelidan  **To'rtburchak (Rectangle)** asbobini tanlang.
- Asboblar panelidagi  **Colors** (Ranglar) bo'limida kontur rangini o'rnatib
- **Properties (Xususiyatlar)** panelida quyidagini o'rnatib:
- kontur qalinligi,
- kontur uslubi,
- rangni to'ldirish.

Ish maydonida sichqonchanning chap tugmasi yordamida bir nechta to'rtburchaklar va kvadratlarni chizing:

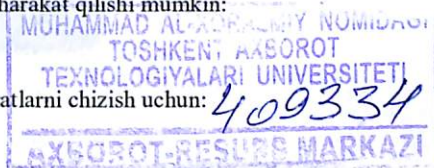
1. ish joyida sichqonchanning chap tugmachasini bosib va uni qo'yib yubormasdan sichqonchani istalgan yo'nalishga torting (klaviaturada **Shift** tugmasi bosilganda kvadratchalar paydo bo'ladi),

2. sichqonchanning chap tugmasini qo'yib yuboring.

Quyidagilar ham to'ldirish va kontur sifatida harakat qilishi mumkin:

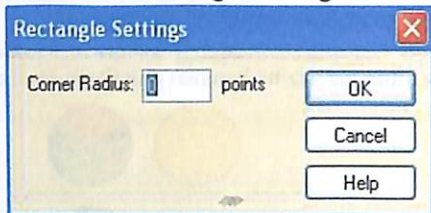
- rang,
- gradient.

Yumaloq burchakli to'rtburchaklar va kvadratlarni chizish uchun:



• **Tools** panelining **Options** bo'limida **Round Rectangle** radius modifikatorini bosing.

• Ko'rsatilgan oynada radiusni o'rnatish.  OK tugmasini bosing.



2.2.6-rasm. To'rtburchak (Rectangle) uskunasiidan foydalanish.

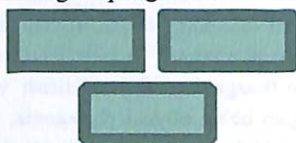
Yoki chizish paytida strelkali tugmachalarini bosib turing.

Dumaloq burchakli to'rtburchaklar chizing.

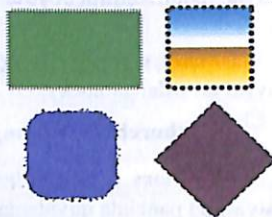
**Boshqa bir variant.**

**Properties** paneliga qarang. Keling, kvadrat/to'rtburchakning burchaklari bilan yana bir oz

tajriba qilaylik. Keling, **Join** ro'yxatidan foydalanamiz. Quyida har bir elementga misollar keltirilgan. Avval sichqonchani chap tugmasi bilan to'rtburchakni tanlang.



**Vazifa :** quyida keltirilgan misollarni chizishga harakat qiling.



**Arrow (Strelka)** uskunasiidan foydalanib, siz shakllarning shaklini o'zgartirishingiz mumkin (Kontur chiziqclarini tortosh):

• **Arrow (Strelka)**  vositasini tanlang.

• Kursorni shakl chegarasiga olib borib, **Arrow** yonida yoy paydo bo'ladigan **Arrow (Strelka)** asbobining holatini toping. Sichqonchani chap tugmachasini bosing va uni qo'yib yubormasdan, sichqonchani istalgan yo'nalishda torting.

• Sichqonchani chap tugmachasini qo'yib yuboring.



2.2.7-rasm. Arrow (Strelka) uskunasiidan foydalanish

## ✓ Eslatma

Biz bir qatlamda chizilgan bo'lsak, quyidagilarni hisobga olish kerak: avvalroq chizilgan to'rtburchaklar yoki kvadrat quyida joylashgan bo'ladi, ya'ni. barcha yangi to'rtburchaklar yoki kvadratlar ilgari chizilganlarning ustiga qo'yilganga o'xshaydi.

Ob'ektlarni guruhlash kerak bo'lsa, **Modify** -> **Group** menyusiga o'ting.

## Misollar

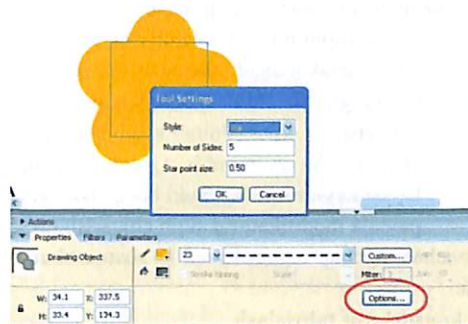
E'tibor bering, kvadrat asbobning yonida kichik pastga qaragan uchburchak mavjud. Agar siz kvadrat asbobda sichqonchani chap tugmachasini bossangiz, uni biroz bosib



turing, keyin siz ko'pburchak chizishingiz mumkin (PolyStar)

Keling, shunga o'xshash shaklni chizishga harakat qilaylik:

- Buning uchun **Многоугольник** (PolyStar)ni tanlang.
- **Properties** panelida **Options** tugmasi. Tool Settings dialog oynasi ochiladi.
- **Style** bo'limi da **star** (yulduzcha) ni tanlang. Burchaklar soni 5. OK tugmasini bosib.
- Keyin, **Properties** panelida konturning qalinligini 23 ga o'rnatib. Rangingizni didingizga qarab belgilang;).
- Kontur turiga e'tibor bering (-----) – uzulgan chiziqlar. □ Endi ko'pburchak gul chizing.



2.2.8-rasm. Ko'pburchak PolyStardan foydalanish.

Agar siz boshqa chiziq turini tanlasangiz (Hatched ro'yxatdagi oxirgi), qalinligi 44 (masalan,), siz natijada sayoh rang chiziqli yulduzni olasiz.

## Arrow vositasi

- - Tool **Arrow** (strelka): grafik ob'ektlarni tanlaydi, belgilaydi, harakatlantiradi, shaklini o'zgartiradi.

**Arrow** (strelka) vositasi va uning modifikatorlari imkoniyatlarini ko'rib chiqing .

## Ob'ektlarni o'chirish

Ob'ektlarni ish maydonidan olib tashlashning bir necha yo'li mavjud:

Hammasini olib tashlash uchun:

1. Asosiy menyudan **Edit** -> **Select All** (yoki **Ctrl+A**) tanlanadi.
2. Klaviaturadagi **Delete** (o'chirish) tugmasini bosish orqali.

Bir nechta ob'ektlarni o'chirish uchun:

1. **Arrow (strelka)** asbobi tanlanadi.
2. Sichqonchani chap tugmasi bilan ish maydonining yuqori chap burchagida, o'chirmoqchi bo'lgan ob'ekt yoki ob'ektlar ustiga bosiladi va sichqoncha tugmachasini qo'yib yubormasdan, sichqonchani ish maydonining pastki o'ng burchagiga tortiladi. Sichqoncha ko'rsatkichi tomonidan yaratilgan to'rtburchak o'chirmoqchi bo'lingan ob'ektlarni qoplashi kerak.
3. Sichqonchani chap tugmachasini qo'yib yuboriladi. Ob'ekt yoki ob'ektlar ajratib ko'rsatiladi.

4. Klaviaturadagi **Delete (o'chirish)** tugmasi bosiladi.

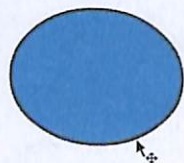
Alohida ob'ektni o'chirish uchun:

1. **Arrow (strelka)** asbobi tanlanadi.
2. Sichqonchani chap tugmasi bilan ob'ektga bosiladi.
3. Klaviaturadagi **Delete (o'chirish)** tugmasini bosiladi.

### Harakatlanuvchi ob'ektlar

1. Ob'ektni, masalan, ovalni chizish.
2. **Arrow (strelka)** vositasini tanlanadi.
3. Sichqonchani chap tugmasi bilan ob'ektga bosiladi.
4. Sichqoncha kursorini uning ustiga olib borilganda ob'ekt yonida (+) belgisi paydo bo'lganda, ob'ektni ko'chirish mumkin:

**Muhim!!** Doira shaklidagi o'zgartiruvchiga e'tibor beraylik. Modifikator yoqilgan bo'lsa, doira/kvadrat chizilganda kontur va to'lg'azish birga birlashtiriladi va chizilgan doirani

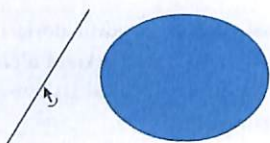


ish maydoni bo'ylab sudrab olib borilganda, to'ldirish hamda kontur birga harakatlanadi va agar siz o'zgartirgichni bo'shatib chizgan bo'lsangiz, ob'ekt alohida tortiladi. Ammo hamma narsani birgalikda o'tkazish qobiliyati doira / kvadrat markazida ikki marta bosish orqali amalga oshirish mumkin. **Ob'ekt to'r bilan qoplangan bo'ladi**. Bu haqda keyinroq. Dastlabki bosqichda, agar u yoqilgan bo'lsa, chizish paytida modifikatorni siqib chiqarish tavsiya qilinadi.

5. Sichqonchani chap tugmachasini bo'shatmasdan, ob'ektni kerakli yo'nalishda harakatlantiriladi.

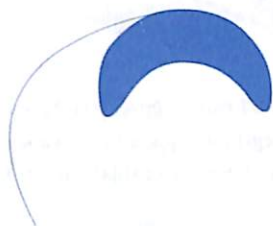
### Chiziqlar va konturlarni tahrirlash

1. Ovalga o'xshash chiziq va ba'zi shakllar tortiladi.
2. **Arrow (strelka)** asbobi tanlanadi.
3. Sichqonchani chap tugmachasini bosmasdan kursorni ob'ektning chizig'i yoki konturiga olib boriladi.
4. Sichqoncha ko'rsatkichini ob'ekt ustiga o'tkazishda uning yonida yoy paydo bo'lganda, ob'ekt shaklini o'zgartirish mumkin bo'ladi:



2.2.9-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

5. Sichqonchani chap tugmachasini bosiladi va sichqonchani kerakli yo'nalishda harakatlantiriladi:




2.2.10-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

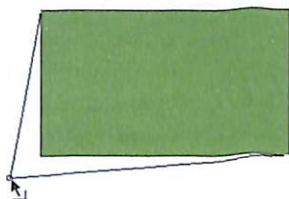
Bu tasvir quyidagicha chiqdi: to'g'ri chiziq yuqoriga chapga tortildi, oval konturining pastki qismi oval ichida yuqoriga tortildi.

Bu chizish uchun ishlatilishi mumkin bo'lgan juda qiziqarli texnologiya.

### Harakatlanuvchi burchak nuqtalari

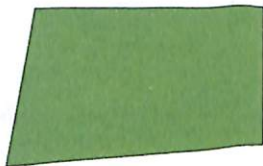
Flashdagi burchak nuqtasi to'g'ri yoki egr segmentlarning kesishish nuqtasidir.

1. To'rtburchak chiziladi.
2.  **Arrow (strelka)** asbob tanlanadi.
3. Sichqonchani chap tugmachasini bosmasdan, kursorni ob'ektning burchak nuqtasiga - bu holda uning burchagiga olib boriladi.
4. Sichqoncha ko'rsatkichi bilan ob'ekt ustiga sichqonchani olib kelinadi va uning yonida to'g'ri burchak paydo bo'lsa, burchak nuqtasini sudrab obyekt shaklini o'zgartirish mumkin. Buning uchun:



2.2.11-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

To'g'ri burchakli kursor ko'rinadigan bo'lsa, sichqonchani chap tugmasini bosib va sichqonchani kerakli yo'nalishda harakatlantiring. Sichqonchani chap tugmachasini qo'yib yuboring.



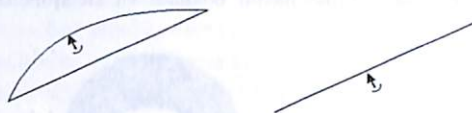
2.2.12-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

Sizning ob'ektingiz shakli o'zgaradi.

### Burchak nuqtalarini yaratish

To'g'ri chiziq chizing.

**Arrow (strelka)** asbobidan foydalanib, to'g'ri chiziqni yoyga aylantiring.



2.2.13-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

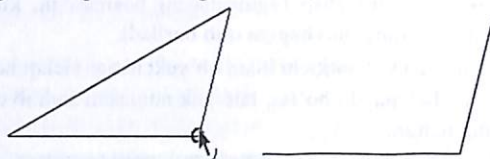
Sichqoncha kursorini egri chiziqqa qo'ying va ko'rsatgich yonida yoy paydo bo'lganda, klaviaturadagi **Ctrl** tugmasini bosib va ushlab turing. **Ctrl** tugmachasini qo'yvormasdan,



2.2.14-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

sichqonchani chap tugmasini bosib va ushlab turing, sichqonchani torting. Egri chiziqda burchak nuqtasi yaratiladi, bu chiziqda paydo bo'lgan doira bilan ko'rsatiladi. Sichqonchani chap tugmachasini qo'yib, kursorni yangi burchak nuqtasiga olib borganingizda, kursor tasviri yonida o'ng burchak paydo bo'ladi - burchak nuqtasi belgisi.

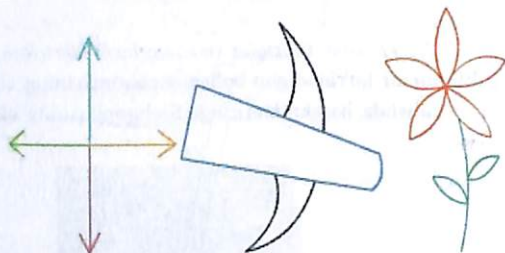
Xuddi shu narsa to'g'ri chiziq bilan amalga oshirilsa, burchak hosil bo'ladi.



2.2.15-rasm. Chiziqlar va konturlarni tahrirlash jarayoni.

Bir vaqtning o'zida burchak nuqtalarini yaratish va harakatlantirish, shuningdek, chiziqlar va konturlarni o'zgartirish juda qiziqarli grafik ob'ektlarni chizishga yordam beradi.

**Vazifa:** Yuqoridagi texnikadan foydalangan holda oddiy chizmalarni chizishga harakat qiling, masalan:



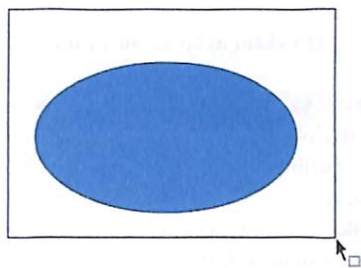
2.2.16-rasm. Chiziqlar va konturlarni tahrirlash jarayoni uchun vazifa.

### Ob'ektni masshtablash

• **Free Transform (Scale)** - ob'ektlarni o'zgartirishga imkon beradi: hajmini o'zgartirish, aylantirish va h.k.

**Ob'ektni chizish:**

Masshtablashtirmoqchi bo'lgan ob'yektni (ob'yektlarni) tanlang: buning uchun ob'yekt ustidagi ish maydonining yuqori chap qismidagi sichqonchaning chap tugmasini bosing va tugmani qo'yib yubormasdan sichqoncha ko'rsatgichini pastki o'ng burchagiga olib boring.



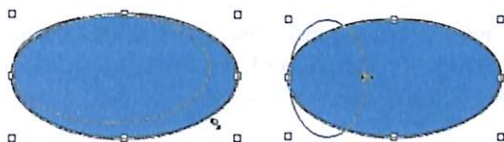
2.2.17-rasm. Ob'yektni masshtablash.

Sichqonchaning chap tugmachasini qo'yib yuboring.

☒ **Free Transform** (Masshtab) Erkin o'zgartirish vositasini tanlang.

Ob'ekt atrofida burchaklari va yon tomonlarida 8 ta kichik kvadratchalar bo'lgan masshtabli oyna paydo bo'ladi:

Agar burchak kvadratlari uchun sichqonchaning chap tugmachasini tortsangiz, proporsional masshtablash amalga oshiriladi, agar masshtablash ramkasining yon tomonlarida bo'lsa - nomutanosib bo'lsa:

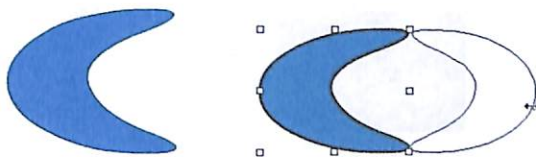


2.2.18-rasm. Free Transform asbobidan foydalanish.

Sichqonchaning chap tugmachasini qo'yuvoring - ob'ekt masshtablangan bo'ladi.


**Free Transform** (masshtab) vositasidan foydalanib, ob'ektni o'z o'qi atrofida aylantirishingiz mumkin.

Masalan, **Arrow** (strelka) asbobidan foydalanib, ob'ekt shaklini kamida bitta o'q bo'ylab simmetrik bo'lmasligi uchun o'zgartiring:



2.2.19-rasm. Arrow (strelka) asbobidan foydalanish.

• Ob'ektni tanlang.


•  asbobni tanlang va kattalashtirish maydonining o'ng tomonidagi kvadratni chapga yoki chap tomonda o'ngga torting.

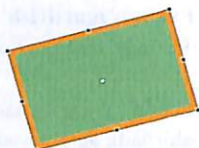


• Aylanish vertikal o'q atrofida amalga oshiriladi. Ob'ektni gorizontal o'q atrofida aylantirish uchun siz masshtablash oynasining gorizontal tomonida joylashgan kvadratni ob'ekt hajmini kamaytirish yo'nalishi bo'yicha sudrab olishingiz kerak.

### Ob'ektni aylantirish va burish

Ob'ektlarni gorizontal va vertikal o'qlar atrofida aylantirish **Scale (masshtab)** modifikatori yordamida amalga oshiriladi, ular o'z markazlari atrofida aylantirilishi yoki asosiy burilish nuqtalariga nisbatan burilishi mumkin.

1. Ob'ektni chizish.
2. **Arrow (strelka)** asbobini tanlang.
3. Masshtab qo'yimoqchi bo'lgan ob'ektni tanlang.
4. Sichqonchaning chap tugmachasini qo'yib yuboring.
5.  asbobini tanlang. Ob'ekt atrofida ramkaning burchaklari va yon tomonlarida 8 ta kichik kvadrat paydo bo'ladi:



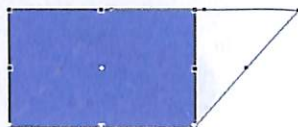
2.2.20-rasm. Ob'ektni aylantirish va burish.

Ushbu ob'ektni aylantirish uchun endi sichqoncha ko'rsatgichini burchaklardan biriga olib boring, kursor aylantiruvchi o'q shaklini oladi. Endi siz sichqonchaning chap tugmachasini bosib, uni qo'yib yubormasdan ob'ektni aylantirishingiz mumkin.





2.2.21-rasm. Ob'ektni aylantirish va burish.

Sichqonchaning chap tugmachasini qo'yib yuboring - ob'ekt aylantiriladi yoki buziladi.



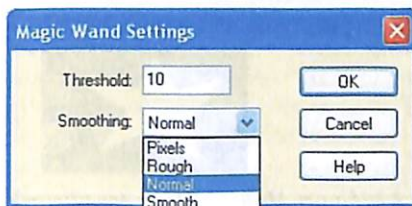
2.2.22-rasm. Lasso(Lasso) vositasi

 **Lasso asbobi (Lasso)** tanlash vositalariga ishora qiladi va erkin shakldagi ob'ektlarning keyingi tahrirlash qismlarini tanlash imkonini beradi.

Agar siz  **Lasso (Lasso)** asbobini tanlasangiz, o'zgartirishlar vositasi **Lasso (Lasso)** asboblari panelining pastki qismida paydo bo'ladi **Options (Parametrlar)** :



- **Poligon Mode** (Polygon rejimi) - to'g'ri chiziqlar bilan tanlash
- **Magic Wand** (Sehrli tayoqcha) - bitmap va plombalarda bir xil rangdagi yoki o'xshash rangdagi maydonni tanlash imkonini beradi.
- **Magic Wand Properties** (Sehrli tayoqchanning xususiyatlari). Agar siz ushbu tugmani bossangiz, Sehrli tayoqcha sozlamalari oynasi paydo bo'ladi:



2.2.22-rasm. Magic Wand Properties(Sehrli tayoqchanning xususiyatlari).

**Threshold** (bo'sag'a) - bir hududga birlashtirilishi mumkin bo'lgan ranglarning yaqinlik darajasini belgilaydi. Agar bu parametr 0 ga teng bo'lsa, u holda faqat bir xil rangdagi piksellar birlashtiriladi. Bu raqam qanchalik katta bo'lsa, piksellar shunchalik o'xshash ranglar bilan birlashtiriladi. Odatiy bo'lib, bu parametr 10 ga teng.

**Smoothing** (Silliqlash) - tanlangan maydonning chegarasini tekislash darajasini aniqlaydi:

- o **Pixel** (Pixel) - anti-alias bajarilmaydi, hududning chegaralari aniq konturlarga ega;
- o **Rough** (Qo'pol) - yengil silliqlash;
- o **Normal** (Oddiy) - o'rtacha darajani tekislash;
- o **Smooth** (Silliqlash) - silliqlashning yuqori darajasi.

✓ **Eslatma:** Amaliyotdan so'ng sehrli tayoqcha yordamida faqat rastri tasvirlarni tanlash mumkin: Asosiy menyu bandi **Modify** -> **Break Appart** (qismlarga ajratish).

Keling, misolda **Lasso** (**Lasso**) asbobining harakatini ko'rsatamiz : qandaydir ob'ektni, masalan, to'rtburchakni chizamiz:



2.2.23-rasm. Lasso (Lasso) asbobi yordamida to'rtburchakni chizish.

1. **Modify** -> **Break Appart** qismlarga ajratish.
2. **Lasso** vositasini tanlang.
3. Ish joyidagi ob'ektda sichqonchanning chap tugmasi bilan ob'ektning bir qismini tortib, ixtiyoriy shakldagi yopiq maydonni chizing:



2.2.24-rasm. Modify -> Break Appart(qismlarga ajratish).

Agar sichqonchanning chap tugmachasini qo'yib yuborsangiz, ob'ektning Lasso asbobi bilan chizilgan maydonga tushadigan qismi tanlanadi.

4. Ob'ektning tanlangan qismiga har qanday tahrirlash operatsiyalari qo'llanilishi mumkin, masalan, klaviaturada **Delete(o'chirish)** tugmasini bossangiz, ob'ektning tanlangan qismi o'chiriladi.



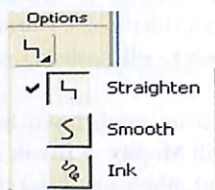
2.2.25-rasm. Modify -> Break Apart(qismlarga ajratish).

### Pencil (Карандаш) asbobi



**Карандаш** (Pencil) asbobi - tekis va egri chiziqlar chizish uchun.

1. **Карандаш** (Pencil) vositasini tanlang
2. Qalamning rangini, uning qalinligini va chizilgan chiziqning uslubini **Line** (Line) asbobida bo'lgani kabi o'rnatish.
3. **Straighten (To'g'rilash)** modifikatorini tanlang.



2.2.26-rasm. Straighten (To'g'rilash) modifikatorini tanlash.


4. Ishchi sohada sichqonchanning chap tugmasi yordamida istalgan konfiguratsiya chizig'ini chizing: ish maydonida sichqonchanning chap tugmachasini bosib va uni qo'yib yubormasdan, xuddi qalam bilan chizayotgandek sichqonchani torting.

5. Sichqonchanning chap tugmachasini qo'yib yuboring.


Agar sizga chiziqaringizni yumshoq tuzatish kerak bo'lsa, silliqlash – **Smooth(silliqlash)** modifikatorini tanlang.

Modifikatorni tanlaganingizda **Ink** (Siyoh) Flash chizilgan chiziqni tuzatmaydi.

✓ **Eslatma**

Chiziqning boshi va oxiri bir-biriga to'g'ri kelishi uchun (keyinchalik to'ldirish uchun yopiq konturni yaratish kerak bo'lganda), ob'ektlarni tarmoq tugunlariga avtomatik yopishtirish rejimini yoqing:  tugmasini bosib (**Snap to Objects**) asosiy menyuning piktoграмма tugmasi satrida yoki asosiy menyudan **View (Ko'rish)** - **Snap to Objects** ni tanlang.

### Кисть (Brush) asbobi.

**Кисть (Brush)**  asbobi chizmachilik asbobidir. Rang, gradient va rastr bilan to'ldirish uchun xizmat qiladi.

**Кистъ (Brush)** vositasi o'z modifikatorlariga ega:



- **Fill Color** - bu modifikator chiziq rangini, shu jumladan gradientlarni tanlash imkonini beradi;



**Размер** kist (Brush Size) o'lchamini o'rnatuvchi shakl maydoni;



**Форма** kist (Brush Shape) shaklini o'rnatuvchi shakl maydoni;



- to'liq bo'yashni blokirovka qilish (Lock Fill) quyish - agar ushbu parametr yoqilgan bo'lsa, bir xil gradientning barcha zarbali bitta butunning bir qismi bo'ladi, agar parametr o'chirilgan bo'lsa, hatto bitta gradientning har bir zarbasi alohida bo'ladi.

Oddiy chizish opsiyalaridan tashqari, Flashda (Brush Mode)  rejimi mavjud:



Paint Normal



Paint Fills



Paint Behind



Paint Selection



Paint Inside

**Paint Normal** - ish joyining istalgan qismi bo'yalgan

**Paint Fills** bilan to'ldirish - to'ldirilgan joylar bo'yalgan, lekin chiziqlar emas

**Paint Behind** - shakl orqasidagi fonni bo'yaydi


**Paint Selections** - tanlangan to'ldirilgan maydonlarning ichki qismlari bo'yalgan

**Paint Inside** - to'ldirish maydoni ichidagi birinchi bo'yashdan so'ng, konturlar emas, balki faqat ichki qism bo'yalgan.

### 2.2.27-rasm.Brush Mode rejimi.

1. **Кистъ (Brush)** vositasini tanlang.
2. **Paint Normal** chizish rejimini tanlang.
3. Qatlamga to'ldirishni bering. Rang maydonini bosish orqali paydo bo'lgan palitrada qandaydir gradientni tanlang.
4. **Кистъ** ning o'lchamini tanlang: avval kattaroqni oling.
5. **Кистъ** shaklini tanlang. Yassi tayoqchani taqlid qiladigan **Кистъ** shaklini tanlashga harakat qiling.
6. Biror narsa yozishga yoki chizishga harakat qiling.

### Paint Bucket (Zalivka, Vedro) asbobi.

**Paint Bucket (Zalivka, Vedro)**  asbobi yopiq yo'lga yoki kichik bo'shliqqa ega bo'lgan yo'lga ega bo'lgan shakllarni to'ldirish uchun ishlatiladi.

**Paint Bucket (Zalivka, Vedro)** asbobini tanlang, **Options** (Parametrlar) maydonida uning modifikatorlari paydo bo'ladi (qo'shimcha vositalar yoki parametrlarni o'zgartirish uchun asboblari):



- **Fill Color** (Fill color)- ob'ektlarning ichki qismini to'ldirish uchun o'zgartiruvchi.



- **Gap Size** - bo'shliq o'lchami menyusini ochish uchun o'zgartirish tugmasi.

Agar siz ushbu tugmani bossangiz, to'ldirishni amalga oshirish mumkin bo'lgan kontur uchlari orasidagi ruxsat etilgan bo'shliqlar miqdorini tanlashingiz kerak bo'lgan menyuni ochiladi:

Agar sichqonchanning chap tugmachasini qo'yib yuborsangiz, ob'ektning Lasso asbobi bilan chizilgan maydonga tushadigan qismi tanlanadi.

4. Ob'ektning tanlangan qismiga har qanday tahrirlash operatsiyalari qo'llanilishi mumkin, masalan, klaviaturada **Delete(o'chirish)** tugmasini bossangiz, ob'ektning tanlangan qismi o'chiriladi.



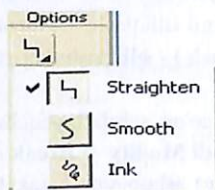
2.2.25-rasm. Modify -> Break Apart(qismlarga ajratish).

### Pencil (Карандаш) asbobi



**Карандаш (Pencil)** asbobi - tekis va egri chiziqlar chizish uchun.

1. **Карандаш (Pencil)** vositasini tanlang
2. Qalamning rangini, uning qalinligini va chizilgan chiziqning uslubini **Line (Line)** asbobida bo'lgani kabi o'rnatang.
3. **Straighten (To'g'rilash)** modifikatorini tanlang.



2.2.26-rasm. Straighten (To'g'rilash) modifikatorini tanlash.


4. Ishchi sohada sichqonchanning chap tugmasi yordamida istalgan konfiguratsiya chizig'ini chizing: ish maydonida sichqonchanning chap tugmachasini bosib va uni qo'yib yubormasdan, xuddi qalam bilan chizayotgandek sichqonchani torting.

5. Sichqonchanning chap tugmachasini qo'yib yuboring.


Agar sizga chiziqaringizni yumshoq tuzatish kerak bo'lsa, silliqlash – **Smooth(silliqlash)** modifikatorini tanlang.

Modifikatorni tanlaganingizda **Ink (Siyoh)** Flash chizilgan chiziqni tuzatmaydi.

#### ✓ Eslatma

Chiziqning boshi va oxiri bir-biriga to'g'ri kelishi uchun (keyinchalik to'ldirish uchun yopiq konturni yaratish kerak bo'lganda), ob'ektlarni tarmoq tugunlariga avtomatik yopishtirish rejimini yoqing:  tugmasini bosib (**Snap to Objects**) asosiy menyuning piktogramma tugmasi satrida yoki asosiy menyudan **View (Ko'rish)** - **Snap to Objects** ni tanlang.

### Кисть (Brush) asbobi.

**Кисть (Brush)**  asbobi chizmachilik asbobidir. Rang, gradient va rastr bilan to'ldirish uchun xizmat qiladi.

**Кистъ (Brush)** vositasi o'z modifikatorlariga ega:



- **Fill Color** - bu modifikator chiziq rangini, shu jumladan gradientlarni tanlash imkonini beradi;




**Размер** kist (Brush Size) o'lchamini o'rnatuvchi shakl maydoni;



**Форма** kist (Brush Shape) shaklini o'rnatuvchi shakl maydoni;



- to'liq bo'yashni blokirovka qilish (Lock Fill) quyish - agar ushbu parametr yoqilgan bo'lsa, bir xil gradientning barcha zarbalari bitta butunning bir qismi bo'ladi, agar parametr o'chirilgan bo'lsa, hatto bitta gradientning har bir zarbasi alohida bo'ladi. Oddiy chizish opsiyalaridan tashqari, Flashda (Brush Mode)  rejimi mavjud:



**Paint Normal** - ish joyining istalgan qismi bo'yalgan



Paint Normal

**Paint Fills** bilan to'ldirish - to'ldirilgan joylar bo'yalgan, lekin chiziqlar emas



Paint Fills

**Paint Behind** - shakl orqasidagi fonni bo'yaydi



Paint Behind

**Paint Selections** - tanlangan to'ldirilgan maydonlarning ichki qismlari bo'yalgan



Paint Selection

**Paint Inside** - to'ldirish maydoni ichidagi birinchi bo'yashdan so'ng, konturlar emas, balki faqat ichki qism bo'yalgan.




Paint Inside

### 2.2.27-rasm.Brush Mode rejimi.

1. **Кистъ (Brush)** vositasini tanlang.
2. **Paint Normal** chizish rejimini tanlang.
3. Qatlama to'ldirishni bering. Rang maydonini bosish orqali paydo bo'lgan palitrada qandaydir gradientni tanlang.
4. **Кистъ** ning o'lchamini tanlang: avval kattaroqni oling.
5. **Кистъ** shaklini tanlang. Yassi tayoqchani taqlid qiladigan **Кистъ** shaklini tanlashga harakat qiling.
6. Biror narsa yozishga yoki chizishga harakat qiling.

### Paint Bucket (Zalivka, Vedro) asbobi.

**Paint Bucket (Zalivka, Vedro)**  asbobi yopiq yo'lga yoki kichik bo'shliqqa ega bo'lgan yo'lga ega bo'lgan shakllarni to'ldirish uchun ishlatiladi.

**Paint Bucket (Zalivka, Vedro)** asbobini tanlang, **Options** (Parametrlar) maydonida uning modifikatorlari paydo bo'ladi (qo'shimcha vositalar yoki parametrlarni o'zgartirish uchun asboblari):



- **Fill Color** (Fill color)- ob'ektlarning ichki qismini to'ldirish uchun o'zgartiruvchi.



- **Gap Size** - bo'shliq o'lchami menyusini ochish uchun o'zgartirish tugmasi.

Agar siz ushbu tugmani bossangiz, to'ldirishni amalga oshirish mumkin bo'lgan kontur uchlari orasidagi ruxsat etilgan bo'shliqlar miqdorini tanlashingiz kerak bo'lgan menyuni ochiladi:



Don't Close Gaps



Close Small Gaps



Close Medium Gaps



Close Large Gaps


2.2.28-rasm. Paint Bucket  
(Zalivka, Vedro) asbobi

**Don't Close Gaps** - bo'shliqlarga ruxsat berilmaydi  
**Close Small Gaps** - kichik bo'shliqlarga ruxsat beriladi  
**Close Medium Gaps** - o'rta bo'shliqlarga ruxsat beriladi  
**Close Large Gaps** - katta bo'shliqlarga ruxsat beriladi.  
Konturda juda katta bo'shliqlar mavjud bo'lsa, to'ldirish sodir bo'lmaydi. Va bu konturni ichini bo'yash uchun uni **Pencil (Karandash)** vositasi yordamida ichini bo'yash kerak bo'ladi.



**Lock Fill** faqat gradient yoki rastr bilan bo'yashda ishlatiladi.

### Ink Bottle (Chernilnitsa) vositasi

**Ink Bottle (Chernilnitsa)** asbobi  - rang, qalinlik, chiziqlar uslubi va shakllarning konturlarini yaratadi va o'zgartiradi.

1. Ish joyida qattiq qora tekis va egri chiziq chizing.
2. Strok parametrlarini o'zgartirish (uslub, rang, qalinlik).
3. Ink **Ink Bottle** (Чернилница asbobini tanlang

Asbobning tasvirini to'kilgan siyohning uchi chiziqqa tegishi uchun chiziqqa keltiring, sichqonchani chap tugmasini bosib. Chiziq yoki kontur uning parametrlarini o'zgartiradi:

Agar shakl avvaldan kontur olib tashlangan bo'lsa yoki u mavjud bo'lmasa, uni **Ink Bottle** (Чернилница) asbobi yordamida tiklash yoki qo'shish mumkin.

### Dropper (Pipetka) asbobni.



**Dropper (Pipetka)** asbobi - ob'ektni to'ldirish yoki chiziq uslubidan nusxa ko'chiradi.

Misol:

Turli xil rangda bo'yalgan va konturli ikkita ob'ektni chizish.



2.2.29-rasm. Dropper(Pipetka) asbobidan foydalanish.

**Dropper** (Pipet) asbobini tanlang va sichqoncha kursorini chap ob'ektning to'ldirilgan joyiga olib boring.

Pipetka tasviri yonida cho'tka paydo bo'ladi, bu **Dropper** (Pipet) asbobi ob'ektning ichki qismini to'ldirish atributlarini nusxalash rejimida ishlashga tayyorligini ko'rsatadi. Sichqonchani chap tugmasini bosib.

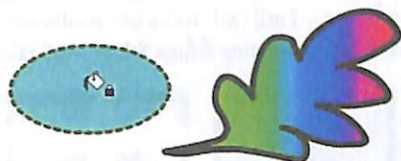


2.2.29-rasm. Dropper(Pipet) asbobi ob'ektning ichki qismini to'ldirish atributlarini nusxalash rejimida ishlashga tayyorligi.

Kursor tasviri cho'tka bilan pipitka bo'lgan joyda o'zgaradi. Y yerda qulfli chelak bo'ladi.

Va asboblari panelida tanlangan vosita avtomatik ravishda **Paint Bucket** (Zalivka, Vedro) asbobiga o'zgaradi.

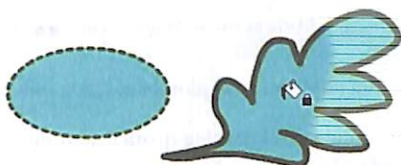
Bu ushbu belgi joylashgan ob'ektning to'ldirish atributlari nusxalanganligini ko'rsatadi.



2.2.30-rasm. Paint Bucket (Zalivka, Vedro) asbobi

Qulfli chelak tasvirini o'ng ob'ektga olib boring va sichqonchani chap tugmasini bosib.

To'g'ri ob'ekt to'ldirilishi o'zgartiriladi. U **Dropper** (Pipetka) vositasi bilan biz nusxa ko'chirganimiz bilan bir xil sozlamalarga ega bo'ladi.



2.2.31-rasm. Dropper (Pipetka) asbobidan foydalanish.

Xuddi shu tarzda, siz konturni siljitishingiz mumkin.

### Eraser (Lastik) asbobi.

**Eraser (Lastik)** asbobi chiziq, to'ldirilgan bo'yoqlarni va shakllarni qisman yoki to'liq o'chirish uchun ishlatiladi.

**Eraser (Lastik)** asbobini tanlang, **Options (Parameters)** maydonida uning modifikatorlari paydo bo'ladi (qo'shimcha vositalar yoki parametrlarni o'zgartirish uchun asboblari):



**Eraser Mode** (Rejim lastika). Ushbu modifikator tugmachasini bosish o'chirish rejimi menyusini ochadi, bu yerda siz quyidagi o'chirish rejimlaridan birini tanlashingiz mumkin:



Erase Normal



Erase Fills



Erase Lines



Erase Selected Fills



Erase Inside

**Erase Normal** (Oddiy o'chirish) - chiziq, kontur va to'ldirishlar o'chiriladi;

**Erase Fills** (to'ldirishni o'chirish) - konturlarni qoldirib, faqat to'ldirishlar o'chiriladi;


**Erase Lines** (chiziqni o'chirish) - chiziq va kontur plombalarga ta'sir qilmasdan o'chiriladi;


**Erase Selected Fills** - to'ldirishning tanlangan qismlarini o'chirish;

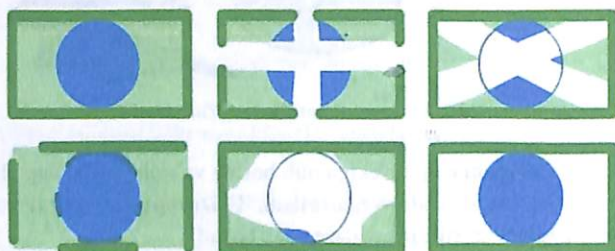
**Erase Inside** (Ichkarida o'chirish) - o'chirish faqat o'chirish boshlangan kontur ichida sodir bo'ladi.

2.2.32-rasm. Eraser (Lastik) asbobidan foydalanish.




 - **Faucet (Kran)** - qismlarni yoki butun konturni yoki plombani tezda olib tashlash uchun modifikator;

 - **Erase Shape** (o'chirgich shakli) - o'chirgichning o'lchami va shakli (dumaloq, kvadrat) o'rnatilgan maydon. Turli xil o'chirish usullarini namoyish qilish uchun aniq ko'rinadigan konturlari bilan bir-birining ichiga ikkita oval chizing:



2.2.33-rasm.Erase Shape(o'chirgich shakli).

Qismlarni yoki butun konturni yoki plombani tezda olib tashlash uchun modifikator

 - **Faucet (Kran)** - aslida ob'ektlarning qismlarini olib tashlash uchun vositadir. Agar siz ushbu modifikatorni tanlasangiz va uni birinchi navbatda to'rtburchakning konturida, so'ngra doira qismlarida bossangiz, konturlar va ularning qismlari o'chiriladi.

### Pen (Pero) asbob.



- **Pen (Pero)** asbob egri segmentlarni chizish uchun ishlatiladi.

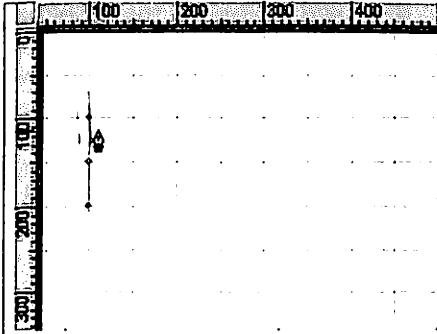
Qanday qilib aniqroq chizishni o'rganish uchun siz Flashda taqdim etilgan o'lchagichlar va to'rlardan foydalanishingiz mumkin. Chizgichlarni ko'rsatish uchun asosiy menyudan **View (Ko'rish) - Rulers (Linyeka)**ni tanlang. Va To'r(setka) ustida chizish uchun uni ish maydonida ko'rsating.

Asosiy menyudan tanlang: **View (Ko'rish)-> Grid (Setka)-> Show Grid (To'rni ko'rsatish)**.

Agar **View (Ko'rish)-> Grid (To'r)-> Edit Grid (To'rni tahrirlash)** ni tanlasangiz, **Grid (To'r)** oynasi ochiladi, unda quyidagi parametrlar o'rnatiladi:

- katakchanning vertikal va gorizontal o'lchami 50 piksel (piksel),
- katakchalarni belgilang:
  - Show Grid (To'rni ko'rsatish) maydonida;
  - Snap to Grid maydonida;
  - yacheyka o'lchamlari;
  - OK tugmasini bosing.

Asosiy menyudan **View (Ko'rish)-> Rules (Linyeka)** tanlang: Ushbu barcha manipulyatsiyalardan so'ng, To'r va o'lchagich paydo bo'ladi. Siz o'lchagichlar yordamida to'rga aniq chizishni boshlashingiz mumkin.



Tools (Asboblari) panelida (sichqonchani chapsig tugmasini bosish orqali) Pen (Pero) asbobini tanlang. Sichqoncha ko'rsatkichini koordinatali to'rning tuguniga o'rnatib:

$X=100$

$Y=150$

Sichqonchani chapsig tugmasini bosib va uni qo'yib yubormasdan, ko'rsatkichni koordinatali tugunga qo'yib, yuqoriga torting:

$X=100$

$Y=100$

Sichqonchani chapsig tugmasini qo'yib yuboring.

Keling, yangi bog'lanish nuqtasini yarataylik:  
Koordinatali to'r tuguniga ko'rsatkichni o'rnatib:

$X=200$

$Y=150$

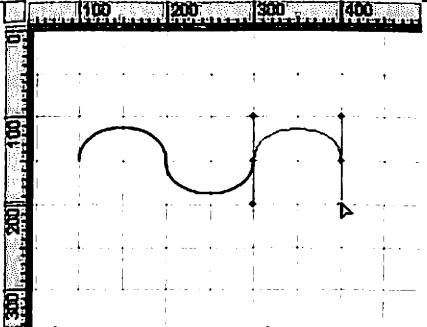
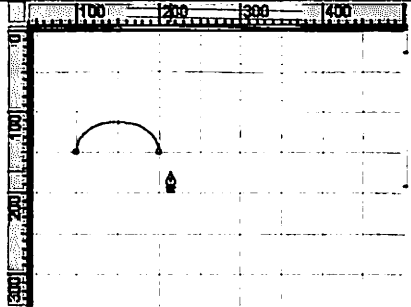
Sichqonchani chapsig tugmasini bosib va pastga torting, kursorni koordinatali tugunga qo'ying

$X=200$

$Y=200$

Sichqonchani chapsig tugmasini qo'yib yuboring.

Ishchi maydonda joyida egrichizikli segment paydo bo'ladi.



Uchinchi tugunlar nuqtasini yaratamiz:  
Koordinatali to'r tuguniga ko'rsatkichni o'rnatib:

$X=300$

$Y=150$

Sichqonchani chapsig tugmasini bosib va ko'rsatkichni koordinatali tugunga qo'yib, yuqoriga torting

$X=300$

$Y=100$

Sichqonchani chapsig tugmasini qo'yib yuboring.

Ishchi maydonda ikkinchi egrichizikli segment paydo bo'ladi.

Keling, to'rtinchi tugunlar nuqtasini yaratamiz:

Koordinatali to'rt tuguniga ko'rsatgichni o'rnatg:

$X=400$

$Y=150$

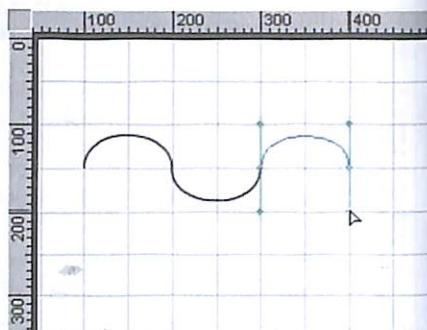
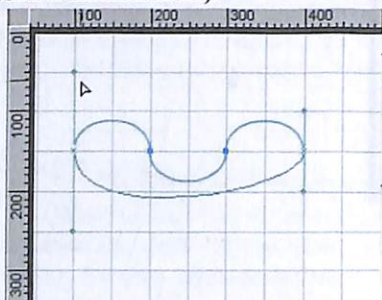
Sichqonchani chap tugmachasini bosib va pastga torting, kursorni koordinatali tugunga qo'ying

$X=400$

$Y=200$

Sichqonchani chap tugmachasini qo'yib yuboring.

Ishchi maydonda uchinchi egrichizikli segment paydo bo'ladi. Ushbu texnikadan foydalanib, siz keyingi shakllarni yaratish uchun yopiq yo'llarni yaratishingiz mumkin (Buning uchun yopiq yo'l rang yoki gradient bilan to'ldirilishi kerak).



Keling, konturni yopamiz.

Ko'rsatkichni birinchi tugun nuqtasiga koordinatali to'rtli tuguniga o'rnatg:

$X=100$

$Y=150$

Sichqonchani chap tugmachasini bosib va ko'rsatkichni koordinatali tugunga qo'yib, yuqoriga torting

$X=100$   $Y=50$

Sichqonchani chap tugmachasini qo'yib yuboring.

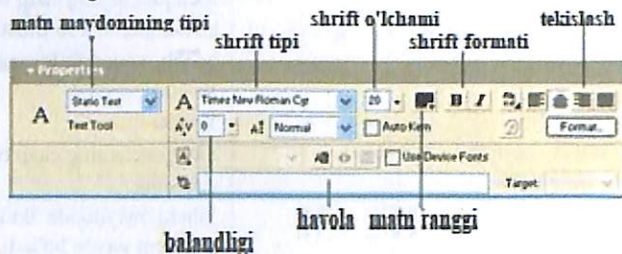
Ishchi maydonda konturni yopadigan to'rtinchi egri chizikli segment paydo bo'ladi.

**Pen (Pero)** vositasi tomonidan yaratilgan egri chiziqlar va konturlarni tahrirlash uchun **Subselect (Ostki tanlash)** vositasidan foydalaniladi.

**Text (Matn)** vositasi

**Text (Matn)** vositasi matn va matn maydonlarini yaratish va tahrirlash uchun ishlatiladi. **Tools (Asboblar)** panelida sichqonchani chap tugmasi bilan bosish orqali **Text**

**(Matn)** **A** asbobini tanlang.



2.2.34-rasm. Pen(Pero)da Subselect (Ostki tanlash) vositasi.

Sichqonchanning chap tugmasi bilan ishchi maydonni bosing va paydo bo'lgan bo'sh matn yorlig'iga matnni kiriting - Internet.

Tanlovni olib tashlash uchun matn maydonidan tashqaridagi ish maydonini sichqonchanning chap tugmasi bilan bosing.

Matnning tanlangan qismiga (matn maydoniga ikki marta bosing va kerakli harfni tanlang), **Text (Matn)** vositasi uchun yangi o'zgartiruvchi qiymatlarni qo'llang, masalan, tanlangan harfning o'lchamini, rangini o'zgartiring:



2.2.35-rasm. Flash matn harflarini vektor grafikasiga aylantirish

Flash matn harflarini vektor grafikasiga aylantirish uchun qiziqarli usullarga ega. Bu bir qator qo'shimcha funksiyalarni beradi:

- o Keyin matnni grafikalar - gradientlar kabi "to'ldirish" mumkin,
- o Siz harflarni o'zingiz o'zgartirishingiz mumkin.

Buning uchun:

1. Asboblardan Arrow (Strelka) asbobini tanlang.
2. Matn blokini tanlang - sichqonchanning chap tugmasi bilan ustiga bosing.
3. Asosiy menyudan **Modify -> Break Apart** (O'zgartirish - qismlarga ajratish) ni tanlang. Buni ikki marta bajaring, chunki birinchi marta matn alohida harflarga bo'linadi.
4. Grafikdan tashqaridagi ish maydonida sichqonchanning chap tugmachasini bosish orqali olingan grafikdan tanlovni olib tashlang. Endi har bir harf alohida grafik figuradir va unga nisbatan grafikaga qo'llaniladigan barcha texnologiyalar qo'llanilishi mumkin, masalan, allaqachon ma'lum bo'lgan "Kontur chiziqclarini tortish" usulidan foydalanib, siz har bir harfni grafik shakl sifatida o'zgartirishingiz mumkin. har qanday tarzda va raqamlarni to'ldiring - harflar, masalan, gradient bilan.

#### 4. Nazorat savollari



1. **Flash MX 8** ilova interfeysi qanday qismlardan (panellardan) iborat ?
2. Qaysi panel eng pastda joylashgan?
3. Chizish uchun qaysi panel javobgar?
4. (chizish) asboblardan paneli nechta blokdan (bo'limdan) iborat?
5. Xususiyatlar panelini qanday faollashtirish mumkin?
6. Ish stolidan (maydonda) qanday qilib to'r(setka) ni o'rnatish mumkin?
7. Ob'ektning konturini qanday bo'yash (rangini belgilash) mumkin?
8. **Lasso** vositasining vazifasi nimadan iborat?
9. **Dropper (Pepitka)** vositasining vazifasi nimadan iborat?
10. **Ink Bottle (Siyohdon)** asbobining vazifasi nimadan iborat?
11. **Pen (Pero)** vositasining vazifasi nimadan iborat?
12. **Инструмент Кист (Brush)** vositasining vazifasi nimadan iborat?

13. **Arrow** (strelka) asbobining vazifasi nimadan iborat?
14. **A** Belgisi bo'lgan asbobning vazifasi nimadan iborat?
15. **Flash MX 8** da qanday cho'tka rejimlari mavjud?

### 1. 3. Foydalanuvchi ishini tashkil qilish vositalari. Asboblarni panelini o'zgartirish. Tezkor tugmalar birikmalarini sozlash.

#### Reja:

1. Asboblarni panelini o'zgartirish.
2. Yordamchi panellar.
3. Tezkor tugmalar birikmalarini sozlash.
4. Nazorat savollari.

**Kalit so'zlar:** *asboblarni paneli, boshqaruv tugmalari, belgilash, sozlash.*

#### 1. Asboblarni panelini o'zgartirish.

**Flashga** kiritilgan foydalanuvchi ishini tashkil qilish vositalari ushbu dastur bilan ishlashni tezlashtirish, vaqtni qisqartirish, shuningdek, foydalanish qulayligi va samaradorligini oshirish uchun mo'ljallangan. Ushbu maqsadlarga erishish uchun yordam tizimi zarur. Ushbu yordam tizimi bir necha usullar bilan tashkil etilgan:

- HTML formatidagi interaktiv axborot tizimi;
- HTML sahifalar ko'rinishidagi interaktiv kitob;
- 10 qismli Flash film ko'rinishidagi interfaol darslar;
- Tezkor veb-quvvatlash;
- Flash video misollar to'plami (bu misollar nafaqat Flash imkoniyatlarini ko'rsatibgina qolmay, balki yangi videolarni yaratish uchun ham asos bo'lib xizmat qiladi).

Flash yaratuvchilari tomonidan yaratilgan barcha qulayliklarga qaramay, barcha foydalanuvchilarning ehtiyojlarini qondirish, albatta, qiyin. Foydalanuvchining tayyorgarlik darajasi, maqsadli videoning xususiyatlari va boshqalar ta'sir qiladi. Shuning uchun Flash yaratuvchilari foydalanuvchilarga muharrirning tashqi ko'rinishini moslashtirish imkoniyatini yaratdilar, bu qulay. Eng ko'p ishlatiladigan variantlardan ba'zilari:

- ekranda joylashgan asboblarni paneli tarkibi va formatini o'zgartirish;
- ish stolidagi tasvirlarning ko'rinishini o'zgartirish;
- tugma belgilari

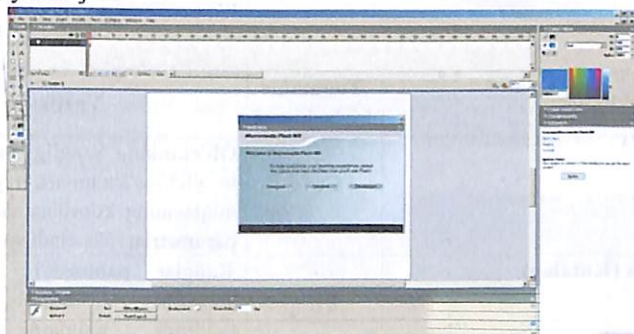
Flash texnologiyasi 3 xil konfiguratsiyaga ega:

- konstruktor (Designer-dizayner) uchun; □ yaratuvchi uchun (Developer - ishlab chiquvchi); □ universal konfiguratsiya uchun (General-umumiy).

Har bir konfiguratsiya haqida qisqacha ma'lumot olishni istasangiz, "Помощь" menyusidan "Добро пожаловать" buyrug'ini tanlang. Ekranda Flash yordamida yaratilgan dialog oynasi paydo bo'ladi. U uchta bo'limdan tashkil topgan menyudan iborat bo'lib, bu bo'limlar mos keladigan konfiguratsiyani o'rnatish uchun mo'ljallangan. O'rnatish rejimiga o'tish uchun quyidagilarni bajaring:

Asosiy oynada **Window** (Окно) menyusidan **Panel Sets** (Panel to'plamlari)ni tanlang.

Ochiladigan menyudan kerakli variantni tanlang, standart tartib elementi **Default Layout** universal konfiguratsiyaga ishora qiladi. Designer va Designer variantlarida uchta modifikatsiya mavjud.



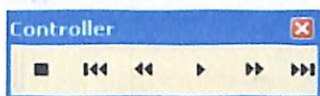
3.1.1-rasm.Help (Yordam) dialog oynasi.

Barcha panellar ro'yxati asosiy oynaning **Window** («Окно») bo'limida joylashgan. Panellarni o'zingiz xohlagan tarzda joylashtirganingizdan so'ng, tanlangan konfiguratsiyani keyingi ish seanslarida foydalanish uchun saqlashingiz mumkin. Konfiguratsiyani saqlash uchun quyidagi amallarni bajaring. «Окно» menyusidan **Save Panel Layout (Panel maketini saqlash)**ni tanlang. Foydalanuvchi paneli konfiguratsiyasi ma'lumotlari belgilangan nom ostida **Flash \ First Ruu \ Panel Sets** papkasida saqlanadi. Shuning uchun konfiguratsiya nomi siz foydalanayotgan tizim talablariga mos kelishi kerak.

## 2.Yordamchi panellar

### Kontroller

**Kontroller** ya'ni tekshirish moslamasi filmni tinglashni boshqarish va vaqt jadvalidagi kadrlar bo'ylab harakatlanish uchun ishlatiladi.



3.1.2-rasm. Kontroller moslamasi.

Asosiy menyudan chaqirish uchun quyidagi ketma-ketlik bajariladi: **Window - Toolbars - Controller** (Oyna - Asboblar - Kontroller).

Sichqonchanning chap tugmasi bilan kontroller piktogrammalarini bosganingizda, jadvalda tasvirlangan amallar bajariladi:

Piktogramma	Harakat
	Film ijro qilishini to'xtatish
	Filming boshiga o'tish
	Orqaga qadam
	Film ijrosini boshlash
	Oldinga qadam
	Filming oxiriga o'tish

3.1.3-rasm. Kontroller piktogrammalari.

**Panellar (палитры)**

Barcha panelar asosiy Window menyusidan chaqiriladi.

*3.1.1-jadval. Panellar.*

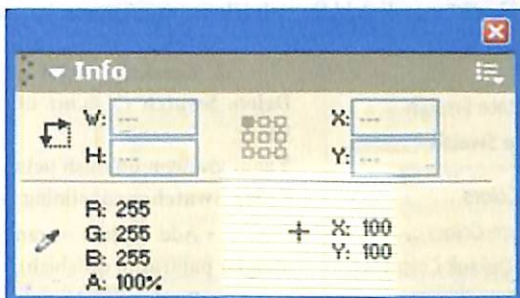
Panel	Tugmalar birlashmasi	Vazifasi
<b>Info</b> (Ma'lumot)	Ctrl-I	Ob'ektning kengligi va balandligi bo'yicha o'lchamlari, ro'yxatga olish nuqtasining koordinatalari bo'yicha parametrlari aks etadi va tahrirlanadi
<b>Color Swatches (Katalog)</b>	Yo'q	Ranglar palitrasini chaqirish, yaratish, va saqlash, tanlangan palitrada ranglarni o'chirish, kiritish, almashtirish va saralash mumkin
<b>Color Mixer</b>	Yo'q	RGB, HSB, Hex rejimlarida yangi ranglarni olish mumkin. Shu yerning o'zida siz rang va shaffoflikni (ranglar raqamli ifodasini ham o'rnatishingiz mumkin, bu turli aralashmasi) xil fayllarda bitta rang sxemasini yaratishda, gradientning rang sxemasini o'zgartirishda juda muhimdir
<b>Components</b>	Ctrl-F7	Turli xil komponentalarni qo'shish
<b>Align</b>	Ctrl-K	Belgilangan ob'ektni gorizontal, vertical, ixtiyoriy chet bo'ylab yoki markaz bo'ylab tekislash mumkin. Barcha belgilangan ob'ektlarning balandligini va kengligini birinchi tanlangan ob'ektning shakliga mos keltirish mumkin
<b>Scene</b> (sahna)	Yo'q	Bor sahnani o'chirish yoki qayta nomlash uchun yangi sahna qo'shish mumkin
<b>Transform</b>	CTRL-T	(Constrain) gorizontal va vertikal ravishda foizda yoki proporsional masshtabda ob'ekt masshtabini o'zgartirish, (Rotate) ob'ektni graduslarda aylantirish, (Skew) ob'ektni buzilishi
<b>Library</b>	F11	Belgilar kutubxonasi

### Info (Информация) paneli

Info panelining tahrirlanadigan oynalarda ob'ektning quyidagi parametrlari aks etadi va tahrirlanadi:

- kengligi bo'yicha o'lchamlar (W),
- balandligi bo'yicha o'lchamlar (H),
- ro'yxatga olish nuqtasining koordinatalari (X, Y).

Ushbu misolda ob'ektni ro'yxatga olish nuqtasi koordinatalari ish maydonining yuqori chap burchagiga nisbatan berilgan.



3.1.4-rasm. Info (Информация) paneli

Panelning pastki qismida quyidagilar aks etadi:

- RGB formatidagi rangning raqamli ifodasi,
- shaffoflik (A),
- kursor joylashuvi koordinatalari (XY).

### Color Swatches paneli


Color Swatches (Rang namunalari) (Образцы) panelida rang va gradientni o'rnatish mumkin. Siz tanlangan ranglar palitrasida ranglar palitrasini chaqirishingiz, yaratishingiz va saqlashingiz, ranglarni o'chirishingiz, kiritishingiz, almashtirishingiz va saralashingiz mumkin.



3.1.5-rasm. Color Swatches paneli

Color Swatches (Obrazsi) standart paneli ikki qismdan iborat:



- yuqori qismi - 216 Web - xavfsiz rang palitrasi, shuningdek siz yaratgan har qanday ranglar, shu jumladan qisman shaffof ranglar.
- pastki qismi - gradient bo'lib, u siz tomoningizdan yaratadigan standart gradientlar to'plamlardan iborat.  tugmani bosilganda panel bilan ishlash uchun menyusti buyruqlari paydo bo'ladi:

1. Agar siz palitradagi namunadan farqli rangni, masalan, shaffoflikni xohlasangiz, avval rangni tanlashingiz kerak. Ushbu rang **Color Mixer** panelida paydo bo'ladi, bu yerda siz, masalan, rangning shaffofligini o'zgartirishingiz mumkin, so'ngra **Color Mixer** paneli tugmasini bosish orqali **Add Swatch (Добавит образец)** buyrug'ini tanlang.

**Color Swatches** paneliga o'tsangiz, siz ushbu yangi yarim shaffof rangni ko'rasiz.

2. Keraksiz rangni olib tashlash uchun uni tanlab,

Duplicate Swatch

Delete Swatch

Add Colors...

Replace Colors...

Load Default Colors

Save Colors...

Save as Default

Clear Colors

Web 216

Sort by Color

Help

Maximize Panel

Close Panel

**Delete Swatch (Удалит образец)** buyrug'ini tanlashingiz kerak.

Yangi gradient qo'shish uchun amallar bir xil.

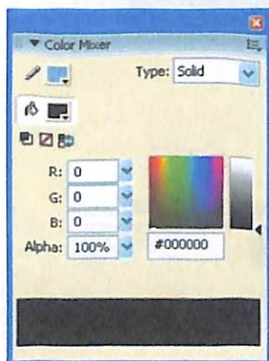
**Color Swatches** panelining boshqa buyruqlari:

- Add Colors – ranglar qo'shish (fayldan boshqa ranglar palitrasini qo'shish),
- Replace Colors - ranglarni almashtirish (boshqa rang palitrasi faylidan),
- Load Default Colors - standart ranglar palitrasini yuklash,
- Save Colors - ranglarni saqlash,
- Save As Default - standart sifatida saqlash,
- Clear Colors - aniq ranglar (qora va oqdan tashqari barcha ranglarni olib tashlash uchun),
- Web 216 - xavfsiz 216-rangli Web palitrasini yuklash,
- Sort by Colors – ranglar bo'yicha tartiblash.

3.1.6-rasm. *Color Swatches* panel buyruqlari

### Color Mixer (aralashtirgich) paneli

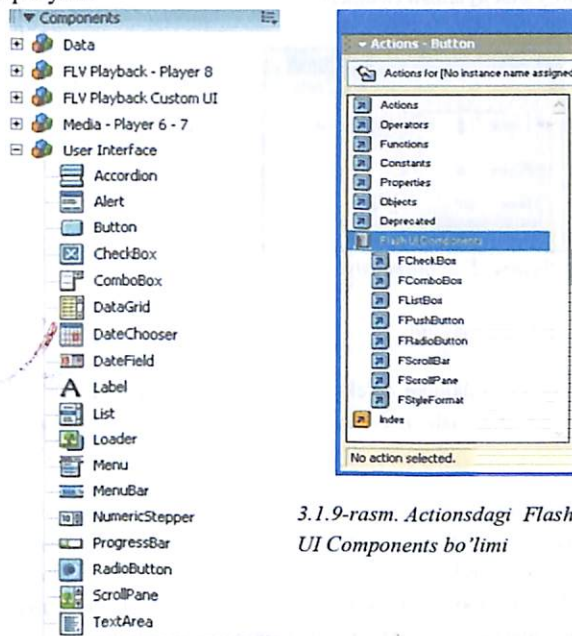
**Color Mixer** panelida siz RGB, HSB rejimlarida ranglarni ko'rsatishingiz, o'rnatishingiz va tahrirlashingiz mumkin. Bu yerda siz raqamli kodda rang va shaffoflikni ham o'rnatishingiz mumkin, bu turli xil fayllarda bitta rang sxemasini yaratishda, gradientning rang sxemasini o'zgartirishda juda muhimdir.



3.1.7-rasm. *Color Mixer (aralashtirgich) paneli*

## Components (Komponentlar) paneli

Components deb nomlangan maxsus panel yordamida filimga komponentlarni kiritish eng qulaydir.



3.1.9-rasm. Actionsdagi Flash UI Components bo'limi

Flash MX da komponentlar 5 ta bo'limga birlashtirilgan. Biz uchun eng ko'p talab qilinadigan (User Interface) foydalanuvchi interfeysi bo'limida:

- **CheckBox** - belgilash katakchasi;
- **ComboBox** - kombinatsiyalangan oyna (ya'ni elementlarni tahrirlash imkoniyatiga ega ochiladigan ro'yxat);

3.1.8-rasm. Komponentdagi User interface bo'limi

Flash MX da komponentlar 5 ta bo'limga birlashtirilgan. Biz uchun eng ko'p talab qilinadigan foydalanuvchi interfeysi bo'limida:

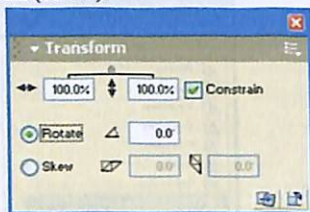
- **List** – ro'yxat;
- **CheckBox** - belgilash katakchasi;
- **ComboBox** - kombinatsiyalangan oyna (ya'ni elementlarni tahrirlash imkoniyatiga ega ochiladigan ro'yxat);
- **List** – ro'yxat;
- **PushButton** - tugma;
- **RadioButton** - almashtirish;
- **ScrollBar** – aylantirish paneli;
- **ScrollPane** - oyna.

Komponentlardan eng keng tarqalgan foydalanish Web-sahifada joylashtirilgan shaklning analogini yaratishdir. Shu bilan birga, komponentlar asosida siz Windows ilovalari uchun Flash filmingiz uchun "an'anaviy" foydalanuvchi interfeysini yaratishingiz mumkin. Bunday holda, foydalanuvchining film bilan ishlashi oddiy dastur bilan ishlashdan unchalik farq qilmaydi.

## Transform paneli (Transformatsiya)

Transform panelida quyidagi parametrlar aks etadi va o'rnatadi:

1. ob'ekt miqyosidagi o'zgarishlar: gorizontaal va vertikal ravishda foiz sifatida yoki proporsional o'lchov (Constrain),
2. ob'ektni gradus bo'yicha aylanish (Rotate).
3. ob'ektni buzilishi (Skew).



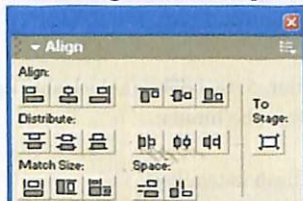
3.1.10-rasm. Transformatsiya(Transform) paneli

Palitraning pastki qismida piktogramma mavjud:

□ buferga nusxa ko'chiring va tanlangan ob'ektlar guruhini yoki panelda ko'rsatilgan o'zgartirish parametrlari bilan ob'ektni ish maydoniga joylashtiring; □ nusxalash va joylashtirishni bekor qilish.

## Align (Viravnivanie) panelni

Align panelida siz tanlangan ob'ektlarni gorizontaal, vertikal, istalgan chekka yoki markazga tekislashtirish mumkin. Siz barcha tanlangan ob'ektlarning balandligi va kengligini birinchi tanlangan ob'ektni shakliga moslashtirishingiz, ob'ektlar orasidagi teng masofani o'rnatishingiz va ob'ektlarni sahna markaziga nisbatan taqsimlashtirishingiz mumkin.



3.1.11-rasm. Tekislash(Align) paneli

Belgilangan tugmani bosganingizda, tanlangan ob'ektlarga nisbatan quyidagi amallar bajariladi:

### Align (ob'ektlar chegaralari va o'qlariga nisbatan tekislash)

- chap tomon chetiga tekislash
- markazlarni gorizontaal o'qlarda tekislash
- o'ng tomon chetiga tekislash
- yuqori chegara bo'yicha tekislash
- vertikal o'qlari bo'yicha markazga tekislash
- quyi chegara bo'yicha tekislash

### Distribute (bo'shliqlar)

- yuqori chegaralar orasidagi teng masofalar
- markazlar orasidagi teng vertikal masofalar

- pastki chegaralar orasidagi teng masofalar
- chap chegaralar orasidagi masofa teng
- gorizontal markazlar orasidagi teng masofalar
- o'ng chegaralar orasidagi teng masofalar

### Match size (o'lcham bo'yicha tekislash)

Barcha tanlangan ob'ektlarning o'lchamlari ularning eng kattasiga teng bo'ladi:

- kengligi bo'yicha
- balandli bo'yicha
- kengligi va balandligi bo'yicha

### Spase (teng oraliq)

- vertikal bo'yicha teng oraliq
- gorizontal bo'yicha teng oraliq

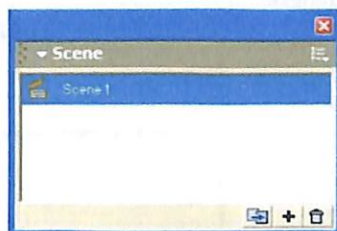
### To stage (sahnaga nisbatan)

- ob'ektni tarqatish rejimini yoqish
- sahnaga nisbatan

Bu muhim panel. Bu yerda har bir belgining imkoniyatini tekshirishni tavsiya qilinadi. Buning uchun siz, masalan, aylana chizishingiz, uni 5 marta takrorlashingiz, ma'lum bir tartibda qo'yishingiz va chapga, o'ngga va hokazolarga nisbatan markazlashingiz mumkin.

**Scene (Sahna) paneli** Scene (Sahna) panelida siz:

- tanlangan sahnaning nusxasini yaratish
- yangi sahna qo'shing
- tanlangan sahnani o'chirish



3.1.12-rasm. Sahna(Scene) paneli

Sahna nomini o'zgartirish uchun:

- sichqonchanning chap tugmasi bilan sahna nomini ikki marta bosing,
- yangi nom bering
- OK tugmasini bosing.

Agar sizning filmingiz katta bo'lsa, uni mustaqil semantik blokli sahnalarga bo'lish tavsiya etiladi.

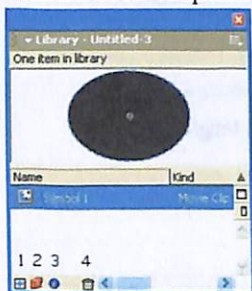
### Library paneli

**Library** paneli siz ishlayotganingizda yaratilgan ob'ektlarni saqlash uchun ishlatiladi. Ushbu ob'ektlar boshqa loyihalarda ham qo'llanilishi mumkin.

Library palitrasini boshqarish uchun piktogramma tugmalari:

1. ob'ektni qo'shish
2. papka yaratish
3. belgilar xususiyatlari

4. ob'ektni kutubxonadan olib tashlang ya'niy o'chirish Keyinchalik bu panelni batafsil ko'rib chiqamiz. Bu juda muhim!!!

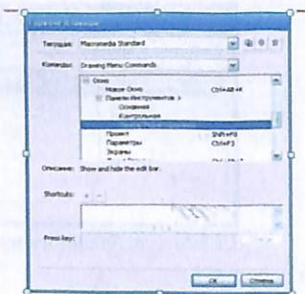


3.1.13-rasm. Kutubxona (Library) paneli

### 3. Tezkor tugmalar birikmasini sozlash

Grafik foydalanuvchi interfeyslarining tarqalishi bilan ilovalarda bajariladigan ko'plab funksiyalar "sichqoncha" rejimiga o'tkaziladi. Ammo amaliyot shuni ko'rsatadiki, ba'zi tez-tez ishlatiladigan buyruqlarni bajarish uchun "tezkor tugmalar" (shortcut keys yoki Hotkeys) dan foydalanish osonroq

Flashda foydalanuvchi alohida tezkor tugmalar birikmalarini belgilash imkoniyatiga ega. Ushbu sozlama Edit («Правка») menyusiga kiritilgan Keyboard Shortcuts... orqali chaqiriladigan maxsus dialog oynasi orqali amalga oshiriladi. Bu oyna quyidagi interfeys elementlaridan tashkil topgan.



3.1.14-rasm. Keyboard Shortcuts muloqot oynasi.

Current Set ro'yxatida ("Joriy to'plan") siz quyidagi grafik muharrirlardagi protsessorlarga tanish bo'lgan oldindan belgilangan yorliq birikmalaridan birini (Shortcut) tanlashingiz mumkin :

- Macromedia Fireworks 4;
- Macromedia Freehand 10; □ Adobe Illustrator 10; □ Adobe Photoshop 6.

Macromedia Standard va Flash 5 ning variantlari ham mavjud. Buyruqlar (buyruqlar ro'yxati bilan) quyidagi to'rt turdagi buyruqlarni o'z ichiga oladi:

- Drawing Menu Commands (chizish menyusi buyruqlari);
- Drawing Tools (chizma asboblari);
- Test Movie Menu Commands (videolarni sinash uchun menyu buyruqlari);

- **Actions Panel Commands** ( Actionscript muharririning menyu buyruqlari ). Quyidagi ro'yxatda paydo bo'ladigan buyruqlar to'plami tanlangan variantga bog'liq. **Drawing Menu Commands** opsiyasi tanlangan, Flash asosiy menyusining barcha buyruqlari ( **Fayl** , **Tahrirlash** va h.k.) ko'rsatiladi. Kerakli variantning panel menyusiga kirish uchun menyusning chap tomonidagi [x] belgisini tanlang.

#### 4. Nazorat savollari



1. Foydalanuvchilarni ishini tashkil qilish vositalari nima uchun zarur?
2. Flashda yordam tizimi qanday tashkil etilgan ?
3. Flashda foydalanuvchining ish maydonini sozlashda qaysi sozlamalar ko'proq qo'llaniladi?
4. Flash texnologiyasi nechta konfiguratsiyaga ega?
5. Flash texnologiyasi qanday konfiguratsiyadan iborat?
6. Flashdagi umumiy konfiguratsiya qanday nomlanadi?
7. Dizaynerlarning Flashda ishlashi uchun konfiguratsiya qanday nomlanadi?
8. Dasturchilarning Flashda ishlashi uchun konfiguratsiya qanday nomlanadi?
9. Flashdagi universal konfiguratsiyaga ishora qiluvchi shablon tartibi qanday nomlanadi?
10. Flashdagi barcha panellar ro'yxati qayerda ?
11. Flashda buyruqlar nechta guruhga bo'lingan?
12. Foydalanuvchi paneli konfiguratsiya ma'lumotlari (sozlamalari) qayerda saqlanadi?
13. Konfiguratsiya ma'lumotlari foydalanuvchi tomonidan oxirgi marta qaysi papkada ishlatilgan?
14. Qanday qilib alohida tezkor tugmalar birikmalarini belgilashim mumkin?
15. Tezkor tugmalarni sozlash uchun dialog oynasini qanday chaqirishim mumkin?
16. Qanday yordamchi panellarni bilasiz?
17. Panellarning spmboxi qayerda joylashgan (Ular asosiy menyusning qaysi variantida joylashgan)?
18. Transform panelining maqsadi.

#### 1.4. Macromedia flashda animatsiya yaratish. Interfaollik.

##### Reja:

1. Animatsiya tushunchasi va turlari.
2. Vaqt diagrammasi – Timeline.
3. Qatlamlar bilan ishlash.
4. Flash dasturida kadrma-kadr animatsiya.
5. Hisoblangan animatsiya. Motion Tweening animatsiyasi.
6. Traektoriya bo'ylab animatsiya. Yo'naltiruvchi qatlamlar.
7. Nazorat savollari.

**Kalit so'zlar:** animatsiya, kadr, qatlam, vaqt diagrammasi, TimeLine, kadrma-kadr animatsiyasi, hisoblangan animatsiya, sprite animatsiya, morfing, shakl o'zgartirish, motion tweening.

## 1. Animatsiya tushunchasi va turlari

Insoniyat paydo bo'lgandan boshlab harakatni o'z san'atida aks ettirishga harakat qilgan. Harakatni chizmada aks ettirishga birinchi urinishlar miloddan avvalgi 2000 yilga to'g'ri keladi (Misr).

Bugungi kunda harakatni tasvirlash animatsiya yordamida amalga oshirilishi mumkin.

**Animatsiya** - bu tasvirlarni yaxlit vizual idrok etishni ta'minlaydigan chastotada chizmalar yoki kadrlar ketma-ketligini ko'rsatish orqali kino, televizor yoki kompyuter grafikasida harakatni sun'iy tasvirlash.

Animatsiya, uzluksiz harakatdan foydalanadigan videodan farqli o'laroq, ko'plab mustaqil chizmalardan foydalanadi. "Animatsiya" so'zining sinonimi - "animatsiya" - mamlakatimizda juda keng tarqalgan. Animatsiya va animatsiya bir xil san'at turining turli xil ta'riflaridir.

Biz uchun ko'proq tanish bo'lgan atama lotincha "multi" so'zidan kelib chiqqan - juda ko'p va an'anaviy tasvirni ko'paytirish texnologiyasiga mos keladi, chunki qahramon "jonlanishi" uchun uning harakatini ko'p marta takrorlash kerak: soniyada 10 dan 30 gacha chizilgan kadrlar. Dunyoda qabul qilingan "animatsiya" ning professional ta'rifi (lotincha "anima" - ruh, "animatsiya" - jonlanish) animatsion kinoning barcha zamonaviy texnik va badiiy imkoniyatlarini eng aniq aks ettiradi, chunki animatsiya ustalari shunchaki emas balki o'z qahramonlarini jonlantiradi va ular ijodiga qalbining bir qismini qo'shadi.

### Animatsiya yaratish texnologiyalari

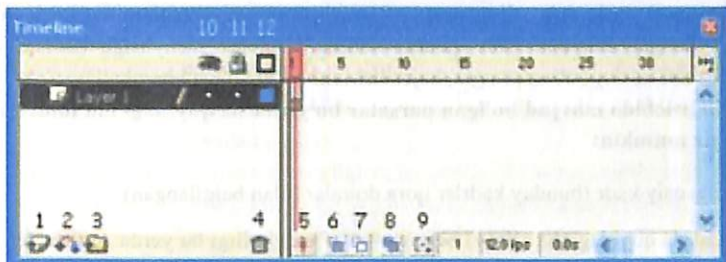
Hozirgi vaqtda turli xil animatsiya yaratish texnologiyalari mavjud:

1. **Klassik (an'anaviy)** animatsiya chizmalarning muqobil o'zgarishi bo'lib, ularning har biri alohida chiziladi. Bu juda ko'p vaqt talab qiladigan jarayon, chunki animatorlar har bir kadri alohida yaratishlari kerak.
2. **Stop-kadr** animatsiyasi. Ishchi maydonga joylashtirilgan ob'ektlar kadrqa o'rnatiladi, shundan so'ng ularning pozitsiyasi o'zgartiriladi va yana o'rnatiladi.
3. **Sprite** animatsiyasi dasturlash tili yordamida amalga oshiriladi.
4. **Morfing** - ma'lum miqdordagi oraliq kadrlarni hosil qilish orqali bir ob'ektni boshqa ob'ektga aylantirish.
5. **Rangli animatsiya** - ob'ektning o'rnini emas, balki faqat rangini o'zgartirganda.
6. **3D-animatsiya** maxsus dasturlar (masalan, 3D MAX) yordamida yaratiladi. Rasmlar sahnani ko'rsatish orqali olinadi va har bir sahna ob'ektlar, yorug'lik manbalari, teksturalar to'plamidir.
7. **(Motion Capture)** Harakatni suratga olish - animatsiyaning birinchi yo'nalishi bo'lib, u real vaqtda tabiiy, real harakatlarni yetkazish imkonini beradi.

## 2. Vaqt diagrammasi – Timeline

**(Window->TimeLine)** vaqt diagrammasi. Bu panel Flash film yaratish va tahrirlash uchun asosiy panel hisoblanadi. Flash ko'rinishini birinchi marta yaratishni boshlaganingizda

, birinchi sahnaning vaqt diagrammasi (Scene 1) bo'sh (oq) kadrlar bilan bitta qatlamni (Layer 1) o'z ichiga oladi.



4.2.1-rasm. Vaqt diagrammasi(Timeline).

Vaqt diagrammasi tugma-piktogrammalari

1. Qatlam qo'shish.
2. Yo'naltiruvchi qatlamini qo'shish.
3. Qatlamlarni jildga birlashtirish.
4. Qatlamni o'chirish.
5. Kadrlni markazlashtirish.
6. Oldingi kadrlarni ko'rsatish (yarim shaffof).
7. Xuddi shunday, lekin faqat oldingi kadrlarning konturlari.
8. Bir vaqtning o'zida bir nechta kadrlarni tahrirlash.
9. Ko'rish yoqilgan paytda oldingi kadrlarni ko'rsatish sozlamalari.
10. Qatlam tarkibini namoish qilish - yashirish.
11. Disable - qatlamda tahrirlashni yoqish.
12. Qatlamdagi ob'ektlarning faqat konturlarini ko'rsating.

Agar ish maydonida grafik tarkib tahrirlanganda, vaqt diagrammasida aslida Flash film skripti yoziladi.

Vaqt diagrammasining asosiy qismlari:

**Chizg'ich(lineyka)** - u barcha kadrlarni o'sish tartibida ko'rsatadi. Har beshinchi kadr raqam bilan belgilanadi. Chizg'ichda qizil marker ko'rinadi;



4.2.2-rasm. Chizg'ich(Lineyka).

**marker** - oynada ko'rsatilgan joriy kadrda ishora qiladi. Kadrlar qatlamidagi istalgan kadrlni bosganingizda, marker avtomatik ravishda unga o'tadi va tezlik ko'rsatkichining chap tomonida

aniq kadr raqamini bilib olishingiz mumkin;

**vaqt diagrammasi ko'rsatkichi** – Ishchi sohasidagi Стрелка (Arrow) vositasi bilan bir xil funktsiyani bajaradi. U bilan keyingi ishlash uchun kadr tanlaydi.






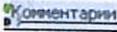
4.2.3-rasm. Marker.



**Qatlam(layer)lar** - vaqt diagrammasining chap tomonida qatlamlar ro'yxati mavjud. Uning ostida qatlamlarni qo'shish, olib tashlash, papkalarini yaratish imkonini beruvchi tugmalar mavjud; **kadrlar diagrammasi** - oddiy va asosiy kadrlarni qo'shish va olib tashlash mumkin bo'lgan maydon. Har qanday kadrda kontekst menyusiga (sichqonchanning o'ng tugmachasini bosib) chaqirsangiz, bajarilishi mumkin bo'lgan harakatlar ro'yxatini ko'rasiz.

**Kadr ichida mavjud bo'lgan narsalar bo'yicha siz quyidagi ma'lumotlarni olishingiz mumkin:**

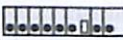
- asosiy kadr (bunday kadrlar qora doiralar bilan belgilangan)


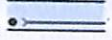
Kadrda quyidagicha alfa( $\alpha$ ) belgisi  mavjudligi bu yerda Action Script harakatlari mavjudligini bildiradi.  nishon (qizil bayroq va undan keyin nishon nomi)  yoki izoh  mavjud.

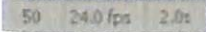
### Kadr rangi uning turini bildiradi

- bo'sh kadrlar og fonda ko'rsatiladi. Bo'sh asosiy(kalit) kadrlarda bo'sh doiralar mavjud. Flashda yangi film yaratganingizda, vaqt diagrammasi palitrasida siz bo'sh doira bilan belgilangan yagona kalit kadri ko'rasiz,

- kul rang - bu asosiy kadri (**keyframe**) aynan takrorlaydigan kadrlar , masalan, agar u fon bo'lsa.

- Kadrima-kadr animatsiyasi  ning, zangori-binafsha yoki yashil rangda

 ajratib ko'rsatishi, bu kadrlarda Flash yordamida yaratilgan animatsiya mavjudligini bildiradi: shakl animatsiyasi, harakat oralig'i , to'q ko'k rang kadr tanlanganligini bildiradi. agar biron sababga ko'ra oraliq animatsiya hosil bo'lmasa (masalan, kerakli asosiy kadr yo'q bo'lsa), u holda strelka o'mniga oraliq kadrlar orqali chiziqli chiziq o'tadi. Keyin fon rangi mo'ljallangan animatsiya turiga mos keladi.

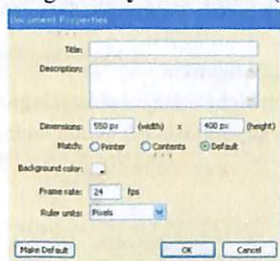
Axborot maydonlari: 

- **Current Frame** (Joriy kadr) - joriy kadrning raqamini ko'rsatadi;
- **Frame Rate** – (Kadr chastotasi) dialog oynasida o'rnatilgan filmning kadr tezligi Xususiyatlar **Movie Properties** (filmning xususiyatlari). Bu oyna asosiy menyudan ochiladi:

**Modify - Document** ( Ctrl -J);

- **Elapsed Time** (O'tilgan vaqt) - film boshidan joriy kadrgacha bo'lgan soniyalardagi vaqt. **Film vaqtlarini o'rnatish**

Filmi tahrirlash oynasini oching **Modify - Document** (Ctrl-J)



4.2.4-rasm. Hujjat xususiyatlari(Document Properties)paneli.

**Title va Description** maydonlarida film tavsifini kiritishingiz mumkin.

Sahna o'lchami (ish maydoni) - **Dimension (Wight - kenglik, Height - balandlik)**, shuningdek fon rangi (**Background Color**.)

**Match** kalitlari yordamida siz ish maydonining avtomatik o'lchamlarini belgilashingiz mumkin: bosma, kontent va sukut bo'yicha.

Kadr parametri **Rate (Kadr tezligi)** filmni ijro etish tezligi va sifatini belgilaydi. **fps** birligi ( frames per second - soniyada kadrlar).

**Ruler Units** maydonida sahnaning kengligi va balandligini o'lchash birliklarini o'rnatadi.

**Flash ko'rinishini o'ynatish** (. swf) - bu barcha bir qatlamli filmlarni bir vaqtning o'zida ketma-ket (o'tish skriptlari bo'lmasa) kadrma-kadr ko'rsatish.

Filmda bir nechta sahnalar (qismlar) bo'lishi mumkin. Ularning barchasiga, shuningdek, qatlamlarga nom berishingiz mumkin. Har bir sahnaning o'z vaqti bor.

### Kadrlar bilan ishlash

Kadr quyidagicha bo'lishi mumkin:

- qo'lda yaratilishi,
- yoki Flash dasturi tomonidan yaratilishi mumkin.

**Ularning farqlarini aniq tushunish muhimdir. Kadrlarning har xil turlari mavjud:**

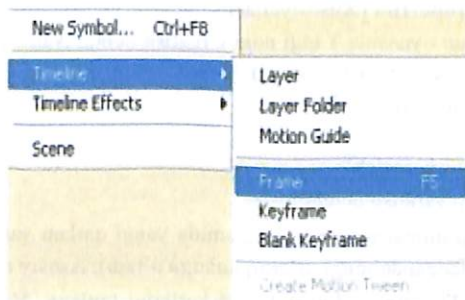
- bo'sh asosiy kadrlar (kulrang kadr),
- bo'sh asosiy kadrlar (oq maydondagi oq kadr),
- asosiy kadrlar (qora nuqta)
- va kadrlar ko'rinmayotgan bo'lishi mumkin (kadrlar shkalasi ko'rinadi).

Flash animatsiyasida muhim tushuncha mavjud - asosiy kadrlar (keyframes). Bu animatsiya jarayonida Flashni o'zgartirishga ruxsat berilmagan kadrlardir. Siz bu asosiy kadrlarni yaratasiz, Flash esa ular orasidagi oraliq kadrlarni yaratadi.

Animatsiyada ikki xil oraliq kadrlar mavjud:

- geometriya o'zgarishlariga asoslangan kadrlar (**shape tweening**) (och yashil kadrlar),
- o'zgaruvchan belgilarga asoslangan kadrlar (**motion tweening**) (och-pushti kadrlar).

Kadrlar bilan ishlash asosiy menyudan, Insert (Qo'shish) menyusidan amalga oshiriladi. Asosiy menyuda **Insert** (Qo'shish) tugmasini bosganingizda yoki klaviaturada Alt + I tugmalarini ketma-ket kiritganingizda, ochiladigan menyu paydo bo'ladi.



4.2.5-rasm. Qo'shish(Insert)menyusi ostki(sub)menyusi.

4.2.1-jadval. *Qatlamlar bilan ishlash uchun eng ko'p ishlatiladigan buyruqlar.*

Buyruqlar	Tugmalar	Amallar
<b>Kadre</b>	F5	Kadr kiritish (buning uchun vaqt diagrammasidan bo'sh kadrni belgilash kerak). Bir vaqtning o'zida bir nechta kadrlarni kiritish uchun Shift tugmasidan foydalanishingiz va bir nechta bo'sh kadrlarni tanlashingiz kerak, so'ngra F5
<b>Keyframe</b>	F6	Bir yoki bir nechta kadrlarni asosiy yoki asosiy kadrlarga aylantirish. Birinchi asosiy kadrning mazmuni boshqa barcha bo'sh joylarga ko'chiriladi.
<b>Blank Keyframe</b>	F7	Keyframe buyrug'iga o'xshaydi, lekin birinchi asosiy kadrning mazmunini boshqalarga ko'chirmaydi.
<b>Create</b>	yo'q	Oraliq jarayonlarni yaratishni boshlash
<b>Motion Tween</b>		twinning bilan to'ldirilgan bo'lsa (harakat, effektlar, masshtablash, aylantirish) yoki transformatsiya (shaklni buzish)

Agar vaqt diagrammasidagi istalgan kadrni sichqonchani o'ng tugmasi bilan bossangiz, u bilan bajarishingiz mumkin bo'lgan amallar ro'yxatini ko'rasiz.

Qatlamlar bilan ishlash uchun eng ko'p ishlatiladigan buyruqlar va qatlamlarning variantlari (parametrlari) vaqt diagrammasida piktogramma tugmachalari shaklida ko'rsatiladi.

### 3. Qatlamlar bilan ishlash

#### Qatlam nomini o'zgartirish

Qatlam nomini o'zgartirish mumkin. Buning uchun:


1. Sichqonchani chap tugmasi bilan qatlam nomini ikki marta bosib.
2. Ko'rsatilgan tahrirlash maydoniga yangi nom kiriting.
3. OK tugmasini bosib.

Yoki vaqt diagrammasida:

1. O'zgartirmoqchi bo'lgan qatlam nomini sichqonchani o'ng tugmasi bilan bosib.
2. Ochiladigan menyuda buyruqni tanlang (sichqonchani chap tugmachasini bosish orqali) **Properties** (Xususiyatlar).
3. Qatlam oynasida Yangi nom o'rnatish uchun Name (Nom) maydonidagi **Layer Properties** (qatlamning xususiyatlari).
4. OK tugmasini bosib.

#### Yangi qatlam yaratish

Yangi qatlam yaratish uchun:

- Vaqt diagrammasining yuqori qismida yangi qatlam yaratmoqchi bo'lgan qatlamni tanlang (qatlam tanlanganda rangi to'qroq holatga o'tadi). Asosiy menyudan
- **Insert - Timeline** - Layer ketma-ketligini tanlang. Vaqt diagrammasidagi Insert tugmasini bossangiz ham xuddi shunday bo'ladi  **Insert Layer** (qatlam qo'shish).

### Qatlamlarni sudrab olish

Qatlamlarni bir-biriga nisbatan siljitish, ularning mazmunini oldinga olib chiqish yoki aksincha uni yashirish mumkin. Tanlangan qatlamni sichqoncha yordamida Timeline'dagi qatlamlar ro'yxatida yuqoriga yoki pastga sudrab borish kifoya. Qatlamni sudrab borish uchun sizga vaqt diagrammasida kerak bo'ladi:

sichqonchani chap tugmasi bilan qatlam nomini bosib,

sichqoncha tugmasini bo'shatmasdan, qatlamni kerakli joyga (qatlamlar orasida) torting, qatlamni sudrab, chap sichqonchani qo'yib yuboring. tugmasi.

### Qatlamlarni o'chirish

Qatlamni o'chirish uchun:

Vaqt diagrammasida o'chirmoqchi bo'lgan qatlam nomini o'ng tugmasini bosib. Ochiladigan menyuda O'chirish buyrug'ini tanlang **Delete Layer** (qatlamni o'chirish). Xuddi shu narsani qilish mumkin, agar sichqonchani chap tugmasi bilan vaqt diagrammasida kerakli qatlamni tanlab, **Delete Layer** tugmasini bosib. Vaqt diagrammasining pastki qismidagi qatlam (qatlamni o'chirish).

### Qatlam xususiyatlari

Qatlam xususiyatlari Layer dialog oynasida aks ettiriladi va tahrirlanadi **Layer Properties** (Qatlam xususiyatlari).

Agar siz Vaqt diagrammasidagi qatlam nomini ustida sichqonchning o'ng tugmasini bossangiz, ochiladigan menyuda paydo bo'ladi. Ochiladigan menyuda buyruqni tanlang (sichqonchani chap tugmachasini bosish orqali) **Layer Properties** (Xususiyatlar).

Qatlam dialog oynasida **Layer Properties** (Qatlam xossalari) quyidagi qatlam parametrlari o'rnatiladi va aks ettiriladi:

**Name** (Ism) - bu maydonga qatlam nomi (nomi) kiritiladi,

**Show** (Ko'rsatish) - bu katak belgilansa, qatlam ko'rinadi,

**Lock** (Qulflash) - agar bu katak belgilansa, u holda qatlam qulflangan, ya'ni boshqa, qulfdan chiqarilgan qatlamda tahrirlash qulflangan qatlamga ta'sir qilmaydi.

Qatlamning mazmuni takomillashtirilganda va tahrirlashni talab qilmasa, qulflash tasviri ostidagi tanlangan qatlamga mos keladigan nuqtani bosish orqali qatlamni qulflang. Va agar qatlamning mazmuni sizning ishingizga xalaqit bersa, tanlangan qatlamdagi ko'zning tasviri ostida bu safar bosish orqali uni o'chiring.



4.3.1-rasm. Qatlam xususiyatlari paneli.

## 4. Flash dasturida kadrma-kadr animatsiya

**Animatsiya** - bu Macromedia Flash dagi tasvirlar (kadrlar) ketma-ketligi bo'lib, ketma-ket ko'rilganda ob'ektning harakati yoki xususiyatlarining o'zgarishi illyuziyasi paydo bo'ladi.

## **Flash dagi animatsiya turlari**

**Kadrma -kadr va hisoblangan Flash animatsiyasi**

**Hisoblangan animatsiya ikki usulda amalga oshirilishi mumkin:**

**Shakl animatsiyasi - Tween Shape**

**Harakat animatsiyasi - Tween Motion**

Bu usullarning tubdan farqi shundaki, birinchi usulda Flash grafik ko'rinishlarar (chiziqlar, yo'ylar, doiralalar, to'rtburchaklar, plomba va boshqalar) to'plami sifatida grafik bilan ishlaydi, ikkinchi holatda esa Flash faqat guruhlangan ob'ektlar yoki belgilar bilan ishlaydi.

Multifilmlar yaratishda kadrma-kadr animatsiyasidan foydalanish muqarrar. Bunday holda, siz va men shakl yoki harakatni o'zgartirish jarayonini aks ettiruvchi barcha kadrlarni chizishimiz kerak bo'ladi. Hisoblangan animatsiyadan foydalanilganda, biz faqat asosiy kadrlarni chizishimiz kerak bo'ladi va asosiy kadrlar orasida joylashgan kadrlardagi tasvir Flash tomonidan dasturiy ravishda yaratiladi, bu bizning ishimizni sezilarli darajada osonlashtiradi.

Hisoblangan animatsiyani yaratishda biz manipulyatsiya qilishimiz mumkin bo'lgan elementar o'zgarishlarga quyidagilar kiradi:

- koordinatalarni,
- o'lchamlarni,
- masshtabni,
- aylanish burchagini,
- shaklini,
- konturning rang atributlarini va ob'ektni to'ldirishni o'zgartirish.

Animatsiya qilishning ikkita printsiplal jihatdan bir-biridan farq qiladigan usuli mavjud:

1. Birinchisi, rastri grafiklarni har bir kadrqa chizish yoki import qilish va Flash dan tasvirlar bo'ylab tez aylanib chiqish, film tomosha qilish imkonini beruvchi vosita sifatida foydalanish. Bu kadrma-kadr animatsiya.

2. Ikkinchisi, Flashni asosiy tarkibga ega va bir-biridan istalgan miqdordagi kadrlar bilan ajratilgan asosiy kadrlar (keyframes) orasidagi oraliq kadrlarni avtomatik ravishda hisoblashga majburlash. Bu tweening (tweening) – animatsiya harakati, xususiyatlar yoki shakllanish animatsiyasi (shaping) - rang, shaffoflik, yorqinlik va boshqalar kabi ob'ektning shakli yoki xususiyatlarini o'zgartirish. Flash dasturidagi animatsiyaning o'zini shartli ravishda bir necha turlarga bo'lish mumkin:

1. Kadrlar bo'yicha animatsiya
2. harakatining hisoblangan animatsiyasi
  - to'g'ri chiziqda
  - traektoriya bo'ylab
3. Obyekt xossalari hisoblangan animatsiyasi (rang, shaffoflik, yorqinlik) 4.

Hisoblangan animatsiya shaklidagi transformatsiya.

Animatsiya yaratish uchun siz vaqt diagrammasi ( timeline ), qatlamlar (layers), kadrlar ( frames) va belgilar ( symbols) bilan ishlashni bilishingiz kerak.

## **Afzalliklari:**

1. Kadrma-kadr animatsiya ma'lum ma'noda animatsiyani ko'proq boshqarish imkonini beradi. Bu har qanday o'zgarishlarni aniqroq takrorlash imkonini beradi.

2. Bu butunlay mustaqil tasvirlarni o'zgartirishni tashkil qilishning yagona yo'li - slyadshou (masalan, Flash yordamida muntazam banner yaratish ).

3. Va boshqa hamma narsa, buning uchun har bir kadрни qo'lda chizish kerak bo'ladi.

#### **Kamchiliklari:**

- Kadrlar bo'yicha animatsiyani o'zgartirish qiyin va ko'p vaqt talab etadi. Ayniqsa, bu diskret tasvirlar to'plami emas, balki bog'langan animatsiya bo'lsa. Barcha kadrlarni o'zgartirishingiz kerak.

- Kadrlar bo'yicha animatsiya juda katta hajmni egallaydi, chunki har bir kadr haqida ma'lumot saqlashingiz kerak. Kadrlar animatsiyasi - bu butunlay asosiy kadrlardan tashkil topgan animatsiya. Unda siz o'zingiz kadrning mazmunini ham, uning "davomiyligini" ham aniqlaysiz (ya'ni, tasvir qancha shunday statik kadrlarni egallashi).

### **5. Hisoblangan animatsiya. Motion Tweening animatsiyasi**

Ushbu animatsiya usuli yordamida Flash siz ko'rsatgan asosiy kadrlar o'rtasida avtomatik o'zaro o'zgarishlarni amalga oshiradi.

Siz ob'ektni chizasiz, so'ngra boshqa asosiy kadrda o'zgarishlar kiritasiz va Flash ushbu ikkita asosiy kadrlar orasida joylashgan kadrlarni hisoblab chiqadi va siz silliq animatsiyaga ega bo'lasiz.

Hisoblangan animatsiyalarning ikki turi mavjud:

twinning ( Tweening - harakat) va shakllantirish ( Shaping - shakl).

Twinning ( Tweening )dagi ob'ektni o'zgartirish deganda biz buni tushunamiz:

ko'chirish,

masshtablash,

aylantirish,

buzish.

Ob'ektning konturlarida tub o'zgarishsiz (faqat konturlarning xarakterli nuqtalarining koordinatalari o'zgaradi, ularning soni o'zgarmaydi). Transformatsiyaning o'zgarishi (shakllanishi ) konturlar tizimidagi tub o'zgarish sifatida tushuniladi. Animatsiya tezligi va silliqliqi quyidagilarga bog'liq: o tahrirlash uchun ajratilgan asosiy kadrlar orasidagi hisoblangan kadrlar soni, o Flash filmingiz (movie) tezligi.

#### **Motion Tweening animatsiyasi**

Bu yerda **Motion Tweening** dan foydalanganda o'zgartirilishi mumkin bo'lgan belgi parametrlari keltirilgan:

- o'lcham (ham mutanosib, ham proporsional bo'lmagan - alohida balandlik va kenglik),
- egilish,
- joylashish,
- aylanish burchagi,  rang effektlari,
- ob'ektning harakat yo'lini o'rnatish uchun hidoyat qatlamlaridan foydalanishingiz mumkin.

**Motion Tweening** ni yoqish bir necha usul bilan amalga oshirilishi mumkin.

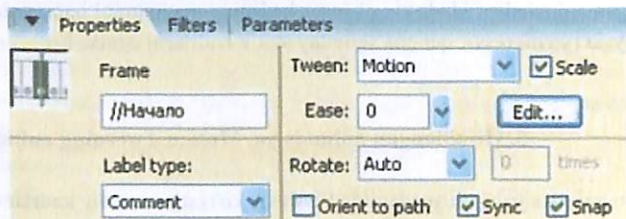
**Motion Tweening** yoqish uchun siz o'tishning boshlang'ich kadrini faol qilishingiz kerak, keyin sichqonchani o'ng tugmachasini bosib va kontekst menyusidan **Create Motion Tween**

(hisoblangan animatsiya yaratish)

Xuddi shu narsani asosiy menyusidan **Insert - Timeline - Create Motion Tween** (hisoblangan animatsiya yaratish) ni tanlash orqali amalga oshirish mumkin.

Keling, hisoblangan animatsiya uchun **Properties** panelini batafsil ko'rib chiqaylik.

- Frame - kadr nomi, izohlar...
- Label Type - kadr turi (nom, izoh...)
- Tween - animatsiya turi

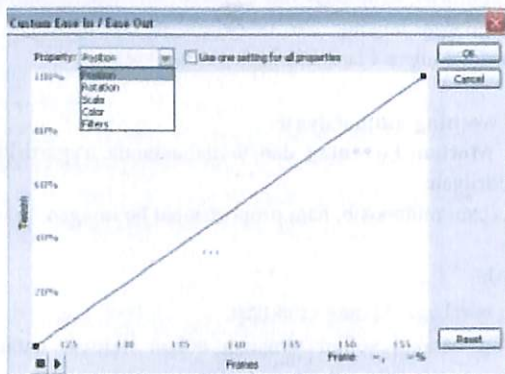


4.5.1-rasm. Animatsiya uchun xususiyatlar paneli.

- Ease – tezlik
- Scale - masshtabni silliq o'zgartirish
- Rotate - soat yo'nalishi bo'yicha yoki soat yo'nalishiga qarshi aylantirish
- Orient to Path – mo'ljal bo'ylab yo'naltirish
- Sync - ob'ektga bog'lash
- Snap - ob'ektga bog'lash

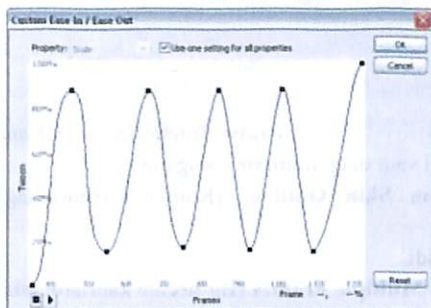
### Harakat silliqligini tahrirlash

Shuni ta'kidlashimiz lozimki, faqat harakat tezligi va ob'ektning pozitsiyasi a va b nuqtalari orasida qat'iy tartibga solinadi. Shunday qilib, (keyframe parametrli) Ease yonida (asosiy kadr parametri) Ease (Edit...) tugmasi mavjud bo'lib, u Ease imkoniyatlarini kengaytiradi:



4.5.2-rasm. Harakat tezligi va ob'ektning pozitsiyasi.

X shkalasi - keyingi asosiy kadrda keyingi kadrlar, Y shkalasi - har bir parametr uchun yo'lning foizi (Position, Rotation, Scale, Color, Filters) siz o'zingizning egri chiziqlaringizni o'rnatishingiz mumkin.



4.5.3-rasm. X va Y shkalasi bo'yicha egri chiziqlarni o'rnatish..

### Bir nechta ob'ektlarning animatsiyasi

Eng muhimi. Agar filmingiz bir nechta harakatlanuvchi ob'ektlardan foydalansa, har bir ob'ekt uchun alohida qatlamdan foydalanishingiz kerak. Agar chizish paytida siz tasodifan bir qatlamda bir nechta ob'ektlarni chizgan bo'lsangiz va keyin ularni bir nechta qismlarga bo'lish kerakligi aniqlansa, tarqatish buyrug'idan foydalaning. Distribute To Layers (Obyektni kontekst menyusida sichqonchani o'ng tugmasi bilan bosish orqali tanlang,

Agar siz bir nechta ob'ektni jonlantirishingiz kerak bo'lsa, bu ob'ektlarning har biri alohida qatlamda bo'lishi kerak!

## 6. Traektoriya bo'ylab animatsiya. Yo'naltiruvchi qatlamlar.

Yo'naltiruvchi qatlamlari (Guide layers) ob'ekt harakatlanishi kerak bo'lgan egri chiziqni (traektoriyani) o'z ichiga olgan qatlamlardir.

Har qanday qatlamni o'z xususiyatlarida ko'rsatish orqali yo'riqnoma yoki kerakli qatlamni sichqoncha bilan yo'riqnoma ostida bo'lishi uchun sudrab yo'riqnoma qilish mumkin.

Traektoriya bo'ylab harakatni yaratish texnologiyasi quyidagicha:

1. Yangi fayl yaratish ( Ctrl -N).

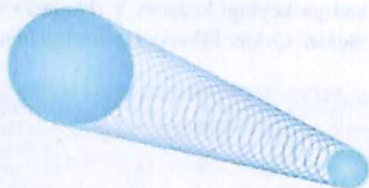
**Ishchi maydonida:**

2. Ish maydonining yuqori chap burchagida aylana chizing va uni radial gradient bilan to'ldiring (birinchi kadrda asosiy kadr paydo bo'ladi).

**Vaqt diagrammasida:**

3. 30 ta kadr uchun oddiy hisoblangan animatsiya yarating:
  - Birinchi kadr tanlangan bo'lishi kerak, agar bo'lmasa - uni tanlang. Asosiy menyudan Insert - Timeline - Create Motion Tweenni tanlang (yoki kontekst menyusida). Endi kadr och binafsha rangda.
  - 30-kadrda asosiy kadr yarating.
4. Agar u tanlanmagan bo'lsa, 30-tugma kadsini tanlang va ish joyida **Arrow** vositasi yordamida to'pni pastga va o'ngga siljiting. Agar kerak bo'lsa, uni Scale modifikatori yordamida kattalashtirish yoki **Properties** (Xususiyatlar) paneli yordamida burish mumkin.





4.6.1-rasm. Traektoriya bo'ylab animatsiya

Tugmalardagi vaqt diagrammasini bosganingizda

- **Onion Skin Outlines** (Kontur ko'rinishidagi qo'shni kadrlar) - qo'shni kadrlarning konturlari ko'rsatiladi,

- **Edit Multiple Frames** (Bir nechta kadrlarni tahrirlash) - bir vaqtning o'zida bir nechta kadrlarni tahrirlash rejimi - o'rnatilgan ob'ekt holatlari ko'rinadi - to'p va konturlar - egizaklashdan foydalanish bilan ta'minlangan barcha hisoblangan holatlar (ochiq binafsha fonda qora o'q)

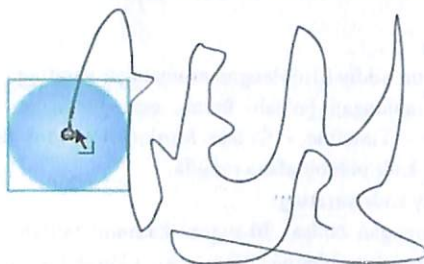
5. Yo'naltiruvchi qatlamlarni qo'shish uchun siz o'zingizning belgingiz (to'p) joylashgan qatlamni tanlashingiz kerak va sichqonchaning o'ng tugmachasini bosib, ochiladigan menyuda **Add Motion Guide** (yo'riqnoma qatlamini qo'shish) -ni tanlang. Bunday holda, to'p bilan boshlang'ich qatlam boshqariladi (guided layer). Bundan tashqari, qatlamlar ostidagi tugma yordamida yo'naltiruvchi qatlamini qo'shishingiz mumkin (qatlamlar haqidagi mavzuga qarang).

6. Endi Guide qatlamida siz harakat traektoriyasini chizishingiz kerak. Yo'l to'ldirish maydoni bo'lmagan har qanday egri chiziq bo'lishi mumkin.

7. Ushbu qatlamdagi birinchi kadrni tanlang va ish maydonida istalgan egri chiziqni chizing. Tekshirish qatlami tayyor. Ishlash uchun qulayroq qilish uchun uni tahrirlash uchun o'chirib qo'yishingiz mumkin va kelajakda uni butunlay ko'rinmas holga keltiring.

8. Traektoriya bo'ylab animatsiya ishlashi uchun siz belgini yo'lga bog'lashingiz kerak. Buni amalga oshirish uchun siz (sichqonchaning chap tugmasi bilan) belgini markaziy nuqtaga olib, uni yo'lga sudrab olishingiz kerak.

9. Belgida kichik doirani ko'rishingiz bilanoq, bu uning yo'lga "ilgaklanganini" bildiradi. Belgisi unga "ilgaklanganini" his qilasis va u qanday sirpanishini ko'rasiz. 30-kadrdagi ikkinchi kalit kadr uchun ham xuddi shunday qilish kerak - belgini yo'lning oxiriga yopishtiring. Agar siz ob'ektning yo'l bo'ylab aylanishini va shunchaki uning bo'ylab harakatlanishini istasangiz, **Properties** panelida **Orient to path** katagiga belgi qo'yishingiz kerak.



4.6.2-rasm. Traektoriya bo'ylab animatsiya hosil qilinishi.

10. Va belgining yo'lga bog'langanligini tekshirish uchun tugmachalarni bosing - - **Onion Skin Outlines** (Kontur ko'rinishidagi qo'shni kadrlar) va - **Edit Multiple Frames** (Bir nechta kadrlarni tahrirlash). Agar yo'l bo'ylab ramz konturlarini ko'rsangiz, u holda ulanish muvaffaqiyatli bo'ldi. Agar yo'q bo'lsa, ushbu misolning 8-bandidan qayta urinib ko'ring.

11. Animatsiyangizni sinab ko'ring.

12. Agar siz ob'ektning shunchaki u bo'ylab harakatlanishini emas, balki yo'l bo'ylab aylanishini istasangiz, u holda **Properties** panelida **Orient** to pathkatagiga belgi qo'yishingiz kerak.

## 7. Nazorat savollari



1. "Animatsiya"ga ta'rif bering
2. Animatsiya yaratish uchun qanday texnologiyalar mavjud?
3. Vaqt diagrammasi qanday boshqaruv elementlaridan iborat?
4. Hisoblangan animatsiya deganda nimani tushunasiz?
5. Macromedia Flashda animatsiyaning qanday turlari mavjud?
6. Motion Tweening animatsiya texnologiyasi haqida gapirib bering.
7. Tracktoriya bo'ylab animatsiya yaratish texnologiyasi haqida gapirib bering.

## XULOSA

Ushbu bobda flash-texnologiyalar bilan ishlashning asosiy usullari bilan tanishtirildi. Bundan tashqari bobda Flash texnologiyasiga kirish, uning imkoniyatlari, afzalliklari va ko'lami bilan tanishtirildi.

Macromedia Flash Professional v8.0 ilovasining interfeysi bilan tanishtirildi va chizish asboblari panelini ko'rib chiqildi. Flash dasturining ishchi oynasini qanday sozlash kerakligi haqida ma'lumot berildi.

### 2.1. Actionscript dasturlash tilining imkoniyatlari va harakatlar paneli bilan ishlash haqida asosiy ma'lumotlar

#### Reja:

1. ActionScript haqida umumiy ma'lumot.
2. ActionScript terminologiyasi.
3. ActionScript ob'ekt modeli. 4. Skriptni bajarish konteksti.
5. ActionScript paneli.
6. Normal Mode va Expert Mode rejimi.
7. Nazorat savollari.

**Kalit so'zlar:** *ActionScript, ob'ekt, ob'ekt namunasi, sinf, ramka, belgi, harakat, hodisa, harakatlar paneli, vaqt jadvali, skript, OYD.*

#### 1. Actionscript haqida umumiy ma'lumot

**ActionScript** sizga klip, tugma yoki ramkani skript qilish imkonini beradi. Har bir bunday skript (ya'ni, aslida ActionScript dasturi) filmning mos keladigan elementiga ulangan. Filmni nashr qilganingizda, boshqa film elementlari kabi skript matni SWF fayliga eksport qilinadi. Ammo, agar xohlasangiz, uni alohida **.as faylida saqlashingiz mumkin** (bu oddiy matnli fayl), keyinroq boshqa filmda foydalanish uchun (yoki uni do'stingizga berish).

Skriptdan siz har qanday boshqa kino ob'ektiga (tugmalar, kliplar va ramkalar) va hatto boshqa filmga yoki ba'zi tashqi tarmoq manbalariga murojaat qilishingiz mumkin. Misol uchun, siz o'qish boshi 10-sonli ramkaga yetganda, [www.sound.ru](http://www.sound.ru) saytida joylashgan ovoz faylini yuklab olishingiz kerakligini belgilashingiz mumkin. Shunday qilib, Flash skriptdagi ob'ektlar qanday "aloqalar" borligini bilishi kerak. Ob'ektga noto'g'ri kirish, albatta, skriptda xatolikka olib keladi. Aytaylik, so'ralgan audio fayl ma'lum bir klipning 10-kadrdan ovoz chiqarilishi o'rniga, film xronologiyasining 10-kadrida ijro etiladi.

**ActionScript** - bu dastur kodiga asoslangan ob'ektga yo'naltirilgan dasturlash tili. Kod Flashga nima qilish kerakligini ko'rsatadigan ko'rsatmalardir. ActionScript ko'rsatmalari ActionScript dasturlash tilida yozilgan. Bir nechta ko'rsatmalarni ketma-ket yozsangiz, dastur kodini yaratasiz. Muayyan parametrlarni kiritish orqali kodingiz qismlarini boshqarishingiz mumkin.

Dastur kodi alohida matn fayliga yoki to'g'ridan-to'g'ri filmingizning kerakli ramkasiga (yoki ramkalariga) yozilishi mumkin. Filmingizda ActionScript uchun alohida qatlam yaratish yaxshi amaliyotdir. Ushbu qatlamdagi birinchi freymga barcha kodingizni yozishingiz kerak. Bunday qatlamni yaratgandan so'ng, siz Harakatlar panelini (kod) ochishingiz va ko'rsatmalar yozishni boshlashingiz mumkin. Ko'pgina ishlab chiquvchilar foydalanish qulayligi uchun ushbu qatlamni Xronologiya (Timeline) ning eng yuqori qismiga aylantiradilar.

Dastur kodi faqat matn formatida yozilganligi sababli, buning uchun Notepad kabi oddiy matn muharriridan foydalanish mumkin. Biroq, bu yondashuv bir qator kamchiliklarga ega:

- kodni ajratib ko'rsatish yo'q, bu ko'rsatmalarni o'qishni qiyinlashtiradi (Actionscript sintaksisi)

- avtomatik chiziqni yakunlash imkoniyati mavjud emas, bu esa ishlashni kamaytiradi.

Adobe ActionScript 3.0 ni qo'llab-quvvatlaydigan va flesh-ilovalar uchun professional rivojlanish muhitini ta'minlovchi ikkita dasturni tarqatadi: Dreamweaver va Flash Builder.

Dreamweaver-da siz alohida ActionScript 3 fayllarini yaratishingiz mumkin va sizga cheklangan darajada bo'lsa ham kodni ajratib ko'rsatish va satrni avtomatik yakunlash taqdim etiladi.

Flash Builder Dreamweaverga qaraganda ancha rivojlangan ActionScript OOP dasturidir. U qaysidir ma'noda Flash Professional-dan ustun bo'lgan kuchli IDE-ni taqdim etadi.

## 2. ActionScript terminologiyasi

Har qanday skript tillari singari, ActionScript ham muayyan sintaksis qoidalariga amal qiladigan maxsus terminologiyadan foydalanadi. Quyidagi ro'yxat eng muhim ActionScript atamalariga kirishni taqdim etadi.

**Actions** (buyruqlar) - klipga ijro paytida nima qilish kerakligini ko'rsatuvchi operatorlar. Masalan, gotoAndStop ijroni ma'lum bir kadrda o'tkazadi.

**Arguments** (argumentlar) deb ham ataladigan funktsiyalarga qiymatlarni o'tkazish imkonini beradi.

**Classes** (sinflar) - yangi turdagi ob'ektlarni aniqlash uchun yaratilishi mumkin bo'lgan ma'lumotlar turlari. Ob'ektlar sinfini aniqlash uchun konstruktor funktsiyalari yaratiladi.

**Constants** (konstantalar) - o'zgarmas elementlar. Masalan, Math.PI doimiyi har doim ma'lum bir qiymatga ega. Konstantalar qiymatlarni solishtirish uchun foydalidir.

**Constructors** (konstruktorlar) - sinfning xossalari va usullarini aniqlash uchun ishlatiladigan funktsiyalar.

**Data types** (ma'lumotlar turlari) - qiymatlar va operatsiyalar to'plami. Satrlar, raqamlar, haqiqiy va noto'g'ri ( mantiqiy) qiymatlar, kliplar ActionScript ma'lumotlar turlaridir.

**Events** - klipni ijro etish paytida sodir bo'ladigan harakatlar. Misol uchun, klip yuklanganda, player freymni o'ynaganda, foydalanuvchi klipdagi tugmani bosganda yoki klaviaturadagi tugmachani bosganda turli hodisalar sodir bo'ladi.

**Expressions ( iboralar )** - qiymatlarga ega bo'lgan bayonotlarning har qanday qismlari. Masalan,  $2 + 2$  ifodadir.

**Functions** argumentlar (parametrlar) uzatilishi mumkin bo'lgan va qiymatlarni qaytarishi mumkin bo'lgan qayta ishlatiladigan kod bloklari. Masalan, getProperty funksiyasiga xususiyat nomi va klip nomi uzatiladi va bu xususiyat qiymatini qaytaradi. getVersion funksiyasi klip o'ynatilyotgan Flash Player versiyasini qaytaradi.

**Handlers** - mouseDown yoki yuklash kabi hodisalarni boshqaradigan maxsus buyruqlar.

**Identifiers** (identifikatorlar) - o'zgaruvchilar, xususiyatlar, ob'ektlar, funktsiyalar va usullarga murojaat qilish uchun ishlatiladigan nomlar. Birinchi belgi harf, pastki chiziq ( `_` ) yoki dollar belgisi ( `$` ) bo'lishi kerak. Har qanday keyingi belgi harf, pastki chiziq ( `_` ), raqam yoki dollar belgisi ( `$` ) bo'lishi kerak. Masalan, o'zgaruvchining nomi firstName.

**Instances** (instances) - ma'lum bir sinfga tegishli ob'ektlar. Sinfning har bir namunasi shu sinfning barcha xossalari va usullarini o'z ichiga oladi. Barcha kliplar MovieClip sinfning xususiyatlariga (masalan, `_alpha` va `_visible`) va usullariga (gotoAndPlay va getURL kabi) ega misollardir.

**Instance names** - skriptlarda klip nusxasini belgilash imkonini beruvchi noyob nomlar. Masalan, kutubxonadagi asl belgi hisoblagich deb nomlanishi mumkin va uning ikkita nusxasi scorePlayer1 va scorePlayer2 deb nomlanishi mumkin.

**Keywords** (kalit so'zlar) maxsus ma'noga ega zahiralangan so'zlardir. Masalan, var - mahalliy o'zgaruvchilarni e'lon qilish uchun ishlatiladigan kalit so'z.

**Methods** (usullar) - ob'ektga xos bo'lgan funktsiyalar. Funktsiya aniqlangandan so'ng, uni ushbu ob'ektga usul sifatida chaqirish mumkin.

**Objects** (ob'ektlar) - har biri o'z nomi va qiymatiga ega bo'lgan xususiyatlar to'plami. Ob'ektlar har xil turdagi ma'lumotlarga kirishni ta'minlaydi. Masalan, oldindan belgilangan Date ob'ekti tizim soatidan ma'lumot beradi.

**Operators** (operatorlar) - bir yoki bir nechta qiymatlardan yangi qiymatlarni hisoblash. Masalan, qo'shish operatori (+) yangi qiymat olish uchun ikki yoki undan ortiq qiymatlarni birga qo'shadi.

**Properties** - ob'ektni belgilaydigan atributlar. Misol uchun, \_visible barcha kliplarning xususiyati bo'lib, u klipning ko'rinadigan yoki ko'rinmasligini aniqlaydi.

**Variables** (o'zgaruvchilar) - har qanday turdagi ma'lumotlarning qiymatini o'z ichiga olgan identifikatorlar. O'zgaruvchilar yaratilishi, o'zgartirilishi va yangilanishi mumkin. Ulardagi qiymatlar skriptda foydalanish uchun so'ralishi mumkin.

### 3. ActionScript ob'ekt modeli

Ob'ekt modeli - bu skriptda ishlatilishi mumkin bo'lgan ob'ektlar turlari to'plami va ular orasidagi bo'ysunish munosabatlari.

Eslatib o'tamiz, dasturlash tillarida ob'ekt atributlar (xususiyatlar) to'plami va ushbu ob'ektga qo'llanilishi mumkin bo'lgan usullar (protseduralar) ro'yxati bilan tavsiflanadi. Har bir ob'ekt sinfining o'ziga xos atributlari va usullari mavjud. Masalan, Button obyekti uchun atribut sifatida yorliq, geometrik o'lchamlar, koordinatalar, usul sifatida esa tugmani bosish reaksiyasidan foydalanish mumkin. Aniq ob'ekt mos keladigan sinfning namunasi. Xuddi shu sinfning ikkita misoli atribut qiymatlarida bir-biridan farq qilishi mumkin.

Ob'ektlardagi harakatlarni tavsiflash, shuningdek, ob'ektning bo'ysunishini ko'rsatish uchun odatda "nuqta belgisi" deb ataladi. Masalan, Button\_1 tugmasiga onPress() usuli qo'llanilishi kerakligini belgilash uchun konstruksiyadan foydalaning.

#### Button\_1.onPress().

Agar Button\_1 Clip\_2 ga tegishli ekanligini ko'rsatish talab qilinsa, yozuv quyidagicha ko'rinishi mumkin:

#### Clip\_2.Button\_1.

ActionScript-da ishlatiladigan ob'ekt modeli ko'p jihatdan boshqa skript tillarining ob'ekt modellariga (masalan, JavaScript) o'xshaydi. Biroq, ma'lum farqlar ham mavjud. Asosiysi, ActionScript-da ob'ektlar ierarxiyasi veb-brauzer oynasida ko'rsatilgan HTML hujjatiga nisbatan emas, balki Flash playerga nisbatan qurilgan.

Misol uchun, agar HTML hujjatida Input\_1 tugmasi bo'lgan Form\_1 shakli mavjud bo'lsa, JavaScript-da tugmaning "bo'ysunishi" quyidagicha tavsiflanadi:

#### Document.Window.Form\_1.Input\_1.

Bunday holda, agar tugma bir xil brauzer oynasida ko'rsatilgan joriy hujjatdan foydalanilsa, u holda ierarxiyaning dastlabki ikki darajasi aniq ko'rsatilmaydi, lekin nazarda

tutiladi. Shunga ko'ra, tugmachaga kirish uchun qurilishdan foydalanish mumkin **Form\_1.**  
**Input\_1.**

Xuddi shunday qoidalar ActionScript uchun ham amal qiladi. Masalan, to'g'ridan-to'g'ri asosiy film ramkasiga kiritilgan klipga murojaat qilish uchun siz quyidagilarni yozishingiz mumkin: **\_root.Clip\_1.**

**\_root** kalit so'zi asosiy film nomi sifatida ishlatiladi va bu holda uni o'tkazib yuborish mumkin. Agar klip boshqa "ota-ona" klipning bir qismi bo'lsa, **\_parent** kalit so'zi uning bo'ysunishini ko'rsatish uchun ishlatilishi mumkin, masalan:

**\_parent.Clip\_1.**

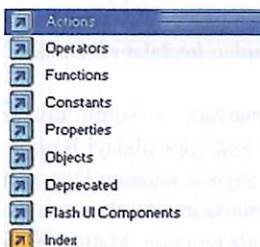
ActionScript-da ob'ektlarga murojaat qilish haqida qo'shimcha ma'lumot olish uchun keyingi bo'limga qarang.

Endi ActionScript obyekt modelining o'ziga qaytaylik.

Yuqorida aytib o'tilganidek, har qanday stsenariyda bevosita mavjud bo'lgan asosiy ob'ekt Flash playerdir. Keyingi bosqichda ob'ektlarning to'rtta sinfi mavjud:

- **Movie (Kino obyektlari);**
- **Core (yadro ob'ektlari);**
- **Client /Server (Mijoz-server ijro muhitining ob'ektlari);** □

**Authoring (ishlab chiqish muhiti ob'ektlari).**



1.3.1-rasm.ActionScript obyekt modeli.

**Movie** klassi ob'ektlari skriptda kino elementlari (ya'ni film tuzilishi) o'rtasidagi munosabatlarni tasvirlash hamda kino elementlarining xossalari va harakatini boshqarish imkonini beradi. Bu sinf, xususan, quyidagi ob'ektlarni o'z ichiga oladi:

- **Button** (tugma) - skriptdagi tugmani ifodalovchi ob'ekt; bunday ob'ekt uchun, masalan, kirish fokusini o'rnatish tartibi dinamik ravishda o'zgartirilishi mumkin;
- **MovieClip** (Clip) - skriptdagi klipni ifodalovchi ob'ekt; u dinamik ravishda o'zgarishi mumkin, masalan, pastki kliplar sonini;
- **Mouse** (Sichqoncha) - skriptdagi sichqoncha ko'rsatkichini ifodalovchi ob'ekt; bu sizga ko'rsatgichning ko'rinishi va uslubini boshqarish imkonini beradi;
- **Key** (Klaviatura) - skriptda filmni boshqarish uchun ishlatiladigan tugmachalarni ifodalovchi ob'ekt;
- **Color** (Rang) - skriptdagi kliplar palitrasini ifodalovchi va shunga mos ravishda ushbu palitrani o'zgartirish imkonini beruvchi ob'ekt;
- **Sound** (Ovoz) - skriptda tugma yoki klip bilan bog'langan tovush belgisini ifodalovchi ob'ekt;
- **Stage** (jadval) - jadvalning ba'zi parametrlarini (xususan, tasvir masshtabini) boshqarish uchun mo'ljallangan ob'ekt;

- **TextField** (Matn maydoni) - skriptdagi dinamik matn maydoni yoki tahrirlash maydonini ifodalovchi ob'ekt;

- **TextFormat** (Matn formati) - matn maydonining shrift sozlamalarini skriptdan boshqarish imkonini beruvchi ob'ekt.

ActionScript-da Button va MovieClip ob'ektlari o'rtasidagi bo'ysunish munosabatlarini aniqlash qiyin: ular bir filmdan ikkinchisiga va hatto bir sahnadan ikkinchisiga o'zgarishi mumkin. Masalan, bitta sahnada tugma klipning bir qismi bo'lishi mumkin, boshqa sahnada esa klip tugma tasvirini "jonlantirish" uchun ishlatilishi mumkin, Film sinfining qolgan ob'ektlari (mumkin bo'lgan istisnolardan tashqari). TextField ) Button va MovieClip obyektlariga bo'ysunuvchi deb hisoblanishi mumkin, chunki ular ularning bir qismi bo'lishi mumkin, lekin aksincha emas. **Izoh**

E'tibor bering, ActionScript na kadr, na kino sahnasi uchun mos ob'yekt sinfini taqdim etmaydi. Buning sababi, ActionScript ushbu film elementlarining biron bir xususiyatlarini o'zgartirishga ruxsat bermaydi. Skript yordamida siz faqat freymga (sahnaga) uning nomi yoki raqami orqali murojaat qilishingiz, shuningdek, uning yuklangan yoki yuklanmaganligini tekshirishingiz mumkin. Odatda, Harakatlar va usullar parametrlari sifatida ramka yoki sahnadan foydalaniladi. Masalan, 20-ramkaga o'tish uchun gotoAndPlay(20) amalini yozishingiz mumkin.

Core sinfining ob'ektlari kino obyektlari bilan ishlash uchun mo'ljallangan va yordamchi rolga ega. Kino ob'ektlarini manipulyatsiya qilish (xususiyatlarini o'zgartirish) uchun standart usullar etarli bo'lmaganda ulardan foydalanish kerak. Core sinfi, xususan, quyidagi ob'ektlarni o'z ichiga oladi:

- **Math** (matematika) - sonli qiymatlar bilan ishlash uchun skriptda foydalaniladigan ob'ekt; ActionScript tilidagi boshqa barcha sinflardan farqli o'laroq, Math ob'ekti ham ushbu sinfning yagona namunasidir; ya'ni bu sinfning usullari to'g'ridan-to'g'ri Math ob'ektiga qo'llanilishi mumkin; masalan, cos usuli yordamida kosinus qiymatini olish uchun quyidagilarni yozishingiz mumkin: Math.cos(3)

- **Number** (Raqam) - raqamli qiymatlar bilan ba'zi maxsus operatsiyalarni bajarish uchun taqdim etilgan ob'ekt; masalan, ma'lum diapazondagi eng katta raqamni aniqlash uchun ishlatilishi mumkin;

- **String** (Satr) - satrlar bilan ishlash uchun skriptda foydalaniladigan ob'ekt;

- **Date** (Sana)- kalendar sanalari va vaqtlari bilan ishlash uchun skriptda foydalaniladigan ob'ekt.

Client/Server sinfining ob'ektlari XML-da yozilgan hujjatlar (veb-sahifalar) bilan ishlash uchun mo'ljallangan. Ushbu ob'ektlarning ba'zilari XML hujjatlarini yuklash, qayta ishlash va jo'natish bilan shug'ullanadi, boshqalari esa rozetka ulanishlarini boshqaradi. Shunday qilib, ActionScript Flash filmlarni o'z ichiga olgan XML hujjatlarini server tomonida qayta ishlashni qo'llab-quvvatlaydi.

Authoring sinfining ob'ekti Flash filmlarni ishlab chiqish muhiti parametrlarini boshqarish uchun mo'ljallangan. Uning ikkita asosiy usuli bor: **o'rnatish** va **o'chirish**.

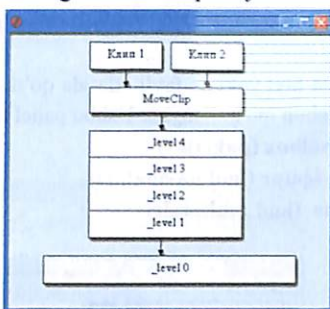
#### 4. Skriptni bajarish konteksti

Skriptni bajarish konteksti skriptda ishlatiladigan ob'ektlar va boshqa dastur qiymatlari (o'zgaruvchilari) mavjudligi va nisbiy manzillarini aniqlaydi.

Agar skript bir xil nomdagi bir nechta ob'ektlar yoki o'zgaruvchilardan foydalansa, kontekstni ko'rib chiqish zarurati paydo bo'ladi. Misol uchun, agar siz asosiy filmni ijro etish tezligini saqlash uchun framePS o'zgaruvchisini yaratsangiz va keyin ma'lum bir klip uchun bir xil nomdagi o'zgaruvchini yaratsangiz (bu shunday emas), keyin ertami-kechmi skript ishlamaydi. kutilgan.

Shu munosabat bilan ActionScript ning quyidagi xususiyatini hisobga olish kerak. Undagi o'zgaruvchining qamrovi hozirda faol bo'lgan vaqt diagrammasi bilan belgilanadi. Misol uchun, agar asosiy filmning 10-kadrlarida u bilan bog'langan skript mavjud bo'lsa, u holda ushbu skriptdagi barcha o'zgaruvchilar asosiy filmga emas, balki ushbu klipning vaqt rejimiga ta'sir qiladi. O'zgaruvchilarga havolada yuqori darajadagi vaqt diagrammasiga havola bo'lmasa (bunday havola qanday ko'rinishi mumkin, bu avvalgi bo'limga aytilgan edi).

Har bir vaqt jadvali u bilan bog'langan maxsus parametrga ega, vaqt jadvalining darajasi ( \_ darajadagi kalit so'zi bilan belgilanadi), bu uning Flash Player-ga yuklangan boshqa vaqt belgilariga nisbatan o'rnini belgilaydi. Odatiy bo'lib, asosiy film xronologiyasi nolga o'rnatiladi. Undan chaqirilgan har safar diagrammaga bir kattaroq daraja beriladi.



1.4.1-rasm. Flash playerga yuklangan vaqt diagrammalarining ierarxiyasi

Shunday qilib, Flash filmidagi o'zgaruvchilar doirasi yuqoridan pastgacha cho'ziladi: asosiy film vaqtini belgilash skriptida yaratilgan o'zgaruvchining nomi, agar u xuddi shu nomdagi o'z o'zgaruvchisiga ega bo'lmasa, keyingi darajadagi skriptda "ko'rinadi".

Bundan tashqari, jingalak qavsli skript blokida mahalliy o'zgaruvchilar faqat shu blok ichida "ko'rinadigan" deb e'lon qilinishi mumkin.

ActionScript o'zgaruvchilar doirasini boshqarish uchun "maqsadli yo'l" tushunchasidan foydalanadi. Belgilangan yo'l ob'ekt yoki o'zgaruvchining Flash Player-ga yuklangan ma'lum bir vaqt jadvaliga tegishli ekanligini aniqlash imkonini beradi.

Belgilangan yo'lni to'g'ri yozish uchun siz ActionScript ob'ekt modeli va vaqt diagrammalarining nisbiy o'rnini hisobga olishingiz kerak. Misol uchun, agar siz asosiy filmning 10-kadrasiga o'tish va uni o'ynashni xohlayotganingizni aniqlamoqchi bo'lsangiz, ushbu konstruksiyani yozishingiz mumkin:

#### **Level0.gotoAndPlay(1 0).**

Belgilangan yo'lni belgilashning ikkita varianti mavjud: mutlaq va nisbiy.

Mutlaq yo'l odatda vaqt diagrammasi darajasiga qarab hisoblanadi (yuqoridagi misolda bo'lgani kabi). Diagramma darajasini boshlang'ich nuqtasi sifatida belgilash orqali siz qo'ng'iroq qilingan skript fragmentini ko'chirsangiz ham, Flash Player manzilini to'g'ri topishiga ishonch hosil qilishingiz mumkin.



Nisbiy yo'l qo'ng'iroq qilingan diagramma pozitsiyasini hisobga olgan holda hisoblanadi. Ushbu parametr yanada ixcham, ammo kamroq ishonchli. Masalan, joriydan ikki daraja yuqorida joylashgan diagramma ramkasiga murojaat qilish uchun siz quyidagi konstruksiyani yozishingiz mumkin:

`_parent._parent.muClip.`

Biroq, klipni yuqori darajaga o'tkazgandan so'ng, masalan, nisbiy havola bekor bo'ladi.

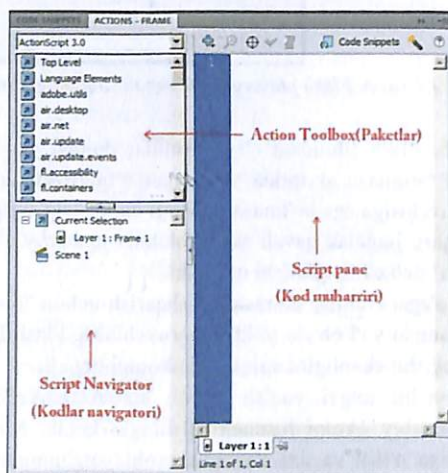
Shunday qilib, ActionScript-da skriptlarni yaratishda, hajmi bo'yicha farq qiluvchi uch turdagi o'zgaruvchilardan foydalanish mumkin:

- har qanday vaqt diagrammasi skriptida mavjud bo'lgan global (Global variables);
- Maqsadli yo'ldan foydalanish sharti bilan istalgan vaqt jadvalidan foydalanish mumkin bo'lgan vaqt jadvali (Global variables) o'zgaruvchilari.
- mahalliy (Local variables), ular faqat ular e'lon qilingan dastur blokida "ko'rinadigan".

## 5. ActionScript paneli

Harakatlar paneli Fla fayl yoki as-fayli sifatida qo'shadigan ActionScript kodini yozish, formatlash va tahrirlash uchun mo'ljallangan. Ushbu panel tizimli ravishda 3 qismga bo'lingan:

- **Actions Toolbox (paketlar)**
- **Script Navigator (kod navigatori)**
- **Script pane (kod muharriri)**



1.5.1-rasm. Actions paneli

**Actions** paneli ikkita ish rejimini taqdim etadi.

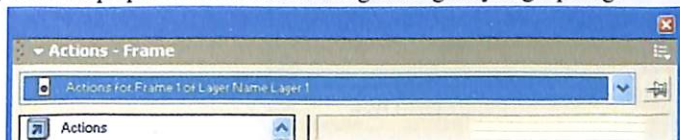
Oddiy rejimda (Normal Mode) harakatlar tavsifi mos keladigan parametr maydonlarini to'ldirish orqali amalga oshiriladi. Ekspert rejimida (Expert Mode) siz an'anaviy matn muharririda bo'lgani kabi, skript oynasida to'g'ridan-to'g'ri harakatlarni yozib olishingiz va tuzatishingiz mumkin.

Odatiy bo'lib, panelga birinchi marta kirganingizda Oddiy rejim o'rnatiladi.

### Actions panelini chaqirish

Actions panelini ko'rsatishning bir necha usullari mavjud. Biroq, eng ishonchli va to'g'ri yo'l - bu siz ssenariy yaratmoqchi bo'lgan film elementi "nomidan" panelni chaqirishdir.

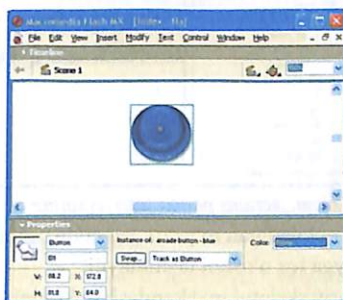
Kadrlar skriptini yaratish uchun Actions panelini ochish uchun tegishli kalit freymning katakchasidagi vaqt diagrammasi ustiga sichqonchani o'ng tugmachasini bosishingiz va kontekst menyusidan Actions buyrug'ini tanlashingiz kerak. Panel ekranda paydo bo'lgandan so'ng, siz yaratgan skript aynan siz xohlagan ramka bilan bog'lanishiga ishonch hosil qiling. Buning uchun faqat panel sarlavhasi va uning ostidagi ro'yxatga qarang.



1.5.2-rasm. Kadr uchun skript yaratishda Actions panelining sarlavhasi

Tugma yoki klip bilan bog'langan skript yaratish uchun Actions panelini ochish uchun:

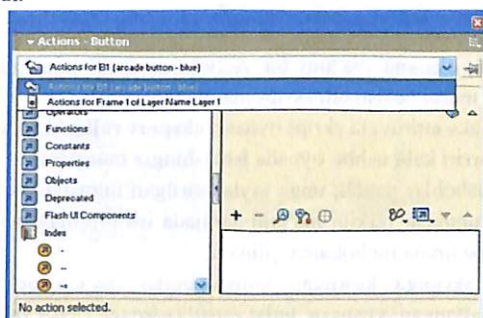
1. Jadvalda tegishli ob'ektni tanlang.
2. Xususiyatlar inspektori panelini oching.
3. Xususiyatlar inspektori panelining o'ng chegarasi yaqinida joylashgan **Edit the action script** tugmasini bosing.



1.5.3-rasm. Tugma yoki klip uchun harakatlar panelini chaqirish tugmasi

Ushbu kichik tugma qiziqarli xususiyatga ega: agar ushbu ob'ekt uchun skript bo'lmasa, u oq rangga ega, skript yaratildandan keyin u ko'k rangga aylanadi.

Tugma yoki klip uchun **Actions** panelini chaqirganingizda, uning sarlavhasida tegishli ma'lumotlar ko'rsatiladi.



1.5.4-rasm. Tugma uchun skript yaratishda Actions panelining sarlavhasi.

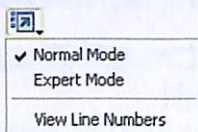
## Izoh

Skript kutubxonada saqlangan haqiqiy belgi bilan emas, balki faqat stolga joylashtirilgan tugma yoki klipning ma'lum bir nusxasi bilan bog'lanishi mumkin. Skriptni tugma yoki klip namunasiga tayinlash boshqa misollarning xususiyatlariga ta'sir qilmaydi.

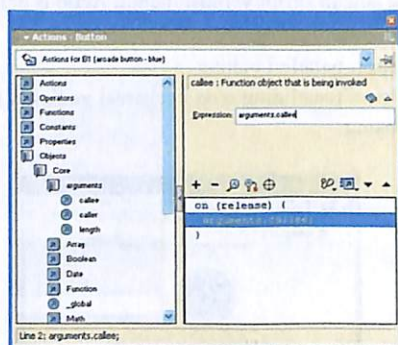
Panel ochilgandan so'ng siz ish rejimini tanlashingiz mumkin.

Rejimni tanlash bir xil nomdagi panel tugmasini bosish orqali ochiladigan **View Options (Format Setting)** menyusining tegishli buyruqlari yordamida amalga oshiriladi.

Tanlangan rejimdan qat'i nazar, panel quyidagi asosiy interfeys elementlarini o'z ichiga oladi:



1.5.5-rasm. Actions panelining ish rejimini tanlash menyusi.



1.5.6-rasm. Actions paneli interfeysining asosiy elementlari.

• film stsenariylarining ochiladigan ro'yxati; jadvalda ushbu stsenariy bog'langan ob'ektni tanlamasdan boshqa stsenariyga tez o'tishni ta'minlaydi;

• **Script Pin** tugmasi joriy skriptni mahkamlash imkonini beradi; biz panelning mazmuni stolda tanlangan kino ob'ektiga bog'liqligi haqida gapiramiz; skriptni tuzatgandan so'ng, siz stolda boshqa ob'ektni tanlashingiz mumkin va panelning mazmuni o'zgar olmaydi;

• **kitob ko'rinishidagi tugma ActionScript tili** uchun o'rnatilgan yordam tizimiga qo'ng'iroqni ta'minlaydi; Shuni ta'kidlash kerakki, bu yordam HTML yordamiga qaraganda ancha batafsilroq;

• **Action Toolbox** paneli (Harakatlar uchun asboblarning to'plami); ushbu pastki oynada taqdim etilgan har bir element ma'lum bir ActionScript tili konstruktsiyasiga mos keladi; element belgisini ikki marta bosish uni skript matniga kiritadi;

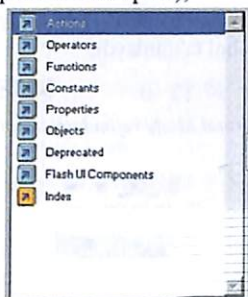
• skript matnini aks ettiruvchi skript oynasi; **ekspert rejimida Expert Mode** ishlaganda, siz oddiy matn muharriri kabi ushbu oynada ishlashingiz mumkin;

• skript oynasi asboblarning paneli; unga joylashtirilgan tugmalar skript matni bilan ishlash qulayligini oshiradi; tugmalar tarkibi ma'lum darajada ish rejimiga bog'liq; ularning maqsadi keyingi ikkita kichik bo'limda muhokama qilinadi;

• holat paneli; skriptga kiritilgan komponentlar haqida qo'shimcha ma'lumotlarni ko'rsatish uchun mo'ljallangan; xususan, holat satrida qisqacha xato xabari ko'rsatiladi;

• **Action Toolbox** ro'yxatida skript yaratishda foydalanish mumkin bo'lgan quyidagi ActionScript tili komponentlari ro'yxati mavjud;

• **Actions** (Harakat) – “universal” xarakterga ega bo'lgan, ya'ni obyektlar sinfi xususiyatlariga bog'liq bo'lmagan ActionScript tili protseduralari (muayyan sinfga mansub obyektlar usullaridan farqli o'laroq); ushbu turkumga shuningdek, ko'pgina yuqori darajadagi dasturlash tillari uchun an'anaviy mantiqiy boshqaruv operatorlari kiradi (mantiqiy shartlarni tekshirish, ttakrorlanishni tashkil qilish va boshqalar);



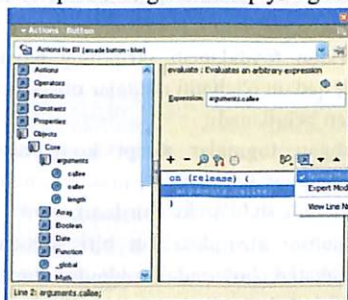
1.5.7-rasm. ActionScript tili komponentlarining toifalari.

- **Operators** (Operatorlar) - mantiqiy va arifmetik amallar (qo'shish, ayirish, taqqoslash va boshqalar) operatorlari;
- **Functions** (Funktsiyalar) - oldindan belgilangan (kutubxona) funktsiyalar (masalan, ular Flash playerning versiya raqamini tekshirish funktsiyasini o'z ichiga oladi);
- **Properties** (Xususiyatlar) - ob'ektlarning sinfga bog'liq bo'lmagan xususiyatlari (masalan, tasvirning shaffoflik darajasi);
- **Objects** (Ob'ektlar) - skriptda ishlatilishi mumkin bo'lgan oldindan belgilangan ob'ektlar (masalan, sana va vaqt bilan ishlash uchun ob'ekt mavjud); xuddi shu bo'lim moslashtirilgan ob'ekt sinflarini o'z ichiga oladi;
- **Deprecated** (Yaroqsizga chiqarilgan) - skriptlarda foydalanish tavsiya etilmaydigan konstruktsiyalar (eskirganligi uchun);
- **Flash UI Components** (Foydalanuvchi interfeysi komponentlari).

Ro'yxatdagi bo'limlarga qo'shimcha ravishda, **Action Toolbox ro'yxatida Indeks** bo'limi ham mavjud bo'lib, unda barcha til komponentlarining alifbo tartibida ro'yxati mavjud.

## 6. Normal Mode va Expert Mode rejimi

Ushbu rejimda ishlashda Action panelning formati quyidagi rasmda ko'rsatilgan.



1.6.1-rasm. Normal Mode rejimdagi Action paneli formati.

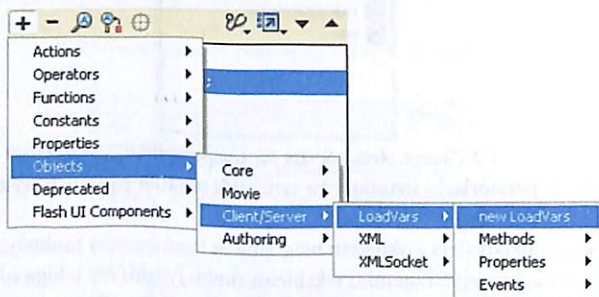
Parametrlarni kiritish maydoni skript matnida joylashtirilgan konstruksiyani to'g'ri bajarish uchun zarur bo'lgan parametrlarni belgilash uchun mo'ljallangan; hududning formati skript matnida tanlangan (ta'kidlangan) konstruksiyaning sintaksisiga bog'liq. Ba'zi konstruksiyalarda parametrlar umuman yo'q. Bunday holda, parametr kiritish maydoni bo'sh.

Skript oynasi asboblar panelida quyidagi tugmalar mavjud:

- **Add new item** tugmasi "+" belgisi bilan belgilangan (**Yangi element qo'shish**) ActionScript tili komponentlari ierarxiyasini aks ettiruvchi kaskadli menyular tizimiga qo'ng'iroq qilishni ta'minlaydi; eng past darajadagi menyular bandini tanlash tegishli komponentni skriptga kiritishni ta'minlaydi;



1.6.2-rasm. Normal Mode rejimidagi skript oynasi asboblar paneli.



1.6.3-rasm. Add new item tugmasi yordamida komponentni kiritish

- **"-"** belgisi bilan belgilangan tugma komponentlar oynasida tanlangan skriptni matndan olib tashlashni ta'minlaydi;
- **Find** tugmasi (Search) dialog oynasini ochadi, unda siz kerakli belgilar ketma-ketligini kiritishingiz mumkin;
- **Replace** tugmasi (almashtirish) satrni (belgilar ketma-ketligini) almashtirish parametrlarini o'rnatish uchun dialog oynasini ochishga imkon beradi;
- **Insert a target path** tugmasi kerakli film elementiga murojaat qilish uchun maqsad yo'lini tavsiflash mumkin bo'lgan dialog oynasini taqdim etadi; **Insert a target path** oynasining formati quyida muhokama qilinadi; bu tugma skript oynasida maqsad yo'li ishlatilishi kerak bo'lgan komponent tanlangan bo'lsa va qo'shimcha ravishda parametrlarni sozlash oynasining tegishli maydoniga kiritish fokasi o'rnatilgan bo'lsa, mavjud bo'ladi;
- **Debug Options** (Nosozliklarni tuzatish parametrlari) tugmasi to'xtash nuqtalarini (yoki to'xtash nuqtalarini) joylashtirish va o'chirish buyruqlarini o'z ichiga olgan menyuni ochadi; nazorat nuqtalaridan foydalanish skriptning bajarilishini bosqichma-bosqich kuzatish imkonini beradi; qadam o'lchami nuqtalar orasidagi masofa (nuqtalar orasidagi skript chiziqlari soni) bilan belgilanadi;
- strelkalar bilan belgilangan tugmalar skript komponentlarini bir-biriga nisbatan ko'chirishga imkon beradi (agar komponent, masalan, uchta qatordan iborat bo'lsa, u holda uchta qator bir marta bosish bilan ko'chiriladi).

Oddiy rejimning eng muhim afzalliklaridan biri shundaki, u qo'lda dasturlashdan foydalanmasdan skriptga Harakatlar bo'limidan oldindan belgilangan amallarni qo'shish imkonini beradi. Hammasi bo'lib, o'ndan ortiq bunday harakatlar mavjud:

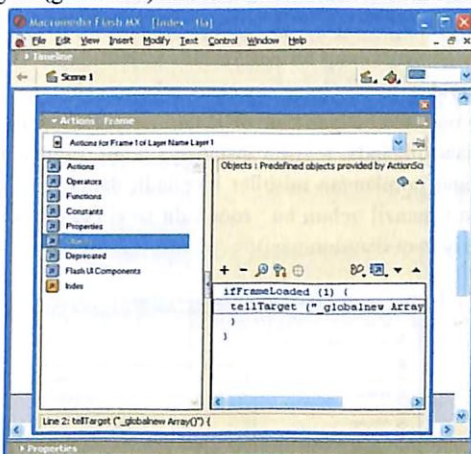
- Go To - berilgan kadr yoki sahnaga o'tish;
- Play va Stop - mos ravishda filmni o'ynatishni boshlash va to'xtatish;
- Toggle High Quality - filmni ijro etish sifatini nazorat qilish;
- Stop All Sounds - saundtrekni o'chirish; URLni olish - berilgan URL manziliga o'ting;
- FSCommand - filmni o'ynatish paytida Flash playerni boshqarish;
- Load Movie va Unload Movie – mos ravishda o'quvchi brauzeri orqali filmni yuklash va tushirish;
- Tell Target - Flash filmlar va kliplar o'rtasida navigatsiyani boshqarish.

Ushbu harakatlarning ba'zilari keyingi bo'limlarda muayyan turdagi ob'ektlar (kliplar, tugmalar va ramkalar) bilan bog'liq holda batafsilroq ko'rib chiqiladi.

Shunday qilib, Oddiy rejimda stenariy yaratish yoki unga biron bir komponent qo'shish uchun siz quyidagi amallarni bajarishingiz kerak:

1. Jadvalda skript bog'lanishi kerak bo'lgan tugma yoki klip (yoki vaqt jadvalidagi ramka) misolini tanlang.
2. Sichqonchaning chap tugmasi bilan Action Toolbox-da kerakli bo'limni oching.
3. Tanlangan elementni skript oynasiga element nomini yoki belgisini ikki marta bosish yoki shunchaki sichqoncha bilan sudrab ko'chirish; tegishli ActionScript skripti fragmenti oynada ko'rsatiladi.
4. Kerakli parametr qiymatlarini tegishli maydonlarga kiriting.

Tugma yoki klipga harakat tayinlaganingizda, Flash avtomatik ravishda ko'plab ob'ektga yo'naltirilgan tillarda "voqea ishlovchisi" (yoki "selektor") deb ataladigan maxsus boshqaruv protsedurasini (ishlab chiqaruvchi) yaratadi. Ishlovchi ma'lum bir turdagi hodisa sodir bo'lganligini tekshiradi va agar shunday bo'lsa, tegishli harakatlar ketma-ketligini boshlaydi. ActionScript-dagi ishlov beruvchining qo'shimcha roli ma'lum bir hodisa bilan bog'liq harakatlarni yig'ish (guruhlash) hisoblanadi. Ishlov beruvchi format rasmda ko'rsatilgan.



1.6.4-rasm. Hodisalarni qayta ishlash formati

Tugmalar uchun ishlov beruvchi on, kliplar uchun esa **onClipEvent** deb ataladi. Ishlovchi nomi ishlov beruvchi javob berishi kerak bo'lgan voqealar ro'yxatidan oldin keladi. Masalan, konstruktsiya on(release, KeyPress:"A") ishlov beruvchining ikki turdagi hodisalarga javob

berishi kerakligini bildiradi: bo'shatish (sichqoncha tugmasi bo'shatiladi) va KeyPress:"A" ("A" tugmasi bosiladi). klaviatura).

Umuman olganda, ActionScript tugmalari quyidagi turdagi hodisalarga ega:

- **Press** - kursor tugma (film) tasviri ustida turganda foydalanuvchi sichqoncha tugmasini bosadi.
- **Release** - foydalanuvchi sichqoncha tugmasini bosadi va keyin qo'yib yuboradi (ko'rsatgich tugma tasviri ustida joylashgan);
- **Release Outside** - Ko'rsatkich tugma tasviri ustida bo'lmaganda foydalanuvchi sichqoncha tugmasini bo'shatadi.
- **Key Press** - foydalanuvchi klaviaturada ma'lum bir tugmani bosadi; ushbu hodisa uchun siz qo'shimcha ravishda kalitni ko'rsatishingiz kerak;
- **Roll Over** - foydalanuvchi ko'rsatgichni tugma tasviri ustiga olib boradi;
- **Roll Out** - foydalanuvchi ko'rsatgichni tugma tasviridan tashqariga siljitadi;
- **Drag Over** - ko'rsatgich tugma ustida turganda foydalanuvchi sichqoncha tugmasini bosadi, uni tugmadan tashqariga olib chiqadi va yana tugma ustiga o'rnatadi;
- **Drag Out** - kursor tugma ustida turganda foydalanuvchi sichqoncha tugmasini bosadi va keyin uni tugmadan tortib chiqaradi.

Kino elementi javob berishi kerak bo'lgan voqealar ro'yxatini o'zgartirish uchun siz ishlov beruvchining nomini o'z ichiga olgan skript satrini bosishingiz kerak; bu parametrlar oynasining formatini o'zgartiradi va unda belgilash katakchalari guruhi paydo bo'ladi, ularning yordami bilan siz kerakli hodisalarni tanlashingiz mumkin (quyidagi rasimga qarang).

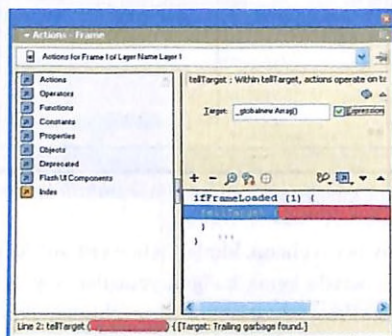
Tanlangan tanlov darhol skript oynasida ko'rsatiladi.

Agar skriptga qandaydir konstruktsiya qo'shish xatolikka olib kelsa yoki buning uchun kerakli parametr kerak bo'lsa, u holda konstruktsiya skript oynasida qizil rang bilan ajratib ko'rsatiladi; holat satrida bir xil konstruktsiya takrorlanadi.

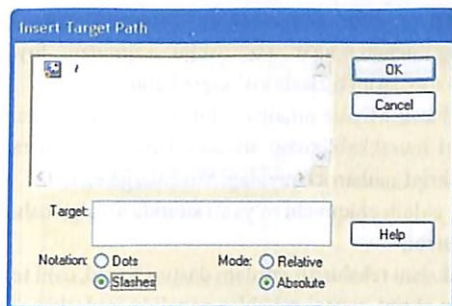
Ba'zi kino elementining maqsad yo'lini (ya'ni manzilini) tasvirlash mumkin bo'lgan "Maqsadli yo'lni qo'shish" oynasining formati rasmda ko'rsatilgan.

Ushbu oynada quyidagi asosiy interfeys elementlari mavjud:

- skriptdan kirish mumkin bo'lgan kino ob'ektlari ierarxiyasini aks ettiruvchi kino daraxti; chunki ob'ektga murojaat qilganda, tegishli instansiya nomi (tugmacha, klip, matn maydoni) ishlatiladi, daraxtga faqat nomlangan misollar kiritiladi; daraxtning ildiz elementi adreslash usuliga bog'liq: absolyut manzil uchun bu `_root` kalit so'zi (ya'ni asosiy film), nisbiy manzil uchun bu so'z (ya'ni joriy vaqt diagrammasi);



1.6.5-rasm. Noto'g'ri konstruktsiyani belgilash



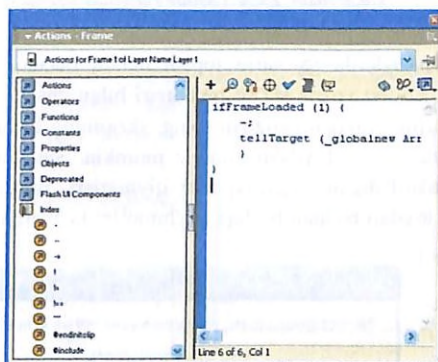
1.6.6-rasm. Insert a target path oyna formati

- ko'p qatorli matn maydoni Target (Mo'ljal), maqsad yo'lini kiritish va sozlash uchun mo'ljallangan; agar siz kino daraxtida biron bir elementni tanlasangiz, u holda Target maydonida mos keladigan marshrut paydo bo'ladi, uni qo'lda o'zgartirish yoki qo'shish mumkin;
- adreslash usulini tanlash imkonini beruvchi bir juft Mode kalitlari (nisbiy - Relative yoki mutlaq - Absolyut); sukut bo'yicha nisbiy adreslash;
- Belgilangan manzilni kiritish formatini tanlash imkonini beruvchi bir juft Notation radio tugmalari:
  - **Dotes** opsiyasi nuqta belgilaridan foydalanishni bildiradi (ActionScript ob'ekt modeli kichik bo'limiga qarang);
  - **Slashes** opsiyasi qiya chiziq yordamida belgilanishga mos keladi (bu "eskirgan" uslub ActionScript ning oldingi versiyalari bilan moslik uchun ishlatiladi); slashdan foydalanganda, yozish formati HTML hujjatlarida URL-manzillarni yozish uchun ishlatiladigan formatga o'xshaydi; masalan, bir darajaga ko'tarilish uchun ../ dan foydalaning.

### Expert Mode rejimi

Ekspert rejimi skript yaratishning ikkita usulini birlashtirish imkonini beradi: ActionScript tili konstruksiyalarini klaviaturadan bevosita kiritish va ularni Actions Toolbox ro'yxatidan skript oynasiga joylashtirish.

Ekspert rejimida ishlaganda parametr oynasi ko'rsatilmaydi. Ushbu rejimda skript qatorlarini o'chirish va ko'chirish tugmalaridan ham foydalanish mumkin emas:



1.6.7-rasm. Expert Mode rejimida Action paneli formati



Ekspert rejimi birinchi navbatda ActionScript dasturlash bilan tanish bo'lganlar uchun mo'ljallangan. Shuning uchun, agar siz ushbu rejimdan foydalanmoqchi bo'lsangiz, ActionScript dasturlash asoslarini o'zlashtirishingiz kerak.

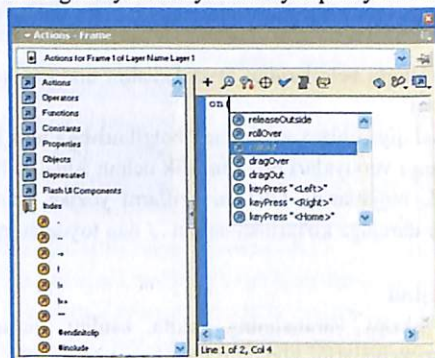
Shu bilan birga, Flash MX-da amalga oshirilgan foydalanuvchilarni qo'llab-quvvatlash darajasi turli darajadagi murakkablikdagi stsenariylarni mustaqil ravishda yaratishga imkon beradi. Buning uchun skript muharriri quyidagi vositalarga ega:

- parametrlarning qalqib chiquvchi ro'yxati sifatida amalga oshirilgan til konstruksiyalari uchun dinamik maslahatchi;
- o'rnatilgan sintaksisni tekshirish yordam dasturi Sintaksisni tekshirish.

Ikkala vosita ham skript oynasi asboblari panelida joylashgan mos tugmalar yordamida faollashtiriladi. **Izoh**

E'tibor bering, Ekspert rejimida "Qo'shish va maqsadli yo'l" tugmasi har doim mavjud.

Sintaksisni tekshirish tugmasini bosish bevosita sintaksis xatolarini tekshirish jarayonini faollashtiradi. Agar ular topilsa, maxsus dialog oynasi ko'rsatiladi. Uning yordami bilan siz nafaqat aniqlangan xatoning tabiati, sabablari va joylashuvi haqida to'liq ma'lumotga ega bo'lasiz, balki Sintaksisni tekshirish yordam dasturining parametrlarini ham sozlashingiz mumkin. Ushbu sozlash ochiladigan oyna menyulari buyruqlari yordamida amalga oshiriladi.

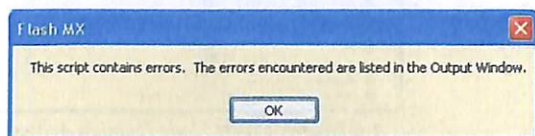


1.6.8-rasm. Til konstruksiyalari uchun dinamik ko'rsatmalar

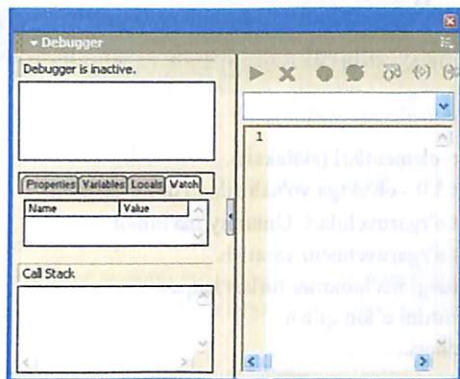


1.6.9-rasm. Expert Mode rejimida stsenariy oynasi asboblari paneli

Murakkab stsenariylarda xatolarni topish uchun maxsus tuzatuvchi, Debugger dan foydalanish mumkin. Ushbu vosita skript muharriri bilan chambarchas bog'langan: muharrir bilan to'xtash nuqtalarini o'rnatganingizdan so'ng, skriptning bajarilishini bosqichma-bosqich o'tkazish uchun tuzatuvchidan foydalanishingiz mumkin. Shu bilan birga, disk raskadrovka oynasida skriptda ishlatiladigan o'zgaruvchilar qiymatlari, funktsiya chaqiruvlari stekasi va dasturchi uchun juda foydali bo'lgan boshqa ma'lumotlar ko'rsatiladi.



1.6.10-rasm. Check Syntax dialog oynasi




1.6.11-rasm. Skriptni tuzatish oynasi

Filmi nosozliklarni tuzatish rejimiga o'tish uchun asosiy oynaning Control menyusidan Debug Movie buyrug'ini tanlash kifoya. Ilgari, yuqorida aytib o'tilganidek, skriptda to'xtash nuqtalarini joylashtirish foydalidir.

Ekspert rejimi ba'zi ActionScript dasturlash ko'nikmalarini talab qilganligi sababli, quyidagi barcha skript misollari Oddiy rejimda qamrab olingan.

## 7. Nazorat savollari

1. Flash MX 8 da kinokliplar uchun skriptlarni yozish uchun qaysi dasturlash tilidan  foydalaniladi.
2. **Events** (hodisalar) atamasi deganda nimani tushunasiz?
3. **Expressions** (iboralar) atamasi deganda nimani tushunasiz?
4. **Functions** (funksiyalar) atamasi deganda nimani tushunasiz?
5. **Handlers** atamasi deganda nimani tushunasiz?
6. **Instances** (nusxalar) atamasi deganda nimani tushunasiz?
7. **Instance names** atamasi deganda nimani tushunasiz?
8. **Keywords** (kalit so'zlar) atamasi deganda nimani tushunasiz?
9. **Methods** (usullar) atamasi deganda nimani tushunasiz?
10. **Objects** (obyektlar) atamasi deganda nimani tushunasiz?
11. **Operators** (operatorlar) atamasi deganda nimani tushunasiz?
12. **Properties** (xususiyatlar) atamasi deganda nimani tushunasiz?
13. ActionScript obyekt modelining ikkinchi darajasida qanday obyekt sinflari joylashgan?
14. Movie sinfiga qanday obyektlar kiradi?
15. Core sinfiga qanday obyektlar kiradi?
16. `_level` kalit so'z nimani anglatadi va u nimani belgilaydi?
17. ActionScript paneli qanday qismlarga bo'lingan?
18. ActionScript panelida nechta ish rejimi mavjud?

## 2.2. Actionscriptda o'zgaruvchilar. Actionscriptda o'zgaruvchilar turi. O'zgaruvchilarni e'lon qilish va ularga qiymatlarni belgilash

### Reja:

1. ActionScript elementlari (sintaksis).
2. ActionScript 3.0 - ob'ektga yo'naltirilgan dasturlash tili.
3. ActionScript o'zgaruvchilari. Umumiy ma'lumot.
4. ActionScript o'zgaruvchisini yaratish.
5. ActionScriptdagi ma'lumotlar turlari haqida.
6. Ma'lumotlar turini e'lon qilish.
7. Nazorat savollari.

**Kalit so'zlar:** *OYD, ob'ekt, sinf, usul, xususiyatlar, e'lon qilish, ishga tushirish, tavsif, massiv, o'zgaruvchi, ma'lumotlar turlari, hodisa.*

### 1. Actionscript elementlari (sintaksis)

Keling, avvalo quyidagi skriptni ko'rib chiqaylik - u har qanday tipik skriptda topilishi mumkin bo'lgan juda muhim til elementlarini o'z ichiga oladi. Shundan so'ng, biz ushbu elementlarni va ularning dasturni bajarishdagi rolini muhokama qilamiz. Bu yerda ko'rsatilgan skript tugmachaga tayinlanishi mumkin:

```
on (release) {  
    // qalam narxini belgilang mugCost = 5.00;  
    // savdo belgisi foizlarda belgilanadi  
    taxPercent = .06; //dollardagi belgini aniqlang  
    totalTax = mugCost * taxPercent; //bitimning umumiy qiymatini aniqlash totalCost  
    = mugCost + totalTax;  
    //xabar ko'rsatiladi  
    myTextBox.text = "Xaridingizning umumiy qiymati" + totalCost;  
    //cashRegister kino-ramz namunasini 50-kadrga o'tkazish  
    cashRegister.gotoAndPlay(50);  
}
```

Ehtimol, bir qarashda, bularning barchasi lotin tiliga o'xshaydi, ammo bu elementlarning ba'zilari bilan tanishib, bu juda murakkab ekanligini topa olmaysiz.

**Eslatma** Skript yaratishning boshqa elementlari (masalan, ob'ektlar, funksiyalar, ttakrorlanishlar, xususiyatlar va usullar) mavjud bo'lib, biz ularni ushbu kursda batafsil ko'rib chiqamiz.

### Hodisalar

Bu skriptni ishga tushirish uchun filmni ijro etish paytida sodir bo'ladigan narsa. Bizning misolimizda skriptni boshlaydigan voqea yoqilgan – on (release). Bu shuni anglatadiki, ushbu tugmachaga birlashtirilgan skript tugma bo'shatilgandan so'ng darhol bajariladi ( release ) bosgandan keyin. Har qanday skript voqea tomonidan ishga tushiriladi va filmingizda bunday

hodisalar ko'p bo'lishi mumkin - tugmani bosishdan tortib, ovoz yozishning oxirigacha va boshqalar. Amallar

Ular senariyning qalbini tashkil qiladi; odatda "harakat" deganda buyruq, ko'rsatma bo'lgan har qanday qator tushuniladi Flash - biror narsa qilish, tayinlash, yaratish, o'zgartirish, yuklab olish yoki o'chirish.

Yuqoridagi misol skriptidagi harakatlarga bir nechta misollar:

```
mugCost = 5.00;
```

```
cashRegister.gotoAndPlay (50);
```

Umuman olganda, harakatlar skriptni tashkil etuvchi satrlarning asosiy qismidir; ular ko'pincha jingalak qavslar ( { } ) ichiga olinadi va deyarli har doim nuqtali vergul bilan tugaydi ( ; - pastga qarag).

### Operatorlar

Ular turli belgilar bilan belgilanadi ( =, <, >, +, -, \*, && va boshqalar) va ikkita skript elementi o'rtasidagi o'zaro ta'sirning turli usullarini anglatadi. Quyidagi misollarni ko'rib chiqing:

```
taxPercent = .06; //taxPercent deb nomlangan o'zgaruvchiga .06 qiymatini
belgilaydi amountA < amountB // amountA ni amountB dan kichikligini
tekshiradi value1 * 500 //qiymat1 ni 500 ga ko'paytiradi
```

### Kalit so'zlar

Bularga sintaksisda saqlangan so'zlar kiradi Maxsus maqsadlar uchun ActionScript . Shuning uchun ularni nom sifatida ishlatish mumkin emas o'zgaruvchilar, funktsiyalar va teglar. Masalan, on so'zi kalit so'z bo'lib, u skriptda faqat skriptni boshlagan voqeani ko'rsatish uchun ishlatiladi, masalan, on(press), on( rollover ), on(rollOut) va hokazo. Skriptlarda kalit so'zlarni faqat ular mo'ljallangan maqsadlarda foydalaning, aks holda siz natijada faqat xato xabarini olasiz. Boshqa kalit so'zlarga quyidagilar kiradi: break, case, continue, delete, do, else, for, function, if, in, instanceof, new, return, switch, this, typeof, var, void, while va with.

### Ma'lumotlar

Dinamik skript bajarilish jarayonida doimiy ravishda barcha turdagi ma'lumotlarni yaratadi, ishlatadi yoki yangilaydi. Ma'lumotlar "zarralari" ning eng keng tarqalgan turi o'zgaruvchidir: bu o'ziga xos nomga ega bo'lgan ma'lumotlarning bir qismidir. O'zgaruvchini yaratganingizdan va unga qiymat berganingizdan so'ng, siz ushbu qiymatni skriptning istalgan joyidan oddiygina almashtirish orqali kirishingiz mumkin. o'zgaruvchan nomi .

Bizning misolimizda biz mugCost nomli o'zgaruvchiga 5,00 qiymatini tayinladik . Keyin, xuddi shu skriptda biz ushbu o'zgaruvchining nomini o'z ichiga olgan qiymatga murojaat qilish uchun ishlatdik.

### Figurali qavslar

Umuman olganda, jingalak qavslarni ochish va yopish orasidagi hamma narsa skriptning ushbu mustaqil qismi ishga tushirilganda bajarilishi kerak bo'lgan harakat yoki harakatlar to'plamidir. Buni shunday o'ylashingiz mumkin: "Buning natijasida - {buni qilish}". Masalan:

```
(chiqarish) {
//krujka narxini belgilang
mugCost = 5.00; //savdo belgisini foizlarda belgilang
taxPercent = .06;
}
```

## Nuqtali vergul

Skript satrining oxiridagi nuqta-vergul bir harakatni boshqasidan ajratib turadi, xuddi vergul bilan murakkab jumlaning qismlarini ajratib turadi. Quyidagi misolda nuqta-vergul bilan ajratilgan oltita alohida harakat mavjud:

```
mugCost = 5.00;  
taxPercent = .06;  
totalTax = mugCost * taxPercent;  
totalCost = mugCost + totalTax;  
myTextBox = "The total cost of your transaction is " +  
totalTax; cashRegister.gotoAndPlay(50); Nuqta
```

Nuqtalar ( . ) skriptlarda ikkita qo'llanishga ega.

**Birinchidan**, ular ma'lum bir kartonga yo'lni belgilashda ishlatiladi. Misol uchun, `_root.usa.indiana.bloomington` "Indiana" film belgisidagi "blumington" nomli film belgisiga ishora qiladi, bu esa o'z navbatida asosiy vaqt jadvalida (`_root`) joylashgan "usa" film belgisida joylashgan.

**Ikkinchidan**: `ActionScript` - bu ob'ektga yo'naltirilgan til bo'lib, ko'pchilik interaktiv vazifalar ob'ektning xususiyatlarini (xususiyatlarini) o'zgartirish yoki ob'ektga biror narsa qilishni buyurish orqali (usulni chaqirish orqali) amalga oshiriladi. Xususiyatni o'zgartirish yoki usulni chaqirishda nuqta ob'ekt nomini kerakli xususiyat yoki usul nomidan ajratish uchun ishlatiladi. Misol uchun, barcha xarakterli filmlar ob'ektlardir; keyin, `wheel` nomli kino belgisi misolining aylanish xususiyatini aniqlash uchun quyidagi sintaksisdan foydalanishimiz kerak bo'ladi:

```
gildirak_aylanishi = 90;
```

Qarang, nuqta ajratiladi qiymat belgilanayotgan xususiyatdan ob'ektning nomi. O'ynashni boshlash uchun kino belgisining xuddi shu misolini aytish uchun (`play()`), biz yozishimiz kerak:

```
gildirak.play()
```

Va yana chaqirilgan usuldan ob'ekt nomi nuqta orqali ajratiladi.

## Qavslar

Ular `ActionScript` da turli maqsadlar uchun ishlatiladi. Asosan, harakatni amalga oshirish uchun zarur bo'lgan har qanday aniq qiymatni ko'rsatish zarur bo'lgan joyda. Namuna skriptining oxirgi qatoriga qarang, u `cashRegister` kino belgisi namunasi 50-ramkada o'ynashni boshlashni aytadi:

```
cashRegister.gotoAndPlay(50);
```

Agar qavs ichidagi 50 raqamini 20 bilan almashtirsak, u holda topshiriqning ma'nosi o'zgarmaydi (aniq ramkaga o'tish), lekin u yangi qiymatga muvofiq bajariladi. Shunday qilib, qavslar yordamida biz harakatni ularning ichida joylashgan narsalar bilan ishlashni aytamiz.

## Qo'shtirnoq

Qo'shtirnoq matn ma'lumotlarini bildiradi. Skriptning o'zi matn bo'lganligi sababli, tirnoq belgilari dasturga qaysi ko'rsatmalar yoki ma'lumotlar va qaysi oddiy so'zlar ekanligini aniqlashga yordam beradi. Masalan, `Derek` (tirnoqsiz) o'zgaruvchining nomini bildiradi. Ammo "Derek" `Derek` so'zidan boshqa hech narsani anglatmaydi.

## Izohlar

Skriptda bu ikki qiyshiq chiziq (`//`) bilan boshlanadigan qatorlardir. Skriptni bajarish orqali, Flash bunday chiziq'larga e'tibor bermaydi. Sharhlar o'sha paytda skript aynan nima qilayotgani haqida eslatmalar va tushuntirishlar bo'lib xizmat qilishi mumkin. Agar dastur

sharhlar bilan ta'minlangan bo'lsa, uni yozilganidan bir necha oy o'tgach, uni ko'rganingizda, uning ishining mantiqini osongina eslab qolishingiz mumkin.

### **bo'sh joylar(probellar)**

Aslida, ular mutlaqo kerak emas, lekin sintaksisda bo'shliqlar va bo'shliqlardan foydalanishga amal qilinadi. Masalan,

```
on(release){
    mugCost = 5.00;
}
Xuddi shunday amalga oshiriladion(release) {
    mugCost = 5.00;
}
```

Biroq, kodni chekinishlar bilan o'qish ancha qulayroq ekanligiga rozi bo'lasiz. Bir vaqtning o'zida ketma-ket bajariladigan buyruqlardan iborat kod bloki bo'lgan jingalak qavslar ichiga olingan har bir qatorni cheklash yaxshi dasturlash amaliyotidir. Bunday holda, ba'zi bloklar boshqa bloklar ichida bo'lishi mumkin, keyin esa ular qo'sh chiziqqa ega bo'ladi - siz buni keyingi mashqlarda uchratasiz. Aslida, bu haqda ko'p tashvishlanishingiz shart emas. Flash-da avtomatik formatlash funksiyasi mavjud bo'lib, u avtomatik ravishda barcha kerakli bo'shliqlarni tartibga soladi.

Qoida tariqasida, skriptda bo'shliqlar ham e'tiborga olinmaydi. Mana, masalan, bu qator: `totalCost = mugCost + totalTax` ; Xuddi shu tarzda amalga oshiriladi `totalCost=mugCost+totalTax`;

Ba'zi dasturchilar kodni bo'shliqlar bilan yaxshiroq o'qish mumkin, deb hisoblashadi, boshqalari bo'shliqlarni qo'shish juda uzun deb hisoblashadi. O'zingiz uchun qaror qiling. Biroq, bo'shliqlar bilan bog'liq ikkita qoida mavjud: birinchidan, o'zgaruvchilar nomlarida bo'sh joy bo'lishi mumkin emas, ikkinchidan, ob'ekt nomi va unga bog'liq xususiyat yoki usul orasiga bo'sh joy qo'yib bo'lmaydi. Bu to'g'ri imlo bo'ladi: `myObject.propertyName` Lekin bunday emas:

`myObject. propertyName`

### **Katta va kichik harflar**

ActionScript sintaktik sezgir katta va kod kichik harf bo'lishi uchun mo'ljallangan. Agar o'zgaruvchi nomi bilan faqat kichik harflar foydalangan bo'lsa, keyin, kelajakda ushbu o'zgaruvchining nazarda, faqat kichik harf, uning nomini yozish kerak.

ActionScript-ga o'rnatilgan funktsiyalar, o'zgaruvchilar, kalit so'zlar, xususiyatlar va usullar har doim tuya korpusida yoziladi. Bu shuni anglatadiki, bir nechta so'zlardan tashkil topgan barcha nomlar kichik harf bilan boshlanadi va nomdagi har bir keyingi so'z bosh harf bilan boshlanadi. Masalan, ob'ektni o'lchash uchun foydalanishimiz mumkin bo'lgan xususiyat nomini yozishda biz buni shunday qilamiz: `scaleX`

E'tibor bering, `X` - bosh harf.

Sinf nomlari ham shu qoidaga amal qiladi, birgina farqi shundaki, ularda har doim nomning birinchi harfi bosh harf bilan yoziladi (Paskal uslubi deb ataladi, unga boshqa ko'plab dasturlash tillari amal qiladi).

### **MovieClip**

Yuqoridagilardan kelib chiqadiki, o'zingizning shaxsiy sinflaringizni, o'zgaruvchilarni, funktsiyalarni va boshqalarni yaratishda. bir xil qoidalarga rioya qiling. Keyin sizning kodingiz

umume'tirof etilgan uslubga mos keladi va shuning uchun foydalanish oson va o'qish oson bo'ladi.

## 2. ActionScript 3.0 - ob'ektga yo'naltirilgan dasturlash tili

ActionScript 3.0 - ob'ektga yo'naltirilgan dasturlash tili. OYDdagi asosiy qurilish bloki bu sinfdir. Sinf ob'ektni tavsiflaydi, lekin uning aniq namunasi emas. Va ob'ektlardan foydalanishni boshlashdan oldin, siz ular uchun sinf yaratishingiz kerak.

Sahnadagi ob'ektlar sinf misollaridir. Ular bilan ishlashni boshlashdan oldin, ActionScript o'zgaruvchilari, actionscript ma'lumotlar turlari va hokazolar yordamida ularni batafsil tavsiflovchi sinf yaratishingiz kerak. Shunday qilib, siz sinfning namunasini yaratganingizda, siz ob'ektni yaratasiz, ya'ni. ilovangizda foydalanishingiz mumkin bo'lgan haqiqiy narsa.

Agar siz Flashda sahnaga movieclip yoki tugma nusxalarini joylashtirgan bo'lsangiz, siz ob'ektlardan foydalangansiz. Flashda rasmni belgiga aylantirish orqali siz sinf yaratasiz. Belgini sahnaga sudrab olib chiqsangiz, uning namunasini yaratasiz.

Tugma sinfi o'z turida noyobdir. Siz uning misollarini faqat sahnada yaratishingiz mumkin. Movie Clip, dinamik matn va boshqa komponentlar sahnada yoki kodda yaratilishi mumkin bo'lgan sinflarga misoldir. Faqat kod bilan yaratilgan yana o'nlab sinflar mavjud.

ActionScript OYD da oldindan tuzilgan sinflarga qo'shimcha ravishda siz o'zingizning shaxsiy sinflaringizni yaratishingiz mumkin. Yuqorida aytib o'tilganidek, siz yangi belgi yaratganingizdan so'ng yangi sinf yaratasiz. Biznes ilovasidagi xarid savatini maxsus sinf sifatida ham aniqlash mumkin. Ularning yaratilishi va izlanishlariga keyingi darslarda qaytamiz.

ActionScript OYD dagi sinflar tizimli ravishda paketlarga ajratilgan. Har bir sinf alohida ActionScript faylida saqlanadi, paket esa barcha tegishli sinflarni o'z ichiga olgan jilddir.

### ActionScript OYD dagi xususiyatlar va usullar

Haqiqiy dunyoda ob'ektlar ularni belgilaydigan xususiyatlarga ega. Misol uchun, uy to'rtburchaklar asosga ega, ma'lum miqdordagi xonalarga ega, u asosiy nuqtalarga nisbatan ma'lum bir tarzda joylashgan, o'ziga xos rang va boshqa ko'plab xususiyatlarga ega. ActionScript OYD da bu xususiyatlarning barchasi xususiyatlar deyiladi. Xuddi shu xususiyatlarga ega uyni tasvirlaganingizdek, masalan, shakli, rangi, shaffofligi, o'lchami va sahnadagi joylashuvi (ActionScript sintaksisi) haqida gapirib, kinoklipni tasvirlashingiz mumkin.

Keling, uy o'xshashligiga qaytaylik. Siz yashayotgan uy odatda boshqa uylar joylashgan turar-joy hududida joylashgan. Ulardan ba'zilar konstruktiv jihatdan siznikidan farq qiladi, garchi ularda mavjud bo'lgan ko'plab xususiyatlar sizning uyingiznikiga o'xshash - to'rtburchaklar asos, u qurilgan material, qavatlar soni va boshqalar. Biroq, ularning ba'zi xususiyatlari bir-biridan farq qiladi - rangi, tom yopish va boshqalar. Shu ma'noda, har bir uy o'ziga xos va qo'shnisidan biroz farq qiladi. Bundan tashqari, sizning uyingiz o'ziga xos manzilga ega.

ActionScript OYD dagi obyekt misollari ham o'xshash xususiyatlarga ega, garchi ularning har biri o'ziga xos tarzda noyobdir. Sahnada filmingiz maqsadlari uchun qancha kerak bo'lsa, shuncha ko'p kino klip nusxalarini yaratishingiz mumkin. Va ularning har biri qolganlari bilan bir xil asosiy shaklga ega bo'ladi. Biroq, ularning barchasi sahnada joylashishi bilan farq

qiladi - ularning x va y qiymatlari boshqacha bo'ladi. Bundan tashqari, siz ularning o'lchamlarini o'zgartirishingiz mumkin, ya'ni. ularning scaleX va scaleY xossalari.

Ob'ektlarning xususiyatlarini aniqlashdan tashqari, ular bilan ba'zi harakatlarni bajarishingiz mumkin. Siz uyingizga kirishingiz va chiqishingiz, unda kechki ovqat tayyorlashingiz, uxlashingiz, televizor ko'rishingiz, kompyuterda ishlashingiz mumkin. Xuddi shu tarzda, siz onlayn-do'koningiz uchun belgilagan sinf savatiga narsalarni qo'shishingiz va olib tashlashingiz mumkin. Filmingizda rasmlarni jonlantirishingiz mumkin. Siz ularga qo'llashingiz mumkin bo'lgan harakatlar usullar (usul) yoki funktsiyalar (funktsiya) deb ataladi.

Ko'pgina hollarda, bir xil ob'ektning misollari bir xil usullarga ega. Muayyan misol uchun maxsus usullarni belgilash mumkin bo'lsa-da, biz ushbu kurs maqsadlari uchun bunga kirmaymiz.

### ActionScript OYD da statik sinflar

Yana mavzuda davom etishdan oldin, ActionScript OYD da statik (static) deb nomlangan maxsus sinf mavjudligini tushunish kerak. Siz ushbu sinfga asoslangan misollar yarata olmaysiz, chunki Bu shunchaki mantiqqa to'g'ri kelmaydi. Buning o'rniga siz ularning xususiyatlari va usullariga to'g'ridan-to'g'ri kirishingiz mumkin. Bunday sinfning yorqin misoli Math dir. U, masalan, pi kabi matematik konstantalarni va trigonometrik funktsiyalar kabi ma'lum matematik hisoblarni bajaradigan usullar to'plamini belgilaydi. Statik sinfning yana bir misoli - Stage (sahna) - filmning barcha ob'ektlari joylashgan sahna bitta va siz boshqa sahna yarata olmaysiz. Keyingi misol (Sichqoncha) Mouse ob'ekti - siz ko'pgina kompyuterlarda ikki yoki undan ortiq sichqoncha kursorlaridan foydalana olmaysiz. Yuqorida aytilganlarning barchasi klaviatura (Keyboard) uchun amal qiladi.

## 3.ActionScript o'zgaruvchilari. Umumiy ma'lumot

AS-o'zgaruvchilari keyinchalik foydalanish uchun ma'lumotlarni saqlashga imkon beradi. Masalan, siz mijozingiz nomini Actionscript o'zgaruvchisi sifatida yoki o'yinchi olgan ballar sonini saqlashingiz mumkin.

O'zgaruvchilar var kalit so'zi yordamida e'lon qilinadi . Undan so'ng darhol bo'sh joy bilan ajratilgan uning ismi. U tavsiflovchi bo'lishi kerak, ya'ni. O'zgaruvchida qanday ma'lumotlar saqlanishini aniq ko'rsatish yaxshi. Kelajakda bu u bilan ishlashni osonlashtiradi. Ismni belgilashda bir nechta majburiy qoidalarga rioya qilish kerak:

- Actionscript-dagi o'zgaruvchining nomi harf bilan boshlanishi kerak
- unda harflar, raqamlar va pastki chiziq bo'lishi kerak
- o'zgaruvchining nomi katta-kichik harflarga sezgir; xatolikka yo'l qo'ymaslik

uchun uni har doim bir xil yozishingiz kerak

Ko'pchilik dasturchilar o'zgaruvchilar nomlarini yozishda oldingi darsda ta'kidlanganidek, (camel casing) uslubiga amal qilishadi. Lekin siz ham xuddi shunday qilishingiz shart emas. Biroq, ActionScript-ga o'rnatilgan o'zgaruvchilarning nomlari ushbu uslubda yozilganligi sababli, xuddi shu tarzda davom ettirish va kodingizda umumiy uslub birligini saqlash oqilona bo'lardi. Bu ismlarni yaxshiroq o'qish va eslab qolishga yordam beradi.

Yangi yaratilgan o'zgaruvchi uchun ActionScript ma'lumotlar turini e'lon qilishingiz kerak. Nomdan keyin darhol ikki nuqta qo'yiladi va undan keyin o'zgaruvchiga tayinlangan ma'lumotlar turi yoziladi. Agar u so'zlar, iboralar yoki harflar va raqamlar birikmasidan iborat



bo'lsa, bunday o'zgaruvchilar uchun turni String ga o'rnatishingiz kerak. Raqamlar uchun ma'lumotlar turini Number ga o'rnatish.

Siz o'zgaruvchiga tenglik belgisini va undan keyin shu qiymatni yozish orqali qiymat belgilashingiz mumkin. Satr ma'lumotlari har doim qo'shtirnoq ichida yoziladi. Raqamlar uchun ma'lumotlar turini Number o'rnatish.

#### 4. ActionScript o'zgaruvchisini yaratish

Agar zarur bo'lsa, yangi fayl yarating.

Actions panelini oching.

Klaviaturada var kalit so'zini, so'ngra o'zgaruvchining nomini kiriting, masalan, myName.

Ikki nuqta qo'ying. Ma'lumotlar turlari ro'yxati paydo bo'lishi kerak.

String ma'lumotlar turini kiriting. Agar koddagi maslahat kerakli turga ishora qilsa, Enter tugmasini bosishingiz ham mumkin.

“Tenglik” belgisini kiriting.

O'zgaruvchi uchun qiymat kiriting.

Nuqtali vergul qo'ying.

Sizing ekraningizda quyidagilar hosil bo'lishi kerak:

```
var myName:String = "Valeriy";
```

Tabriklaymiz! Siz hozirgina birinchi ActionScript o'zgaruvchingizni yaratdingiz.

Va endi yana bir bor har bir element haqida alohida.

- > var kalit so'z bo'lib, “o'zgaruvchi” ma'nosini bildiradi.
- > myName - o'zgaruvchining nomi
- > : - ikki nuqta o'zgaruvchi nomini unga tayinlangan ma'lumotlar turidan ajratib turadi
- > String - o'zgaruvchan ma'lumotlar turi
- > = - teng belgisi o'zgaruvchiga qiymat beradi
- > " - qo'shtirnoqlar faqat String (string) tipidagi o'zgaruvchining qiymatini yozishda ishlatiladi.
- > Valeriy - o'zgaruvchan qiymat
- > ; - nuqtali vergul gapni tugatadi (odatdagi gapdagi nuqtaga o'xshash).

Endi olingan bilimlar bilan mashq qilish yaxshi fikr. Bir so'zdan ikki yoki uch so'zgacha bo'lgan bir nechta ActionScript o'zgaruvchilar nomlarini o'ylab toping. Ularning ismlarini yozish uchun tuya qobig'idan (camel casing) foydalaning. Ular uchun ma'lumotlar turini belgilang va ularga qiymatlarni belgilang. Har bir gapning oxiriga nuqta-vergul qo'yishni unutmang!

#### 5. ActionScriptdagi ma'lumotlar turlari haqida

ActionScript 3 - bu o'zgaruvchilar yaratishda ActionScript ma'lumotlar turini yozishni talab qiladigan dasturlash tili. Bu o'zgaruvchiga qanday turdagi ma'lumotlar tayinlanganligini taniydi va siz ushbu tilda dasturlashda ushbu yondashuvga amal qilishingiz kerak degan ma'noni anglatadi.

**ActionScript ma'lumotlarining asosiy turlari**

ActionScript ma'lumotlar 3 ta asosiy turi mavjud: Null, void va Object.

Null (nol) faqat bitta qiymatga ega bo'lishi mumkin - qiymat yo'qligi. Bu o'zgaruvchining qiymati yo'qligini anglatadi. Shu bilan birga, null 0 emas va bo'sh satr emasligini unutmang, chunki ikkinchisi ma'lum bir ma'noga ega.

Null ma'lumotlar turi kabi, void (aniqlanmagan) bitta qiymatga ega: undefined (qiymat aniqlanmagan). Hech qanday qiymat qaytarmaydigan funksiyalar bilan ishlashda ko'pincha bu turga duch kelasiz. Biz bu masalani tegishli darsda batafsil muhokama qilamiz.

Object (ob'ekt) ma'lumotlar turi barcha sinflarning barcha namunalarini, shu jumladan ActionScript-ga allaqachon o'rnatilganlarini o'z ichiga oladi. Bizning darslarimiz davomida aynan shunday ob'ektlarning namunalari qachon yaratilishiga oid ko'plab misollarni ko'rasiz.

Yuqoridagi asosiy Actionscript ma'lumotlar turlariga qo'shimcha ravishda, har qanday sinf unga o'rnatilgan yoki siz yaratgan ma'lumotlar turi sifatida tan olinadi. Shuning uchun, keyinroq ko'rsatilgandek, o'zgaruvchilarga String, Number, Array va boshqalar kabi ma'lumotlar turlarini belgilashingiz mumkin. Bu o'z navbatida siz String, Number va Array sinflarining namunalari bo'lgan o'zgaruvchilar yaratayotganingizni anglatadi.

### **Oddiy ma'lumotlar turlari**

O'zgaruvchiga oddiy ma'lumotlar turi tayinlanishi mumkin. AS3 (ActionScript 3) da bular String (satrlı), Boolean (mantiqiy), Number (raqam), int (butun son) va uint (musbat butun son).

Satrlar (strings) - bu harflar, so'zlar, jumlarlar, matn qismlari bo'lib, ularda harflar, raqamlar va bo'shliqlar mavjud. Satrlar har doim kotirovka qilinadi! Bu ularning xususiyati va yozishda boshqa turdagi ma'lumotlardan farqidir. Satrlarni qanday birlashtirish haqida alohida darsda "Stringlarni birlashtirish (birlashtirish)" tasvirlangan.

Mantiqiy (boolean) ma'lumotlar turi fizik Jorj Bul sharafiga nomlangan. Ushbu turdagi o'zgaruvchiga faqat ikkita qiymat berilishi mumkin: true (1) yoki false (0). Boshqa dasturlash tillarida bo'lgani kabi, ActionScript-da true har qanday nolga teng bo'lmagan raqamga, bo'sh bo'lmagan satrga mos keladi.

Ma'lumotlar turi Har qanday Number qiymatlar bilan ishlash uchun raqam (satrlar emas!). U 2 ta kichik turni o'z ichiga oladi: int (integer) - ham musbat, ham manfiy bo'lishi mumkin bo'lgan butun son va uint (unsigned integer) - faqat musbat butun son, 0.

### **Murakkab ma'lumotlar turlari**

Ba'zan o'zgaruvchida bir nechta qiymatlarni saqlashingiz kerak bo'ladi, ya'ni. bitta emas. Bu murakkab ma'lumotlar turlari bilan amalga oshirilishi mumkin. Bunday turdagi misollardan biri

Array (massiv) bo'lib, biz uni ActionScript massivini yaratish, ActionScript massiviga elementlar qo'shish va o'chirish va boshqa o'quv qo'llanmalarida ko'rib chiqamiz.

## **6. Ma'lumotlar turini e'lon qilish**

O'zgaruvchiga ActionScript ma'lumotlar turini tayinlash o'zgaruvchi e'lon qilingan vaqtda sodir bo'ladi. O'zgaruvchi nomidan keyin siz ikki nuqta qo'ying va darhol ma'lumotlar turini yozing. Ko'pgina hollarda (lekin har doim ham emas) ActionScript-da ma'lumotlar turi qandaydir sinf bo'ladi va shuning uchun uning nomi bosh harf bilan boshlanishi kerak. Masalan,

siz myName o'zgaruvchisini yaratishga va unga String ma'lumotlar turini berishga qaror qildingiz. Natijada siz quyidagilarni olishingiz kerak:

```
var Ism:String;
```

```
Turlarga ajratilmagan ma'lumotlar
```

Muayyan holatlarda siz o'zgaruvchining qiymatini yozishdan qochishingiz kerak bo'ladi.

Buni qilish oson: hech qanday ma'lumot turini tayinlamang. Quyidagi misoldagi kabi: var myData;

Biroq, shunday qilish orqali siz kodingiz ustida ishlayotgan boshqa dasturchilarni buni ataylab qildingizmi yoki xato bo'lganmi degan savolni qoldirishingiz mumkin. Shuning uchun, agar siz haqiqatan ham o'zgaruvchiga hech qanday tur belgilamoqchi bo'lsangiz, uni yulduzcha \* (yulduzcha) bilan yozishingiz kerak. Masalan:

```
var myData:*;
```

Yuqoridagi ikkala misol ham xuddi shunday ishlaydi. Bu yerda yana bir bor eslatib o'tmoqchimanki, turlarga ajratilmagan ma'lumotlardan faqat o'ta zarur hollarda foydalanish kerak!

Muayyan ma'lumotlar turiga tayinlangan o'zgaruvchilar faqat shu turdagi ma'lumotlarni o'z ichiga olishi kerak. Shuning uchun, agar siz ushbu qoidaga rioya qilsangiz, kelajakda koddagi xatolarga yo'l qo'ymaslik va agar mavjud bo'lsa, ularni tuzatish oson bo'ladi. Har safar siz tuzatish juda oson bo'lgan qator raqami bilan xato xabarini olasiz. Agar biron bir o'zgaruvchi yozilmagan bo'lsa, u holda hech qanday xabar olinmaydi, lekin dastur ishlamaydi. Xato qayerda ekanligini topish juda qiyin bo'ladi.

## Ma'lumotlar turini o'zgartirish

Agar siz Actionscript o'zgaruvchisining ma'lumotlar turini o'zgartirishingiz kerak bo'lsa, buni "casting" deb nomlangan usul bilan qilishingiz mumkin. Siz shunchaki yangi ma'lumotlar turini, keyin o'zgaruvchining nomini qavs ichiga yozasiz. Masalan:

```
var Ism:String = "Valeriy";
```

```
Boolean(myName);
```

Bu misolda myName o'zgaruvchisiga String ma'lumotlar turi berilgan va Valeriy qiymati o'rnatilgan. Keyin uning turini Booleanga o'zgartiramiz.

Shuni ta'kidlash kerakki, bunday ActionScript ma'lumotlar turini o'zgartirish operatsiyasi har doim ham muvaffaqiyatli bo'lavermaydi. Yuqorida ko'rsatilgan misolda kodni qayta ishlash bilan bog'liq muammolar bo'lmaydi. Endi myName o'zgaruvchisining qiymati rost bo'ladi. Biz allaqachon bilamizki, har qanday bo'sh bo'lmagan satr mantiqiy turga kelganda rost deb baholanadi. Biroq, quyidagi misol, ba'zida quyish paytida yuzaga keladigan ba'zi qiyinchiliklarni ko'rsatadi.

```
var Ism:String = "myName";
```

```
Number(myName);
```

Ushbu amaldan so'ng myName o'zgaruvchisining qiymati raqam emas, NaN (Not a Number) ga teng bo'ladi! Bular umuman biz kutgandek emas. Bu harflardan iborat qatorni raqamga aylantirib bo'lmasligi sababli sodir bo'ldi. Teskari operatsiya mumkin, chunki satrlarda raqamlar bo'lishi mumkin.

```
var sum:Number = 356;
```

```
String(sum);
```



1. Dasturning kodi qaysi elementlarda yoziladi?
2. ActionScript tilida Izoh qatori qaysi belgi bilan boshlanadi?
3. ActionScript dasturlash tilidagi “harakat” atamasini aniqlang?
4. ActionScript dasturlash tilida “voqea” atamasini aniqlang?
5. ActionScript dasturlash tilidagi “o‘zgaruvchi” atamasiga ta’rif bering?
6. ActionScript dasturlash tilida “obyekt” atamasini aniqlang?
7. O‘zgaruvchilarni e’lon qilish qaysi kalit so‘z bilan boshlanadi?
8. Asosiy ActionScript ma’lumotlar turlarining nomini sanab bering?
9. Ob’ekt yoki o‘zgaruvchiga nom berishda qanday qoidalarga amal qilish kerak?
10. ActionScript dasturlash tilida qanday oddiy ma’lumotlar turlari mavjud?

### 2.3. Trace() funksiyasi. Oddiy arifmetik operatorlar. Satrlar. Satrlarda bajariladigan amallar. typeof() funksiyasi.

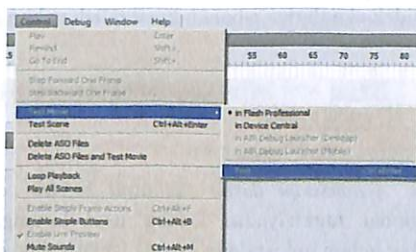
#### Reja:

1. ActionScript xatolarini tekshirish.
2. trace() funksiyasi.
3. Oddiy arifmetik operatorlar.
4. Satrlar. Satrlarda bajariladigan amallar.
5. typeof() funksiyasi.
6. Nazorat savollari

**Kalit so'zlar:** trace(), typeof(), string, nazorat, raqam, test, uzunlik, substr, split, charAt(), charCodeAt(), concat(), toLowerCase(), toUpperCase()

### 1. ActionScript xatolarini tekshirish

Kodingizni ActionScript xatolariga tekshirish uchun Flash Player yordamida fleshfilmingizni ko'rishingiz kerak. Buni istalgan vaqtda qilishingiz mumkin. Buning uchun Flash CS5 dasturining asosiy menyusida Control (Boshqarish) menyu bandini bosing, so'ng Test Movie (sinov filmi) va Test (test) ni bosing.



3.1.1-rasm. ActionScript xatolarini tekshirish uchun Boshqarish(Control) menyu bandi

Bundan tashqari, klaviaturadagi **Ctrl + Enter** tugmachlar birikmasidan foydalanishingiz mumkin .

Sizing flesh-filmigiz to'g'ridan-to'g'ri Flash CS5 da ishlaydi, ya'ni. integratsiyalashgan rivojlanish muhitida (IDE).

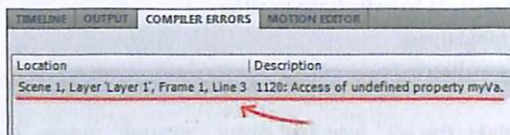
Agar ActionScript kodingizda xatolar bo'lsa, kompilyator panelida ActionScript xato xabari paydo bo'ladi, unda siz xatolik yuz bergan koddagi satr raqamini va uning qisqacha tavsifini ko'rasiz. Actionscript xatosini hal qilish uchun sizga kerak:

- Flash Playerni yoping
- Xatolarni to'g'rilang
- flesh-filmigizni yana sinab ko'ring

Sinov paytida bir nechta xatolar bo'lsa, ularni ketma-ket tuzatishingiz kerak, har safar keyingi Actionscrip xatosini tuzatgandan so'ng sinovdan o'tkazing, chunki. koddagi bitta xato bir nechta xabarlarga olib kelishi mumkin. Agar ActionScript kodingizda kuzatuv funksiyasi mavjud bo'lsa, uning ishining chiqishi Chiqish panelida ko'rsatiladi.

## 2. trace() funksiyasi.

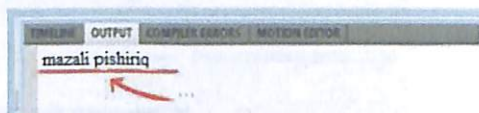
Afsuski , biz hammamiz xato qilishga moyilmiz, shuning uchun kod bilan ishlashda siz ham xato qilasiz. ActionScript o'zgaruvchisi nomini noto'g'ri yozish yoki til sintaksisini noto'g'ri ishlatish Flash Player tomonidan darhol aniqlashga yordam beradigan eng keng tarqalgan xatolardir . Agar bu sodir bo'lsa, kompilyator xatolar panelida tegishli xabarni ko'rasiz. Masalan:



3.2.1-rasm. ActionScript xatolarini tekshirish uchun kompilyator xatolari(Compiler errors) bandi

Mantiqiy xatolar (**logical errors**) deb ataladigan bunday xatolarni aniqlash ancha qiyin . Mantiqiy xato - bu sizda barchsi "grammatik jihatdan" to'g'ri yozilgan bo'lsa, lekin kodda xatolik mavjud bo'lib, u noto'g'ri ishlaydi. Misol uchun, siz o'zgaruvchiga boshqa qiymatga ega bo'lsa, bitta qiymat berilgan deb o'ylashingiz mumkin. Va aynan shu qiymatni tekshirib, sinab ko'rishga harakat qilganingizda, siz kutgan narsaga erisha olmaydiz.

Har qanday o'zgaruvchining qiymatlarini tekshirishga yordam beruvchi maxsus **trace()** ActionScript usuli mavjud, uni funksiya yoki buyruq deb ham atash mumkin. Flesh-filmigizni sinab ko'rayotgan paytda ular Chiqish (Output) panelida paydo bo'ladi. Ushbu test barcha qiymatlar kutilganidek ekanligiga ishonch hosil qilish uchun aniq amalga oshiriladi.



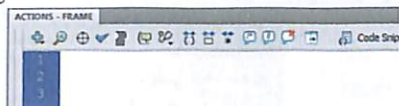
3.2.2-rasm. ActionScript dastur natijasini ko'rish uchun Chiqish(Output)bandi.

ActionScript-da trace global funksiyaadir. Ushbu usul kodingizni yozishning istalgan bosqichida mavjud, shuning uchun uni istalgan vaqtda ishlatishingiz mumkin. Ushbu usulning nomini har doim kichik (kichik) harflar bilan yozing va har doim sinab ko'rmoqchi bo'lgan

qiymatni qavs ichiga qo'ying. Agar siz oddiy matnni sinovdan o'tkazayotgan bo'lsangiz (String turidagi o'zgaruvchi emas) u qo'sh tirnoq ichiga " " olinishi kerak, qolgan barcha holatlarda qo'shtirnoq ishlatilmaydi.

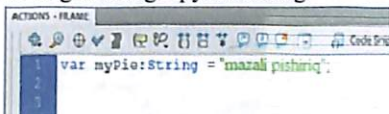
### O'zgaruvchilarni ActionScript trace usuli bilan tekshirish

Actions panelini oching



3.2.3-rasm. ActionScript dasturini yozish uchun Actions bandi.

O'zgaruvchi yarating va o'zgaruvchiga qiymat bering



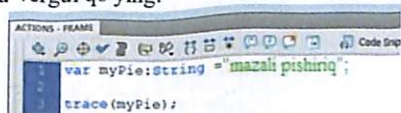
3.2.4-rasm. ActionScript dasturi yozilgan Actions bandi.

Trace so'zini yozing.

Qavs oching.

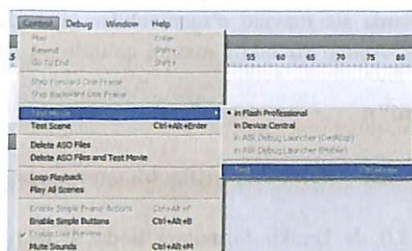
Yaratilgan o'zgaruvchining nomini kiriting.

Qavsni yoping va nuqta-vergul qo'ying.



3.2.5-rasm. ActionScriptda trace funksiyasidan foydalanish.

Flash CS5-ning asosiy menyusidagi Controlni, so'ng Test Movie sinab ko'rish va keyin Testni bosing.



3.2.6-rasm. ActionScript dastur kodini ishga tushirish.

Flash Playerda video-tasvir o'z harakatini qayta boshlaydi.

Output (Chiqish) paneli ochiladi, unda tekshirilayotgan o'zgaruvchining qiymati ko'rsatiladi.

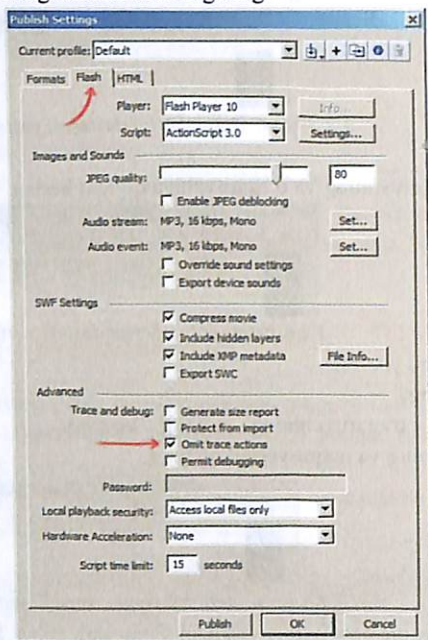


3.2.7-rasm. ActionScript dastur natijasi.

ActionScriptda trace - bu kodingizni xatolarini tekshirish uchun juda yaxshi vosita, ammo kodingizning oxirgi versiyasida foydalanuvchilaringizni chalkashtirib yubormaslik uchun undan xalos bo'lishingiz kerak bo'ladi. Ko'pincha, u shunchaki o'chiriladi yoki sharhlanadi (bu holda ActionScript uni qayta ishlash kerak bo'lgan kod deb hisoblamaydi), lekin siz filmingiz

chop etilganda Flashning trace (kuzatish) usulini eksport qilishiga ham yo'l qo'ymasligingiz mumkin. Buning uchun quyidagilarni bajaring:

- asosiy menyuda Faylni, keyin pastki menyuda **Publish Settings**ni bosing
- Flash yorlig'ini tanlang
- Trace Actions opsiyasi yonidagi katakchani belgilang



3.2.8-rasm. Chop qilish sozlamalari(Publish Settings).

Aksariyat hollarda siz mavjud o'zgaruvchilar qiymatlarini ActionScript trace usuli yordamida tekshirasiz. Ammo siz oddiy matnni qo'shtirnoq ichiga qo'yish orqali ham sinab ko'rishingiz mumkin:

```
trace("Salom!");
```

### 3. Oddiy arifmetik operatorlar

ActionScript 3.0 da kerakli formulani ifoda qiymati sifatida kiritish orqali oddiy matematik amallarni bajarishingiz mumkin. Oddiy matematik amallarga qo'shish, ayirish, bo'lish va ko'paytirish kiradi. Ularni ifodalash uchun ishlatiladigan belgilar standart bo'lib, ular **ActionScript** iboralari deb ataladi. Ular kompyuter olamida keng qo'llaniladi: " + " qo'shish uchun, " - " ayirish uchun, " \* " ko'paytirish uchun, " / " bo'lish uchun. ActionScript 3.0 da "qoldiq" uchun operator ham mavjud - %.

ActionScript oddiy matematika bilan bir xil ish jarayonidan foydalanadi. Agar formulangizda barcha to'rtta ActionScript operatori mavjud bo'lsa, ya'ni 4 ta arifmetik amal bajariladi bular:

- birinchi bo'lib ko'paytirish amalga oshiriladi (agar bir nechta bo'lsa, chapdan o'ngga tartibda), ➤ keyin bo'lish amali,
- keyin qo'shish amali
- va nihoyat ayirish amali.

Siz ifoda qismlarini qavs ichiga qo'yib, ularni baholash tartibini ta'minlash uchun guruhlashingiz mumkin: avval qavs ichiga olingan amallar birinchi navbatda bajariladi.

### Matematik operatorlar

#### 3.3.1-jadval. Matematik operatorlar

Operator	Amal	Misol
+	qo'shish	$8 + 7 = 15$
-	ayirish	$150 - 45 = 105$
*	ko'paytirish	$22 * 4 = 88$
/	bo'lish	$36 / 9 = 4$
%	qoldiqni hisoblash	$25 \% 3 = 1$
+=	qo'shish va o'zlashtirish	agar $x=5$ bo'lsa, $x += 3$ , jami $x=8$
-=	ayirish va o'zlashtirish	agar $x=5$ bo'lsa, $x -= 3$ , jami $x=2$
*=	ko'paytirish va o'zlashtirish	agar $x=5$ bo'lsa, $x *= 3$ , jami $x=15$
/=	bo'lish va o'zlashtirish	agar $x=5$ bo'lsa, $x /= 2$ , jami $x=2,5$
%=	qoldiq va o'zlashtirish	agar $x=5$ bo'lsa, $x \% = 2$ , jami $x=1$
++	1 ga orttirish	agar $x=5$ bo'lsa, $x++$ , jami $x = 6$
--	1 ga kamayishi	agar $x=5$ bo'lsa, $x--$ , jami $x=4$

Matematik operatsiyalar o'zgaruvchilarning qiymatlari sifatida (var) yoki raqamlar ishlatilishi kerak bo'lgan boshqa joyda (Number) ishlatilishi mumkin.

Hisoblash natijasining to'g'riligini tekshirish uchun trace buyrug'idan foydalaning.

Masalan: var natija: Number = 2 + 3 \* 2 / 4 - 1; trace(natija);

Bundan tashqari, siz matn ob'ektini yaratishingiz va hisoblash natijasini Flash matn maydoniga joylashtirishingiz mumkin.

#### Number raqamli turi

Flashda faqat bitta raqamli tur mavjud, bu Number. Number turida biz istalgan raqamli qiymatlarni yozishimiz mumkin: butun son va suzuvchi nuqta. Kodda juda katta bo'lgan butun son konstantalari (yoki fiksirlangan nuqta konstantalari) suzuvchi nuqta qiymatlariga aylantiriladi. Number turiga yozilishi mumkin bo'lgan modulning maksimal va minimal qiymatlari Number.MAX\_VALUE va Number.MIN\_VALUE maxsus konstantalarida saqlanadi.

Shunday qilib, MAX\_VALUE raqami 1,79769313486231e+308 va Number.MIN\_VALUE raqami 4,94065645841247e-324.

Number turi nafaqat raqamli, balki unchalik ham raqamli bo'lmagan qiymatlarni saqlashi mumkin. Ushbu maxsus qiymatlardan uchtasi mavjud. Ulardan ikkitasi cheksizlikni bildirish uchun xizmat qiladi. Bundan tashqari, cheksizlik turlicha bo'ladi: musbat va manfiy. Bu qandaydir ma'noga ega, chunki musbat cheksizlik faqat Number.MAX\_VALUE dan kattaroq raqamni bildiradi. Manfiy esa (-Number.MAX\_VALUE) dan kichik sonidir. Musbat cheksizlik Number.POSITIVE\_INFINITY konstantasiga mos keladi. Va manfiy esa ravishda -Number.NEGATIVE\_INFINITY konstantasiga mos keladi. Satrga aylantirilganda, bu konstantalar Infinity va -Infinity ga aylanadi. Bu qiymatlardan foydalangan holda bir nechta misollar: trace("-5/-0 = " + -5/-0);



```

trace("-5/Number.MIN_VALUE = " + -5/Number.MIN_VALUE);
trace("-5/-Number.MIN_VALUE = " + -5/-Number.MIN_VALUE);
trace("Math.log(0) = " + Math.log(0));
trace("Math.sqrt(-1) = " + Math.sqrt(-1));

```

Natijada biz quyidagilarni olamiz:

```

5/0 = Infinity
-5/0 = -Infinity
-5/-0 = -Infinity
-5/Number.MIN_VALUE = -Infinity
-5/-Number.MIN_VALUE = Infinity
Math.log(0) = -Infinity
Math.sqrt(-1) = NaN

```

Ulardan oxirgisi (bu bilan tez-tez shug'ullanish juda keng tarqalgan) Bu raqam.NaN (satrga aylantirilganda "NaN" ga tarjima qilinadi). "Not a Number" buning qisqargan holati uchun nomi - "raqam emas". NaN haqiqiy (cheksiz bo'lsa ham) raqam sifatida ko'rsatib bo'lmaydigan ifoda qiymatlari uchun ishlatiladi. Biz hozirgina misolni ko'rdik - Math.sqrt(-1) ifodaning qiymati Number.NaN.

#### 4. Satrlar. Satrlarda bajariladigan amallar

String sinfida matn satrlari bilan ishlash imkonini beruvchi usullar mavjud. Ko'p ob'ektlar bilan ishlashda satrlar muhim rol o'ynaydi. Bu yerda tasvirlangan usullar TextField, StaticText, XML, ContextMenu va FileReference kabi ob'ektlarda qo'llaniladigan satrlar bilan ishlashda foydalidir. Satrlar belgilar ketma-ketligidir. ActionScript 3.0 ASCII va "Unicode" belgilarini qo'llab-quvvatlaydi.

Dasturlash tilida satr matn qiymatini, ya'ni bitta qiymatga birlashtirilgan harflar, raqamlar yoki boshqa belgilar ketma-ketligini bildiradi. Masalan, ushbu kod String ma'lumotlar turiga ega o'zgaruvchini yaratadi va unga satr literalini tayinlaydi.

```
var albumName:String = "Three for the money";
```

Ushbu misoldan ko'rinib turibdiki, ActionScriptda satr qiymatining matni qo'sh yoki bitta tirnoq ichiga olingan. Quyida boshqa misol qatorlari keltirilgan. "Hello" "555-7649"

```
http://www.adobe.com/
```

ActionScriptda matnning istalgan qismini manipulyatsiya qilganingizda, siz satr qiymati bilan ishlaysiz. ActionScriptda String sinfi matn qiymatlari bilan ishlash uchun ishlatilishi mumkin bo'lgan ma'lumotlar turidir. String nusxalari ko'pincha xususiyatlari, usullar parametrlari va hokazolar ActionScriptning boshqa sinflari sifatida ishlatiladi.

#### Satrlarni yaratish

String sinfi ActionScript 3.0 da satr (matn) ma'lumotlarini ko'rsatish uchun ishlatiladi. ActionScript satrlari ASCII va "Unicode" belgilarini qo'llab-quvvatlaydi. String literal yordamida eng oson aniqlanadi. Satrni literal deb e'lon qilish uchun to'g'ri qo'sh qo'shtirnoq (") yoki bitalik qo'shtirnoq (')dan foydalanish mumkin. Masalan, quyidagi ikkita satr bir-biriga ekvivalentdir: var str1:String = "hello"; var str2:String = 'hello';

Quyida ko'rsatilganidek, yangi operator yordamida qatorni e'lon qilishingiz ham mumkin.

```

var str1:String = new String("hello");
var str2:String = new String(str1); var
str3:String = new String(); // str3 = ""

```

Quyidagi ikkita qator bir-biriga ekvivalentdir:

```

var str1:String = " hello ";
var str2:String = new String("hello");

```

Bittalik tirnoqlardan (') chegaralovchi sifatida bitta tirnoq (') bo'lgan satr literalida foydalanish uchun teskari chiziq(slesh belgisi- \) dan foydalanish zarur. Xuddi shunday, bitta qo'shtirnoq (") bilan ajratilgan satrda qo'sh tirnoqlardan (") foydalanish uchun teskari chiziq belgisi (\)dan foydalanish mumkin. Quyidagi ikkita qator bir-biriga ekvivalentdir: var str1:String = "That's \"A-OK\""; var str2:String = "That\'s \"A-OK\"";

Bitta yoki qo'sh tirnoqlarni tanlash quyida ko'rsatilganidek, satr literalida qaysi qo'shtirnoqlar mavjudligiga asoslanishi kerak. var str1:String = "ActionScript 3.0";

```

varstr2:String = 'banana' ;

```

Quyidagi jadvalda ko'rsatilganidek, teskari chiziq belgisi (\) satr harflarida boshqa belgilarni aniqlash uchun ishlatilishi mumkin.

3.4.1-jadval. Satrlarni yaratish.

ESC-ketma-ketlik	Belgi
\b	Qaytish
\f	Yangi sahifa
\n	Yangi qator
\r	Yangi so'z boshi
\t	tabulyasiya
\unnnn	"Unicode" belgilar kodida nnnn ifodasining O'noltilik sanoq tizimida masalan \u263a
\xnn	ASCII belgilar kodida nn ifodasining O'noltilik sanoq tizimida
\'	Bittalik tirnoq
\"	Qo'shtirnoq
\\	Bitta teskari yotiq chiziq

### length xususiyati

Har bir satr satrda belgilar sonini ifodalovchi length (uzunlik) xususiyatiga ega.

```

var str:String = "Adobe"; trace(str.length); // output: 5

```

Quyidagi misolda ko'rsatilganidek, bo'sh satr va null satr nolga teng uzunlikka ega. var str1:String = new String(); trace(str1.length); // output: 0  
str2:String = "";  
trace(str2.length); // output: 0

### Satrlardagi belgilar bilan ishlash

Satrdagi har bir belgi satrda indeks pozitsiyasiga ega (butun son). Birinchi belgining indeks pozitsiyasi 0 ga teng. Masalan, keyingi qatorda y belgisi 0 pozitsiyasida va w belgisi 5 pozitsiyasida.

```

"yellow"

```

Quyida ko'rsatilganidek, charAt() va charCodeAt() usullaridan foydalangan holda satrning turli pozitsiyalarida individual belgilarni tahlil qilishingiz mumkin.

```
var str:String = "salom dunyo!";
trace(str.charAt(0), "-", str.charCodeAt(i));
trace(str.charAt(1), "-", str.charCodeAt(i));
trace(str.charAt(2), "-", str.charCodeAt(i));
```

Ushbu kod bajarilganda, quyidagi chiqish qaytariladi. h-104 e-101 l-108

Quyidagi kodda bo'lgani kabi, fromCharCode() usuli yordamida qatorni aniqlash uchun belgilar kodlaridan ham foydalanishingiz mumkin. var myStr:String = String.fromCharCode(104,101,108,108,111); // myStr-ni "salom!"

### String taqqoslash

Satrlarni solishtirish uchun quyidagi operatorlardan foydalanish mumkin: <=, !=, ==, => va >.

Ushbu operatorlar satrlar bilan foydalanilganda, ActionScript kodi quyida ko'rsatilganidek, ularni chapdan o'ngga taqqoslab, satrdagi har bir belgining kod qiymatini hisobga oladi.

```
trace("A" < "B"); //
true trace("A" < "a"); //
true trace("Ab" < "az");
// true
trace("abc" < "abza"); // true
```

### Boshqa ob'ektlarning string ko'rinishini olish

Har qanday turdagi ob'ektning String ko'rinishi olishingiz mumkin. Buning uchun barcha ob'ektlar toString() usuliga ega.

```
var n:Number = 99.47;
var str:String = n.toString(); // str == "99.47"
```

Birlashtirish operatori + dan String va simsiz ob'ektlar kombinatsiyasi bilan foydalanilganda toString() usuli talab qilinmaydi.

### Satrlarni birlashtirish

Satrlarni birlashtirish (birlashtirish) uchun plyus operatoridan foydalanilganda, string boshqa barcha turlardan ustun turadi. Bu shuni anglatadiki, agar kerak bo'lsa, boshqa turdagi ifodalarning qismlari satrlarga aylantiriladi. Quyidagi misolga qarang.

```
trace((5 > 7) + 10 + "String");
trace(10 + "String" + (5 > 7));
trace(10 - "String" - (5 > 7));
```

Uni bajarish uchun ishga tushirsak, biz buni konsolda ko'ramiz:

10 qator

10 qator false

NaN

Birinchi ikkita kuzatuv bayonotida trace satr qo'shilgan operandlar satrlarga aylantirildi. Faqat birinchi holatda (5 > 7) + 10 ifoda qiymati birinchi bo'lib hisoblandi. Shu bilan birga, tipni o'zgartirish ham amalga oshirildi: agar ifodada Boolean va Number paydo bo'lsa va ifoda

semantikasi bir vaqtning o'zida arifmetik amallarni nazarda tutsa, u holda Number ga o'tkazish amalga oshiriladi. Bunda true 1 ga, false esa 0 ga aylantiriladi (Agar mantiqiy amallar bajarilsa, aksincha, Number Boolean ga aylantiriladi: nolga teng bo'lmagan har qanday raqam rost, nol esa yolg'oni bildiradi). Shunday qilib, bizning misolimizda birinchi trace qo'ng'irog'ida mantiqiy ifodaning qiymati birinchi bo'lib hisoblab chiqilgan, u false bo'lib chiqdi, u nolga aylandi va o'n bilan birga yana o'nni berdi. Va shundan keyingina bu o'nta trace bilan bog'landi. Ikkinchi kuzatuv chaqiruvida mantiqiy ifoda satrga qo'shildi, shuning uchun biz u yerda aniq false ni oldik.

Nihoyat, trace kuzatuv uchun uchinchi chaqiruvda ifoda "minus" operatsiyasini o'z ichiga olgan, bu esa satrga o'tkazishni anglatmaydi. Shuning uchun, qatorni raqamga aylantirishim kerak edi, buning natijasida NaN paydo bo'ldi.

Satrlarni birlashtirish - bu ikki qatorni bir qatorga birlashtirish. Masalan, ikki qatorni birlashtirish uchun + operatoridan foydalanishingiz mumkin.

```
var str1:String = "green";
var str2:String = " fruits ";
var str3:String = str1 + str2; // str3 == "greenish"
```

Quyidagi misoldagi kabi natijani olish uchun += operatoridan ham foydalanishingiz mumkin.

```
var str:String = "green"; str += " fruits "; // str == "greenish"
```

Bundan tashqari, String quyidagi tarzda ishlatilishi mumkin bo'lgan concat() usulini o'z ichiga oladi.

```
var str1:String =
"Bonjour"; var
str2:String = "from"; var
str3:String = "Paris";
var str4:String = str1.concat(" ", str2, " ", str3);
// str4 == "Bonjour from Paris"
```

+ (yoki +=) operatori String ob'ekti va string bo'lmagan ob'ektni birlashtirish uchun ishlatilsa, ActionScript quyida ko'rsatilganidek, ifodani baholash uchun satsiz ob'ektni avtomatik ravishda String ob'ektiga aylantiradi. var str:String = "Area = "; var area:Number = Math.PI \* Math.pow(3, 2); str = str + area; // str == "Area = 28.274333882308138"

Biroq, quyidagi misoldagi kabi + operatori uchun kontekstni ta'minlash uchun atamalar qavslar yordamida guruhlanishi mumkin.

```
trace("Total: $" + 4.55 + 1.45); // output: Total: $4.55145
trace("Total: $" + (4.55 + 1.45)); // output: Total: $6
```

Satlardagi satrostilar va shablonlarni qidirish

Satrosti - bu satrdagi belgilar ketma-ketdir. Masalan, "abc" qatori quyidagi satrostilarga ega:

"", "a", "ab", "abc", "b", "bc", "c". Stringdagi satrostilarni topish uchun ActionScript usullaridan foydalanishingiz mumkin. ActionScript-da shablonlar satrlar yoki oddiy ifodalar yordamida aniqlanadi. Masalan, quyidagi muntazam ibora ma'lum bir qolipni belgilaydi: A, B va C dan keyin raqam keladi (qiyshiq chiziq muntazam ifodalarda chegaralovchi hisoblanadi).

```
/ABC\d/
```

ActionScript satrlardagi naqshlarni qidirish va topilgan mosliklarni o'zgartirish pastki qatorlari bilan almashtirish usullarini o'z ichiga oladi (o'zingiz o'rganing).

Oddiy iboralar sizga juda murakkab naqshlarni aniqlash imkonini beradi (o'zingiz o'rganing).

### Belgilar joylashuvi bo'yicha satrostini topish

`Substr()` va `substring()` usullari umumiy jihatlariga ega. Ularning ikkalasi ham berilgan satrning satrostini qaytaradi. Ularning ikkalasi ham ikkita parametrga ega. Ikkala usulda ham birinchi parametr berilgan satrdagi boshlang'ich belgining o'rnini belgilaydi. Biroq, `substr()` usulida ikkinchi parametr uzunlik bo'lib, u qaytarilgan pastki satr uzunligini o'z ichiga oladi, `substring()` usulida esa ikkinchi parametr end parametr bo'lib, u oxiridagi belgi o'rnini o'z ichiga oladi. pastki qatorning (qaytarilgan qatorga kiritilmagan). Ushbu misol ikki usul o'rtasidagi farqni ko'rsatadi.

```
var str:String = "Hello from Paris, Texas!!!";
trace(str.substr(11,15)); // output: Paris, Texas!!!
trace(str.substring(11,15)); // output: Pari
```

`slice()` usuli `substring()` usuliga o'xshaydi. Parametr sifatida manfiy bo'lmagan butun sonlar berilganda, u xuddi shunday ishlaydi. Biroq, `slice()` usuli manfiy raqamlarni ham parametr sifatida qabul qilishi mumkin. Bunday holda, belgilarning o'rnini quyidagi misoldagi kabi qator oxiridan boshlab hisoblanadi.

```
var str:String = "Hello from Paris, Texas!!!";
trace(str.slice(11,15)); // output:
Pari      trace(str.slice(-3,-1)); //
output: !! trace(str.slice(-3,26));
// output: !!!
```

```
trace(str.slice(-3,str.length)); // output: !!!
trace(str.slice(-8,-3)); // output: Texas
```

`slice()` usulining parametrlari sifatida siz manfiy va manfiy bo'lmagan sonlar kombinatsiyasini belgilashingiz mumkin.

### Mos keladigan satrostining belgilar o'rnini topish

`indexOf()` va `lastIndexOf()` usullari quyidagi misolda bo'lgani kabi satrda mos keladigan pastki qatorlarni topish uchun ishlatilishi mumkin.

```
var str:String = "The moon, the stars, the sea, the
land"; trace(str.indexOf("the")); // output: 10
```

E'tibor bering, `indexOf()` usuli katta-kichik harflarga sezgir. Quyida ko'rsatilganidek, qidiruvni boshlash uchun satrdagi indeks o'rnini belgilash uchun ikkinchi parametrga belgilashingiz mumkin. var

```
str:String = "The moon, the stars, the sea, the land"
trace(str.indexOf("the", 11)); // output: 21 lastIndexOf() usuli satrdagi
pastki qatorning oxirgi takrorlanishini topadi. var var str:String = "The
moon, the stars, the sea, the land" trace(str.lastIndexOf("the")); //
output: 30
```

`lastIndexOf()` usuli uchun ikkinchi parametrga belgilasangiz, qidiruv belgilangan joydan

teskari tartibda (o'ngdan chapga) boshlab amalga oshiriladi.

```
var str:String = "The moon, the stars, the sea, the
land" trace(str.lastIndexOf("the", 29)); // output: 21
```

Ajratuvchilar asosida bo'lingan satrosti massivini yaratish

Split() usulidan foydalanilgan ajratuvchi belgilar asosida ajratilgan pastki qatorlar qatorini yaratish uchun foydalanish mumkin. Masalan, vergul yoki yorliq bilan ajratilgan qatorni bir nechta pastki qatorlarga ajratish mumkin. Quyidagi misolda massivni chegaralovchi sifatida ampersand

(&) yordamida ichki qatorlarga qanday ajratish mumkinligi ko'rsatilgan.

```
var queryStr:String =
"first=joe&last=cheng&title=manager&StartDate=3/6/65";
var params:Array = queryStr.split("&", 2);
// params == ["first=joe", "last=cheng"]
```

Split() usulining ikkinchi, ixtiyoriy parametri qaytarilgan massivning maksimal hajmini belgilaydi.

#### Mos keladigan satrostilarni topish

Search() usuli quyidagi misolda bo'lgani kabi berilgan naqshga mos keladigan birinchi pastki qatorning indeks o'rnini qaytaradi. `var str:String = "The more the merrier."; // (This search is case-sensitive.)` `trace(str.search("the"));` // output: 9

Mos keladigan satrostilarni almashtirish `replace()` usulidan foydalanib, siz kerakli naqshni satrda topishingiz va quyidagi misolda bo'lgani kabi mos keladiganlarni berilgan almashtirish satri bilan almashtirishingiz mumkin.

```
var str:String = "She sells seashells by the seashore."; var
pattern:RegExp = /sh/gi; trace(str.replace(pattern, "sch")); //sche
sells seaschells by the seaschore.
```

#### Satrlarni katta yoki kichik harflarga aylantirish

Quyidagi misolda ko'rsatilganidek, `toLowerCase()` va `toUpperCase()` usullari satrning harfiy belgilarini mos ravishda kichik yoki katta harflarga aylantiradi.

```
var str:String = "Dr. Bob Roberts, #9."
trace(str.toLowerCase()); // dr. bob roberts, #9. T
trace(str.toUpperCase()); // DR. BOB ROBERTS, #9.
```

Ushbu usullar bajarilgandan so'ng, asl satr o'zgarishsiz qoladi. Asl satrni aylantirish uchun quyidagi koddan foydalaning:

```
str = str.toUpperCase();
```

Ushbu usullar faqat a-z va A-Z emas, balki kengaytirilgan belgilarni qo'llab-quvvatlaydi.

```
var str:String = "José Barça";
trace(str.toUpperCase(), str.toLowerCase());
```

## 5. typeof() funksiyasi

FLASH MXda ikkita maxsus tur mavjud bo'lib, ular uchun ob'ektning qiymati uning turi bo'yicha yagona tarzda beriladi.

#### null turi

Bu tip null bo'lib, u bo'shliqqa murojaat qilish uchun maxsus ishlatiladi. Null kalit so'zining mavjudligi Java kod qismlarini FLASH MX ga o'g'irishni osonlashtiradi.

**undefined turi undefined** - yana bir maxsus tur. Bu tipda qiymati avval o'rnatilmagan o'zgaruvchiga (obyekt, funksiya) tegishli ifoda mavjud. Biroq, agar u satrga aylantirilishi kerak bo'lgan ifodada aniqlanmagan bo'lsa, u bo'sh qatorga aylantiriladi. Raqamga aylantirilganda 0 olinadi, mantiqiy turga - false. Agar o'zgaruvchi yaratilgan bo'lsa va o'chirish bilan o'chirilgan

bo'lsa, bu o'zgaruvchining qiymatini so'ragandan so'ng, biz yana aniqlanmagan bo'lamiz. Esda tutingki, null va undefined ni == yoki != operatorlari bilan solishtirganda, biz ikki qiymat o'rtasidagi farqni ayta olmaymiz. Ammo bu qiymatlarning turli xil bo'lishi bizga Flash MXga xos bo'lgan == va != "qat'iy tenglik - tengsizlik" operatorlari yordamida muammoni hal qilishga yordam beradi.

### O'rnatilgan turlar ro'yxati. Typeof operatoridan foydalanish

Biz o'rnatilgan turlarni faqat maxsus typeof operatori turli qiymatlarni ishlab chiqaradiganlarini chaqiramiz.

**typeof operatoridan foydalanish sintaksisi: typeof <ifoda>**

yoki

**typeof(<ifoda  
a>)**

Ushbu operator o'tkazilgan typeof ifodasi tomonidan qaytarilgan qiymat turi nomini o'z ichiga olgan qatorni ishlab chiqaradi. Keyin 1-jadvalda ko'rsatilganidek, satr kichik harflarga qisqartiriladi. Jadvalda foydalanuvchi "qo'lda" yaratishi mumkin bo'lgan ob'ektlar - ya'ni kliplar, tugmalar va matn maydonlari qanday qabul qilinishi ham ko'rsatilgan. Ko'rinish turibdiki, vizual tarzda yaratilgan ob'ektlardan faqat kliplar typeof operatori bilan ajralib turadi. ActionScript tilining o'ziga kelsak, faqat Number, Mantiqiy, String, Ob'ekt, Funktsiya, undefined va null turdagi ob'ektlar har xil (noyob usullar to'plamiga kamaytirilmaydi) xatti-harakatlariga ega. Bundan tashqari, Arrayda qurish uchun maxsus operatorlar mavjud. Va faqat Number, Boolean, String, undefined va null turlari ibtidoiy ("faqat o'qish uchun").

### Turi bo'yicha qaytarilgan qiymatlar

3.5.1-jadval. Turi bo'yicha qaytarilgan qiymatlar

typeof qiymat turi keltirilgan	
tur	typeof qaytaradi
Number	Number
Boolean	Boolean
String	String
Object	Object
Function	Function
Array	Array
MovieClip	Movieclip
Button	Object
Text field	Object
undefined	Undefined
null	Null

Keling, ushbu operatorning ishini misollar bo'yicha kuzatamiz. Biz quyidagi kodni yozamiz:

```
trace(typeof(5 + "")) = ' + typeof(5 +
  "")); trace(typeof(a) = ' + typeof(a)); a
  = new Object();
```

```

trace(typeof(a) = ' +
typeof(a)); a = null;
trace(typeof(a) = ' +
typeof(a)); a = {x: 20, y:
"chiziq"}; trace(typeof(a) =
' + typeof(a));
trace(typeof(ax) = ' + typeof(ax));

```

Natija :

```

typeof(5 + "") = string
typeof(a) = undefined
typeof(a) = object
typeof(a) = null
typeof(a) = object
typeof(a.x) = number

```

## 6. Nazorat savollari



1. ActionScript xatolari uchun kod qanday tekshiriladi ?
2. `trace()` funksiyasining vazifasi?
3. ActionScript da arifmetik amallarni bajarasizmi ?
4. Qanday arifmetik amallarni qisqartirilgan holda yozish mumkin?
5. **Number** tipidagi o'zgaruvchiga qanday maksimal qiymat o'rnatilishi mumkin?
6. **Number** tipidagi o'zgaruvchiga qanday minimal qiymat qo'yish mumkin?
7. **Number** turining qanday maxsus qiymatlarini bilasiz?
8. String tipidagi o'zgaruvchini qanday yaratish mumkin ?
9. **length** usulining maqsadi?
10. Satrlarni birlashtirish qanday amalga oshirilishi mumkin?
11. Qanday qilib satrostini tanlash mumkin?
12. Qatordagi belgini qanday izlash mumkin?
13. `charAt()` va `charAt()` usullarining maqsadi ?
14. `typeof()` funksiyasining vazifasi?

## 2.4. Flash dasturida matematik hisob-kitoblarni amalga oshirish. Matematik ob'ektning metodlari va xususiyatlari. ParseFloat funksiyasi.

Reja:

1. Number (Sonlar uchun) turi. ActionScriptdagi sonlar turlari.
2. Sonlarni ifodalash xususiyatlari.
3. Satrlarni sonlarga aylantirish. `parseInt()` va `parseFloat()` funksiyalari.
4. Matematik funksiyalarni amalga oshirish xususiyatlari.
5. Matematik funksiyalar va konstantalar.
6. Nazorat savollari.



**Kalit soʻzlar:** *parseInt()* , *parseFloat()* , *floatingpoint* , *Color Mixer* , *sanoq tizimi*, *Math.E*, *Math.PI* , *NaN* , *isFinite()*, *Math.pow()*, *Math.abs()*, *Math.pow()*, *Math.sin()*, *Math.cos()*, *Math.tan()*, *Math.log()*.

## 1. Number(Sonlar uchun) turi . ActionScriptdagi sonlar turlari

Kompyuterda hisoblashlar tavsifida sonlar butun(integer) va haqiqiy ( real ), yoki suzuvchi nuqtali (floatingpoint) sonlarga boʻlinishi odatiy holdir . Birinchi tur faqat raqamlar (masalan, 122 yoki -896), ikkinchisi - raqamlar va kasr nuqtasi (5,8905 yoki 0,9067) bilan tuzilgan belgilarga mos keladi.

ActionScriptda real yoki integer son turlari mavjud emas, barcha sonlar 64-bitli suzuvchi nuqtali number turiga tegishlidir. Butun sonlar va kasrli sonlar oʻrtasidagi farq yaqqol koʻrinarli. Tilning ayrim operatorlari va funksiyalari faqat butun sonlar bilan ishlashi mumkin. Number turi suzuvchi nuqtali son boʻlganligi sababli, tilning ushbu vositalarini toʻgʻri ishlashini taʼminlash uchun 32 bitli va 16 bitli butun sonlarga aylantiruvchi ichki metodlari mavjud. Bundan kelib chiqadiki, ActionScript -da hali ham butun sonlar va suzuvchi nuqta raqamlariga boʻlinish mavjud, lekin faqat amalga oshirish darajasida. Oʻnli sanoq sistemasiga qoʻshimcha ravishda siz ActionScript da sakkizlik va oʻn oltilik sanoq sistemalaridagi butun sonlar bilan bevosita ishlashingiz mumkin . Tizim sonni sakkizlik sifatida qabul qilishi uchun u noldan boshlanishi kerak. Tabiiyki, qolgan raqamlar yettidan oshmasligi kerak. Satrga aylantirilganda sakkizlik son 10 asosga aylantiriladi (shuning uchun uni argument sifatida trace() funksiyasiga beriladi, Output panelida oʻnlik son chiqariladi).

**trace(0210); // Natija 136 (  $0 * 8^0 + 1 * 8^1 + 2 * 8^2 = 136$  )**

Amaliyotda qiziqroq va muhimroq oʻn oltilik sonlar formati hisoblanadi. Undagi raqamlar 0 dan 9 gacha boʻlgan raqamlar va lotin alifbosining A dan F gacha boʻlgan harflaridir. ActionScript da oʻn oltilik sonlar "0x" prefiksi yordamida belgilanadi. 16 asosli son avtomatik ravishda satrlarga aylantirilganda, sakkizlik sonlar kabi oʻnlik sanoq sistemasidagi songa aylantiriladi.

**trace(0xFF); // Natija: 255 (  $15 * 16^0 + 15 * 16^1$  )**

Oʻn oltilik sonlar birinchi navbatda ranglarni belgilash uchun ishlatiladi. Color Mixer panelida tanlangan rang uchun formulaga eʼtibor bering - olti xonali oʻn oltilik son. Uning birinchi ikkita raqami qizil komponentning intensivligi uchun, uchinchi va toʻrtinchi raqamlar yashil komponent uchun, beshinchi va oltinchi raqamlar koʻk komponent uchun javobgardir. Masalan, oq 0xFFFFFFFF, qora 0x000000, yashil 0x00DD00 va hokazo.

Tilni amalga oshirish darajasida sakkizlik va oʻn oltilik sonlar 32 bitli butun sonlarga mos keladi. Bu ular  $232 = 4294962795$  dan kattaroq qiymatni ifodalay olmasligini anglatadi. Shuning uchun 0xFFFFFFFFD5A kabi yozuv maʼnoga ega emas . Odatdagidan tashqari, maxsus sonli qiymatlar mavjud. Ular cheksizlikni, minimal va maksimal hisoblash qiymatini, yaroqsiz qiymatni belgilash uchun xizmat qiladi.

## 2. Sonlarni ifodalash xususiyatlari

Matematikada sonlar toʻplami cheksiz deb hisoblanadi. Bu son istalgancha katta yoki istalgancha kichik boʻlishi mumkinligini anglatadi. U har qanday raqamlar soni bilan tuzilishi mumkin. Demak, cheksiz kasrli sonlar va cheksiz transsendental doimiyalar mavjud (masalan, p

soni yoki e soni ). Lekin haqiqiy dunyoda hammasi chegaralangan. Hatto koinotning ham chegaralari bor. Kompyuter xotirasi ham cheklangan. Sonning har bir raqami uchun bir nechta bit ajratish kerak bo'lganligi sababli, har qanday kompyuter cheklangan aniqlik bilan hisoblaydi. Hisob-kitoblarning aniqligi har bir sonni saqlash uchun qancha xotira ajratilganligi bilan belgilanadi. Zamonaviy kompyuterlar suzuvchi nuqtali sonlarni ifodalash uchun 64 bitli ikki aniqlik formatidan foydalanadi (mos ravishda 32 bitli bitta aniqlik formati). Bu 2<sup>64</sup>-18 446 744 073 709 551 616 butun sonni bevosita kodlash imkonini beradi . Juda kam, shunday emasmi?

Amalda siz juda ko'p sonli qiymatlar bilan ishlashingizga to'g'ri keladi. Qanday qilib barcha kerakli sonlarni oddiy 64 bitga "siqish" mumkin? Ta'riflangan murakkablikning yechimi suzuvchi nuqta yordamida sonlarni ifodalash formatidir. Uning asosiy g'oyasi quyidagicha: Son ikki qismga bo'linadi - muhim qism (mantissa) va daraja. Masalan, 123 sonini  $1.23 \cdot 10^2$  deb yozish mumkin, 123000 soni –  $1.23 \cdot 10^5$ , 0,123 soni –  $1.23 \cdot 10^{-1}$ . Shu bilan birga, 64 bit orqali uzatiladigan raqamlar uchun mumkin bo'lgan variantlar soni keskin ortadi. To'g'ri, sonlarni ifodalashning aniqligi bundan oshmaydi - u har doim mantisning oxirgi belgisining soniga teng. Ammo bizni deyarli har doim cheklangan aniqlik natijalari qiziqirganligi sababli (ko'pincha verguldan keyin uchinchi raqamgacha), suzuvchi nuqtali sonlar modeli juda maqbuldir. Umuman olganda, son s . m . 2<sup>e</sup> shaklga ega, bu yerda s – sonning belgisi (1 bit), m – mantisani aniqlaydigan butun son (uning uchun 53 bit ajratilgan), e – -1074 dan 971 gacha daraja ko'rsatkichi (10 bit). Keling, berilgan raqamlarni tahlil qilaylik. Agar sonning muhim qismini 53 ta ikkilik raqam bilan aniqlash mumkin bo'lsa, unda uning o'nlik raqamlari soni  $15 \cdot \lg(2) \approx 15$  dan oshmasligi kerak. Bu shuni anglatadiki, hisoblash aniqligi 15 ta haqiqiy raqam bilan cheklangan. Agar ishlov berish uchun olingan sonning mantissining uzunligi 15 belgidan oshsa, u oddiygina yaxlitlanadi:

**trace (1,5555555555555557); // Natija: 1.55555555555556**

Agar butun son 15 dan ortiq raqamdan tuzilgan bo'lsa, u holda 15 ta muhim raqamga yaxlitlash ham amalga oshiriladi. Bunday holda, mantis o'nli kasr sifatida yoziladi va tartib quyidagi formatida beriladi:

**tarce (1234568901234567890); // Natija: 1.2345679012346e+19**

Kasr sonlar xuddi shu tarzda aylantiriladi:

**trace (0,0000001234567890123956789 ); // Natija: 1.2345678901234 e -7**

Suzuvchi nuqtali sonning ko'rsatkichini saqlash uchun cheklangan miqdordagi bitlar ajratilganligi juda katta yoki juda kichik qiymatlar bilan ishlashning mumkin emasligini anglatadi.

Hisob-kitoblarda qatnashishi mumkin bo'lgan maksimal songa mashina cheksizligi deyiladi. O'rnatilgan konstantalarga murojaat qilmasdan ham uni aniqlash juda oson. Suzuvchi nuqtali raqamlarni tasvirlashning tavsiflangan tamoyillariga asoslanib, u maksimal mumkin bo'lgan mantissani maksimal mumkin bo'lgan darajaga ko'paytirishga ega bo'lishi kerakligi aniq. Ikkilik tasvirda bu haqiqiy qismi 53 birlikdan tashkil topgan raqam va tartib 2 ning 971 darajasiga teng bo'ladi. Keling, bu sonning kasr qiymatini aniqlashga harakat qilaylik:

**var n = 0;**

**// Takrorlanish yig'indini aniqlaydi**

**// 252 + 251 +250 +249 +...+21 +20 - mantissaning maksimal qiymati**

**for ( i =0; i <53; i ++)**

**{ n += Math.pow(2, i ); }**

// Maksimal mantissani maksimal darajaga  
ko'paytiramiz. `trace(n*Math.pow(2, 971));` //  
Natija : 1,797 b 9313486231 e +308

Olingan qiymat - qidirilayotgan mashina cheksizligining taxminiy qiymati.

ActionScript da uni belgilash uchun maxsus doimiy Infinity dan foydalaniladi. Agar ishlov berish uchun olingan son (absolyut qiymatda) mashinaning cheksizligidan oshib ketganligi aniqlansa, u bilan almashtiriladi:

```
trace(1e+400); // Natija: Infinity
```

Infinity qiymati ko'pincha hisob-kitoblarda qaytariladi. Ko'p holatda nolg'a bo'linishdir:  
`trace(-1/0);` // Natija: - Infinity (cheksizlik manfiy bo'lishi mumkin)

Infinity so'zi interpretator tomonidan mos keladigan sonli qiymat sifatida qabul qilinadi.

Shuning uchun siz uni hisob-kitoblarda oddiy sonlar bilan bir xil tarzda ishlatishingiz mumkin:  
`iz (1/ Infinity );` // Natija: 0      `iz ( Infinity *0);` // Natija: NaN (bunday ko'paytma aniqlanmagan)

Ba'zan sonning chekli ekanligini aniqlash kerak bo'ladi. Buni argument muvaffaqiyatsizlikka yoki noaniqlikka olib keladigan cheksizlikni o'tkazishda qilish kerak (masalan, sinusi Infinity da chekli qiymatga ega emas). Bu muammoni ikki yo'l bilan hal qilish mumkin. Birinchi yo'li, ishlov berish uchun olingan qiymatni Infinity bilan taqqoslashingiz mumkin :

```
var value =  
Infinity; if (value =  
Infinity) {  
    trace ("Qiyamat noto'g'ri"); // Natija: Qiyamat  
    noto'g'ri }  
Ikkinchi yo'li, isFinite( ) maxsus global funksiyasidan foydalanishingiz mumkin. Bu
```

funksiya , agar unga berilgan son chekli bo'lsa true, cheksizlikka teng bo'lsa false qiymatini qaytaradi. Bu funksiyaning kamchiligi shundaki, u argument son bo'lmaganda ham false qaytaradi.

```
var value = infinity; agar  
( isFinite (value) = false) {  
    trace ( "Noto'g'ri qiyamat" ) // Natija: Noto'g'ri  
    qiyamat }  
Suzuvchi nuqtali sonning musbat qiymatli darajalari uchun emas, balki manfiylar uchun
```

ham cheklov mavjud. Bu shuni anglatadiki, barcha kichik sonlar 0 sifatida qabul qilinadigan son mavjud. Bu son mashina noli deb ataladi va quyidagiga teng:

```
1 . 2 -1024 ≈ 4,94065645841247 . 10 -324  
trace (1e-400); // 0 chiqadi
```

E'tibor bering, Number sinfining metodini songa qo'llash uchun u umuman ushbu inf obykti bo'lishi shart emas. toString() metodi tub sonlar uchun ham yaxshi ishlaydi. Number() sinfining ikkita mavjud metodlaridan ikkinchisi, sinf ob'ektlarini number elementar turiga aylantirish uchun xizmat qiluvchi valueOf() metodidir. Bu metod amaliy ahamiyatga ega emas, shuning uchun biz uni batafsil ko'rib chiqmaymiz. Number klassi shuningdek, bir nechta o'rnatilgan xususiyatlarga ega. Ular cheksizlik, maksimal va minimal belgilangan sonlar, NaN kabi eng muhim sonli konstantalarni saqlash uchun mo'ljallangan. Biz ushbu qiymatlarning barchasiga duch kelganimiz sababli, biz ularga mos keladigan xususiyatlarni sanab o'tamiz:

**Number.MAX\_VALUE** // Maksimal belgilangan son (1.79769313486231e+308)

**Number.MIN\_VALUE** // Minimal belgilangan son (4.940656458412476-324)

**Number.NEGATIVE\_INFINITY** //  $-\infty$  (-Infinity ni qaytaradi) **Number.POSITIVE\_INFINITY** //  $+\infty$  (+Infinity ni qaytaradi)

**Number.NaN** // Noto'g'ri sonli qiymat (NaN ni qaytaradi). Number klassi nima uchun foydali bo'lishi mumkin, bu foydalanuvchi matematik metodlarini yaratish uchundir. Masalan, sonning faktorialini hisoblash funksiyasini yozish qiyin emas (faktorial bu sonning o'zigacha bo'lgan butun sonlar ko'paytmasidir) va u toString() kabi o'rnatilgan metodlar kabi ishlaydi. /\* Sinf metodi - bu o'z konstruktor prototipida saqlanadigan funksiya va avtomatik ravishda shu sinf obyektlarining har birining meros zanjiriga o'tadi, Number sinf konstruktori prototipining "manzili" " Number.prototype" bo'ladi. Yaratilgan metodni faktorial deb ataymiz.\*/ Number.prototype.factorial = function() { /\*Metodga beriluvchi son faktorial hisoblash qoidasiga to'g'ri kelishini tekshirish kerak. U chekli qiymatga ega bo'lishi kerak ( NaN yoki Infinity emas ), manfiy bo'lmasligi va butun son bo'lishi kerak. Oxirgi shartni argumentdan Math.round() metodi bilan yaxlitlash natijasini ayirish orqali tekshirish mumkin. Agar son butun son bo'lsa, yaxlitlangan qiymat asl qiymat bilan bir xil bo'ladi va ularning farqi 0 ga teng bo'ladi. Argument sifatida this kalit so'zidan foydalanishingiz kerak : u metodni chaqirgan obyektga ishora qiladi.\*/ if (isNaN(this) == false && isFinite(this) == true && this >= 0 && this - Math.round(this) == 0){

/\* { Agar son qiymatida belgilangan cheklovlar bajarilsa, keyin faktorial hisoblaymiz. Buning uchun biz 1 dan this gacha (metodni chaqirgan son) butun o'zgaruvchi ustida takrorlanishni o'tkazamiz va har bir iteratsiyada oldingi qiymatni factor o'zgaruvchisiga ko'paytiramiz. factor uchun boshlang'ich qiymat 1 bo'lishi kerak.\*/ var factor = 1, for (var i = 1; i <= this; i++) { factor \*= i ; }

// return buyrug'i yordamida faktorial qiymatini qaytariladi

return factor; }

/\* Agar metodni chaqirgan son noto'g'ri qiymatga ega ekanligi aniqlansa, NaN qiymati qaytariladi\*/ else { return NaN ; }

};

var a=0, b=5, c=-3; trace(a.faktorial()); // Natija: 1

trace(b.faktorial()); // Natija: 120

trace(c.faktorial()); // Natija: NaN

Ko'rib turganingizdek, biz yaratgan metod xuddi o'rnatilgan ActionScript metodi kabi ishlaydi. ActionScript obyekt modelining imkoniyatlaridan foydalanish ko'p vazifalarni faqat "oddiy dasturlash" dan foydalanishdan ko'ra ancha oqilona hal qilish imkonini beradi.

Number sinfining metodlari to'g'ridan-to'g'ri songa uni o'zgaruvchida saqlamasdan qo'llashi mumkin. To'g'ri, buning uchun uni qavs ichida olish kerak, aks holda nuqta xususiyatlar va metodlarga kirish operatori sifatida emas, balki kasr qismiga o'tish belgisi sifatida qabul qilinadi. Misol:

trace ((1000).toString (16)); // Natija: 3e8

### 3. Satrlarni sonlarga aylantirish. parseInt() va parseFloat () funksiyalari

Sonli qiymatlarni satrlarda saqlash amalda juda keng tarqalgan usuldir. Masalan, shu tarzda boshqa sanoq sistemalarida ifodalangan miqdorlarni ko'rsatish mumkin. Masalan, rang

kodini o'nlik formatga qaraganda o'n oltilik formatda saqlash ancha qulayroq. Agar siz u bilan son sifatida ishlasangiz, Output panelida va matn maydonlarida faqat o'nlik qiymati ko'rsatiladi. Kodni satrga o'tkazish orqali siz uni kerakli formatda chiqarishingiz mumkin. Tarmoqdagi ma'lumotlarni uzatish uchun sonli ma'lumotlarni satrlarda saqlash alohida ahamiyatga ega. Satrlarni sonlarga aylantirishning eng oddiy funksiyasi allaqachon tanish bo'lgan Number() funksiyasidir. U quyidagi shartlarda o'z vazifalarini to'g'ri bajarishi mumkin: Satrda raqamlardan boshqa belgilar bo'lmasligi kerak, + yoki - son belgisi yoki daraja belgisini ko'rsatish uchun, e harfi daraja kiritish uchun. Son o'nlik formatda bo'lishi kerak.

```
trace(Number("-454e+5")); // Natija: -45400000 (to'g'ri satr formati)
```

```
trace(Number("FFFF00")); // Natija: NaN (funksiya o'n oltilik sonlarni //tanimaydi)
```

• Satrdagi sonlardan oldin probellar va bosh nollar qo'yilishi mumkin. Oxirgi shart, agar sonda 8 yoki 9 sonlari bo'lsagina to'g'ri bo'ladi. Aks holda, u sakkizlik deb talqin qilinadi.

```
trace(Number("00067")); // Natija: 55 (sakkizlik son sifatida tan olingan)
```

```
trace(Number("000678")); // Natija: 678 (8 sonining mavjudligi uchun //o'nlik format)
```

O'n oltilik va sakkizlik qiymatlari bo'lgan satrlar mos ravishda 0x va 0 prefikslari bilan boshlanishi kerak.

```
trace(Number("0x99FA")); // Natija: 39418 (to'g'ri belgi ishlatilgan)
```

```
trace(Number("99FA")); // Natija: NaN ("0x" yo'qligi uchun o'n oltilik son //tanib olinmaydi)
```

Number() funksiyasidan tashqari, ActionScript satrlarni sonlarga aylantirish uchun ancha kuchliroq xususiyatlarga ega funksiyalar mavjud. Ular parseInt() va parseFloat() global funksiyalaridir. parseInt(string, radix) funksiyasi satrdan butun sonni chiqarish uchun ishlatiladi. U ikkita parametrni qabul qiladi: string - aylantirilishi kerak bo'lgan satr, radix - 2 dan 32 gacha bo'lgan son, interpretatorga tanlangan qiymat qaysi sanoq tizimida ekanligini ko'rsatadi (agar u o'rnatilmagan bo'lsa, o'nlik sanoq tizimi deb tushunadi).

```
trace(parseInt("10101101", 2)); // Natija: 173 (ikkilik sanoq tizimi)
```

```
trace(parseInt("10101101")); // Natija: 10101101 (o'nlik sanoq tizimi)
```

parseInt() funksiyasi butun sonni satrdan ajrata oladi, hatto unda sonli bo'lmagan belgilar, + va - belgilar va daraja ko'rsatkichi bo'lsa ham. Bunday holda, birinchi noto'g'ri belgigacha bo'lgan belgilar ketma-ketligi songa aylantiriladi. Agar satr harf yoki operator bilan boshlasa, funksiya NaN ni qaytaradi.

```
trace(parseInt("2747464 - mening telefonim")); // Natija: 2747464 (son // satr boshida)
```

```
trace(parseInt("Mening telefonim - 2747464")); // Natija: NaN (satr // harf bilan boshlanadi, shu uchun NaN)
```

Satrdan tanlangan sondan oldin bo'sh joylar va bosh nol qo'yilishi mumkin. Ammo shuni yodda tutingki, agar nol parametri sondan oldin tushirilsa, u sakkizlik prefiks sifatida talqin qilinadi. parseInt() funksiyasi 0x o'n oltilik prefiksini ham taniy oladi.

```
trace(parseInt("000123")); // Natija: 83 (sakkizlik deb talqin qilingan)
```

```
trace(parseInt("000123", 10)); // Natija: 123 (aniq asos, shuning uchun //bosh nollarga
```

e'tibor berilmaydi) trace(parseInt("0xFFAA")); // Natija: 65450 ("0x" o'n oltilik bazani //bildiradi) parseInt() ga juda o'xshash parseFloat() funksiyasi satrdan suzuvchi nuqta formatidagi sonni ajratish uchun mo'ljallangan. parseFloat() funksiyasi faqat bitta parametrni oladi, o'nlik formatda sonli qiymat berilishi kerak bo'lgan satr.

```
trace(parseFloat("3.456e+7")); // Natija: 34560000
```

```
trace (parseFloat ("-034ABC")); // Natija: 34 (boshlovchi nollar va // bo'shliqlar hisobga olinmaydi)
```

Ta'riflangan funksiyalar orasida parseInt() ayniqsa qimmatlidir, chunki undan foydalanib, ixtiyoriy sanoq tizimlarining qiymatlarini o'nli sonlarga aylantirish mumkin.

#### 4. Matematik funksiyalarni amalga oshirish xususiyatlari

Algoritmarda matematik funksiyalardan foydalanganda, ularning kompyuterda qanday amalga oshirilishi haqida tasavvurga ega bo'lish muhimdir. Gap shundaki, matematik funksiyalar vazifasining o'ziga xos xususiyatlari tufayli ular tomonidan qaytariladigan qiymatlar har doim ham rost qiymatdan uzoqdir. Agar bu e'tiborga olinmasa, algoritim muvaffaqiyatsiz bo'lishi mumkin, uni topish juda qiyin bo'ladi. Suzuvchi nuqtali sonlar ustidagi qo'shish, ko'paytirish yoki bo'lish kabi asosiy arifmetik amallar kompyuter tomonidan apparat darajasida qo'llab-quvvatlanadi, ya'ni ular bevosita protsessor tomonidan amalga oshiriladi. Shuning uchun bunday operatsiyalarning tezligi juda yuqori bo'lishi mumkin. Sinus yoki logarifmni hisoblash bilan bir xil operatsiyalar apparat tomonidan qo'llab-quvvatlanmaydi. Buning sababi, buning uchun zarur bo'lgan algoritimni "apparatda" amalga oshirishning iloji yo'qligida emas. Yo'q, bu mutlaqo mumkin bo'lgan vazifa (bundan tashqari, mikroelektronikaning asosiy aksiomasi har qanday algoritimni apparatda amalga oshirish mumkinligini aytadi). Tegishli elementni protsessorga oddiygina kiritish mantiqqa to'g'ri kelmaydi, chunki faqat apparat tomonidan qo'llab-quvvatlanadigan arifmetik operatsiyalardan foydalangan holda dasturiy jihatdan matematik funksiyalarni yaratish juda oson. Umumiy holatda ifodalab bo'lmaydigan matematik funksiyalarni sinus, kosinus yoki logarifm kabi arifmetik amallar orqali amalga oshirishning zamirida har qanday matematik tahlil kursida isbotlangan juda muhim g'oya mavjud. Uning mohiyati shundan iboratki, ma'lum cheklovlariga duchor bo'lgan funksiya ma'lum bir nuqtaga yaqin joyda darajali qator bilan almashtirilishi mumkin. Darajali qator yig'indili ko'rinishda bo'ladi,

$$a_0 \cdot x^0 + a_1 \cdot x^1 + a_2 \cdot x^2 + a_3 \cdot x^3 + \dots + a_n \cdot x^n, \text{ gde } a_1, a_2, a_3, \dots, a_n$$

Berilgan funksiya uchun unga yaqinlashuvchi qatorni topish (aniqlash) darajali qatorga yoyish deb ataladi.

Yoyilma hadlari qanchalik ko'p topilsa, yaqinlashuvchi funksiya shunchalik aniqroq bo'ladi. Shuningdek, yaqinlashishning to'g'riligi berilgan nuqta yoyish amalga oshirilgan nuqtadan qanchalik uzoqda ekanligiga bog'liq. Masalan, sinus uchun yoyish nuqtasidagi darajali qatori  $a=0$  ko'rinishga ega (12 ta yoyilma hadlari, ulardan 6 tasi 0 ga teng):

$$\sin(x) = 1 \cdot x - \frac{1}{6} \cdot x^3 + \frac{1}{120} \cdot x^5 - \frac{1}{5040} \cdot x^7 + \frac{1}{362880} \cdot x^9 - \frac{1}{39916800} \cdot x^{11}.$$

Funksiya yoyilgan qatorga asoslanib, uning qiymatlarini aniqlash algoritimini yozish qiyin emas. Yagona narsa, yoyilmaning qancha hadlarini olish kerakligini to'g'ri baholashdir, shunda yaqinlashishning aniqligi yetarli bo'ladi. Buning uchun maxsus formulalar mavjud.

Matematik funksiyalar yaqinlashuvchi qator yordamida hisoblanadi. Bunday hisobkitoblarning to'g'riligi ideal holda sonlarni ifodalashning aniqligiga tengdir. Bu mantisaning 1415-belgilarida deyarli xato bo'lishini anglatadi. Albatta, aksariyat hollarda bu muhim emas. Ammo ba'zida bu xato halokatli bo'lishi mumkin. Bu, birinchi navbatda, uzilish nuqtalari va funksiyalarning nollariga tegishli. Masalan, har qanday o'quvchi biladiki  $n$  butun son uchun  $\sin(n \cdot \pi) = 0$  ga teng ekanligi aniq. ActionScript da  $\sin(n \cdot \pi) = 0$  tengligi deyarli kuzatilmaydi (yuqoridagi misolga qarang). Deyarli har qanday nuqtada har qanday

funksiyaning qiymati haqida ham xuddi shunday deyish mumkin: u mantisning 15-belgisiga qadar aniq bo'lmaydi. Masalan :

**trace(Math.tan(Math.PI/2));** // Natija: 1. 63317787283833e+16

// ( Infinity bo'lishi kerak )

**trace(Math.sin ( Math.PI /4)-Math.SQRT1\_2);**

// Natija: -1.11022302462516 e -16 (0 bo'lishi kerak)

Matematik funksiyalarning qiymatlarida deyarli har doim xato bo'lishidan juda yomon oqibatlar kelib chiqadi: funksiya tomonidan qaytarilgan qiymatni qaysidir nuqtada ushbu funksiyaning aniq qiymati bilan solishtirishdan oldin, hech bo'lmaganda yaxlitlash kerak. 12-13 kasr. Aks holda, tenglik hech qachon to'g'ri kelmaydi.

## 5. Matematik funksiyalar va konstantalar

### Darajaga ko'tarish va kvadrat ildiz

ActionScriptda sonni ixtiyoriy darajaga ko'tarish **Math.pow(x, a)** metodi yordamida bajariladi, bunda  $x$  – son va  $a$  – darajadir. Agar  $x$  musbat bo'lsa,  $u$  holda  $a$  mutlaqo har qanday qiymatni qabul qilishi mumkin – daraja ko'rsatkichi ham butun, ham kasrli, ham musbat, ham manfiy bo'lishi mumkin. Masalan: **trace( Math.pow ( 2,256 ) );** // Natija:

1.15792089237316 e +77 **trace( Math . pow ( 2,-1,3\*156));** // Natija: 8.93545926600057 e -62

Agar  $x$  manfiy bo'lsa,  $u$  holda ko'rsatkich faqat butun son bo'lishi mumkin (ham musbat, ham manfiy). Kasr quvvatini hisoblashga urinish NaN ni qaytaradi . Bu juda muhim xususiyat, xususan,

-27 ning kub ildizini hisoblab bo'lmaydi, garchi  $u$  mavjud bo'lib va -3 ga teng bo'lsa ham.

Misolalar: **trace( Math.pow (-2 , 256));** // Natija: 1.15792089237316 e +77

**trace( Math . pow (-2, -256));** // Natija: 8.63616855509445e-78

**trace( Math.pow (-27, 1/3 ) );** // Natija: NaN

**Math.pow ( )** metodidan samarali foydalanish uchun daraja uchun bir belgi boshqasiga qanday almashtirilishini bilish muhimdir. Shunday qilib, ko'pincha amalda har xil turdagi ildizlarni hisoblash kerak. Oddiy ko'rinishidagi yozuvga ildizdan o'tish imkonini beruvchi quyidagi formulani eslab, bunda ko'rib chiqilayotgan metod ishlaydi: Kvadrat va kub ildizlarni hisoblashga misol keltiramiz: **trace( Math.pow (9,1/2));** // Natija : 3 ( $3^2=9$ ) **trace( Math.pow ( 27,1/3));** // Natija : 3 ( $3^3=27$ )

Murakkab kasrlarni hisoblashni soddalashtirishga imkon beradigan quyidagi munosabatni eslab qolish ham foydalidir:  $x^{-n} = 1/x^n$

Biroq, algebraik o'zgarishlar formulalarini juda ehtiyotkorlik bilan qo'llash kerak. Shuni yodda tutingki, yuqorida tavsiflangan sabablarga ko'ra, algebraik jihatdan bir xil ikkita ifoda turli xil natijalar berishi mumkin. Shuning uchun algoritmda bir xil ifoda faqat bitta belgida ifodalaniishi kerak:

**trace(1/ Math.pow ( 9, 1/2)=== Math.pow (9,-1/2));** // Natija: false

**trace(1/Math.pow(9,1/2)-Math.pow(9,-1/2));**

// Natija: -5,55111512312578e-17

Eng keng tarqalgan darajaga ko'tarish operatsiyasi, albatta, kvadrat ildizga oshirishdir. Kvadrat ildizni hisoblash uchun ActionScript da maxsus **Math.sqrt()** metodi mavjud:

**trace(Math.sqrt (9));** // Natija: 3

Math.pow() metodi kvadrat ildizni ham hisoblashi mumkin bo'lsa, Flash dasturchilari nima uchun Math.sqrt() metodini yaratishi kerakligini so'rashingiz mumkin. Gap shundaki, ActionScript bilan ishlaydiganlar orasida matematikani bilish umuman shart bo'lmagan ko'plab dizaynerlar bor, Math.pow() dan foydalanish algebra bo'yicha ma'lum bilimlarni talab qiladi (ya'ni 0,5 daraja kvadrat ildiz ekanligini bilishi kerak). Bundan tashqari, bajarilgan vazifalarning qisqaligi tufayli Math.sqrt () Math.pow () ga qaraganda bir oz samaraliroqdir, bu katta hisobkitoblarni amalga oshiradigan kodda muhim ahamiyatga ega.

### EkspONENTA va natural logarifm

Matematikada e soni (eksponenta) katta ahamiyatga ega, taxminan 2,718 ga teng (aniqroq qiymat Math.E da saqlanadi). Uni o'z ichiga olgan ifodalar juda ko'p (masalan, yadroviy parchalanish). Odatda, eksponenta ma'lum darajada fizik va matematik formulalarga kiradi. Shuning uchun, ActionScript da  $e^a$  ni hisoblash imkonini beruvchi maxsus Math.exp(a) metodi borligi ajablanarli emas.  $a$  - har qanday son. Masalan: `trace (Math.exp (-1/2) ); // Natija: 0.606530559712633` `trace (Math.pow (Math.E,-1/2)); // Natija: 0.606530659712633`

Yuqoridagi misoldan ko'rinib turibdiki, Math.exp () metodi hech qanday muhim emas: uni osonlik bilan Math.pow () metodi bilan almashtirish mumkin. Ekspontensial funksiyadan ko'ra natural logarifm muhimroq ( $e$  asosli logarifm). Uni hisoblash uchun Math.log () metodidan foydalaniladi: `trace(Math.log(Math . E)); // Natija: 1`  
`trace( Math.log (10)= Math .LN10) ; // Natija: rost`

ActionScript da 10 li logarifmni hisoblaydigan metod yo'q. Logarifmlarni ixtiyoriy asosga hisoblash imkonini beradigan metod yo'q. Biroq, agar kerak bo'lsa, tegishli metodlarni osongina yaratish mumkin. Buning uchun  $n$  asosli logarifmni natural logarifmlar bilan ifodalash imkonini beruvchi quyidagi formuladan foydalanish kerak:

$$\log_n(x) = \frac{\ln(x)}{\ln(n)}$$

### Sonning absolyut qiymati

Sonning moduli uning absolyut qiymatidir. ActionScript modulni aniqlash uchun Math.abs() metodidan foydalanadi. Misol:

```
trace( Math.abs (-1)); // Natija: 1
```

Sonning moduli manfiy qiymatdan foydalanish mumkin bo'lmaganda qo'llaniladi. Bundan tashqari, agar ifoda belgisini 1 yoki -1 sifatida ajratib ko'rsatishda foydali bo'ladi. Math.abs() metodidan foydalanishga misol sifatida bu yerda ham musbat, ham manfiy sonlarning kub ildizini hisoblaydigan funksiya keltiriladi ( Math.pow() metodi faqat musbat sonlarning ildizini topishi mumkin):

```
function root1_3( arg ) { if ( arg =0) { //Nolga bo'linishni oldini olish uchun shart return 0; } }
```

```
var znak = Math.abs(arg)/arg; // Argument belgisini chiqarib olamiz
//Argument modulining ildizini hisoblaymiz va natijani
qaytramiz return Math.pow (Math.abs(arg), 1/ 3)* znak ; }
trace(root1_3(27) ); // Natija : 3
trace( root1_3 (-27)); // Natija : -3
```



## Tasodifiy sonlar

Hayotimizdagi ko'p narsalar tasodifiydir. Qor yog'ishi, ruletda g'alaba qozonish, tug'ilgan kun - bularning barchasi tasodifga bog'liq. Kompyuterda biron bir tasodifiy hodisaning modelini yaratishda noaniqlik manbasiga ega bo'lishingiz kerak. Tasodifiy sonlar generatorlari shaklida taqdim etilgan bunday manbalar har qanday dasturlash tilida mavjud. ActionScript dagi tasodifiy sonlar generatori `Math.round()` metodini taqdim etadi. Bu metod 0 dan 1 gacha bo'lgan tasodifiy sonni qaytaradi. 1 000 000 000 000 000 ta mumkin bo'lgan sonlarning har birining paydo bo'lish ehtimoli bir xil va ahamiyatsiz. Xuddi shu tasodifiy sonning doimiy ravishda yaratilishi amalda mumkin emas (shuningdek, ma'lum bir qiymatning tushib qolish ehtimoli ahamiyatsiz). Amalda, odatda N dan M gacha bo'lgan oraliqda tasodifiy qiymatlarni olish kerak bo'ladi, bu yerda N va M 0 va 1 dan farq qiladi. Bunday holda, `Math.random()` bilan birgalikda maxsus multiplikatoridan foydalanish kerak. N va M o'rtasida tasodifiy sonlarni yaratishning umumiy formulasi:  $(M - N)$

× `Math.tasodifiy () + N`. Masalan , 0 dan 100 gacha tasodifiy sonni yaratish uchun siz quyidagilarni yozishingiz kerak:

```
trace(100*Mathematik ).tasodifiy ()); // Natija: 1.4111054223031 -
```

```
10 dan 20 gacha bo'lgan tasodifiy son quyidagi kodni beradi:
```

```
trace(30*Math.random()-10); // Natija: -4.52681495999017 Math.randomO  
tomonidan
```

qaytarilgan son 15-mantisga qadar tasodifiy. Bu har doim ham qabul qilinishi mumkin emas. Ba'zan cheklangan qiymatlar to'plamidan qiymat yaratish kerak bo'ladi. Masalan, "Battleship" o'yinida kemalarni joylashtirishda 1 dan 100 gacha sonlarni yaratish kerak. Ushbu muammoni hal qilish uchun `Math.random ()` metodi yaxlitlash metodlaridan biri bilan birgalikda qo'llanilishi kerak (keyingi kichik bo'limga qarang):

```
trace( Math.ceil ( Math.random()*100)); // Natija: 56
```

Tasodifiy sonlar generatoridan foydalanishga misol sifatida molekulaning Broun harakatini belgilovchi kodni keltiramiz: // Molekulani kichik doira shaklida chizing. // Uni klipga aylantiring va molekula nomini bering . `_root.onEnterFrame = funksiya () { molecula . _x += 20* Math.random()-10; molecula . _y += 20* Matematik.random()-10; }`;

## Eng yaqin butun songa yaxlitlash

Ko'pgina xususiyatlar, parametrlar va xarakteristikalar butun sonlar sifatida ifodalanadi.

Misol uchun, rang 0 dan 16 777 215 gacha butun son sifatida kodlangan (o'n oltilik, 0x000000 dan 0xFFFFF gacha). Rang formulasini tasodifiy yaratish uchun 0xFFFFFni `Math.random()` metodi bilan qaytarilgan tasodifiy songa ko'paytirish kerak, ammo bu murakkablikni keltirib chiqaradi - natijada ruxsat berilmagan kasr son bo'lishi mumkin:

```
trace(0xFFFFF * Math.random ()); // Natija: 966372.747087192 To'g'ri
```

rang formulasini olish uchun hisoblash natijasi butun sonlarga yaxlitlanishi kerak. Buning uchun ActionScript da uchta metod mavjud, ulardan eng samaralisi `Math.round()`:

```
trace( Math.round (0xFFFFF* Math.random ()); // Natija: 11601572
```

`Math.round ()` metodi eng yaqin butun songa yaxlitlaydi . Masalan, 3.6 4 ni beradi; 3.4 esa 3 ni. Agar sonning kasr qismi 0,5 ga teng bo'lsa, yuqori yaxlitlash amalga oshiriladi:

```
trace( Math.round (1.5)); // Natija : 2
```

Yaxlitlashning aniq metodlari `Math.floor()` va `Math.ceil ()`. `Math.floor ()` metodi sonni eng kichik butun son qiymatiga yaxlitlaydi . Masalan : `trace( Math.floor (2.99999999));`

```
// Natija: 2
```

```

    trace(Math.floor(-2.99999999)); // Natija: -3 (-3 -2,99999999 dan kichik)
    Math.ceil() metodi Math.floor() metodiga qarama - qarshidir . U berilgandan kattaroq eng
    yaqin butun sonni qaytaradi: trace(Math.ceil(2.99999999)); // Natija : 3
    trace(Math.ceil(-2.99999999)); // Natija : -2

```

Ko'pincha butun sonlarga emas, balki o'nlik kasrga yaxlitlash kerak bo'ladi. Biroq, ActionScript da bu muammoni hal qilishga imkon beradigan metod yo'q. Ammo agar kerak bo'lsa, uni osongina yaratish mumkin. Shunday qilib, sonni n chi kasrga yaxlitlash uchun uni  $10^n$  ga ko'paytirish kerak, yaxlitlash operatsiyasini bajariladi va natijani  $10^n$  ga bo'linadi. Ushbu yondashuvni amalga oshiradigan funksiyaga misol:

```

function my_round(number,n) {
    var kof=Math.pow(10, n); return
    Math.round(number*kof)/kof;
}
trace(my_round(10.19876784, 3)); // Natija: 10.199

```

### Sonlar juftligini solishtirish

Math obyektini sonlarni solishtirish imkonini beruvchi ikkita metodni saqlaydi. Math.max(x,y) metodi argument sifatida ikkita sonni (yoki hisoblanuvchi ifodalarni) oladi va ulardan kattasini qaytaradi. Shunga ko'ra, Math.min() metodi yuborilgan qiymatlarni taqqoslaydi va ulardan kichikini qaytaradi:

```

trace(Math.max(3,2)); // Natija : 3
trace(Math.min(3,2)); // Natija : 2

```

Math.min () va Math.max () metodlari saralash masalalarini yechish, shuningdek, tanlovda minimal yoki maksimal qiymatni topish uchun foydalidir. Masalan, massivdagi eng katta qiymatni qaytaruvchi funksiya:

```

function maxNumber(array) {
    var max_value = -Infinity; for
    (var i = 0; i<array.length; i++) {
        max_value = Math.max(max_value, array[i]);
    }
    return max_value;
}
trace(maxNumber([0, -567, 567, Infinity])); // Natija: Infinity

```

### Trigonometrik funksiyalar

Trigonometrik funksiyalar geometriyada faol qo'llaniladi, chunki ular boshqa burchaklar va tomonlar orqali figuralarning burchaklari va tomonlarini topishga imkon beradi. Flash aylana yoki spiral harakatni aniqlash, davriy jarayonlarni tavsiflash (masalan, 3-bob loyibasida biz sinus funksiyasi asosida prujina mayatnikini modellashtiramiz), ikkita oddiy shaklning kesishishini aniqlash va boshqa ko'p narsalarni hal qilish uchun trigonometrik funksiyalardan foydalanadi. Eng mashhur trigonometrik funksiyalar sinus, kosinus va tangensdir. ActionScript da ular Math.sin(), Math.cos() va Math.tan () metodlariga mos keladi. Kotangens funksiyasi yo'q, lekin uni o'zingiz osongina yaratishingiz mumkin: function cot (arg) {

```

    return Math.cos(arg)/Math.sin(arg);
}
trace(cot(Math.PI/4)); // Natija : 1

```

ActionScript trigonometrik funksiyalarining o'ziga xos xususiyati shundaki, ular radianlarda qiymatlarni oladi, biroq, masalan, klip, tugma yoki matn maydonining burilish burchagini saqlaydigan aylanish xususiyati burchak o'lchovi bilan ishlaydi.  $360^\circ 2\pi$  radianga teng ekanligini bilib, darajalarni radianga aylantirish juda oson. Bundan kelib chiqadiki, bir daraja  $\pi / 180$  radianga teng. Bu munosabatdan foydalanib, darajalarni radianga aylantiruvchi funksiyani yozish qiyin emas:

```
function deg2rad(arg) {
    return arg*Math.PI/180;
}
```

```
trace(deg2rad(180)); // Natija: 3.14159265358919
```

Radianlardan darajaga o'tkazish ham xuddi shunday oson. Buning uchun zarur bo'lgan kod quyidagicha ko'rinadi:

```
function rad2deg(arg) {
    return arg*180/Math.PI;
}
```

```
trace(rad2deg(Math.PI/3)); // Natija : 60
```

Sinus, kosinus, tangens to'g'ridan-to'g'ri trigonometrik funksiyalar deb ataladi. Tegishli funksiya qiymatiga asoslanib, burchakning qiymatini hisoblash imkonini beruvchi teskari funksiyalar (ko'pincha yoy funksiyalari deb ataladi) mavjud.

ActionScript o'rnatilgan arksinus ( Math.asin() metodi), arkkosinus ( Math.acos() metodi) va arktangens ( Math.atan() va Math.atan2() metodlari) funksiyalariga ega. Teskari trigonometrik funksiyalarning natijasi radianlarda qaytariladi:

```
trace(Math.asin(1)/Math.PI*180); // Natija: 90
```

```
trace(Math.acos(0)/Math.PI*180); // Natija: 90
```

```
trace(Math.atan(Infinity)/Math.PI*180); // Natija: 93
```

Teskari trigonometrik funksiyalar faqat ma'lum bir oraliqda natijani qaytarishini tushunish muhimdir. Arksinus uchun uning chegaralari 0 va  $\pi$ , arkkosin va arktangens uchun  $-\pi/2$  va  $\pi/2$ . Bu xususiyat ko'rib chiqilayotgan funksiyalardan foydalanishga ma'lum cheklovlar qo'yadi.

Tasavvur qilinga, sizda ma'lum bir nuqta atrofida aylanadigan to'p bor. To'pning X koordinatalari va markaz nuqtasi farqining ularning Y koordinatalaridagi farqga nisbati sifatida mos keladigan tangensni hisoblab, so'ngra olingan qiymatni Math.atan() metodiga o'tkazish orqali uning burilish burchagini aniqlay olasizmi? Javob – faqat aylanish burchagi  $-\pi/2$  va  $\pi/2$  orasida bo'lsa. Aks holda, yoy tangensini hisoblash natijasi to'pning haqiqiy joyiga mos kelmaydi. To'pning burilish burchagini uning markazining koordinatalari bo'yicha aniq aniqlash uchun siz Math.atan2(y, x) maxsus metodidan foydalanishingiz kerak (parametrlarni ko'rsatish ketmakteligiga e'tibor bering). Bu metod boshlang'ich (0, 0) va nuqta (x, y) va x o'qi o'rtasidagi burchakni qaytaradi. Masalan:

```
trace(Math.atan2(-1,1)/Math.PI*180); // Natija : -45
```

```
trace(Math.atan2(1,1)/Math.PI*180); // Natija : 45
```

```
trace(Math.atan2(1,-1)/Math.PI*180); // Natija : 135
```

```
trace(Math.atan2(-1,-1)/Math.PI*180); // Natija: -135
```

Nuqtaning (x, y) boshiga nisbatan emas, balki ixtiyoriy nuqtaga (X0, Y0) nisbatan burilish burchagini aniqlash uchun quyidagi formuladan foydalanish kerak:

```
Math.atan2(x-X0, y-Y0).
```

`Math.atan2()` `Math.atan()` ga qaraganda ko'p qirrali, chunki u sizga o'rtasidagi burchaklarni aniqlash imkonini beradi.  $-\pi$  dan  $\pi$  gacha, `Math.atan()` esa faqat  $-\pi/2$  va  $\pi/2$  oralig'ida. Bu butun aylanish oralig'ida burilish burchagini aniqlash uchun foydalanish imkonini beradi.

### Matematik konstantalar

`Math` obyekti eng muhim matematik konstantalarni saqlaydigan bir qancha xususiyatlarga ega. Keling, ularni sanab o'tamiz:

- `Math.PI`.  $\pi$  soni. Trigonometrik funksiyalar radianlarda qiymatlarni qabul qilishi (va teskari trigonometrik funksiyalar natijani qaytaradi) tufayli hisob-kitoblarda faol qo'llaniladi.

Taxminan 3,14159265358979 ga teng.

- `Math.E`.  $e$  soni, eksponenta. Taxminan 2,71828182845905 ga teng.

- `Math.SQRT2`. Ikkining ildizi. U ikkitaning ildizi tugun nuqtalarida (masalan,  $30^\circ$ ,  $60^\circ$ ) trigonometrik funksiyalar qiymatlari uchun ifodalarga kiritilganligi sababli faol qo'llaniladi. Taxminan 1,4142135623731.

- `Math.SQRT1_2`.  $1/2$  ildiz. Foydali, chunki bu sinus va kosinusning qiymati  $45^\circ$ da. Taxminan 0,707106781186548.

- `Math.LN2`. 2 ning natural logarifmi. Natural logarifmdan 2 asosli logarifimga o'tganda foydali. Taxminan 0,693147180559945.

- `Math.LN10`. Natural logarifm 10. Natural logarifmlardan o'nlik logarifmlarga o'tish imkonini beradi. Taxminan 2,30258509299405.

- `Math.LOG2E`.  $e$  ning 2 asosli logarifmi. Taxminan 1,44269504088896.

Barcha sanab o'tilgan matematik konstantalar cheksiz irratsional kasrlardir. `Math` obyektining xossalari sifatida saqlanadi, ularning qiymatlari taxminiydir. Bu hisob-kitoblarda qo'shimcha xatolarni keltirib chiqaradi va, masalan, `Math.sin(Math.PI)` ning aynan nolga teng kelmasligi sabablarining bir qismidir.

## 6. Nazorat savollari



1. Sonli tur ActionScriptda qanday nomlanadi?
2. Butun sonning qiymati qaysi sanoq tizimiga berilganligini qanday aniqlash mumkin?
3. Sonli satrni songa aylantirish usullarini bilasizmi?
4. `parseInt` global funksiyasining vazifasi nima?
5. `parseFloat` global funksiyasining vazifasi nima?
6. Cheksizlik ( $\infty$ ) ga qanday qiymat mos keladi?
7. ActionScript da daraja ko'rsatgich uchun mo'ljallangan matematik funksiyalar?
8. ActionScript da qanday trigonometrik funksiyalarni bilasiz?
9. ActionScript da logarifmlarni hisoblashning qanday funksiyalarini bilasiz?
10. Matematik konstantalar ( $\pi$ ,  $e$  va boshqalar) qayerda saqlanadi?

## 2.5. Munosabat va tarmoqlanuvchi operatorlar

### Reja:

1. Munosabat operatorlari.
2. Tarmoqlanuvchi operatorlar. **if** shart operatori.
3. **else** operatori va **else if** operatorlarining kombinatsiyasidan foydalanish.
4. Mantiqiy operatorlar **VA (&&)**, **YOKI (||)**.
5. **SWITCH-CASE** tanlash operatori.
6. Nazorat savollari.

**Kalit soʻzlar:** *operatorlar, munosabat operatorlar, tarmoqlanuvchi operatorlar, tanlash operatorlari, if, else, switch, case, true, false, AND operatori, OR operatori, NOT (inkor) operatori.*

### 1. Munosabat operatorlari

An'anaviy dasturlash tillari ham, skript tillari ham taqqoslashni amalga oshirish va bir qator muqobil harakat yo'nalishlarini yozish imkonini beruvchi tuzilmalarni taqdim etadi va ActionScript ham bu borada istisno emas. Ushbu maqsadlar uchun faqat ikkita qiymatni qabul qilishi mumkin bo'lgan mantiqiy turdagi o'zgaruvchilardan foydalaniladi: **true** ( to'g'ri ) yoki **false** ( noto'g'ri ). O'zgaruvchilarning ikkita qiymatini solishtirish uchun oddiy mantiqiy ifodalarni qurishda quyidagi taqqoslash operatorlari qo'llaniladi:

5.1.1-jadval. Munosabat operatorlari

Operatorlar	Natija
==	Tengmi
!=	teng emas
>	Katta
<	Kichik
>=	Katta yoki teng
<=	Kichik yoki teng
%	Bo'lishning qoldiq qismini qaytaradi

### 2. Tarmoqlanuvchi operatorlar. **if** shart operatori

**ActionScript 3** ko'pgina boshqa dasturlash tillari kabi, u kontekstga qarab harakat yo'nalishini tanlash imkoniyatini beradi. Buning uchun eng ko'p ishlatiladigan **if** operatoridir.

Bu operator o'zgaruvchi yoki ifoda qiymatini rost( true ) yoki yolg'on(false ) uchun tekshirish imkonini beradi. **if (<shart>) { <operator1>;}**

Agar if buyrug'idan keyin qavs ichiga olingan ifoda rost bo'lsa, u holda qavsdan so'ng darhol figurali qavs ichiga olingan amal yoki amallar qatori bajariladi. Agar u noto'g'ri bo'lsa, figurali qavslar ichiga olingan kod bajarilmaydi, lekin quyidagi ko'rsatmalar bajariladi.

Mutlaqo hamma narsani tekshirishingiz yoki solishtirishingiz mumkin:

- bir o'zgaruvchining qiymatidan katta, kichik yoki unga teng, boshqasining qiymati
- o'zgaruvchining qiymati umuman aniqlanganmi
- matn maydonida biron bir matn bormi □ simbol nusxasi yaratilganmi yoki yo'qmi □ va hokazo.

O'zgaruvchilar qiymatlarini taqqoslashda == (ketma-ket ikkita teng) sifatida yozilgan tenglik operatoridan foydalanishda ehtiyot bo'lishingiz kerak. Yagona tenglik belgisi (=), allaqachon bilganingizdek, uning o'ng tomonidagi o'zgaruvchining chap tomonidagi qiymatni belgilash operatoridir. Ikkita teng belgi (==) chapdagi qiymatni o'ngdagi qiymat bilan solishtiradi. **Misol.** Tenglik uchun ikkita qiymatni taqqoslaydigan ko'rsatma yozing.

**ActionScript Number turidagi num1 va num2 o'zgaruvchilarni yaratamiz**

```
var num1:Number = 5;
```

```
var num2:Number = 5; Matn maydonini yaratamiz
```

```
var txtCompare:TextField = new TextField();
```

```
Sahnaga matn maydoni qo'shish
```

```
addChild ( txtCompare
```

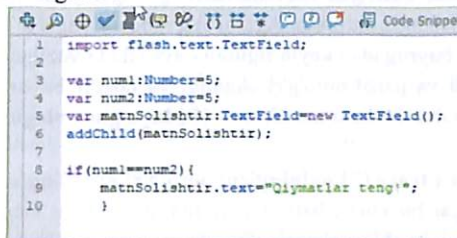
```
); if operatoridan foydalanish
```

```
if (num1 == num2){ txtCompare.text = " Qiymatlar teng!";}
```

solishtirilgan o'zgaruvchilarning qiymatlari teng bo'lsa, flesh-playerda ko'rsatiladigan matn chiqadi.

```
txtCompare.text = "Qiymatlar teng!";
```

Quyidagi natijani olishingiz kerak:

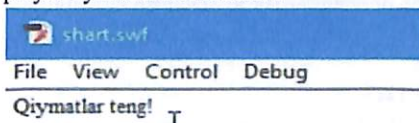


```
1 import flash.text.TextField;
2
3 var num1:Number=5;
4 var num2:Number=5;
5 var matnSolishtir:TextField=new TextField();
6 addChild(matnSolishtir);
7
8 if(num1==num2){
9     matnSolishtir.text="Qiymatlar teng!";
10 }
```

5.2.1-rasm. ActionScriptda Number turidan foydalanish.

if iborasining ishlashini tekshirish . Ctrl + Enter ni bosing .

Matn flesh -player oynasida ko'rsatiladi .



5.2.2-rasm. ActionScriptda Number turidan foydalanish natijasi.

Shunday qilib, "Qiymatlar teng!" iborasi num1 va num2 o'zgaruvchilar qiymatlari teng bo'lgani uchun paydo bo'ldi. Flash playerni yoping .

Ikkinchi o'zgaruvchining qiymatini 10 ga o'zgartiring.

Ctrl + Enter ni bosing.

Flash player oynasida hech narsa ko'rsatilmaydi , chunki o'zgaruvchilarning qiymatlari teng emas.

O'zgaruvchilar qiymatlarini solishtirish uchun faqat teng operatoridan emas, balki quyida ko'rsatilgandek boshqa operatorlardan ham foydalanishingiz mumkin:

```
if (num1 > num2) // num1 num2 dan katta ekanligini tekshiring if
(num1 < num2) // num1 num2 dan kichik yoki yo'qligini tekshiring if
(num1 >= num2) // num1 num2 dan katta yoki teng ekanligini
tekshiring if (num1 <= num2) // num1 num2 dan kichik yoki teng
ekanligini tekshiring if (num1 != num2) // num1 num2 ga teng
emasligini tekshiring
if ( num1 is String) // num1 turini tekshirish
if ( num1===num2) // o'zgaruvchan qiymatlarni ma'lumotlarni konvertatsiya
qilmasdan
solishtirish. Agar num1 satr tipidagi o'zgaruvchi bo'lsa va 4 ga
va // num2 son turidagi 4 ga teng bo'lsa, u holda num1 == num2 rost bo'ladi
```

### 3. else operatori va else if operatorlarining kombinatsiyasidan foydalanish

if ... else konstruksiyasidan shart yolg'on(false) bo'lgandagi holatlarni bajarish uchun foydalanishingiz mumkin.

```
if(<shartlar >) { <operator1>;} else{<operator2>;}
```

Misol tariqasida quyidagilarni ko'rib chiqing. Tasavvur qiling-a, saytga kirish uchun formaning ishlashini boshqaruvchi kodni yozgansiz. Sizda ikkita ssenariy bor:

- birinchisi – mehmon to'g'ri foydalanuvchi nomi va parolni kiritdi
- ikkinchisi - mehmon noto'g'ri ma'lumotlarni kiritdi

Bunday holatda, kiritilgan ma'lumotlarning to'g'ri yoki yo'qligini tekshirish uchun if iborasidan foydalanishingiz mumkin. Agar u to'g'ri bo'lsa, tegishli ko'rsatmalar bajariladi. Agar noto'g'ri bo'lsa, else buyrug'idan keyin figurali qavs ichida yozilgan gaplar bajariladi. Masalan, foydalanuvchi nomi va parol noto'g'ri ekanligi va tashrif buyuruvchiga kirishni takrorlashi kerakligi haqidagi xabar bo'lishi mumkin. `if ( in_login ==login) {trace(" Parolni kiriting ");}`

```
else { trace ("Loginigizni noto'g'ri kiritdingiz. Iltimos, qayta kiriting");}
```

Bundan tashqari, agar bir emas, balki bir nechta qiymatlarni ketma-ket tekshirishingiz kerak bo'lsa, buning uchun else if iboralarining kombinatsiyasidan foydalanishingiz mumkin. if , else if har biri o'zining figurali qavslariga ega. Birinchisi tekshirilishi kerak bo'lgan qiymatni o'z ichiga oladi, ikkinchisida birinchisi to'g'ri bo'lsa, bajariladigan ko'rsatmalar mavjud..

Kodingizda boshqa ko'plab operatorlardan foydalanishingiz mumkin if , else qaysi birini xohlasangiz, lekin ular shartli if ifodasidan keyin va else dan oldin mavjud bo'lsa kelishi kerak.

Masalan: var

```
numPencils: Number = 12;
```

```
if( numPencils < 9 ){
```

```
    trace( "Qutida 9 tadan kam qalam bor." );
```

```
}
```

```
else if( numPencils == 9 ){
```

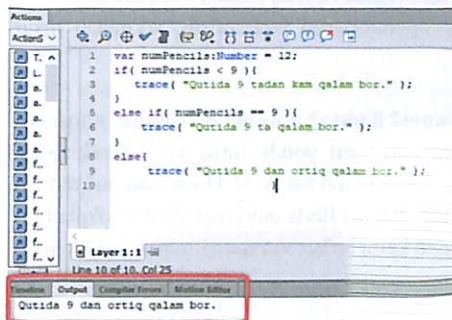
```
    trace( "Qutida 9 ta qalam bor." );
```

```

}
else
{
    trace( "Qutida 9 dan ortiq qalam bor." );
}

```

numPencils o'zgaruvchisiga qanday qiymat berganingizga qarab if, else iboralari bilan bog'langan uchta ko'rsatmalardan biri bajariladi. if yoki else va tegishli ibora Chiqish panelida paydo bo'ladi. Bizning holatda:



5.3.1-rasm. ActionScriptda shart operatoridan foydalanish.

Har qanday ifodani if iborasi orqali rost yoki noto'g'ri ekanligini tekshirishi mumkin. Mantiqiy o'zgaruvchilarning standart qiymatlari true yoki false bo'lishi mumkinligini bilamiz. Shunday qilib, agar sizda, o'zgaruvchi bo'lsa

```

var adminStatus:Boolean = true;
test natijasi
if( adminStatus ) true bo'ladi. Qarama-qarshi natijani tekshirish uchun " ! "
belgisidan foydalaning.
if( !adminStatus )

```

false bilan mos keladi. Boshqa turdagi o'zgaruvchilarning qiymatlari xuddi shu tarzda sinovdan o'tkazilishi mumkin, ammo bu qiymatlar ActionScript tomonidan qanday talqin qilinishini aniq bilishingiz kerak. Misol uchun, agar num1 turi Number ga o'rnatilgan bo'lsa, agar num1 0 yoki null bo'lmasa, if( num1 ) true deb baholanadi. Aks holda false bo'ladi.

#### 4. Mantiqiy operatorlar AND (&&), OR (||)

ActionScript 3 dagi shartli bayonotlar yordamida faqat bitta qiymat yoki ifodani emas, balki bir vaqtning o'zida bir nechta rostlikka tekshirish mumkin. Buning uchun **mantiqiy operatorlardan foydalaniladi**. Masalan, agar ikki yoki undan ortiq shartlar mos kelsa yoki berilgan shartlardan kamida bittasi to'g'ri bo'lsa, ma'lum ko'rsatmalar bloki bajariladi.

Mantiqiy operator VA (&&) ikki yoki undan ortiq shartlar to'g'ri yoki yo'qligini tekshirish uchun ishlatiladi. Misol uchun, agar kiritilgan login "myLogin" va "12345" parolning qiymatlari to'g'ri yoki yo'qligini tekshirmoqchi bo'lsak, u holda quyidagicha yoziladi:

```

if( login == "myLogin" && password == "12345" )

```

E'tibor bering, ikkala shart ham to'g'ri bo'lishi kerak. Agar ulardan biri yolg'on bo'lsa (false



) - masalan, login to'g'ri, lekin parol noto'g'ri - u holda butun ifoda ham yolg'on bo'ladi.

Agar shartlardan kamida bittasini rost bo'lishiga ruxsat bersangiz, mantiqiy operator **OR** || dan foydalaniladi:

```
if( country == "Uzbekistan" || capital == "Tashkent" )
```

Bunday holda, shartlardan biri noto'g'ri bo'lsa ham, shartlardan biri to'g'ri bo'lgani uchun butun ifoda to'g'ri deb baholanadi. Ammo agar ikkala shart ham noto'g'ri bo'lsa, unda butun ifoda yolg'on bo'ladi.

Mantiqiy **OR** operatori ikkita vertikal chiziq bilan yoziladi ||. Ko'pgina kompyuter klaviaturalarida ushbu belgili tugma buyruq ( Enter ) tugmachasining tepasida joylashgan.

&& va || mantiqiy operatorlarini birlashtirib, ikkitadan ortiq shartlarni ham yozish mumkin.

Masalan:

```
if( num1 == num2 || num2 > num3 && num3 < num4 )
```

Shu bilan birga, quyidagilarni yodda tutishingiz kerak: agar bunday operatorlardan birgalikda foydalansangiz, birinchi navbatda **AND** operatorlari &&, keyin esa **OR** operatorlari || baholanadi. Shunday qilib, yozma ifoda quyidagi tartibda o'qiladi:

```
"Agar num2 num3 dan katta bo'lsa VA num3 num4 dan kichik bo'lsa YOKI num1 num2 ga teng bo'lsa."
```

Ko'pincha, bunday iboralardagi tartib muhim emas, lekin bu haqiqatan ham muhim bo'lganda, bu haqida unutmang. Aks holda, keyinchalik aniqlash oson bo'lmagan xatolar bo'ladi.

Agar mantiqiy **OR** operatori birinchi bo'lib baholanishi kerak bo'lsa, uni qavs ichiga qo'yishingiz mumkin:

```
if( ( num1 == num2 || num2 > num3 ) && num3 < num4 )
```

## 5. SWITCH-CASE tanlash operatorlari

SWITCH-CASE tanlash operatorining umumiy ko'rinishi :

```
switch ( < ifoda > )
```

```
{
```

```
    case <1 -holat > : <1 -operatorlar >
```

```
    case < 2 -holat > : <2 -operatorlar >
```

```
    case < n -holat > : <n -operatorlar >
```

```
    default: < operatorlar >
```

```
}
```

case kalit so'zidan keyingi qiymatlar bilan tenglik uchun solishtiriladi : holat 1, 2, ... n. Agar uning qiymati ushbu holatlardan biriga teng bo'lsa, u holda ushbu blokda ko'rsatilgan ko'rsatmalar bajariladi. Agar shartlarning hech biri mos kelmasa, default yozuvidan keyin yozilgan operatorlar bajariladi. **Break** iborasi switch iborasining ishini tugatish uchun ishlatiladi.

Masalan: var numNed: Number = 2;

```
    switch ( numNed )
```

```
    {
```

```

    case 1 : trace("Dushanba ");
    break;   case 2 : trace("Seshanba
"); break;   case 3 :
    trace("Chorshanba "); break;   case
    4 : trace("Payshanba "); break;
    case 5 : trace("Juma "); break;
    case 6 : trace("Shanba "); break;
    case 7 : trace("Yakshanba "); break;
    default : trace("Hafta kunini xato kiritdingiz ");
}
    trace("Tamom ");

```

switchdan keyin qavs ichidagi ifodaning qiymati 2. Shuning uchun ekranda "Seshanba" yozuvi ko'rsatiladi. Va break operatori switch operatoridan chiqadi va boshqaruvni keyingi operatorga o'tkazadi:

```

    trace(" End ");

```

## 6. Nazorat savollari



1. ActionScript da ikkita qiymatni solishtirish uchun qanday munosabat operatorlar ishlatiladi ?
2. ActionScriptda "teng emas"ni ifodalash uchun qaysi taqqoslash operatoridan foydalanish mumkin?
3. ActionScript da "teng" ni ifodalash uchun qaysi taqqoslash operatoridan foydalanish mumkin?
4. || Mantiqiy operatorning maqsadi?
5. && Mantiqiy operatorning maqsadi?
6. ! Mantiqiy operatorning maqsadi?
7. Murakkab mantiqiy ifodalarni tuzish uchun qanday mantiqiy operatorlar ishlatiladi?
8. ActionScript da qanday tarmoqlanuvchi operatorlar mavjud ?
9. ActionScript dagi if operatorining maqsadi ?
10. ActionScript dagi else operatorining maqsadi ?
11. ActionScript da else if operatorining maqsadi?
12. ActionScript da switch-case konstruksiyasining maqsadi ?
13. ActionScript dagi switch - case konstruksiyasidagi break operatorining maqsadi nima?
14. ActionScript dagi switch - caste konstruksiyasining default operatorining maqsadi nima?
15. if operator konstruksiyasida else if operatorini necha marta ishlatish mumkin?

## 2.6. Takrorlash (takrorlanish) operatorlari. for, while va do while operatorlari

### Reja:

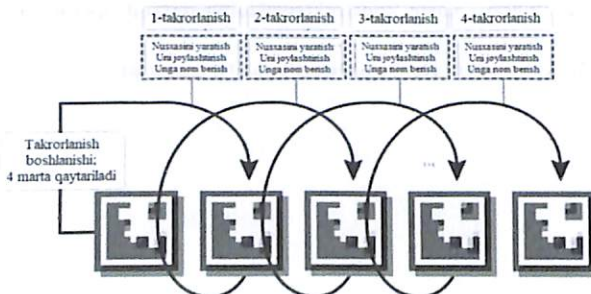
1. Takrorlanishlarni qanday va nima uchun ishlatish kerak.
2. Takrorlanish turlari.
3. Takrorlanish shartlarini o'rnatish.
4. Ichki o'rnatilgan takrorlanishlarni yaratish.
5. Takrorlanishlarni to'xtatish.
6. Nazorat savollari.

**Kalit so'zlar:** takrorlanish operatorlari, menyu, qiymat, tana, ichki o'rnatilgan takrorlanishlar, uzilish, inkrement, dekrement, parametr, takrorlanish, for, while, do.

### 1. Takrorlanishlarni qanday va nima uchun ishlatish kerak

Takrorlanishlar biror harakatni (yoki harakatlar to'plamini) takrorlash uchun ishlatiladi. ActionScript ga bir necha qator qo'shish yordamida Flash ushbu amallarni xohlaganingizcha takrorlashi mumkin! Odatda, takrorlanishlar quyidagi holatlarda qo'llaniladi (aytgancha, ularning ba'zilarida takrorlanishsiz qilish qiyin, hatto imkonsizdir).

- Ochiladigan ro'yxatlarni dinamik yaratish
  - Ma'lumotlarni tekshirish
  - Matn bo'yicha qidirish
  - Xarakterli filmlarning dinamik ikki baravar ko'paytirish
  - Massiv tarkibini boshqa massivga nusxalash
  - O'yinlarda harakatlanuvchi "snaryadlar" ning ob'ektlar bilan to'qnashuvini aniqlash
- Takrorlanishlar ko'p vazifalarni avtomatlashtirishi mumkin. Masalan, xarakterli filmning dinamik nusxasini olaylik. Aytaylik, loyihangizda bir xil klipning 100 ta nusxasini teng ravishda joylashtirmoqchisiz. Buni amalga oshirish uchun kutubxonadan 100 ta misolni barchasini sahnaga chaqirib olib, ularni tekislashning hojati yo'q - faqat bitta misolni chaqirib olib, to'rt-besh qatorli skriptni yozish yetarli - avtomatik ravishda yuztalab dublikat yaratadigan va ularni sahna bo'ylab teng ravishda joylashtiradigan Takrorlanish operatoridir. Bundan tashqari, ushbu Takrorlanish operatori loyihada bir necha marta o'zgartirilishi va ishlatilishi mumkin.



6.1.1-rasm. Takrorlanishlar.

Skriptlarning yana bir afzalligi - bu dinamik ekanligidir. Aytaylik, biz 15 ta elementdan (tugmalardan) iborat menyuni dinamik ravishda yaratish uchun takrorlanish yaratdik. Ushbu takrorlanish bilan skriptni biroz o'zgartirib, hech qanday muammosiz alohida elementlarni dinamik ravishda qo'shishingiz yoki o'chirishingiz mumkin. Agar menyu qo'lda yaratilgan bo'lsa, unda elementlarni qo'shish yoki olib tashlash qiyin bo'ladi, chunki hali ham qolgan elementlarni yuqoriga yoki pastga siljitish kerak, shuningdek, bu yerda va u yerda skriptlarga o'zgartirishlar kiritishingiz kerak.

## 2. Takrorlanish turlari

ActionScript da uch xil takrorlanish operatorlari mavjud. Uchalasining ham bitta umumiy jihati bor: ular ma'lum bir shart rost ( true ) bo'lsa, harakat yoki harakatlar to'plamini bajaradilar

### **while** takrorlanish

**operatori** Ushbu

operatorning sintaksisi:

```
while (someNumber < 10)
```

```
{
```

```
// bajarilishi kerak bo'lgan harakatlar (takrorlanish tanasi)}
```

Bu yerda someNumber < 10 ifodasi - takrorlashlar sonini aniqlaydigan shart (takrorlanish takrorlanishi). Har bir iteratsiyada takrorlanishning tanasiga joylashtirilgan harakatlar bajariladi. Takrorlanish shartini to'g'ri tanlash va o'rnatish haqida keyingi bo'limda - "Takrorlanish shartini tanlash"da gaplashamiz. Quyida while takrorlanishiga misol keltirilgan:

```
i = 0;
```

```
while (++i <= 10) {
```

```
myClip.duplicateMovieClip (" myClip " + i , i );
```

```
}
```

Bu skript myClip klipining 10 ta dublikatlarni yaratadi.

### **for** takrorlanish operatori

for operatori ba'zi bir o'zgaruvchining qiymati har bir iteratsiyada birga ortishi yoki kamayishiga asoslangan takrorlanishlarni dasturlashning ixcham va qulay usulidir. Bu takrorlanish o'zgaruvchisida dastlabki qiymat berish, takrorlanish shartini o'rnatish, inkrement (yoki dekrement) bilan qulay - bularning barchasi ActionScript ning faqat bitta qatorini talab qiladi. For va while operatorlarining sintaksisi har xil, lekin ular ma'no jihatidan o'xshash. Qaysi birini u yoki bu holatda ishlatish muayyan sharoitlarga bog'liq. for operatorining sintaksisi:

```
for (someNumber = 0; someNumber < 10; ++someNumber) {
```

```
// bajarilishi kerak bo'lgan harakatlar (takrorlanish
```

```
tanasi) }
```

Qavslar ichida nuqta-vergul bilan ajratilgan uchta element bajariladigan takrorlash sonini aniqlaydi. Bu misolda someNumber o'zgaruvchisi dastlabki 0 qiymat bilan yaratiladi. So'ngra, someNumber 10 dan kichik bo'lsa, takrorlanishning tanasi takrorlanishi kerakligini bildiradi. Oxirgi element har bir iteratsiyada someNumber 1 ga oshirilishi kerakligini bildiradi

– oxir-oqibat someNumber 10 ga teng bo'ladi.(shuning uchun takrorlanish to'qqiz marta takrorlangandan keyin tugaydi).

for takrorlanishining tuzilishi shuni ko'rsatadiki, u asosan harakatlar to'plamini ma'lum bir necha marta bajarish kerak bo'lgan joylarda qo'llaniladi. Dastkabki misol for operatori yordamida quyidagicha yoziladi:

```
for ( i =0; i <=10; ++ i ) {  
    myClip.duplicateMovieClip ( " myClip " + i , i );  
}
```

Ushbu takrorlanish myClip ning 10 ta dublikatini yaratadi .

**for..in takrorlanish operatori**

Bu operator obyektning barcha xossalari orqali "o'tish" uchun ishlatiladi. Uning sintaksisi quyidagicha:

```
for ( i in _object ) {  
// bajarilishi kerak bo'lgan harakatlar (takrorlanish tanasi)  
}
```

Bu yerda i o'zgaruvchisi keyingi xususiyat nomini vaqtinchalik (har bir iteratsiyada) saqlash uchun ishlatiladi. i qiymati takrorlanish tanasi ichida ishlatilishi mumkin. Quyidagi misolda ushbu operatorning amaliy qo'llanilishini ko'rib chiqamiz:

```
car = new Object();  
car.color = "red";  
car.make =  
"BMW";  
car.doors = 2; for  
(i in car) {  
    result = result + i + ": " + car[i] + newline;  
}
```

Dastlab skriptda car nomli obyektini yaratadi. Keyinchalik tegishli qiymatlar bilan car obyektida xususiyatlar yaratiladi (ularni faqat car obyektini ichidagi o'zgaruvchilar deb hisoblanadi). Keyin for .. in takrorlanishi car obyektining barcha xossalari bo'ylab takrorlanadi va har bir xususiyat nomini o'z navbatida i o'zgaruvchisiga qo'yadi. i qiymati takrorlanish tanasida ishlatiladi. Davr oxirida natija o'zgaruvchisi barcha xususiyatlar va ularning qiymatlari ro'yxatini ko'rsatadigan matn qatorini o'z ichiga oladi.

Birinchi iteratsiyada i ga " doors " satr qiymati beriladi (chunki bu oxirgi belgilanishi kerak bo'lgan xususiyat edi). Shunday qilib, birinchi iteratsiyada takrorlanish tanasidagi ifoda quyidagicha ko'rinadi:

```
result = result + "doors" + ": " + 2 + newline;
```

Birinchi iteratsiyadan so'ng, result o'zgaruvchisi "doors: 2 " qiymatga ega bo'ladi.

Aniqlik kiriritish kerakki, ifodada i o'zgaruvchisi (kvadrat qavslarsiz) xususiyat nomini o'z ichiga oladi ("doors", "make" yoki "color"). Va car[i] elementi bu xususiyat qiymatiga havola; masalan, bu car.doors yozuvi bilan bir xil.

Shunday qilib, butun takrorlanish oxirida result quyidagi satr qiymatiga ega bo'ladi:

```
"doors: 2  
make: BMW  
color: red"
```

Car obyektini uchta xususiyatga ega bo'lganligi sababli, ushbu skriptdagi for .. in takrorlanishi avtomatik ravishda uch marta takrorlanadi.

Ushbu takrorlanish operatori odatda quyidagi hollarda qo'llaniladi:

I

Obyekt car



Xususiyatlar

Iter. 3 `color = "red"` `i = "color" car[i] = "red"`

Iter. 2 `make = "BMW"` `i = "make" car[i] = "BMW"`

Iter. 1 `doors = 2` `i = "doors" car[i] = 2`

6.2.1-rasm. Takrorlanishlarning xilma-xilligi.

- Montaj stoli yoki obyektidagi barcha o'zgaruvchilarning qiymatlarini ochish uchun;
- Montaj stoli yoki obyektidagi barcha obyektning nomlarini aniqlash uchun;
- XML hujjatining barcha atributlarining nomlari va qiymatlarini ochish uchun.



**IMPORTANT**

**Maslahat.** Obyektida o'zgaruvchi yaratilsa, u assotsiativ massiv deb ataladigan joyga joylashtiriladi. Ma'lumki, oddiy massivlarda elementga uning indeksi - raqami orqali murojaat qilinadi (raqamlash 0 dan boshlanadi). Assotsiativ massivda element o'z nomi bilan ataladi. Ta'riflangan takrorlanish operatori ko'rsatilgan montaj stoli yoki obyektga tegishli assotsiativ massivning barcha elementlarini oddiygina takrorlaydi.

### 3. Takrorlanish shartlarini o'rnatish

Takrorlanish shartini tanlash va yozish

Ushbu ma'ruzaning qolgan qismida faqat while operatoridan foydalanamiz. Eslatib o'tamiz, ushbu takrorlanishning tanasidagi harakatlar shart ifodasi true deb baholanguncha bajariladi. Misol keltiramiz: `i = 0;`

```
while (i < 10) {  
    // bajarilishi kerak bo'lgan harakatlar (takrorlanish  
    tanasi) }
```

Ushbu takrorlanishning sharti `i < 10`. `i` 10 dan kichik qiymatgacha, bu ifoda true deb baholanadi va takrorlanish tanasidagi amallar bajariladi. Biroq, shuni ta'kidlash kerakki, bizning takrorlanishimizda eng muhim "ingrediyent" yo'q: ya'ni `i` ning qiymat o'zgarmaydi, ya'ni shart hech qachon false bo'lmaydi va takrorlanish cheksiz ishlaydi. Ilova shu nuqtaga yetib keladiki, "takrorlanishga tushib qoladi" va Flash takrorlanish tugamaguncha boshqa harakatlar qila olmaydi. Bunga yo'l qo'ymaslik uchun, takrorlanishning tanasida `i` ni qiymatni

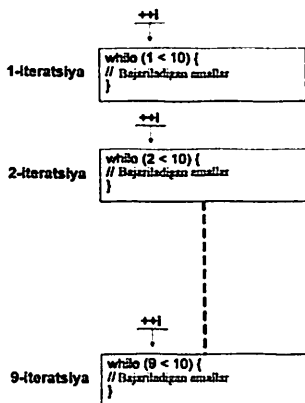
o'shiradigan amalni joylashtirish kerak, shundan keyingina u qachondir 10 ga yetadi - keyin shart false deb baholanadi va takrorlanish tugatiladi. Buni amalga oshirishning eng keng tarqalgan usuli - takrorlanishlarda inkrement(++ ) operatoridan foydalanish.

```
Masalan: i = 0; while (i < 10 )
{ // bajariladigan amallar
```

```
    ++i
}
```

Endi har safar takrorlanish tanasi bajarilganda i ni qiymati bittaga oshiriladi, natijada takrorlanish 10 marta takrorlanadi. Ya'ni: birinchi marta qiymat i 0; har bir iteratsiyada ( takrorlanish tanasining oxirida ) u 1 ga ortadi. O'ninchi iteratsiyada i = 10 ifodasini olamiz. i < 10 shart endi to'g'ri bo'lmaydi va takrorlanish tugaydi. Xuddi shu narsani qisqaroq yozish mumkin:

```
i = 0; while (++i < 10 )
{ // bajariladigan amallar
}
```



6.3.1-rasm. While takrorlanish operatori blok sxemasi.

Bunday holda, takrorlanish to'qqiz marta takrorlangandan keyin tugaydi. Dastlab, i qiymati 0 ga teng va har bir iteratsiyada u 1 ga ortadi - lekin oxirida emas, balki boshida, shartli ifodaning o'zida. O'ninchi iteratsiyadan oldin, i = 10 da, i < 10 ifodasi endi to'g'ri bo'lmaydi va takrorlanish to'xtaydi.

Dekrement operatoridan ham foydalanish mumkin - u quyidagicha yoziladi:

```
i = 10;
while (-- i > 0) {
// bajariladigan harakatlar
}
```

Takrorlanishning sharti qandaydir ortib boruvchi yoki kamayuvchi qiymatga bog'liq bo'lishi shart emas; har qanday narsa bo'lishi mumkin. Funksiya chaqiruvi shart sifatida ham paydo bo'lishi mumkin, bu esa natijani true yoki false qaytaradi, masalan :

```
while (someFunction()) {
// bajariladigan harakatlar
}
```

#### 4. Ichki o'rnatilgan takrorlanishlarni yaratish

Ko'rib turganingizdek, takrorlanishlardan foydalanish takrorlanuvchi vazifalarni avtomatlashtirish imkonini beradi. Biroq, bu hammasi emas. Ichki o'rnatilgan takrorlanish bilan nima qilish mumkinligini tasavvur qiling - ya'ni takrorlanish ichiga joylashtirilgan takrorlanish. Harakatlar to'plami bir necha marta amalga oshiriladi, keyin bu ketma-ketlik takrorlanadi, lekin kichik o'zgarishlar bilan, keyin yana ... Mana, ichki o'rnatilgan takrorlanish:



6.4.1-rasm. Ichma-ich takrorlanishlar.

Ushbu skriptdagi harakatlar to'plami 100 marta bajariladi. Bu yerda ma'no quyidagicha: tashqi takrorlanish (shartda i o'zgaruvchisi yordamida) 10 marta bajariladi. Bunday holda, har bir iteratsiyada j o'zgaruvchisi (ichki holatda ishlatiladi) 0 ga qayta o'rnatiladi va ichki takrorlanish 10 marta bajariladi. Boshqacha qilib aytadigan bo'lsak, tashqi takrorlanishning birinchi iteratsiyasida ichki takrorlanish 10 marta, ikkinchi takrorlashda yana 10 marta aylanadi va hokazo.

Hayotdan ichki o'rnatilgan takrorlanishlarga misol. Tasavvur qiling-a, matn muharririda qandaydir matn yozyapsiz. Bu jarayon ichki o'rnatilgan takrorlanishdan boshqa narsa emas: siz birinchi qatordan boshlaysiz, 100 ta belgi kiritasiz, ikkinchi qatorga o'tasiz, yana 100 ta belgi kiritasiz va hokazo. Agar matn, masalan, 25 qatordan iborat bo'lsa, unda bu ishni shunday skript sifatida ko'rsatish mumkin:

```
var i = 0; while
(++ i <= 25) {
  var j = 0; while
  (++j <= 100) {
    // bitta belgi kiritish
  }
  // keyingi qatorga o'tish
}
```

Shuni ham ta'kidlash kerakki, boshqasi ichidagi bir takrorlanish chegara emas: ona bola qo'g'irchoqlar kabi takrorlanishlarni bir-biriga, kerakli darajada ko'p marta joylashtirishingiz mumkin.



## 5. Takrorlanishlarni to'xtatish

Takrorlanishning iteratsiyasi odatda shart to'g'ri bo'lguncha davom etadi. Biroq, ActionScript bu hatti-harakatni o'zgartirish uchun ikkita amalga ega: continue va break.

Continue harakati joriy iteratsiyaning bajarilishini to'xtatish va to'g'ridan-to'g'ri takrorlanishning keyingi iteratsiyasiga o'tish imkonini beradi. Misol:

```
total = 0;
i = 0;
while (++i <= 20)
{ if (i == 10) {
    continue;
  }
  total +=
  i; }
```

Ushbu skriptda while takrorlanish operatori 20 ta iteratsiyani amalga oshiradi. Har bir iteratsiyada total o'zgaruvchisiga i qiymat qo'shiladi - o'ninchi iteratsiya i 10 ga teng bo'lgandan tashqari. Bu holda continue harakati takrorlanish tanasida keyingi amallarning bajarilishi to'xtaydi va o'n birinchi iteratsiyaga o'tish sodir bo'ladi. Natijada, total o'zgaruvchis quyuidagi sonlar to'plamini o'z ichiga oladi: "1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20". E'tibor bering, ushbu to'plamda 10 raqami yo'q - axir, o'ninchi iteratsiyada takrorlanish to'xtatildi.

Break amali takrorlanishni erta tugatish uchun ishlatiladi - hatto takrorlanishning sharti hali to'g'ri bo'lsa ham. Misol: total = 0;

```
i = 0;
while (++i <= 20) {
  total += i;
  if (total >= 10) {
    break;
  }
}
```

Ushbu skriptda har bir iteratsiyada total o'zgaruvchining qiymati i ga oshiriladi. total qiymati 10 ga yetsa (buni if operatori tekshiradi), break amali bajariladi va while takrorlanishi tugaydi (shart bo'yicha 20 marta bajarilishi kerak bo'lsa ham).

## 6. Nazorat savollari

1. ActionScript da takrorlanish operatorlari qachon ishlatiladi ?



2. ActionScript da necha xil turdagi takrorlanish operatorlari mavjud ?
3. while operatorining sintaksisini tavsiflab bering ?
4. while takrorlanishi operatoridan foydalanishga misol keltiring .
5. for operatorining sintaksisini tavsiflab bering ?
6. for takrorlanish operatoridan foydalanishga misol keltiring .
7. for .. in operatorining sintaksisini tavsiflang ?
8. for .. in takrorlanish operatoridan foydalanishga misol keltiring.
9. Ichma ich o'rnatilgan takrorlanishlar haqida nima deya olasiz?

## 10. ActionScript -dagi takrorlanishni to'xtatish operatorlari qanday ?

### 2.7. Tugma hodisalarini boshqarish. O'tish operatori va kadr yorlig'i. stop() va play() funksiyalari. O'tish operatori va kadr yorlig'i.

#### Reja:

1. Hodisalar bilan ishlovchilar. To'g'ri ishlov beruvchini tanlash.
2. Sichqoncha va klaviatura hodisalari.
3. Klip hodisasini qo'llash.
4. Hodisalarni qayta ishlash usullari haqida.
5. stop() va play() funksiyalari. O'tish operatori va kadr yorlig'i.
6. Nazorat savollari.

**Kalit so'zlar:** *handler, skript foydalanuvchisi, reaksiya, kadr, maxsus belgi, on, scanner, backspace, scan, kursor, sichqoncha, keypresses, hodisani boshqarish, bo'sh joy, maydon, interaktivlik, funksiya.*

#### 1. Hodisalar bilan ishlovchilar. To'g'ri ishlov beruvchini tanlash

Hodisalarini boshqaruvchilar (event handlers) filimingizning interaktiv qismini ijro etuvchi orkestrdir. Har bir ishlov beruvchida faqat ma'lum bir hodisa sodir bo'lganda bajariladigan skript mavjud. Filmdagi har bir ssenariy hodisa tufayli ishga tushiriladi – ular foydalanuvchi sichqoncha ko'rsatgichini tugmaga bosgan yoki klaviaturani bosgan yoki film xronologiyada ma'lum bir kadrga yetgan va hokazo bo'lishi mumkin.

ActionScript da, odatda, har bir skriptning birinchi qatorida hodisa ishlovchisi (kadr bilan bog'liq hodisalar bundan mustasno) ro'yxatga olinadi. Masalan: **Qachon sodir bo'ladi ( hodisa tipi ) { nimadir qil; nimadir qil;**

}

Kadrga asoslangan hodisalar ijro etish paytida vaqt jadvali skriptni o'z ichiga olgan kadrga yetganda sodir bo'ladi. Kadrga joylashtirilgan skriptga hodisa ishlov beruvchisi kerak emas - bunday skript berilgan kadrga yetganda bajariladi. Shunday qilib, agar yuqoridagi skript kadrga birlashtirilgan bo'lsa, u quyidagicha ko'rinadi:

**nimadir qil;**

**nimadir qil;**

Hodisa ishlov beruvchilari tushunchalarini qanchalik yaxshi tushunsangiz, interaktiv elementlar ustidan nazoratingiz shunchalik to'liq bo'ladi. Ishlovchilardan to'g'ri foydalanish orqali foydalanuvchi o'zini huddi o'zini uyida his qiladigandek interaktiv muhitni yaratishingiz mumkin.

Ko'pgina kompyuter dasturlari foydalanuvchiga ekrandagi elementlarni sudrab olib tashlash, oynalar hajmini o'zgartirish, tugmalar yordamida muayyan sozlashlarni amalga oshirish va virtual vositalar yordamida o'zlarining "san'at durdonalarini" yaratish imkonini beradi. Ushbu interaktiv o'zaro ta'sirning barcha usullari dasturiy ta'minotda turli hodisalarini qayta ishlash (sichqoncha tugmachalarini bosish, sichqoncha harakati, klaviaturadan kiritish va boshqalar) qanday amalga oshirilishiga bog'liq.

## Kerakli ishlov beruvchini tanlash

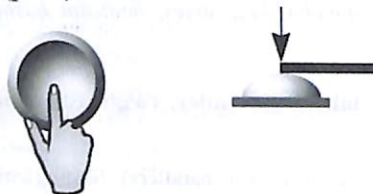
Hodisa ishlov beruvchilaridan to'g'ri foydalanish ActionScript foydalanuvchisi duch keladigan eng qiyin muammolardan biridir, chunki ko'pchiligimiz atrofimizdagi hodisalar qanday va nima uchun sodir bo'lishi haqida o'ylamaymiz. Ammo interaktiv muhitni yaratishda aynan shu narsa muhim - nima bilan nima va qanday aloqada bo'lishini aniq tasavvur qilish.

Flash hodisalari ishlov beruvchilari bilan tanishtirishda ularning real hodisalarga qanday aloqasi borligini ko'rib chiqamiz. Esda tutingki, Flash da har bir hodisa harakatga (va javobga) ega.

## 2. Sichqoncha va klaviatura hodisalari

Ushbu guruhning ishlov beruvchilari ekrandagi tugmalar va film belgilarining misollari orqali sichqonchani manipulyatsiyasi bilan bog'liq hodisalar uchun skriptlarning bajarilishini nazorat qiladi. Boshqa tomondan, Flash da MX sichqoncha hodisalarini tugmalar bilan bir qatorda film belgilariga ham qo'llash mumkin.

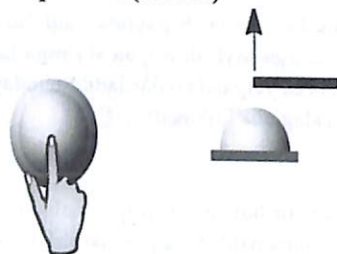
**Bosish, aloqa: on (press)**



7.2.1-rasm. Sichqoncha va klaviatura hodisalari on(press)buyrug'i.

Haqiqiy dunyoda, agar biror narsaga tegsangiz yoki biror narsani bossangiz (bu odam yoki muz kubi bo'lsin), reaksiyani kutasiz: muz kubi eriy boshlaydi, odam ham u yoki bu tarzda javob beradi. On (press) hodisa ishlovchisi teginish, ushlar, bosish kabi hodisalarni taqlid qilish uchun ishlatiladi. Bu ishlov beruvchi kursor tugma yoki film belgisi ustida turganda va sichqoncha tugmasi bosilganda skriptni ishga tushirish uchun ishlatiladi.

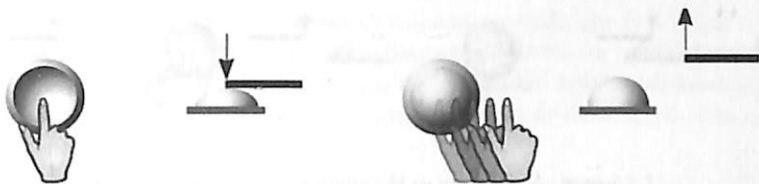
**Qo'yib yuborish, ozod qilish: on(release)**



7.2.2-rasm. Sichqoncha va klaviatura hodisalari on(release)buyrug'i.

Biror narsadan voz kechish, aloqani tugatish orqali odatda obyekt bilan o'zaro aloqani yakunlaydi. Ushbu ishlov beruvchi har qanday ozod qilishni simulyatsiya qiladi - bosishdan yoki tortib olishdan. Foydalanuvchi sichqoncha tugmasini bo'shatganda skriptni ishga tushirish uchun ishlatiladi (u avval tugmachani yoki film belgisi misolini bosgan).

**Siljitish, sudrash: on(releaseOutside)**

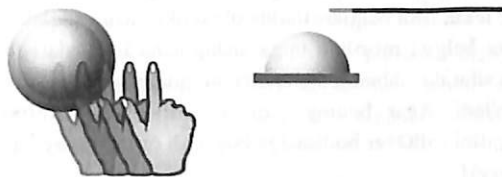


7.2.3-rasm. *Sichqoncha va klaviatura hodisalari on(releaseOutside)buyrug'i.* Stol ustida yotgan kartalar to'plamini tasavvur qiling. Barmog'ingizni yuqori kartaga bosing, uni pastki qismdan siljiting va barmog'ingizni qo'yib yuboring. Flash da foydalanuvchi bosgan, sichqonchani siljitgan va tugmani qo'yib yuborgan tugma haqida tassavvur qilishingiz mumkin. Ushbu ishlov beruvchi skriptni foydalanuvchi obyektidagi sichqoncha tugmachasini (ekrandagi tugma yoki film belgisi) bosganida va uni obyektidan tashqariga qo'yib yuborganida ishlaydi - barcha turdagi harakatlar va siljishlarni dasturlash uchun foydalidir. **Ko'rsatildi, lekin bosilmagan: on(rollOver)**



7.2.4-rasm. *Sichqoncha va klaviatura hodisalari on(rollOver)buyrug'i.* Issiq pechkadan issiqlikni his qilish uchun unga tegish shart emas - shunchaki qo'lingizni ko'taring. Biz taxmin qilamizki, bu ishlov beruvchi masofada harakat qila oladigan obyektlarni - issiqlik, sovuq, yorug'lik va boshqalarni taqlid qiladi. Shuningdek, ushbu ishlov beruvchidan tugma yoki film belgisining nusxasi ustiga kelganingizda ma'lumotni (paydo bo'luvchi izohlar) ko'rsatish uchun ham foydalanishingiz mumkin. Shunday qilib, ushbu ishlov beruvchi foydalanuvchi sichqonchani tugma yoki film belgisining nusxasi ustiga olib kelganida skriptni ishga tushirish uchun ishlatiladi.

**Ko'rsatkich chetga surildi: on(rollOut)**



7.2.5-rasm. *Sichqoncha va klaviatura hodisalari on(rollOut)buyrug'i.* Qo'lingizni pechkadan olib, tabiiy ravishda uning issiqligini his qilishni to'xtatasiz. Ushbu ishlov beruvchi aynan shunday vaziyatni simulyatsiya qiladi. Foydalanuvchi o'z ko'rsatgichini tugma yoki film belgisidan (ilgari sichqonchani ko'tarib turgan) uzoqlashtirganda skriptni ishga tushirish uchun ushbu ishlov beruvchidan foydalanishingiz mumkin.

**"Tepaliklar", "chuqurchalar", ishqalanish: on(dragOver)**



### 7.2.6-rasm. Sichqoncha va klaviatura hodisalari on(dragOver)buyrug'i.

Ushbu hodisa tasodifan biror narsaga tegib, darhol qo'lingizni tortib olgan vaziyatni simulyatsiya qilish imkonini beradi. Ushbu ishlov beruvchidan foydalanuvchi kursorni tugma yoki film belgisi ustiga olib borib, sichqoncha tugmasini bosgan va darhol kursorni yon tomonga surganida (sichqoncha tugmasi bosilganda) skriptni ishga tushirish uchun ishlatilishi mumkin.

#### **Klaviatura boshqaruvi: on(keyPress)**

Ushbu hodisa yordamida foydalanuvchi klaviaturadagi istalgan tugmachani bosganida bajariladigan skriptni dasturlashingiz mumkin. Istalgan tugmachani o'rnatishingiz mumkin - harf, raqam, tinish belgilari va maxsus belgilar, ko'rsatgich tugmalari, Backspace, Insert, Home, End, Page Up va Page Down.

#### **Ko'pgina sichqoncha hodisalari film belgisiga birlashtirilishi mumkin**

Flash 5 va undan oldingi versiyalarida sichqoncha hodisalari faqat tugma nusxalariga birlashtirilishi mumkin edi. Flash MX da sichqoncha hodisasi bilan ishlashni film belgilari uchun ham belgilashingiz mumkin. Ushbu yangi xususiyatdan maksimal darajada foydalanish uchun bir nechta narsalarni e'tibor qilish kerak - ayniqsa avval Flash ning oldingi versiyalari bilan ishlagan bo'lsangiz va tugma nusxalari uchun sichqoncha hodisalarini amalga oshirishga odatlangan bo'lsangiz.

- Film belgisi tugma sifatida ishlashi uchun unga sichqoncha hodisasi ishlov beruvchisini birlashtirish kerak. Aksariyat hollarda sichqoncha hodisalarini ham, klip hodisalarini ham bir misolga birlashtirish mumkin emas; ammo, ishlov berish usullaridan foydalangan holda film belgisi o'sha va boshqa hodisalarga darhol reaksiyaga kirishishi mumkin.

- Sichqoncha hodisalari tayinlangan film belgisi misoli Flash tomonidan tugma sifatida ko'rib chiqiladi, lekin film belgisi sifatida o'z funksiyasini saqlab qoladi.

- Agar film belgisi misolida unga sichqoncha hodisalari tayinlangan bo'lsa va Flash tomonidan tugma sifatida ishlansa, sichqonchani uning ustiga olib borganingizda, kursor qo'l sifatida paydo bo'ladi. Agar buning o'rniga oddiy kursor ko'rsatilishini istasangiz, useHandCursor xususiyatini rolOver hodisasiga bog'lash orqali "false" ga o'zgartiring:

```
on(rollOver){
    this.useHandCursor = false;
}
```

useHandCursor xususiyati tugma nusxalari uchun ham amal qiladi (pastga qarang). □ Sichqoncha hodisalarini film belgilariga (tugmalar kabi) qo'llashingiz mumkinligi, klip hodisalarini tugma nusxalariga qo'llashingiz mumkin degani emas.

- Tugma nusxalari nomlanishi mumkin va film belgilarining xususiyatlariga o'xshash xususiyatlarga ega (masalan, \_alfa, \_rotation, \_y va boshqalar). Shunday qilib, tugma nusxasini myButton deb nomlash orqali uning shaffofligini myButton .\_alpha = 50 harakati bilan o'zgartirishingiz mumkin.

- Tugma nusxalariga nom berish mumkin - natijada ular film belgilari, xossalari va usullarini olish kabi obyektlarga aylanadi. Biroq, tugmalar baribir mustaqil montaj stollari emas (film belgilari kabi) - yo'lni belgilashda this atamadan foydalanish kerakligini yodda tutish muhimdir. Misol uchun, agar siz quyidagi skriptdan foydalansangiz, film qahramoni nusxasi aylanadi: `on(press){`

```

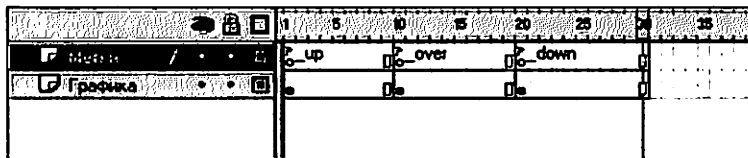
    this._rotation =
    30; }

```

Biroq, agar xuddi shu skript tugmachaga tayinlangan bo'lsa, unda tugmaning o'zi emas, balki uning "ota-onasi" - u joylashgan montaj stoli aylanadi! Ushbu kontseptsiyani esda tuting: film belgisi nusxasi har doim montaj stolning o'zi bo'ladi, hatto unga sichqoncha hodisasi ishlov beruvchilari birlashtirilgan bo'lsa va Flash uni tugma sifatida ishlatsa ham. Tugma, garchi u nusxa nomiga ega bo'lsa ham, xossalari va usullari, u joylashgan montaj stolining faqat bir qismi bo'lib qoladi.

Siz so'rashingiz mumkin, film belgilari endi tugmachalar kabi harakat qilishlari mumkin bo'lsa va film belgilari funksiyalarini saqlab qolsa, nega unda standart tugmalardan foydalanish kerak? Chunki, tugma belgisi o'zining standart uchta holatiga ega - up, over va down (yuqoriga, ustiga, pastga) oddiy tugmani amalga oshirishni ancha tezlashtiradi va osonlashtiradi. Shunday qilib, agar siz shunchaki tugmani xohlasangiz, tugma nusxasidan foydalaning. Agar siz tugmachaga ba'zi "maxsus effektlar" berishni istasangiz, u holda sichqonchanning hodisalari birlashtirilgan film belgisi kerak.

- Agar siz tugma sifatida film belgisidan foydalanayotgan bo'lsangiz, tugmacha film bilan o'zaro aloqada bo'lganda filmning ko'rinishini qanday o'zgartirishini tasvirlashni osonlashtirish uchun uning xronologiyasiga (\_up, \_over, \_down) yoriqlik uchta maxsus kadrni joylashtirishingiz mumkin. (ammo, bu shart emas – film belgisi usiz ham tugma sifatida ishlashi mumkin).



7.2.7-rasm. Klaviatura boshqaruvi.

- standart bosish maydoni (hit area) film tugmasi uchun uning grafik mazmuni bilan belgilanadi, lekin hitArea xususiyatidan foydalanib, bosish maydon istalgan vaqtda o'zgartirilishi mumkin. Masalan, quyidagi amal film tugmasi uchun hitArea sifatida boshqa film belgisini belgilaydi:

```
myClipButton.hitArea = "_root.myHitClip";
```

Bu xususiyat standart tugmalar uchun emas, faqat tugma sifatida ishlatiladigan film belgilari uchun mavjud.

### Kadr hodisalarini qo'llash

Kadr hodisalarini tushunish va o'zlashtirish osonroq. Agar montaj stoli kadriga bir qator amallarni birlashtirgan bo'lsangiz, Flash filmni o'ynatish vaqtida kadriga yetib borishi bilanoq bu amallarni bajaradi. Harakatlarni tetiklash yoki biror narsani hozirda sahnada bo'lgan vizual elementlar bilan sinxronlashtirish kerak bo'lganda, kadr hodisalari juda yaxshi.



**IMPORTANT**

**Maslahat:** Ko'pincha dinamik asosga ega bo'lgan loyihalarda, ijro etish boshlanishi bilan oq, ko'p harakatlar darhol bajarilishi kerak. Initsializatsiya deb nomlanuvchi ushbu jarayonda odatda ma'lumotlar elementlari yaratiladi, filmning turli umumiy

jihatlari aniqlanadi. Shu sababli, ishlab chiquvchilar ko'pincha ishga tushirish jarayoni uchun zarur bo'lgan asosiy montaj stolining 1-freymiga juda ko'p harakatlarni birlashtiradilar. Bu yerda o'zgaruvchilar va massivlar e'lon qilinadi, funksiyalar tavsiflanadi - umuman olganda, filmdagi skriptlar ijro boshlangandan so'ng darhol kerak bo'lishi mumkin bo'lgan hamma narsa joylanadi.

### 3. Klip hodisasini qo'llash

Agar sahnada unga skriptlar birlashtirilgan film belgisining nusxasi mavjud bo'lsa, bunday sahnaning tashqi ko'rinishini, funktsionalligini, foydalanuvchi bilan o'zaro munosabatda bo'lish qobiliyatini o'zgartirish nuqtai nazaridan klip hodisalaridan foydalanish orqali sezilarli darajada oshirish mumkin. Bunday hodisalarga sahnaga kirganda yoki undan chiqib ketganda, sichqonchanning sahna bo'ylab harakatlanishida va hokazolarda harakatlarni boshlashga imkon beradi.

Quyida klipdagi turli hodisalar tavsifi, shu hodisalarga qiyoslash mumkin bo'lgan real o'xshashliklar keltirilgan. Yodda tutingki, klip hodisalarini qayta ishlovchi skriptlar faqat film belgilarining nusxalariga birlashtirilishi mumkin.

#### **Mavjudligi: onClipEvent (load)**

Har qanday joyda (xonada) kimdir (yoki biror narsa) paydo bo'lganda, bu hodisa doimo qandaydir ta'sir qiladi: odamning xatti-harakati o'zgaradi, atrof-muhit yoki muhit u yoki bu tarzda o'zgaradi. Hatto odamning (yoki narsaning) o'zi ham ushbu xonada paydo bo'lishi natijasida o'zgarishi mumkin. Ushbu hodisani qayta ishlash vositasi sahnada film belgisi nusxasi paydo bo'lgan paytda skriptni ishga tushirish orqali shunga o'xshash narsani ishga tushiradi - film qahramonini ishga tushirishni amalga oshirish, sahnadagi nusxa ta'sirini ham, sahnaning unga ta'sirini ham dasturlash uchun foydalidir.

#### **Yo'qligi: onClipEvent (unload)**

Film belgisi paydo bo'lganda belgilangan sahnani (yoki uniki) o'zgartirishi mumkinligi sababli, u yo'qolganda nimadir sodir bo'lishi mumkin. Film belgisi sahnadan yo'qolganda skriptni ishga tushirish uchun ushbu ishlov beruvchidan foydalaniladi.

#### **Harakatlantiruvchi kuch, energiya: onClipEvent (enterFrame)**

Agar obyekt energiya, harakatlantiruvchi kuch bilan ta'minlangan bo'lsa, bu uning ma'lum vaqt davomida biron bir harakatni bajarishga qodir ekanligini anglatadi. Masalan, soatlarni olaylik. Prujinaning harakatlantiruvchi kuchi bo'lmasa, ular harakatsiz va foydasiz bo'lib qoladi. Agar ular ishlasa sonlar aylanasi bo'ylab yuguradi vaqtini ko'rsatib berish uchun. Ushbu hodisa ishlov beruvchisi ba'zi skriptlarni doimiy ravishda takrorlash uchun ishlatiladi - misol uchun filmning kadr tezligida. Agar chastota sekundiga 20 kadr bo'lsa, bu ishlov beruvchining skripti soniyada 20 marta bajariladi. Ushbu ishlov beruvchi uchun juda ko'p foydali foydalanish usullari mavjud.

#### **Ko'chirish: onClipEvent (mouseMove)**

Ushbu ishlov beruvchini harakat detektorini sifatida tasavvur qilishingiz mumkin. Agar sahnada bunday ishlov beruvchi birlashtirilgan film belgisining nusxasi mavjud bo'lsa, amallar

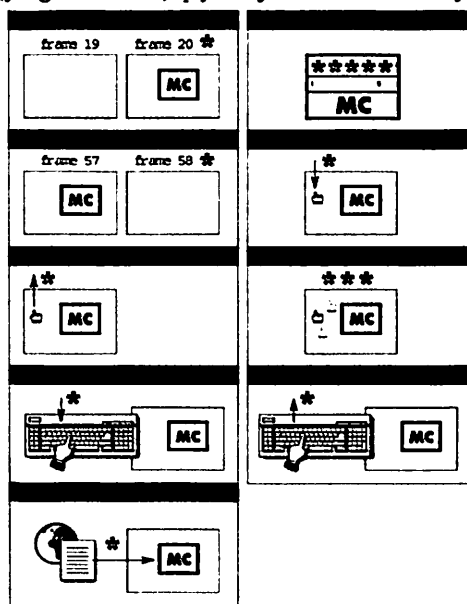
to'plami foydalanuvchi sichqonchani har safar harakatlantirganda (kamida bitta piksel) bajariladi. Bu harakat yo'nalishini, ko'rsatgichning joriy holatini va boshqalarni aniqlash kabi harakat bilan bog'liq interaktiv shovqinlarni dasturlash imkonini beradi.

**Kompyuter bilan o'zaro ta'sir: onClipEvent ( mouseDown ), onClipEvent ( mouseUp ), onClipEvent ( keyDown ), onClipEvent ( keyUp )**

Sichqoncha va klaviatura faqat foydalanuvchining kompyuter bilan o'zaro aloqasi uchun mo'ljallanganligi sababli, bu hodisalarning haqiqiy ekvivalentlari mavjud emas. Ammo ular biron bir tugma yoki sichqoncha tugmasi bosilsa (yoki qo'yib yuborilsa) skriptlarni bajarishga imkon beradi. Ushbu hodisalar ilgari tasvirlangan sichqonchani press, release va keyPress hodisalariga juda o'xshash bo'lsada, ular aslida kuchliroqdir. keyUp va keyDown klip hodisalari ishlov beruvchilaridan foydalanib, ilovangizdagi buyruqlarni tezda tanlash uchun klaviatura yorliqlarini dasturlashingiz mumkin. keyPress sichqoncha hodisasi esa amallarni faqat bitta tugmaga belgilash imkonini beradi. mouseUp va mouseDown klipi hodisalari sichqonchani press va release hodisalaridan farq qiladi, chunki ular foydalanuvchi to'g'ridan-to'g'ri tugmani bosganida skriptni ishga tushiradi, mouseUp va mouseDown hodisalari esa sahnaning istalgan joyida sichqoncha tugmasi bosilishi yoki qo'yib yuborilishiga javob beradi.

**Ko'rsatmalarni olish: onClipEvent ( data )**

Haqiqiy hayotda to'liq bo'lmagan ko'rsatmalar har xil muammolarni keltirib chiqarishi mumkin. Flashda ham huddi shunday. Flash turli xil ma'lumotlarni (o'zgaruvchilar, SWF va JPG fayllari) tashqi manbalardan film belgisi nusxasiga yuklash imkonini berganligi sababli, ushbu hodisa uchun ishlov beruvchi nusxaga birlashtirilgan skript faqat manbadan to'liq yuklab olingan ma'lumotlardan keyin bajarilishini ta'minlash uchun juda muhimdir. Shunday qilib, to'liq bo'lmagan ko'rsatmalar tufayli xatolar paydo bo'lishining oldini olamiz. Agar ma'lumotlar klipga bir necha marta yuklangan bo'lsa, ushbu ishlov beruvchidan harakatlarni (yangilanishlarni) qayta bajarish uchun ham foydalanishingiz mumkin.



7.3.1-rasm. Ko'rsatmalarni olish onClipEvent ( data ).

Klip hodisalarining vizual tasviri. Pastga qaratilgan o'q sichqoncha tugmachasini bosishni bildiradi; yuqoriga o'q - qo'yib yuborish. Yulduzcha hodisa sodir bo'lganligini bildiradi.

**Turli hodisalarning kombinatsiyasi**

Turli xil hodisalarni qayta ishlashingiz mumkinligini bilasiz.

Masalan, foydalanuvchi kursorni harakatlantirganda sichqoncha tugmasini bosgan holda ushlab turishi (press sichqoncha hodisasi) va ekran bo'ylab aylantirish mumkin (mouseMove klip hodisasi). Turli hodisalarni qayta ishlash muvofiqlashtirilgan dasturlash orqali filmning interaktiv imkoniyatlarini yanada oshirishingiz mumkin.



Biz allaqachon ko'plab Flash hodisalarini ko'rib chiqdik, ammo bu, ta'bir joiz bo'lsa, aysbergning faqat uchi. Skriptni sichqonchani o'zaro ta'siriga, vaqt jadvalining ma'lum bir kadrğa yetib borishiga, sahnada film belgisining yangi nusxasi paydo bo'lishiga qanday javob berishni o'rgandik. Biroq, hodisalarni qayta ishlash usullaridan foydalanib, skriptni ovoz yozishning tugashi, sahna o'lchamining o'zgarishi yoki matn maydonining sarlavhasini o'zgartirish kabi narsalarga munosabat bildirishingiz mumkinligini bilarmidingiz? Ishlov beruvchi usullari yordamida ushbu darsda biz allaqachon o'rganib chiqqan hodisalarga munosabatni o'zgartirishingiz mumkin.

Garchi ishlov beruvchi usullar va standart ishlov beruvchilar bir xil maqsadga xizmat qilsalar ham (filmda biror narsa sodir bo'lganda ular skriptni ishga tushirishlari kerak), dasturda ularni amalga oshirish usullari boshqacha.

Hodisa natijasida bajariladigan skriptni qanday yaratishni bilasiz. Aytaylik, quyidagi skript myMovieClip film belgisining nusxasiga biriktirilgan; u sichqoncha tugmasi bosilganda bajariladi, agar bu nusxa sahnada mavjud bo'lsa:

```
onClipEvent ( mouseDown ) {
    _rotation = 45;
}
```

Sichqoncha tugmachasini bosganingizda, skript nusxani 45 darajaga aylantiradi.

Ishlov beruvchi usuldan foydalanib, quyidagi skriptni montaj stoliga qo'yish mumkin, bu xuddi shu narsani bajaradi - sichqoncha tugmasi bosilganda myMovieClip ni aylantiradi.

```
myMovieClip.onMouseDown = function(){
    myMovieClip ._rotation = 45;
}
```

onClipEvent dan foydalanish o'rniga, bu yerda obyekt nomini (bu holda myMovieClip) reaksiya sodir bo'ladigan hodisadan ajratib turadigan nuqta qo'yamiz. Biz ushbu skriptni nusxaga biriktirmasdan, kadrğa joylashtiramiz.

Yuqoridagi skriptda function() kalit so'zi sizni xavotir solmasin. function() so'zi ishlov beruvchi usulini amalga oshirish sintaksisining zarur elementidir.



**IMPORTANT**

**Maslahat:** Funksiyalar bilan yaqindan tanishganingiz sari, ba'zi hollarda ishlov beruvchi usul sifatida ma'lum bir funksiyadan foydalanishingiz mumkinligini tushunasiz, buning uchun sintaksisni shu tarzda o'zgartirishingiz kerak bo'ladi : myMovieClip.onMouseDown = FunctionName;

Skriptning figurali qavslari ichida ko'rsatilgan harakatlar hodisa sodir bo'lganda nima qilish kerakligini tasvirlaydi.

**Eslatma:** E'tibor bering, birinchi skriptda maqsadga yo'l ko'rsatilmagan (\_rotation = 45), ikkinchi holatda esa u ko'rsatilgan (myMovieClip .\_rotation = 45). Gap shundaki, birinchi skript myMovieClip ga biriktirilgan, shuning uchun yo'l kerak emas - film belgisining o'zi maqsad deb taxmin qilinadi. Ikkinchi skript kadrğa joylashtirilgan, shuning uchun Flash qaysi obyektni aylantirish kerakligini tushunishi uchun yo'lni belgilashingiz kerak. Ishlov beruvchi usullardan yuqorida tavsiflangan usulda foydalanilganda, harakat qilinayotgan maqsadga yo'lni to'g'ri belgilashni unutmang.

Ushbu skript myMovieClip nusxasi onMouseDown hodisasiga qanday javob berishini tasvirlab berganligi sababli, ushbu hodisani qayta ishlash usuli aniqlangan vaqtda ushbu nusxa sahnada bo'lishi kerak. Keyin tasvirlangan funktsionallik nusxaga biriktiriladi. Agar obyekt

sahnani tark etgan bo'lsa (yoki boshqacha tarzda o'chirilgan bo'lsa), unga tayinlangan ishlov berish usuli ham o'chiriladi. Agar shundan keyin obyekt yana sahnada paydo bo'lsa, ishlov berish usuli tasvirlanishi va qayta tayinlanishi kerak.

Siz oddiy usullardan farqli ishlov berish usullariga hayron bo'lishingiz, ehtimol hatto birinchisining afzalliklari bor-yo'qligiga shubha qilishingiz ham mumkin. Bu haqida hozir gaplashamiz.

**Eslatma:** Ishlov beruvchi usullari o'z obyektlaringizni yaratishda va ularning hodisalarga qanday munosabatda bo'lishini tasvirlashda juda muhim rol o'ynaydi.

#### **Hodisalar ishlovchilari sifatida usullardan foydalanish**

Barcha standart hodisa ishlov beruvchilarida ekvivalent ishlov berish usullari mavjud.

Mana bir nechta misollar:

**on(press) – TugmaNomi.onPress yoki FilmBelgiNomi.onPress**

**on(release) – TugmaNomi.onRelease yoki FilmBelgiNomi.onRelease**

**on(enterFrame) – FilmBelgiNomi.onEnterFrame**

Bundan tashqari, standart hodisa ishlovchilar orasida ekvivalenti bo'lmagan quyidagi ishlov berish usullari mavjud:

**Tugmalar / Film belgilari**

**MovieNameOrButtons.onKillFocus**

**MovieNameOrButtons.onSetFocus**

**Ovoz**

**SoundObjectName.onLoad**

**nameSoundObject.onSoundComplete**

**Matn maydonlari**

**TextFieldName.onChanged**

**TextFieldName.onKillFocus**

**TextFieldName.onScroller**

**TextFieldName.onSetFocus**

**Obyekt LoadVars**

**ObjectLoadVarsName.onLoad**

**XML**

**XMLObjectName.onData**

**XMLObjectName.onLoad**

**XML Soket**

**XMLSocketObjectName.onClose**

**XMLSocketObjectName.onConnect**

**XMLSocketObjectName.onData**

**XMLSocketObjectName.onXML**

Ko'rib turganingizdek, skriptni ishga tushirish uchun juda ko'p turli hodisalar ishlatilishi mumkin. Ushbu obyektlarning ba'zilari "nomoddiy" bo'lgani uchun (masalan, tovush, LoadVars, XML va boshqalar), ishlov berish usullaridan foydalanish ( montaj stolidagi kalit kadrda) bunday obyekt bilan bog'liq hodisa sodir bo'lganda skriptni ishga tushirishning yagona usuli hisoblanadi . (tugmalar va film belgilaridan farqli o'laroq, ular shunchaki sahnada tanlanishi va to'g'ridan-to'g'ri skriptga birlashtirilishi mumkin).

**Eslatma.** Biz ushbu kursdagi mashqlarda ushbu ishlov berish usullaridan ko'p foydalanamiz. Har biri haqida batafsil ma'lumot uchun ActionScript lug'atiga qarang .

Tugma yoki film belgisi misoliga skriptni biriktirish orqali nafaqat skriptni ishga tushiradigan hodisani tanlashda, balki bajariladigan amallarni tanlashda ham cheklovlariga duch kelasiz. Keling, ushbu misolni olaylik:

```
on(press){
  gotoAndPlay ( 5);
}
```

Agar bunday skriptni tugmachaga biriktirsangiz, u faqat press hodisasiga javob beradi va faqat bunday hodisa sodir bo'lganda ushbu amalni bajaradi. Endi ishlov berish usullarining moslashuvchanligi va kuchi haqida tushuncha berish uchun, keling, sahnada **myButton** deb nomlangan tugma nusxasi mavjud deb faraz qilaylik. Hodisalarga javobni tasvirlash uchun quyidagi skriptni asosiy vaqt jadvalining 1- freymiga joylashtiramiz (agar bu freymda tugma mavjud bo'lsa):

```
myButton.onPress = function() {
  stopAllSounds ();
}
myButton.onRelease = function() {
  myMovieClip ._xscale = 50;
}
```

Tugmani bosish barcha ovozi yozuvlarni tinglashni to'xtatadi; myMovieClip ni qo'yib yuborilganda asl hajmining 50 foizigacha masshtablashtiriladi.

Biroq, montaj stolini 2- kadrda o'tkazish orqali (tugma shu kadrda ham mavjud bo'lsa), biz tugmani qanday ishlashini butunlay o'zgartirishimiz mumkin:

```
myButton.onPress = null
myButton.onRelease = null
myButton.onRollOver = function(
) {
  stopAllSounds ();
}
myButton.onRelease = function() {
  myMovieClip ._xscale = 50;
}
```

Null (bo'sh funksiya) dan foydalanib, tugmani onPress va onRelease hodisalariga javob berishini oldini oldik va buning o'rniga ikkita yangi hodisaga javob berishni aytdik.

Ko'rib turganingizdek, ishlov berish usullaridan foydalangan holda siz istalgan vaqtda tugmaning funksiyalarini va u reaksiyaga kirishadigan hodisalarni o'zgartirishingiz mumkin - bu muhim afzallik!

Ishlov berish usullari dinamik ravishda yaratilgan obyektlar uchun oddiygina ajralmas hisoblanadi. Dinamik yaratish jarayonida filmda dastlab mavjud bo'lmagan obyekt paydo bo'ladi (allaqachon o'ynash jarayonida). Shuning uchun, sahnada belgilab, bunday obyektga bo'lgan hodisalarga reaksiyani tasvirlay olmaysiz (bu hozircha mavjud emas!). Bu yerda ishlov berish usullari yordam beradi. Quyidagi oddiy skriptga qarang va bu qanday amalga oshirilishini tushunasiz:

```
_root.createEmptyMovieClip (" newClip " , 1);
_root.newClip.onEnterFrame = function(){
  myVariable ++;
}
```

```

_root.newClip.onMouseMove = function(){
myCursorClip ._x = _root._xmouse ;
myCursorClip ._y = _root._ymouse ;
}

```

Ko'rib turganingizdek, newClip nomli film belgisining nusxasini yaratgandan so'ng, darhol ishlov beruvchi usullaridan foydalangan holda, ushbu nusxa muayyan hodisalarga qanday javob berishini tasvirlab berdik.

## 5. stop() va play() funksiyalari . O'tish operatori va kadr yorlig'i

ActionScript o'yinchi boshining bu harakatini o'zgartirish uchun stop() usulidan foydalaniladi. Uni to'xtatmoqchi bo'lgan filmning asosiy kadriga qo'shish kerak. Stop() usuli global emas va shuning uchun kodning biron bir qismidan chaqirib bo'lmaydi. U MovieClip sinfining usuli bo'lib, undan MainTimeLine klassi voris qilib olinadi. Film boshlanganda yaratiladigan obyekt MainTimeLine sinf obyektidir. Kadr kodini tahrirlashda this belgisini o'tkazib yuborishingiz mumkin, lekin baribir this.stop() har bir stop() da , this.addChild() - addChild () va hokazo bilan nazarda tutiladi.

### stop() ActionScript usuli. Film to'xtashi

#### Flash CS5 da harakatlanuvchi obyekt yaratish

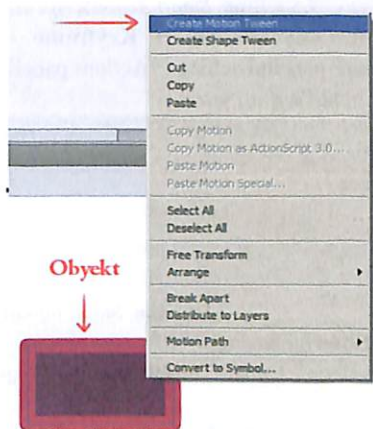
Chizish vositalaridan har qanday foydalanib, sahnada obyekt yarating.

Tanlash vositasini tanlang (selection tool) va chizgan obyektning bosing.

F8 tugmasini bosing. Convert to Symbol dialog oynasi paydo bo'ladi. (belgiga aylantirish).

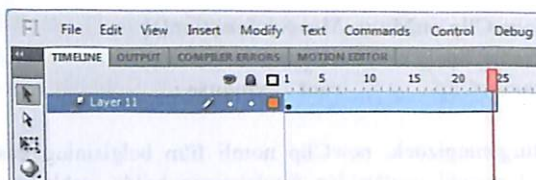
MovieClip turini tanlang va OK tugmasini bosing.

Sichqonchanning o'ng tugmasi bilan belgini bosing. Ko'rsatilgan kontekst menyusida Create Motion Tween ( Motion Tween yaratish).



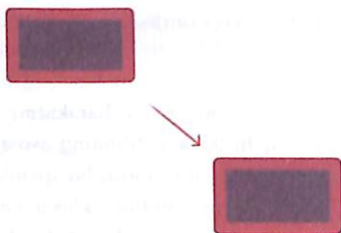
7.5.1-rasm. Flash Motion Tween.

Flash Motion Tweenni vaqt oralig'iga qo'shadi.



7.5.2-rasm. Flash Motion Tweenni vaqt oralig'iga qo'shish.

Obyektni sahnadagi yangi joyga torting.



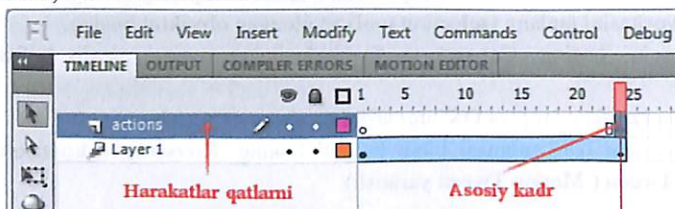
7.5.3-rasm. Obyektni sahnadagi yangi joyga ko'chirish.

### Stop() ActionScript usulidan foydalanish

Yangi qatlam yarating - New Layer tugmani bosning.

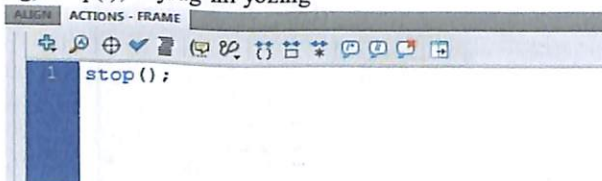
Harakatni to'xtatmoqchi bo'lgan kadrni ustida sichqoncha o'ng tugmasini bosning.

Bizning holatda, bu 25-kadr bo'ladi.



7.5.4-rasm. Stop() ActionScript usulidan foydalanish..

Kontekst menyusida **Insert Keyframe -ni bosning (kalitli kadrni kiritish)**. Actions panelini oching (" Actions panelidan foydalanish" mavzusiga qarang) stop( ); buyrug'ini yozing



7.5.5-rasm. Stop() funksiyasi.

**Ctrl + Enter** ni bosning .

Flash Player videoni o'ynata boshlaydi. Yaratilgan animatsiya ko'rsatilgan kadrda to'xtaydi.

play() usuli bilan qayta boshlashingiz mumkin. Xuddi stop ( ) ActionScript usuli kabi, bu usul hech qanday argument talab qilmaydi, chunki u hech qanday qo'shimcha ma'lumotga muhtoj emas: u faqat o'yinchining boshi to'xtagan kadrda boshlanadi va harakatni davom ettiradi.

AS3 ( ActionScript 3 ) yordamida harakatlanuvchining boshini vaqt oraliq'idagi kerakli kadrda siljitishingiz mumkin. Shundan so'ng, film yo to'xtaydi yoki belgilangan kadrda o'ynay boshlaydi. Bu ikkita global usullardan qaysi biri chaqirilganiga bog'liq: **gotoAndPlay()** **ActionScript** yoki **gotoAndStop()** .

Ikkala usul ham bitta argumentni oladi: o'yinchi boshi harakatlanishi kerak bo'lgan kadr raqami. Ushbu kadrni ikkita usuldan biri orqali belgilashingiz mumkin:

- uning raqamini ko'rsatib
  - unga tayinlangan yorliqni ko'rsatib.
- Flash -da kadrda yorliq belgilash oddiy: □ vaqt oraliq'ida kerakli kadrni tanlang
- xususiyatlar panelida yorliq nomini kiriting

Yorliqlar faqat asosiy kadrlarga tayinlanishi mumkin! AS3 ( ActionScript 3 ) yorliqlari ma lumotlar turiga ko'ra satrlar ( String ) bo'lgani uchun ular bo'shliqlar va boshqa maxsus belgilarni o'z ichiga olishi mumkin. Biroq, yorliq nomidagi harflar holatini ko'rib chiqish muhimdir!

Ko'pincha, ma'lum bir kadrda o'tish film yoki veb-sayt orqali harakat qilish bilan bog'liq. Bu foydalanuvchi sahifadan sahifaga yoki bo'limdan bo'limga o'tadigan tugmalar yordamida amalga oshiriladi. Agar film oxirida to'xtatilgan bo'lsa, oxirgi kadrda boshiga qaytadigan va filmni ko'rish uchun yana ishga tushiradigan tugmani yaratishingiz mumkin.

Bunday navigatsiyani yaratish orqali foydalanuvchilarga saytda qolish yoki film tomosha qilish vaqtini boshqarishga ruxsat berasiz. Ular o'zlarini passiv tomoshabinlar emas, balki jarayonni nazorat qila oladigan odamlar kabi his qiladilar.

Berilgan kadrda o'tish

Filmni Motion Tween bilan to'xtatish

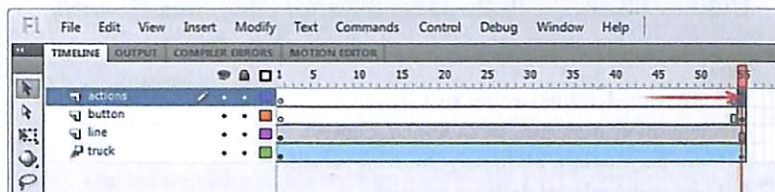
Motion Tween bilan film yaratish.

Uni oxirgi kadrda to'xtatish ("ActionScript stop() usuli . Filmni to'xtatish"). Filmni ijro etish uchun tugma yaratish Yangi qatlam yaratish.

Videongizning oxirgi kadrini o'ng tugmasini bosing. Kontekst menyusidan Inset Keyframe -ni tanlang(asosiy kadr yaratish uchun F6 tugmasidan ham foydalanishingiz mumkin).

Tugma yaratish ( Flash CS5 da ActionScript belgisini yaratish mavzusiga qarang ).

Tugmachaga nom bering, masalan btnReplay. stop() usuli qo'shilgan qatlamda oxirgi freymni bosing.



7.5.6-rasm. Stop() funksiyasi kiritilgandan keyin vaqt diagrammasida qatlam ko'rinishi.

Actions panelini oching.

stop()ActionScript usuli ostidagi satrga tugma nomini kiriting, nuqta qo'ying va keyin

yoziq `addEventListener ( MouseEvent.CLICK ,  
replay);` Ishlov beruvchi hodisalarni yaratinq  
`function replay(event:MouseEvent):void {  
}`

Figurali qavslar ichiga  
`gotoAndPlay (1);`  
`stop();`  
`btnReplay.addEventListener(MouseEvent.CLICK, replay);`  
`function`  
`replay(event:MouseEvent):void{`  
`gotoAndPlay(1); }`

Ctrl + Enter ni bosinq.

Video o'ynay boshlaydi. Oxirgi kadrda u to'xtaydi va tugma paydo bo'ladi.

Tugmani bosinq. Video boshidan o'ynay boshlaydi.

"Boshiga qaytish" tugmasini bosinq.

## 6. Nazorat savollari



1. **On (press)** hodisasi ishlov beruvchisi qachon ishlatiladi ?
2. **On (release)** hodisa ishlov beruvchisi qachon ishlatiladi ?
3. **On (releaseOutside)** hodisa ishlov beruvchisi qachon ishlatiladi ?
4. **On (rollOver)** hodisa ishlov beruvchisi qachon ishlatiladi ?
5. **On (rollOut)** hodisa ishlov beruvchisi qachon ishlatiladi ?
6. **On (dragOver)** hodisa ishlov beruvchisi qachon ishlatiladi ?
7. **On (keyPress)** hodisa ishlov beruvchisi qachon ishlatiladi ?
8. Film tugmasidagi bosish maydonini o'zgartirish uchun qanday xususiyatdan foydalanish mumkin?
9. Hodisa kadrlari qanday qo'llaniladi?
10. Kliplar uchun qanday maxsus ishlov beruvchilar mavjud?
11. Qaysi ishlov beruvchilar kompyuter bilan ishlash uchun mo'ljallangan?
12. Hodisalarni qayta ishlash usullari haqida nimalarni bilasiz?

## 2.8. Rolik va videoroliklarning xususiyatlari. Matn maydonlarini yaratish.

### Roliklarda shartli if, if...else operatorlaridan foydalanish

#### Reja:

1. Rolik va videoroliklarning xususiyatlari.
2. Kliplarni ko'paytirish, bo'sh kliplar yaratish.
3. Matn maydonlari. Matn maydonlarini yaratish.
4. Matn maydonlarini dasturiy yaratish.
5. Roliklarda shartli if, if...else operatorlaridan foydalanish.
6. Nazorat savollari.

**Kalit so'zlar:** *klip, rolik, flesh, ko'paytirish, matn maydoni, obyekt, dinamik maydon, statistik maydon, kiritish maydoni, xususiyat, level, MovieClip, attachMovie, swapDepth.*

## 1. Roliklar va videorolik (klip)larning xususiyatlari

Xususiyatlarni dasturiy tarzda qanday boshqarishni ko'rib chiqamiz, umuman olganda, muharrir orqali sozlagan kliplarimizni keyinchalik algoritmik tarzda o'zgartirishni xohlarmiz. Bunday xususiyatlarga misollar klipning joylashuvi, burilish burchagi, o'zaro perpendikulyar o'qlar bo'ylab cho'zish yoki siqish va shaffoflikdir. Xususiyatlar bilan ishlash sintaksisi oddiy o'zgaruvchilar bilan ishlash kabi oddiy, masalan: `clip._x = 10`. Lekin, oddiy o'zgaruvchidan farqli o'laroq, xususiyatga yangi qiymat berish tegishli harakatni keltirib chiqaradi (masalan, `clip._x = 10` tenglik natijasi klipni shunday holatga o'tkazadiki, uning dastlabki klipning koordinatalari 10 ga teng absissaga ega bo'ladi). Bu yerda eng ko'p ishlatiladigan xususiyatlar mavjud. Kliplar xususiyatlari izohlari bilan:

- **\_alpha** - shaffoflik. Foizlarda. 100 qiymati klipning normal ko'rinishi, 0 qiymati esa butunlay shaffofdir.
- **\_height** - piksellardagi klipning vertikal o'lchami.
- **\_name** - klip nusxasining satrli nomi.
- **\_rotation** - klipning dastlabki holatiga nisbatan burilish burchagi (soat miliga teskari, darajalarda o'lchangan).
- **\_target** (faqat o'qish uchun) - bu klipning ierarxiyadagi joylashuvini tavsiflovchi satr.

Nuqtalar o'rniga (odatdagidek) ierarxiya darajalari oldinga qiyshiq chiziq bilan ajratiladi. Agar klip yo'li `_level 0` bilan boshlansa, `_level 0` so'zining o'zi o'tkazib yuboriladi va yo'l to'g'ridan-to'g'ri qiyshiq chiziq bilan boshlanadi (masalan, `_level 0/clip` o'rniga oddiy `/clip` qilib yoziladi). Agar klipga yo'l `_level 1`, `_level 2` yoki noldan boshqa har qanday darajadan boshlansa, u holda daraja aniq ko'rsatilgan. Eslatib o'tamiz, nolga teng bo'lmagan daraja alohida \*.swf faylidan yuklangan klipni bildiradi. E'tibor bering, klipga to'liq yo'lni ko'rsatishda `_root` so'zi foydasiz, chunki u faqat bitta ma'lum klip ichida ishlaydi va bir-biriga yuklangan kliplarni tasvirlamaydi. Ammo bu holda `_level` so'zi ancha ma'lumotga ega (va shuning uchun u ishlatiladi).

- **\_url** (faqat o'qish uchun) - berilgan klip joylashgan \*.swf faylining URL manzili. Agar boshqa videolarni ma'lum bir videoga (tayyor \*.swf-fayllar ko'rinishida) yuklash mumkinligiga yana bir bor e'tibor qaratsak, ushbu xususiyatning foydaliligi aniqroq bo'ladi. Qo'shimcha video asl ishga tushirilgan joydan butunlay boshqa joydan yuklanishi mumkin (bundan tashqari, videoni yuklash manzili dinamik ravishda, ya'ni bajarish paytida yaratilishi mumkin).

- **\_visible** - mantiqiy xususiyat bo'lib, u klip ko'rsatilgan yoki to'liq yashirilganligini ko'rsatadi. `_visible == false` holati `_alpha == 0` holatidan farq qiladi, chunki ikkinchi holatda klip sichqonchani bosish sodir bo'lganligi to'g'risida xabar olmaydi (va shunga o'xshashlar, ular uchun sichqonchani klip ustida yoki yo'q bo'lishi muhim emas). Belgilar kadrlarida yozilgan kod `_visible` xususiyati qiymatidan qat'iy nazar, har qanday nusxa uchun bajariladi.

- **\_width** - piksellardagi klipning gorizontal o'lchami.



- `_x` - berilgan klipning boshlang'ich pozitsiyasining abtissasi, ota-ona koordinatalarida ifodalangan.

- `_xmouse` (faqat o'qish uchun)

- `_xscale` - joriy gorizontal o'lchamning asl o'lchamga nisbati (foizda ifodalangan). `_xscale` va `_width` xossalari o'zaro bog'liq: biri o'zgariganda ikkinchisi ham o'zgaradi.

- `_y` - berilgan klipning boshlang'ich pozitsiyasining ordinatasi, ota-ona koordinatalarida ifodalangan.

- `_ymouse` (faqat o'qish uchun)

- `_yscale` - joriy vertikal o'lchamning asl o'lchamga nisbati (foizda ifodalangan). `_yscale` va `_height` xossalari o'zaro bog'liq: biri o'zgariganda ikkinchisi ham o'zgaradi.

Qolgan barcha xususiyatlarni o'zgartirish mumkin. Hatto klip nusxasi nomi mavjud bo'lgan `_name` xususiyatini ham! Haqiqatan ham, keling Flash filmi yarataylik, uning sahnasida ixtiyoriy klip qo'yamiz (va bu nusxani klip deb nomlaymiz). Keyin birinchi `_root` kadrda quyidagi kodni yozamiz:

```
a = clip;
trace("clip._name = " + clip._name);
trace("newName._name = " + newName._name);
clip._name = "yangi nom"; trace("-----
");
trace("a._name = " + a._name);
trace("clip._name = " + clip._name);
trace("newName._name = " +
newName._name); Rolikni ishga tushirgandan
so'ng (Ctrl+Enter) konsolda quyidagilarni olamiz:
clip._name = clip newName._name =
-----
a._name = yangi nom
clip._name =
newName._name = newName
```

Videoning nomi haqiqatan ham o'zgardi! Va biz unga yangi nom bilan murojaat qilishimiz mumkin, lekin eskisi bilan - yo'q! Ism o'zgarishidan oldin biz klipga qilgan havola ishlayotganligidan dalolat beradi.

Keling, savol beraylik, agar obyekt yoki xususiyat nomini faqat dastur kodining ishlash vaqtida bilsak, xususiyatlarga kirish mumkinmi? Ya'ni, obyekt va/yoki xususiyatlarning nomlari qandaydir o'zgaruvchida yozilsa (yoki ular shunchaki qandaydir ifodani baholash natijasimi)? Ma'lum bo'lishicha, buni bilish mumkin va buning uchun `setProperty` funksiyasi mavjud. U quyidagi parametrlarni qabul qiladi: klip nomi satrli, xususiyat nomi (qo'shtirmoqsiz!), xususiyatga o'rnatilgan qiymat. Masalan: `setProperty("clip", _visible, false)`;

Ammo qulay kvadrat qavslar bilan ham shunday qilish mumkinmi? Ha. Bu shunday yoziladi: `clip["_visible"] = false`;

Xususiyat nomi bilan satrdan foydalanishga muvaffaq bo'ldik, bu qulay, chunki bizga qaysi xususiyat kerakligi haqidagi ma'lumotni o'zgaruvchiga joylashtirish mumkin. To'g'ri, biz bu yerda klip nomini satr sifatida o'rnatolmadik, lekin buni tuzatish mumkin, masalan, quyidagicha: `_root["clip"]["_visible"] = false`;

Lekin `setProperty` qulayroqdir, chunki birinchi argumentdagi yo'lni

"clip1.innerClip.innerInnerClip" (yoki slesh (qiya chiziq) formatda "clip1/innerClip/innerInnerClip") shaklida berishingiz mumkin. Ya'ni, **setProperty** ni chaqirish uchun ushbu funksiyaga berilgan manzilda boshqa qanday kliplar joylashganini oldindan bilishimiz shart emas.

## 2. Kliplarni ko'paytirish, bo'sh kliplar yaratish

Klip nusxalari dasturiy tarzda ham yaratilishi mumkin. Kliplarni dasturiy ravishda yaratishning uchta usuli mavjud. Ulardan ikkitasi kutubxonada mavjud bo'lgan belgilar namunalarni yaratish uchun mo'ljallangan, ulardan biri **duplicateMovieClip** – hozir ko'rib chiqamiz, ikkinchisi **attachMovie** (uni o'zingiz o'rganishingiz kerak bo'ladi). Uchinchi (**createEmptyMovieClip**) bo'sh klip yaratadi, keyinchalik u turli maqsadlarda ishlatilishi mumkin: uni dasturiy ravishda chizish yoki **attachMovie** yordamida unga bolalar kliplarini birlashtirish mumkin.

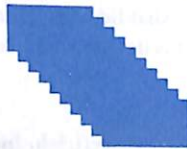
Xo'sh, **duplicateMovieClip** funksiyasi nima? U bizning rolikimizdagi klip misollarini takrorlash uchun mo'ljallangan va uchta argumentni oladi. Birinchisi, biz ko'paytirmoqchi bo'lgan nusxaga yo'l (yo'l klip yo'li xususiyatda saqlangan bir xil formatda o'tkaziladi . **\_target** ). Ikkinchi argument - bu yangi nusxaning nomi (unga yo'l asl nusxaga boradigan yo'ldan faqat shu nom bilan farq qiladi; ya'ni dublikat asl nusxaning egasi bo'lgan bir xil klipga tegishli bo'ladi) . Uchinchi argument - bu darajadagi chuqurlik deb ataladigan narsa. Biroq, bu nom parametrning ma'nosiga to'liq mos kelmaydi: haqiqat shundaki, bu raqam qanchalik katta bo'lsa , klip z-tartibida shunchalik baland bo'ladi. Shuni ham yodda tutingki, chuqurlik tartibi faqat bitta ma'lum bir klipga tegishli bo'lgan nusxalar uchun amal qiladi (xususan, ildiz klip ham ega bo'lishi mumkin ). Agar kliplar egalari boshqacha bo'lsa, u holda z-tartibi egalari qanday buyurtma qilinganiga qarab belgilanadi.

Shu o'rinda muhim terminologik izohni keltiramiz. Biz tayyor roliklarni dinamik yuklash haqida biroz oldinroq gaplashdik va u yerda daraja raqamlari ham paydo bo'ldi. Shunday qilib, kliplar uchun daraja raqamlari kliplar uchun daraja raqamlari bilan bir xil emas . Z-tartibga ta'sir qilish nuqtai nazaridan , eng "katta" - bu rolikning darajasi, keyin esa kliplarning joylashuvi - ztartibidagi egalari va shundan keyingina alohida kliplarni yuklash darajasi keladi (bu bo'lishi kerak). z-tartibdagi joylashuvi bo'yicha solishtirilsa, hisobga olinadi kliplar bir xil egasiga tegishli.

Shunday qilib, keling, amaldagi kliplarni ko'paytirishga harakat qilaylik. Ichkarida chizilgan to'rtburchak belgisini yarataylik (ramkasiz; chizma panelida bunday to'rtburchaklar yaratish uchun "bo'sh" chiziq rangini tanlash qulay - bu rasmning yuqori o'ng palitra qismida chizilgan kvadrat bilan ifodalanadi). Keling, ushbu belgining bir nusxasini yaratamiz va nusxani **sourceClip** deb nomlaymiz. Keyin ildiz klipning birinchi kadrda quyidagi kodni yozamiz:

```
// Rasmimizni buzmasligi uchun original klipni yashiramiz
sourceClip._visible = false;
// Asl klipning 10 ta nusxasini yaratish keyingisini pastga va o'ngga siljiting
for (var i=1; i<=10; i++){
    duplicateMovieClip("sourceClip", "duplicated" + i, i);
    setProperty("duplicated" + i, _x, 50 + 20*i);
    setProperty("duplicated" + i, _y, 50 + 20*i);
}
```

Rolikni boshlaymiz va ko'z uchun yoqimli quyidagi rasmni kuzatamiz:



8.2.1-rasm. Actions Kliplarni ko'paytirish.

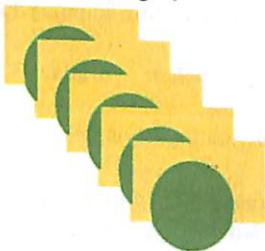
Bizda haqiqatan ham bir-birining ustiga o'rnatilgan 10 ta to'rtburchaklar o'ngga pastga siljishi borligini ko'ramiz.

Klipni allaqachon mavjud darajaga joylashtirishni aytganimizda nima sodir bo'lishini tekshirib ko'raylik. Keling, yana bir klip chizamiz (uning belgisi endi doirani o'z ichiga oladi va biz nusxani **anotherSourceClip** deb nomlaymiz). Va biz ushbu klipni besh nusxada ko'paytiramiz va ularni allaqachon egallab olingan darajalarga joylashtiramiz.

Mana bu vazifani bajaradigan kod:

```
// Rasmimizni buzmasligi uchun original klipni yashiring
    sourceClip._visible = false;
// Asl klipning har biri 10 ta nusxasini yarating
// keyingi siljish pastga va to'g'ri
for (var i=1; i<=10; i++)
{ duplicateMovieClip("sourceClip", "duplicated" + i, i);
  setProperty("duplicated" + i, _x, 50 + 20*i);
  setProperty("duplicated" + i, _y, 50 + 20*i);
}
// Ikkinchi manba klipni ham yashiring
    anotherSourceClip._visible = false;
// Ikkinchi klipning besh nusxasini yarating va ishonch hosil qiling
// ular birinchi klip nusxalari o'rnini egallaydi
// bir xil darajadagi raqamlar bilan
for ( var i =2; i <=10; i +=2){
    duplicateMovieClip("anotherSourceClip", "anotherDup" + i, i);
  setProperty("anotherDup" + i, _x, 50 + 20*i);
  setProperty("anotherDup" + i, _y, 50 + 20*i);
}
```

Natijada, rasm o'zgaradi va shunday ko'rinadi (bu yerdagi asl to'rtburchak kattaroq kontrast uchun ataylab ochroq rangga bo'yalgan):

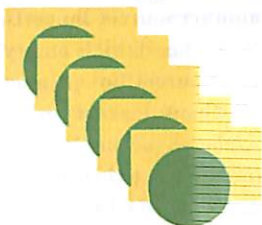


8.2.2-rasm. Actions Kliplarni ko'paytirish.

Ya'ni, teng darajadagi barcha to'rtburchaklar oddiygina doiralalar bilan almashtirildi! Ma'lum bo'lishicha, bir xil darajadagi ikkita klip bo'lishi mumkin emas: ishg'ol qilingan darajaga yangi klipni joylashtirgandan so'ng, muqarrar ravishda eskisini u yerdan tushiramiz. Aytgancha, ushbu xususiyatdan foydalanib, berilgan darajada bo'sh klip yaratadigan **createEmptyMovieClip** usuli haqiqatan ham ishlayotganligini tekshirishimiz mumkin. Albatta, hali to'liq tekshiruvni amalga oshira olmaymiz - bo'sh klip ko'rinmaydi. Unda biror narsani dasturiy ravishda chizishingiz kerak yoki dasturli ravishda unda belgining nusxalarini yaratishingiz kerak. Ammo boshqa tomondan, allaqachon bilamizki, yangi klipning paydo bo'lishi (hatto bo'sh bo'lsa ham) qaysidir darajada bu darajadagi oldingi tarkibni tozalaydi. Shunday qilib, oldingi kodga quyidagi qatorni qo'shing:

```
createEmptyMovieClip("empty1", 3);
```

(ya'ni 3-darajada empty1 nomli bo'sh klip yaratamiz), biz quyidagi rasmni olamiz:



8.2.3-rasm. Actions Kliplarni ko'paytirish createEmptyMovieClip usuli.

Bitta to'rtburchak g'oyib bo'ldi! Va biz, albatta, nima uchun bu sodir bo'lganini tushunamiz: uning o'rnini o'z darajasini olgan bo'sh klip egalladi.

**MovieClip** sinfining bir nechta qo'shimcha foydali usullarini ko'rib chiqaylik. Birinchisi - **swapDepths** usuli. Yoki ba'zi klipga yo'l unga o'tadi yoki bu klip joylashgan daraja. Bunday holda, ko'rsatilgan klip, albatta, biz chaqirayotgan klip bilan bir xil ota-onaga ega bo'lishi kerak. Ushbu usulning ta'siri ikkita klipni z-tartibida almashishdir.

klip qay darajada yotganini aniqlash mumkinmi? Ma'lum bo'lishicha, **getDepth** usuli aynan shu maqsadda. Unga hech qanday argument berish shart emas, shunchaki kerakli klip uchun unga qo'ng'iroq qiling va u daraja raqamini qaytaradi.

Keling, ushbu usul qanday ishlashini ko'rib chiqaylik va buning uchun oxirgi misolga quyidagi to'rt qatorni qo'shaylik:

```
trace("duplicated7.getDepth() = " + duplicated7.getDepth());
trace("anotherDup4.getDepth() = " + anotherDup4.getDepth());
trace("sourceClip.getDepth() = " + sourceClip.getDepth());
trace("anotherSourceClip.getDepth() = " + anotherSourceClip.getDepth());
```

Misolni qayta ishga tushirsak, konsolda quyidagi matn ko'rsatilganligini ko'ramiz:

```
duplicated7.getDepth() = 7 anotherDup4.getDepth() = 4 sourceClip.getDepth()
= -16383 anotherSourceClip.getDepth() = -16381
```

Shunday qilib, biz, birinchi navbatda, **getDepth** klipni joylashtirgan darajani qaytarishiga ishonch hosil qildik. Ikkinchidan, tahrirlash jarayonida "qo'lda" yaratilgan kliplarning ham darajalari borligini ko'rdik, ularning soni z-tartibiga juda mos keladi - ya'ni ular salbiy (chunki qo'lda yaratilgan kliplar dasturiy ravishda yaratilgan kliplar ostida).

Keling, **swapDepths** usuli haqidagi bilimlarimizdan foydalanib almashish uchun (z-tartibida) dastlab dasturiy tarzda yaratilgan ikkita klip, so'ngra muharrirda yaratilgan klip bilan dasturiy tarzda yaratilgan klip. Bizning misolimizga quyidagi kodni qo'shing:

```

anotherDup10.swapDepths(anotherSourceClip.getDepth());
trace("anotherDup10.getDepth() = " + anotherDup10.getDepth());
trace("anotherSourceClip.getDepth() = " + anotherSourceClip.getDepth());

```

Shakllar zanjirining pastki o'ng uchida joylashgan doira fonga o'tganini ko'ramiz va `getDepth` funksiyasi joylardagi barcha o'zgarishlar muvaffaqiyatli bo'lganligini bildiradi. Ikkinchisini konsolda ikkita yangi qator paydo bo'lganidan ko'rish mumkin:

```

anotherDup10.getDepth() = -16381
anotherSourceClip.getDepth() = 10

```

`CreateEmptyMovieClip` funksiyasi salbiy darajalar bilan ham ishlaydi: agar oxirgi uchta kod qatoridan oldin chaqirsak `createEmptyMovieClip("empty2", anotherSourceClip.getDepth());`

keyin oldingi ikkita satr o'rniga konsolda ko'ramizki

```
anotherDup10.getDepth() = 10 anotherSourceClip.getDepth() =
```

bu klip `anotherSourceClip` bo'sh klip bilan almashtirildi `empty2` va shuning uchun kliplar orasidagi darajalarni almashtirib `anotherSourceClip` va `anotherDup10` sodir bo'lmadi. Qizig'i shundaki, kliplarni o'chirish uchun maxsus ishlab chiqilgan xususiyat - `removeMovieClip` (argument sifatida o'chirish uchun klipga yo'l berilgan) faqat ijobiy darajadagi qiymatlarga ega kliplarni olib tashlashi mumkin. Nihoyat, keling, `duplicateMovieClip` funksiyasi ba'zi kliplarning dublikatini yaratishda uni salbiy darajada joylashtirishi mumkinligini tekshirib ko'ramiz.

Keling, misolimizga yana uchta qator qo'shamiz:

```
duplicateMovieClip("anotherSourceClip", "anotherDup_", -10);
setProperty("anotherDup_", _x, 150); setProperty("anotherDup_", _y, 30);

```

va ko'rsatgan joyda yana bir doira paydo bo'lganini ko'ramiz (qisman to'rtburchaklarning birinchisi bilan qoplangan, 1-darajani egallagan). Ya'ni, yangi yaratilgan klip haqiqatan ham salbiy darajada joylashgan.

### 3. Matn maydonlari. Matn maydonlarini yaratish.

Qanday dastur matni chiqarmasdan bajara oladi?

Flash MX da matni ko'rsatish (va kiritish) imkonini beruvchi yagona boshqaruv (boshqaruv elementi) mavjud bo'lib, u barcha standart Flash MX komponentlari tomonidan ham qo'llaniladi va matn maydonlarini chetlab o'tib matni ko'rsatishning hech qanday usuli (shu



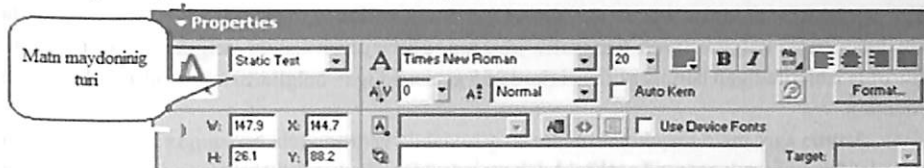
Matn maydonini yaratuvchi uskuna

jumladan dasturiy) yo'q. (ehtimol, uni rasmlardan tartibga solish bundan mustasno). To'g'ri, matn maydonlarini nafaqat "qo'lda", balki dasturiy jihatdan ham yaratishingiz mumkin.

Faqat matn maydonlari bilan ishlashning eng asoslarini ko'rib chiqamiz, ular uchta turga bo'linadi: **statik**, **dinamik** va **matn kiritish maydonlari**. Barcha uch turdagi matn maydonlari asboblari paneli ikkinchi ustunining uchinchi qatorida joylashgan bir xil nomdagi asbob yordamida yaratiladi.

### Statik matn maydonlari

Statik matn maydonlari tag lavhalarini o'z ichiga olishi uchun mo'ljallangan ( flesh-filmda o'zgarmasdir ). Maydonni statik qilish uchun Properties panelining chap tomonidagi ochiladigan ro'yxatdan Static Text ni tanlang.



8.3.1-rasm. Statik matn maydoni hosil qilish uchun Properties panelining ko'rinishi.

Statik maydonlar boshqa matn maydonlari uchun mavjud bo'lmagan ba'zi format sozlamalariga ega, masalan, pozitsiya (indeks, daraja), kerning. Statik maydonlardagi matn dastlab silliqilgan matn bo'sh ko'rsatiladi (agar "Use Device Fonts" katagiga belgi qo'ymasangiz). Shuni yodda tutingki, Flash MX tomonidan amalga oshiriladigan silliqilash matn uchun emas, balki grafiklar uchun mo'ljallangan, shuning uchun Flash silliqilash ishlatilmagan matn yaxshiroq ko'rinadi. Shuni ham yodda tutingki, matnni tahrirlashda har doim Flash MX muhiti tomonidan tekislangan shaklda ko'rsatiladi va film boshlangandan keyingina uni tizim shriftlari bilan ko'rsatish mumkin (agar siz ushbu rejimni o'rnatgan bo'lsangiz).

Yana bir narsani eslatib o'tish kerak (bu faqat statik emas, balki har qanday turdagi matn maydonlariga tegishli). Dastlab Flash tizimi uchun asosiy platforma Apple Macintosh platformasi edi. Buning natijasida ma'lum o'lchamdagi shriftdagi belgilarning (gliflar) haqiqiy hajmini standart ekran o'lchamlari sifatida 72 dpi (Mac OS X paydo bo'lishidan oldin 72 dpi Macintosh operatsion tizimlarida standart ruxsat edi) yordamida hisoblash mumkin edi. Xuddi shu rezolyutsiya standart tipografik ruxsat deb hisoblanadi va undan tipografik nuqtaning o'lchami hisoblanadi. Windowsda yana bir standart ruxsat qabul qilinadi: 96 dpi. Shuning uchun, Windows-da tipografik nuqta odatda o'rtacha 4/3 pikselni egallaydi (shuning uchun 72 nuqta hali ham bir dyuymni tashkil qiladi, bu Windows mashinasida 96 piksel). Masalan, Microsoft Word. Ammo 72 dpi standart Macintosh ruxsatiidan foydalanadigan Flashda bitta tipografik nuqta bir pikselga to'g'ri keladi (bir va uchdan bir emas). Shu sababli, Flash bilan yaratilgan shriftlar ham "kichrayadi" va Microsoft Word dasturidagi 12-shrift ( $12 * 4/3 = 16$  piksel o'lchamga ega) Flashdagi bir xil shriftning 16-shriftiga o'xshaydi.

### Dinamik matn maydonlari

**Dinamik maydonlar** videoni ijro etish paytida o'zgarib turadigan matnni ko'rsatish uchun mo'ljallangan. Ushbu rejimni yoqish uchun **Properties** panelining chap tomonidagi ochiladigan ro'yxatdan **Dynamic Text ni** (**Static Text** o'rniga) tanlang. Matn maydonning qiymatini o'zgartirish uchun ikkita yondashuvdan foydalanish mumkin:

- matn maydonini o'zgaruvchiga bog'lang va o'zgaruvchining qiymatini dasturiy ravishda o'zgartiring (o'zgaruvchi Properties panelining **Var** maydonida o'rnatiladi). Masalan:  $v = \text{"new text"}$ , agar  $v$  **Var** maydonida bo'lsa. O'zgaruvchi matn maydoni bilan bir xil klipda yaratilishi kerak.

- matn maydonning text xususiyati qiymatini dasturiy ravishda o'zgartirish. Misol uchun, `myTextField.text = "some value"`. Bu yerda `myTextField` matn maydonining nomi.

Matn dinamik matn maydonlaridagi tizim shriftida yoki o'rnatilgan shriftida ko'rsatiladi. Shuni esda tutingki, statik matndan farqli o'laroq, o'rnatilgan shriftning barcha belgilari kerakli vaqtda mavjud bo'lishini ta'minlash uchun alohida e'tibor berish kerak (matn dinamik va satrda ilgari bo'lmagan belgilar paydo bo'lishi mumkin). Shriftni joylashtirishni sozlash uchun **Properties** panelidagi **Character** tugmasini bosing. Dialog oynasi paydo bo'ladi, unda shriftidan qaysi belgilar (yoki tipografik jihatdan, gliflar) kiritilishini belgilashingiz mumkin. Shriftida ishlatiladigan barcha gliflarni (kod 255 gacha bo'lgan belgilarni anglatadi) joylashtirish mumkin.

Ammo agar matn maydonlarini dasturiy ravishda yaratmoqchi bo'lsangiz nima bo'ladi (bu **MovieClip** sinf usuli **createTextField** deb ataladi)? Barcha kerakli shriftlarni joylashtirishning biron bir usuli bormi? Buni amalga oshirish uchun hech bo'lmaganda kutubxonaga ("**Export in first frame**") katagiga belgi qo'ygan holda) belgini olishingiz kerak, unda dinamik matn maydoni bo'ladi, bu esa o'z navbatida barcha kerakli belgilarni (ular kerakli shriftning gliflari) joylashtirish uchun belgilanadi. Shuningdek, butun shriftni joylashtirishingiz mumkin, ammo bu tayyor videoning hajmini sezilarli darajada oshiradi. Agar "**Export in first frame**" opsiyasi yoqilgan bo'lsa (uni yoqish uchun, aytmoqchi, "**Export for ActionScript**") opsiyasini ham tekshirishingiz kerak), sahnada ushbu belgining nusxasini tortib ololmaysiz, u yerga o'rnatilgan barcha gliflar avtomatik ravishda filmingizga kiritiladi. Bundan tashqari, agar siz boshlang'ich matn sifatida maydonga turli xil shriftlardan bir nechta gliflarni kiritsangiz va shriftni o'rnatish dialog oynasida "**All Characters**" bandini tanlasangiz, kamida bitta glif kiritilgan barcha shriftlar o'rnatilgan bo'ladi. Albatta, bu hiyla nafaqat dastur asosida yaratilgan matn maydonlari, balki oddiy maydonlar uchun ham foydalidir. Shuning uchun, agar siz bitta matn maydoniga shriftni (yoki uning qismini) joylashtirsangiz, u ushbu videoning barcha sohalarida ishlaydi. Nihoyat, shuni ta'kidlaymizki, ma'lum bir shriftidagi gliflar to'plami bilan statik maydon yaratish, joylashtirish nuqtai nazaridan, aynan shu maxsus gliflarni (tegishli dialog oynasida) joylashtirish uchun belgilashga tengdir. Albatta, o'rnatish sodir bo'lishi uchun ushbu statik maydonda o'rnatilgan shriftlardan foydalanish kerak.

Ekzotik shriftlar bilan ishlashda yana bir muammo paydo bo'ladi: siz ko'p sonli matn maydonlari uchun shriftlarni tanlashingiz kerak deylik. Va agar uslubni o'zgartirishingiz kerak bo'lsa, ularni bir vaqtning o'zida tezda o'zgartiring. Bularning barchasi haqiqatan ham amalga oshirilishi mumkin. Belgilar kutubxonasi panelini oching (**Ctrl + L**), uning yuqori o'ng burchagida, kulrang sarlavha satrida menyuni ochish tugmasi mavjud. **New Font** bandini tanlang va ochilgan menyuda kerakli shriftni belgilang (va shu bilan birga uni ishlatadigan taxallusni kiriting, masalan **MyFont**). Agar hozir matn maydonini yaratsangiz, u holda mavjud shriftlarning ochiladigan ro'yxatida **MyFont** yozuvini ham ko'rasiz (yulduzcha bilan belgilangan - bu shrift kutubxonada aniqlangan taxallus ekanligini eslatishi kerak). Endi belgining xususiyatlarini o'zgartirish unga mos keladigan **MyFont** shrifti bilan bir vaqtning o'zida **MyFont** shrifti o'rnatilgan barcha maydonlarda shriftni o'zgartirasiz (ehtimol, matn maydonlaridagi matn qayta chizilgan bo'lishi uchun kiritish fokusini ish maydoniga o'tkazishingiz kerak bo'ladi). Ammo bularning barchasi shundan so'ng \*.fla-faylingizni saqlashingiz va uni fonti bilan birga boshqa mashinaga o'tkazishingiz mumkin degani emas. Biz ta'riflagan mexanizm \*.fla faylida shriftni saqlamaydi, u shunchaki taxallus yaratadi va boshqa hech narsa yo'q.

### **Matn kiritish maydonlari**

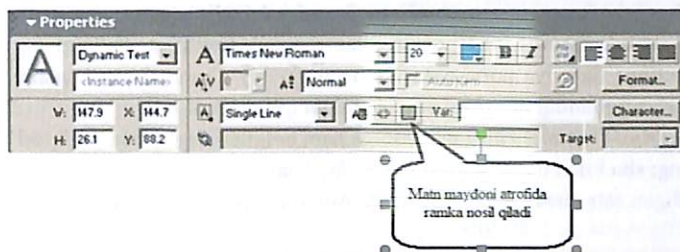
Bu maydonlar foydalanuvchi tomonidan interaktiv matn kiritish uchun mo'ljallangan. Keyin maydon qiymatiga dasturiy ravishda kirish mumkin ( dinamik maydonlarga o'xshash,

buni **Properties** panelidagi "**Var**" maydonidan foydalanish yoki matn xususiyatiga kirish orqali amalga oshirish mumkin). Bunday maydonni yaratish uchun **Properties** panelining chap tomonidagi ochiladigan ro'yxatdan **Input Text** ni tanlashingiz kerak.

**Input Text** turidagi matn maydonlari uchun o'rnatilgan shriftlar bilan bog'liq vaziyat **Dynamic Text** tipidagi maydonlar uchun sodir bo'ladigan holatga butunlay o'xshaydi. Agar hozirgacha yuklangan kamida bitta maydonda o'rnatilgan belgilar orasida biron bir belgi yetishmayotgan bo'lsa, foydalanuvchi bu belgilarni yoza olmaydi.

### Formatni o'zgartirish

Matn maydonlarining uchta turi elementar formatlashni qo'llab-quvvatlaydi (rangni, shriftini, uning o'lchamini va uslubini, tekislashni o'zgartirishingiz mumkin). Bundan tashqari, **Dynamic** va **Input** turlarining matn maydonlari uchun chegarani (**Border**) yoqishingiz yoki o'chirishingiz mumkin:



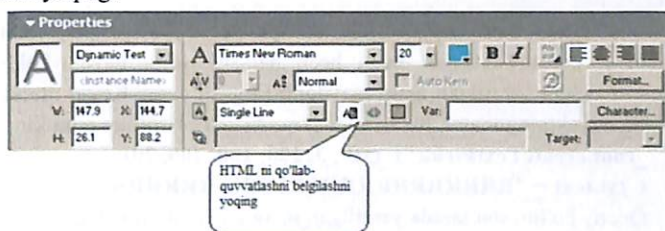
8.3.2-rasm. Matn maydonlarini formatni o'zgartirish paneli..

Ko'pgina parametrlarni dasturiy jihatdan o'zgartirish mumkin (buning uchun **TextField** va **TextFormat** obyektlarining usullari javobgardir).

Muhimi shundaki, statik maydonda har bir alohida belgining formatini o'zgartirishingiz mumkin. To'g'ri, bu dinamik maydonda html belgilaridan foydalangan holda amalga oshirilishi mumkin (keyingi kichik bandga qarang), lekin statik maydon uchun biroz ko'proq imkoniyatlar mavjud (masalan, yuqori va pastki belgilardan foydalanishingiz mumkin).

### HTML maydonlari

Dinamik matn maydonlari HTML renderlashni qo'llab-quvvatlashi mumkin. Buning uchun tegishli variantni yoqing:



8.3.3-rasm. Matn maydonning htmlText xususiyatiga html belgisini yozish.

Shundan so'ng, matn maydonining htmlText xususiyatiga html belgisini yozishingiz mumkin. Masalan:

```
myTextField.htmlText = "< b > Bold </ b > Not bold".
```



Yana bir bor ta'kidlaymiz: tahrirlash jarayonida html formatidagi satrni to'g'ridan-to'g'ri matn maydoniga yozish befoyda. Formatlash faqat satr htmlText xususiyati orqali o'tkazilsa qabul qilinadi.

Albatta, shuni ta'kidlash kerakki, matn maydonlari bilan ishlashda qo'llanilishi mumkin bo'lgan html formatlash juda cheklangan. Shu bilan birga, matnning qalinligi, kursiv, tagiga chizish, ranglar, shrift turi va paragraf sozlamalari (ko'p qatorli matnlar uchun) belgilanishi mumkin. Bundan tashqari, giperhavolalarga ruxsat beriladi.

Qizig'i shundaki, shunga o'xshash belgilar yordamida giperhavoladan JavaScript kodini chaqirish `myTextField.htmlText=<u><a href='javascript:alert(1)'>link</a></u>`

(bu holda havolani bosgandan so'ng dialog oynasi paydo bo'ladi), uni bosish orqali ActionScript funksiyasini ham chaqirishingiz mumkin. Faraz qilaylik, `reactionFunc(arg)` funksiyasini aniqladik. Keyin quyidagi kod bilan belgilangan havolani bosing `myTextField.htmlText =`

```
"<u><a href='asfunction:reactionFunc,fromLink1'>link1</a></u>" reactionFunc  
argument sifatida "fromLink1" qatori bilan chaqiriladi. Ehtiyot bo'ling:
```

funksiya `asfunction`: funksiya nomi o'rtasida hech qanday ajratuvchi bo'lmasligi kerak (aks holda ular funksiya nomining bir qismi sifatida talqin qilinadi).

Nihoyat, `htmlText` ni kiritish maydoniga ham belgilashingiz mumkinligini

unutmang; shu bilan birga, klaviaturadan foydalanuvchi keyingi kiritishi foydalanuvchi belgilar qo'shadigan satr qismining belgilangan formatiga qarab formatlanadi.

#### 4. Matn maydonlarini dasturli yaratish

Biz hozirgina `createEmptyMovieClip` usulini ko'rdik, bu bizga istalgan darajada bo'sh klip yaratish imkonini beradi. Ma'lum bo'lishicha, siz klipni emas, balki berilgan darajadagi matn maydonini ham joylashtirishingiz mumkin (va keyin uning usullarini chaqirib, matnni joylashtiring, ushbu matnni kerakli tarzda formatlang va hokazo). Maydonni yaratuvchi usul ham `MovieClip` sinfiga joylashtirilgan va `createTextField` deb ataladi. Biroq, `createEmptyMovieClip` usulidan ko'ra unga ko'proq parametrlar uzatiladi. Birinchi ikkita parametrga qo'shimcha ravishda - nusxa nomi va daraja raqami (bu ikki usul uchun bir xil) - koordinatalarni ham topshirishingiz kerak. `x` va `y`, shuningdek matn maydonining kengligi va balandligi. Shu bilan birga, koordinatalar `x` va `y` matn maydonining yuqori chap chetiga mos keladi.

`CreateEmptyMovieClip` usulidan farqli o'laroq (bu hozirgina yaratilgan klipga havolani qaytaradi), `createTextField` usuli hech narsani qaytarmaydi. Lekin biz misol nomini berganimiz uchun yaratilgan matn maydonini topish oson. Keyin mulkka biror narsa yozing yaratilgan obyektning matni, biz matnni ekranda ko'rsatishimiz mumkin. Masalan:

```
_root.createTextField("t_txt", 3, 100, 100, 400, 20);
```

```
t_txt.text = "RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR";
```

Odatiy bo'lib, shu tarzda yaratilgan matn maydoni quyidagi xususiyatlarga ega bo'ladi :

```
type = "dynamic",
```

```
border = false,
```

```
background =
```

```
false, password =
```

```
false, multiline =
```

```
false, html = false,
```

```
embedFonts =  
false, variable =  
null, maxChars =  
null
```

Agar sizga matn maydoni atrofida chegara qilish kerak bo'lsa, quyidagicha kerak `t_txt.border = true;`

Agar matnni formatlashni xohlasangiz, yaratilgan obyektga `setTextFormat` usulini chaqirishingiz kerak bo'ladi va unga `TextFormat` sinfining maxsus obyektini o'tkazing (Unda sizga kerak bo'lgan hamma narsani yozasiz). Odatiy bo'lib, matn quyidagi formatda yaratiladi:

```
font="Time New Roman"  
size = 12 textColor  
= 0x000000  
bold = false italic = false  
underline = false url = ""  
target = "" align = "left"  
leftMargin = 0  
rightMargin = 0 indent =  
0 leading = 0 bullet =  
false tabStops = []  
(massiv bo'sh).
```

Masalan, biz yaratgan chiziqning o'ng chetiga tekislashni o'rnatmoqchi bo'lsangiz (bu holda matn maydonining yuqori chap burchagi harakatlanmaydi, lekin undagi chiziq chiziqning o'ng chetiga bosiladi), shuningdek matnni qalin (qalın) qilib qo'ying, keyin quyidagilarni yozishingiz kerak bo'ladi:

```
t_fmt = new  
TextFormat(); t_fmt.bold  
= true; t_fmt.align =  
"right";  
t_txt.setTextFormat(t_fmt  
);
```

Bunday holda, matn ramkada olingani yaxshiroqdir - shunda tekislash o'zgariganini aniqroq ko'rasiz.

## 5. Roliklarda shartli if, if...else operatorlaridan foydalanish

Odatda skript ketma-ket, boshidan oxirigacha bajariladi. Shartli mantiq yordamida ma'lum bir shart bajarilgan taqdirdagina muayyan harakatlarni bajarishga ruxsat berish orqali skriptlarni boshqarishingiz mumkin. Shu tariqa filmlaringiz "o'z hayotini o'tkazishi", vaziyatga qarab qaror qabul qilishi va reaksiyaga kirishishi mumkin bo'ladi. Shartli operatorlar shartli mantiqni skriptlarda amalga oshirish uchun ishlatiladi.

ActionScript - dagi barcha shartli mantiqning o'zagi if operatori tomonidan amalga oshirilgan (turli xil o'zgarishlarda) "agar - keyin" konstruksiyasidir. (if, if-else, if-else if-else). Biroq, bu usul yagona emas; ba'zi hollarda bunday konstruksiyalar boshqa shartli ko'rsatmalarni muvaffaqiyatli almashtira oladi va shu bilan birga dastur matnining o'zi ham sodda va tushunarli bo'lishi mumkin.

Switch operatori if/else konstruktsiyasini o'rinbosari bo'lib xizmat qilishi mumkin. Muayyan harakatlar mos kelishi kerak bo'lgan aniq qiymatlar to'plami mavjud bo'lsa, bu juda qulay.

Mantiqiy qiymatlar uchun (axir ulardan faqat ikkitasi bor - true va false), shartli operatorning keyingi versiyasi ancha mos keladi.

Ternar (yoki uchlik) operator - ? : - eng oddiy if / else konstruktsiyasini bitta qatorga yozish imkonini beradi. Masalan:

```
myMood = (money > 1000000) ? "Happy" : "Sad";
```

Ternar operator ko'pincha ikkita holatga (qiymatga) ega bo'lgan o'zgaruvchilarni "almashtirish" uchun ishlatiladi. Masalan:

```
playSound = (playSound)?false:true;
```

Har safar ushbu skript qatori bajarilganda, playSound o'zgaruvchisining qiymati true dan false ga yoki false dan true ga o'zgaradi .

### Shartlarni tanlash

Loyihada dastur qanday ishlashini va foydalanuvchining o'zaro munosabatini nazorat qilish uchun juda ko'p turli xil sharoitlardan foydalanish mumkin. Shartlar boshqacha ko'rinishi mumkin, ammo asosiy qoida har doim bir xil: shartning natijasi true yoki false bo'lishi mumkin. Shartli operatorlar odatda qo'llaniladigan holatlarga misollar:

- Bir obyekt boshqa obyekt bilan aloqaga chiqadi
- Nimadir yoqiladi yoki o'chadi
- Film belgisining joylashuvi, o'lchami yoki boshqa xususiyati ma'lum bir qiymatga yetadi

(yoki bu qiymatdan kattaroq yoki kichikroq bo'ladi)

- Foydalanuvchi sichqoncha yoki klaviatura yordamida muayyan harakatlarni amalga oshirdi
- Ikki raqamli yoki matn qiymatlarini solishtirish □ Yuqoridagi holatlarning turli kombinatsiyalari

### Bir nechta shartlarga reaksiyani dasturlash

Ushbu misolda biz raketani uchiramiz. Muvaffaqiyatli ishga tushirish uchun ma'lum shartlar talab qilinadi. Eng muhim omillardan biri bu raketaning tezlashishiga, shuningdek, uning boshqarilishi mumkinligiga ta'sir qiluvchi ob-havo. Ushbu mashqda tasodifiy tanlangan ob-havo sharoitlarini yaratishni dasturlaymiz, bizning raketamiz u yoki bu tarzda reaksiyaga kirishadi. Keling, loyiha uchun skript yozaylik.

```
onClipEvent(load) { randomWeather = random(3); if (randomWeather == 0) {
conditions = "Sunny"; _root.rocket.noThrust = 3;
    _root.rocket.thrust = 6;
} else if (randomWeather ==
1) { conditions = "Rainy";
_root.rocket.noThrust = 2;
    _root.rocket.thrust = 4;
} else {
conditions = "Night";
_root.rocket.noThrust = 1;
_root.rocket.thrust = 2;
```

}  
}

Ushbu skript berilgan film belgisining nusxasi yuklanganda bajariladi. randomWeather o'zgaruvchisiga qiymat - tasodifiy tanlangan raqamlar orqali belgilanadi (uchta mumkin bo'lgan raqamdan: 0, 1 yoki 2). Shundan so'ng, if operatori randomWeather qiymatini tahlil qilishni boshlaydi, boshqa uchta o'zgaruvchiga kerakli qiymatlarni belgilash uchun. Agar randomWeather 0 ga teng bo'lsa, joriy montaj stolida conditions o'zgaruvchisi yaratiladi (ushbu film belgisi) va unga satr qiymati tayinlanadi "Quyoshli". Bundan tashqari, raketa filmi montaj stolida ikkita o'zgaruvchi yaratiladi: biri noThrust deb ataladi, ikkinchisi thrust; bu o'zgaruvchilar mos ravishda 3 va 6 qiymatlarini oladi. Agar qiymat bo'lsa randomWeather 0 ga teng bo'lmasa, shartli operatorning bu qismi e'tiborga olinmaydi va boshqaruv qiymatni tekshiradigan else if bo'limiga o'tkaziladi. randomWeather birga teng bo'lishi. Agar shart bajarilsa, u holda bir xil uchta o'zgaruvchi yaratiladi, lekin bu safar ular mos ravishda "Yomg'ir", 2 va 4 ga o'rnatiladi. Agar randomWeather qiymati 1 ga teng bo'lmasa, bu bo'lim ham e'tiborga olinmaydi va else bo'limidagi harakatlar bajariladi. Esingizda bo'lsa, boshqa bo'limda shartli operator, agar operatorning oldingi bo'limlari shartlaridan hech biri bajarilmasa, bajarilishi kerak bo'lgan harakatlar mavjud. Bizning else bo'limida bir xil uchta o'zgaruvchi yaratiladi, faqat bu holda ular "Night", 1 va 2 qiymatlarini oladi.

Voqea joyida turli ob-havo sharoitlarini grafik ko'rsatish uchun qisqa vaqt ichida conditions o'zgaruvchisidan foydalanamiz. Qolgan ikkita o'zgaruvchi ob-havoga qarab raketaning tezligini o'rnatish uchun xizmat qiladi (raketa misoli).

```
gotoAndStop(conditions);  
_root.weatherText.text = "Weather: " + conditions;
```

Agar bu ikkita harakatlar shartli operator if tugagandan so'ng darhol bajariladigan bo'lsa, ikkalasi ham conditions o'zgaruvchining qiymatidan foydalanadi.

Birinchi harakat film belgisining ushbu nusxasini nomi o'zgaruvchan conditions o'zgaruvchisi joylashgan yorliqlik kadrda o'tkazadi. Agar, masalan, conditions "Rainy" qiymatini olsa, klip nusxasi yomg'irli ob-havo tasvirini o'z ichiga olgan Rainy yorlig'iga ko'chiriladi. Ikkinchi amal weatherText matn maydonida ko'rsatilgan yorliqni yaratadi, bu sahnaning pastki o'ng burchagida joylashgan. Conditions yana "Rainy" bo'lsin, keyin ushbu matn maydonida "Weather: Rainy" ko'rsatiladi.

### **Chegara shartlarini o'rnatish**

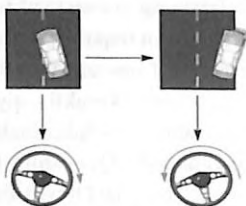
Biz hayotda har doim turli xil chegaraviy sharoitlarga duch kelamiz. Chegaralar har xil bo'lishi mumkin - jismoniy, moliyaviy, hatto bizning ongimizda belgilanishi mumkin; Oxir-oqibat, har bir chegara shunchaki biror narsa chegarasining belgisidir. Keling, amaldagi chegara shartlariga misol keltiraylik. Tasavvur qiling-a, mashina haydayapsiz. Yo'lda belgilanish chiziqlari bizning mashinamiz to'xtashi kerak bo'lgan chegaralarni ifodalaydi. Agar chap chiziqqa (chap chegara) juda yaqin bo'lsak, avtomobilni chiziq ichida ushlab turish uchun rulni o'ngga aylantiramiz; agar o'ng chiziqqa yaqinlashsak, biz rulni teskari yo'nalishda aylantiramiz - harakatdagi shartli mantiq! Agar harakatlarimiz algoritmini ActionScript -da yozsak, shunga o'xshash narsani olamiz:

```
if (car._x > leftOfRoad) {  
    turnWheel ("right"); } else
```

```

if (car < rightOfRoad) {
turnWheel ("left");
}

```



8.5.1-rasm. Flash da chegara shartlarini o'rnatish.

Flash da chegara shartlarini o'rnatish uchun ba'zi cheklovlarni o'rnatishingiz kerak, so'ngra ularni belgilangan chegaralardan oshib ketishiga olib kelishi mumkin bo'lgan hodisa sodir bo'lganda cheklangan elementni tekshiradigan if operatorida ishlatishingiz kerak. Keyingi mashqni bajarayotganingizda, bu jarayon aniqroq bo'ladi.

Chegara shartlari Flashda quyidagi holatlarda qo'llaniladi:

- Film belgisi nusxasi xususiyatlari (x, y, alfa, xscale, yscale va boshqalar) ma'lum qiymatlardan oshib ketishining oldini olish uchun
- Biror narsa belgilangan chegaralarga tushib qolgan yoki aksincha, chegaradan chiqib ketgan vaziyatda harakatni chaqirish (xususiyat yoki usulning qiymatini o'zgartirish)
- Har qanday ma'lumotlarning qiymatlari ma'lum chegaralardan tashqariga chiqmasligini ta'minlash uchun.

### Yoqish/o'chirish

Ba'zi harakatlar faqat biror narsa yoqilgan bo'lsa bajarilishi mumkin bo'lgan holatlar mavjud. Avtomobil bilan bir xil misol: dvigatelni ishga tushirish uchun kontaktni yoqmasdan, siz uzoqqa bora olmaysiz. Filmlaringizda shunga o'xshash narsa sodir bo'lishi mumkin: siz skriptning ayrim qismlariga "o'tkazgichni birlashtirishingiz" mumkin, bu ularni bajarishga ruxsat beradi yoki taqiqlaydi.

Buni shunday tasavvur qiling: skriptning o'zi muayyan vazifani bajaradigan mexanizm, "mashina". Filmda sodir bo'layotgan voqealar ssenariylarning "motori"dir. Shartli mantiq voqea sodir bo'lganda, barcha yoki ba'zi harakatlarni faqat ma'lum bir sharoitda bajarishga imkon beradigan "kalit" sifatida ishlatilishi mumkin. Misol tariqasida ushbu skriptni olaylik:

```

onClipEvent(enterFrame) {
soat._aylanish = soat._aylanish + 1;
}

```

Ushbu skript filmda sodir bo'layotgan boshqa hodisalardan qat'i nazar, har bir kadrda (enterFrame hodisasi) kirgandan so'ng, film soat belgisining bir nusxasini (masalan, soat strelkalarini ifodalaydi) 1 gradusga aylantiradi. Agar soatni ishga tushirish va to'xtatish imkoniyatiga ega bo'lishini istasak, skriptni quyidagicha qayta yozish kerak:

```

onClipEvent (enterFrame)
{ if (power == true) {

```

```

clock_rotation = clock_rotation + 1;
}
}

```

Endi aylanish faqat power o'zgaruvchisi true bo'lsa sodir bo'ladi. Power o'zgaruvchisiga false yoki true qiymat tayinlanishi mumkin, tugma bilan ulangan boshqa klip nusxasiga birlashtirilgan boshqa skript yordamida - bu birinchi skript uchun "yoqib/o'chiruvchi" bo'ladi.

Shunga o'xshash tushuncha ko'pincha ilovalarda qo'llaniladi, bu foydalanuvchiga sozlamalarni tanlash uchun dialog oynalarida dasturning ba'zi xususiyatlarini yoqish va o'chirish imkonini beradi. Flash loyihalarida, qoida tariqasida, asosiy e'tibor animatsiya va multimediyaga qaratiladi, ya'ni "yoqib/o'chiruvchi" (ularni triggerlar deb ham ataladi) ham foydali bo'ladi - ular ushbu skriptlarni (animatsiyalar va butun montaj stolini) "o'chirish" imkonini beradi.

### Foydalanuvchi harakatlariga munosabat

Foydalanuvchi va filmingiz o'rtasidagi o'zaro aloqa asosan ikkita qurilma: sichqoncha va klaviatura orqali amalga oshiriladi va ulardan foydalanishning ko'plab usullari bo'lishi mumkin. Foydalanuvchi sichqonchani siljitishi, tugmalarini bosib qo'yib yuborishi, klaviaturadan foydalanishi mumkin. Shartli mantiq yordamida siz filmni foydalanuvchining harakatlariga u yoki bu tarzda reaksiyaga kirishishi uchun dasturlashingiz mumkin.

Quyidagi misolda loyihada interaktiv o'zaro ta'sir o'tkazish imkoniyatini ko'rib chiqamiz: foydalanuvchi bo'sh joy tugmasini bosib, raketa tezroq ko'tarilishi uchun "kuchaytirgichlarni" yoqishi mumkin.

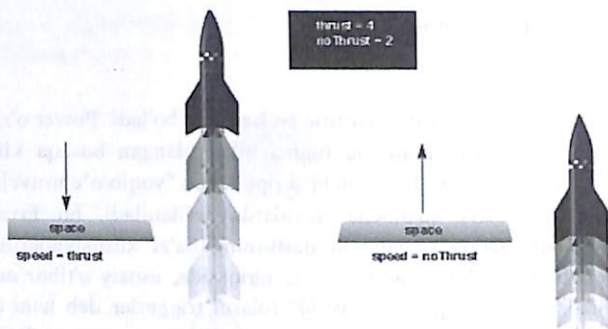
Keling, ikkita muhim nuqtadan boshlaylik. Birinchisi: ob-havo klipi misolini yuklash natijalari randomWeather o'zgaruvchisining qiymatiga bog'liq bo'lgan skript bajariladi. Skript nafaqat ma'lum ob-havo sharoitlariga mos keladigan tasvirni ekranda ko'rsatadi, balki thrust va noThrust o'zgaruvchilariga qiymatlarni ham belgilaydi. thrust o'zgaruvchisi barcha holatlarda noThrust qiymatdan ikki baravar katta qiymatga ega bo'ladi. Shunday qilib, agar noThrust 2 ga teng bo'lsa, unda thrust 4 ga o'rnatiladi. Ikkinchidan, raketa nusxasi yuklanganda speed o'zgaruvchisi (nusxaning montaj stolida joylashgan) noThrust qiymatiga o'rnatiladi; shuning uchun noThrust 2 ga o'rnatilgan bo'lsa, speed o'zgaruvchisi bir xil qiymatga ega bo'ladi. Speed qiymati raketaning ko'tarilish tezligini belgilaydi. Ushbu mashqda buni o'zgartiramiz: speed o'zgaruvchisi yoki noThrust qiymatga tayinlanadi yoki thrust qiymatga - foydalanuvchi "probe"ni bosib ushlab turishiga qarab. Thrust qiymatidan beri noThrust qiymatidan ikki baravar tezroq bo'lsa ham "probe"ni ushlab turish raketaning ikki baravar tez ko'tarilishiga olib

```

onClipEvent (keyDown) {
    if (launch && !
speed = thrust;
    _root.thrustB
}
}

```

Ushbu skript klaviaturadagi istalgan tugma bosilganda bajariladi va shartli operator ikkita shartni tekshiradi. Agar launch qiymatini true bo'lsa va "probe" bosilganda ikkita narsa sodir



Raketaning tezligi speed ning qiymatiga bogliq

8.5.2-rasm. Interaktiv o'zaro ta'sir o'tkazish imkoniyatini (foydalanuvchi sichqonchani siljitishi, tugmalarini bosib qo'yib yuborishi, klaviaturadan foydalanishi mumkin)

bo'ladi. Birinchisi speed o'zgaruvchisi thrust qiymatida bo'lsa - buning natijasida raketa ikki baravar tez ko'tariladi. Ikkinchi harakat thrustBoost film belgisi nusxasiga (Launch tugmasi ostidagi sahnada joylashgan) katta qizil harflar bilan yozilgan "Thrusters" tasvirini o'z ichiga olgan on yorliqli kadrda o'tishni buyuradi.

#### Obyektning to'qnashuvini aniqlash

Ko'pincha Flash ilovalarida, ayniqsa o'yinlarda, sahnada ikkita obyekt bir-biri bilan "uchrashgan" vaziyatlarni aniqlash talab qilinadi. Bu yerda shartli mantiq ham o'ynaydi.

ActionScript ikki obyekt (film qahramonlari nusxalari) o'rtasidagi to'qnashuvni aniqlash uchun MovieClip obyektining hitTest() usulini taqdim etadi. Ushbu usulni shartli mantiq bilan birgalikda qo'llash orqali bir film belgisi boshqasiga to'g'ri kelganda amalga oshiriladigan harakatlarni dasturlashingiz mumkin. Keyingi skript ni ko'raylik:

```
onClipEvent (enterFrame) {
    if (hitTest
        ("wallOfCotton")) {
        pain = 0;
    } else if (hitTest ("wallOfCardboard")) {
        pain = 5;
    } else if (hitTest ("wallOfBricks")) {
        pain = 10;
    }
}
```

## 6. Nazorat savollari



1. Rolik (klip)ning qanday xususiyatlarini bilasiz? Ularni sanab bering.
2. Rolikning shaffofligini qaysi xususiyat boshqaradi?

3. Videoning o'lchamini (ekranning balandligi va kengligi (klip)) qanday xususiyatlardan foydalanish mumkin?
4. Klipping joylashish darajasi haqida ma'lumot beruvchi xususiyat qanday nomlanadi?
5. Rolik xususiyati - `_name` ?
6. Rolik xususiyati - `_xscale` va `_yscale`?
7. Rolik xususiyati - `_x` va `_y` ?
8. Qiymatlarni dasturiy jihatdan o'rnatish yoki o'zgartirish uchun qanday usul ishlatiladi?
9. Kliplarni dasturli yaratishning nechta usuli bor? Ularning ismlari nima ?
10. Bo'sh kliplar yaratish usuli qanday nomlanadi?
11. FLASH MX da nechta turdagi matn maydonlari mavjud?
12. Statistik maydonlar uchun... Gapni davom ettiring.
13. Dinamik matn maydon uchun... Gapni davom ettiring
14. Kirish maydoni uchun... Gapni davom ettiring.
15. Matn kiritish maydonlarining ko'p variantlarini dasturiy ravishda o'zgartirishingiz mumkin. Buning uchun qanday obyekt usullari javobgar?
16. Maydonlarni dasturli yaratish uchun mo'ljallangan usul qanday nomlanadi?
17. Qanday qilib html formatidagi satrni to'g'ridan-to'g'ri matn qutisiga yozish mumkin?
18. Shartli operatorlar qachon ishlatiladi?
19. Qanday shartli operatorlarni bilasiz?
20. Asosan, qaysi hollarda ternar (yoki uchlik) operator ishlatiladi?

## 2.9. Dinamik maydonlarning asosiy xususiyatlari. Aylantirish funksiyasi

### Reja:

1. dynamic text Flash. Flash CS5da dinamik matn yaratish.
2. dinamik ActionScript matnini yaratish (dynamic text).
3. Actionscript 3.0 yordamida dinamik matn yaratish.
4. Flash CS5 va CS6 da kiritish matn maydonini yaratish.
5. TextFieldType.INPUT, TextFieldType.DYNAMIC TextField AS3 sinfining nusxalari.
6. textInput AS3, change, link, scroll - TextField hodisalari.
7. Matn hodisalari bilan ishlash.
8. Nazorat savollari

### 1. dynamic text Flash. Flash CS5 da dinamik matn yaratish

Flash CS5 3 turdagi matn maydonlarini qo'llab-quvvatlaydi:

- static (statik)
- dynamic (dinamik)
- input (kiritish maydoni)



Statik tekst asboblari panelida joylashgan **Text** vositasi yordamida yaratish mumkin. Ushbu turdagi matnlar videoni ijro etish vaqtida o'zgar olmaydi. Misollar: sarlavhalar, navigatsiya tugmalari, kontent va boshqalar.

Roлик/film/o'yinni ko'rsatish vaqtida o'zgarishi kerak bo'lgan matnlar dinamik matn maydonlari (**dynamic text Flash**) yordamida yaratiladi. Ular ham xuddi statik matnlar kabi Text vositasi yordamida kiritiladi. Biroq, Actionscript ushbu turdagi matnni yaratishga ham imkon beradi va buni qanday qilishning ikkita varianti mavjud:

- Ularga matn sahnada matn maydonini yaratishda dastlab joylashtirilishi va keyinroq o'zgartirilishi mumkin, Flash Player Actionscript ostida ishlayotganda
  - matn maydoni bo'sh qoldiriladi va faqat Actionscript 3 bilan kesilgan rolik/o'yin davomida to'ldiriladi.

Dinamik matndan foydalanishga misollar:

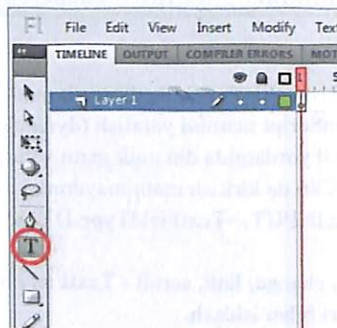
- o'yinchilar tomonidan to'plangan ochkolar soni
- hozirda filmni tomosha qilayotgan foydalanuvchining ismi
- ko'rsatilayotgan film yoki rasmning nomi

Flashda dinamik matn yaratishda, xuddi MovieClip misollari yoki tugmalari (Button) bilan bo'lgani kabi, unga nom berishingizga ishonch hosil qilishingiz kerak. Odatda, txt qisqartmasi matn maydoni nusxasi nomining boshida ishlatiladi, undan keyin u nima xizmat qilishini aniq tavsiflaydi. Masalan, kitob nomini ko'rsatadigan dinamik matn maydoni quyidagicha ko'rinishi mumkin: **txtBookTitle**

Dinamik matn maydonlari (dynamic) va kiritish maydonlari (input) **TextField** sinfining nusxalaridir.

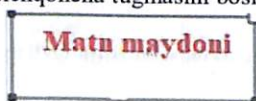
**{jumi[\*13]}**

### dynamic text Flash. Flash CS5 da dinamik matn yaratish Text vositasini bosib

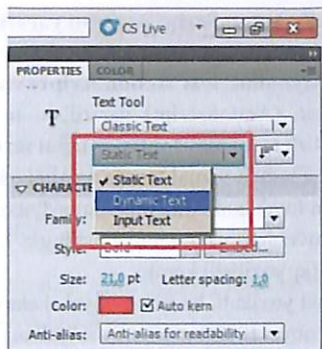


9.1.1-rasm. Flash CS5 da dinamik matn maydonini yaratish

Matn maydonini yaratish uchun sichqoncha tugmasini bosib va torting.

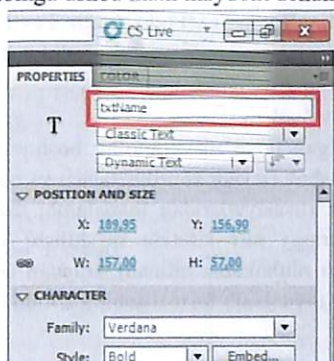


Xususiyatlar panelida Static Text ni bosib. Yordam menyusi ochiladi. Dynamic Text ni tanlang.



9.1.2-rasm. Xususiyatlar paneli

Nusxa nomini kiritish maydoniga ushbu matn maydoni uchun nom yozing.



9.1.3-rasm. Xususiyatlar panelida matn maydonini yaratish

Dasturning asosiy menyusida **Fayl** -> **Save As** ni bosing.

Ko'rsatilgan dialog oynasiga fayl nomini kiriting.

Save tugmasini bosing.

Fayl sahnada dinamik matn maydoni bilan saqlanadi.

### Qo'shimcha ma'lumot

Flash oxirgi marta yaratilgan matn maydonining turini eslab qoladi, shuning uchun yangi matn maydoni avvalgisini bilan bir xil bo'ladi.

Yaratilishdan oldin ham, keyin ham turini o'zgartirishingiz mumkin. Biroq, turini belgilashda ehtiyot bo'ling va sahnada allaqachon tanlangan matn maydonlari mavjudligini tekshiring.

Agar ulardan birini tasodifan tanlangan bo'lsangiz va keyin yangi yaratilgan maydonning turini o'zgartirgan bo'lsangiz, u tasodifiy tanlangan uchun ham o'zgaradi! Bunday xatolikka yo'l qo'ymaslik uchun quyidagilarni bajaring: har safar yangi matn maydonini yaratishdan oldin, sahnaning bo'sh joyini bosing va shu bilan har qanday obyektini, agar mavjud bo'lsa, tanlovni olib tashlang. Bundan tashqari, dasturning asosiy menyusidan **Edit** -> **Deselect All** buyruqni tanlashingiz mumkin. Agar menyu elementi kulrang ko'rinsa, ya'ni faol emas, bu tanlangan obyektlar yo'qligini bildiradi.

## 2. Dinamik ActionScript matnini yaratish (dynamic text)

Matn maydonlari (**dynamic text ActionScript**) vizual tarzda (Flashdagi Text vositasi yordamida) yoki kodlangan (Actionscript) yaratilishi mumkin. Xuddi shunday, siz ikkita usuldan birini ishlatib, kiritish matn maydonlarini(input text field) yaratishingiz mumkin. Statik matn (static text) esa faqat Flashda vizual tarzda amalga oshirilishi mumkin.

ActionScript kodidan foydalanib dinamik matn(dynamic text) yaratish uchun avvalo unga murojaat qiladigan o'zgaruvchini (var) e'lon qilishingiz kerak. Ushbu o'zgaruvchining nomi quyidagi qoidalarga muvofiq yozilishi kerak:

- txt qisqartmasi avval yoziladi, bu matn obyektini ekanligini ko'rsatadi
- keyin ushbu matn obyektini nima uchun mo'ljallanganligi aniq bo'lgan nomning o'zi □ keyin ma'lumotlar turini yozing: TextField
- yangi kalit so'z
- TextField() sinfining konstruktoriga chaqiriladi.

```
var txtMyFavBook:TextField = new TextField();
```

Sahnaga nusxa qo'shish uchun addChild() usulidan foydalaning va unga nusxa nomini argument sifatida bering:

```
addChild(txtMyFavBook);
```

Keyin yangi yaratilgan matn maydonining boshqa xususiyatlarini o'rnating: kenglik, balandlik, sahnadagi joylashuv (x va y koordinatalari) va boshqalar.

Text yoki htmlText xususiyatlaridan foydalanib, matn maydoniga matn qo'shishingiz mumkin. Oddiy matn(string) satr sifatida qo'shilishi mumkin. htmlText xususiyatidan foydalanish keyingi bobda muhokama qilinadi. Bundan tashqari, ushbu matnni TextFormat sinfining xususiyatlaridan foydalanib formatlashingiz mumkin.

## 3. Actionscript 3.0 yordamida dinamik matn yaratish

Actions panelini oching ( F9 tugmasini bosing).

O'zgaruvchi yarating, unga TextField ma'lumotlar turini tayinlang

```
var txtMyBook:TextField
```

Teng belgisini qo'ying, new kalit so'zini kiriting va TextField sinfining konstruktoriga chaqiring

```
new TextField();
```

AddChild() usuli bilan sahnaga matn maydoni nusxasini qo'shing

```
addChild(txtMyBook); text xususiyatidan foydalanib kitob nomini qo'shing
```

```
txtMyBook.text = "Actionscript 3.0 asoslari"; x va y koordinatalarini o'rnatish orqali sahnadagi matn maydonining o'rnini aniqlang txtMyBook.x = 150;
```

```
txtMyBook.y = 150;
```

Kenglik va balandlik xususiyatidan foydalanib, matn maydoniga width va height ni belgilang **txtMyBook.width = 150; txtMyBook.height = 50;**

Agar sizga matn maydoni uchun tashqi kontur kerak bo'lsa, border xususiyatidan foydalaning

```
txtMyBook.border = true;
```

```

1 import flash.text.TextField;
2
3 var txtMyBook:TextField = new TextField();
4
5 addChild(txtMyBook);
6
7 txtMyBook.text = "Actionscript 3.0 asoslari";
8 txtMyBook.x = 150;
9 txtMyBook.y = 150;
10 txtMyBook.width = 150;
11 txtMyBook.height = 50;
12 txtMyBook.border = true;
13

```

Ctrl + Enter ni bosing .

Flash Player ishga tushadi, ekranda matn oynasi paydo bo'ladi.

Actionscript 3.0 asoslari

### Qo'shimcha ma'lumot

Avtomatik rejimda matn maydonining kengligini o'rnatish uchun, ya'ni. kiritilgan matnning kengligi uchun TextFieldAutoSize sinfining autoSize xususiyatidan foydalaning, bunda uchta variant mavjud: LEFT, RIGHT va CENTER. Ulardan birini tanlash maydon ichidagi matnning tekislanishini aniqlaydi. Misol: `txtMyBook.autoSize = TextFieldAutoSize.LEFT;`

Agar sizning matningiz juda uzun bo'lsa va xatboshi kabi formatlanishi kerak bo'lsa, multiline va wordWrap xususiyatlaridan foydalaning

`txtMyBook.multiline = true;`

`txtMyBook.wordWrap = true;`

Yuqorida ko'rsatilgan uchta xususiyat matni shunday formatlaydi: matn maydonining kengligi width xususiyatidan olinadi va balandlik matn miqdoriga qarab avtomatik ravishda o'rnatiladi.

## 4. Flash CS5 va CS6 da kiritish matn maydonini yaratish

Foydalanuvchi zarur hollarda matn kiritishi uchun (saytdagi interaktiv shakllar, turli o'yinlarga kirish uchun ma'lumotlar va boshqalar) **Flash CS5 da** matn kiritish maydonining maxsus turi mavjud - input text field. Maqsadlari bo'yicha u HTML sahifani belgilash tilidagi matn kiritish maydoniga juda o'xshaydi.

Flash CS5 va CS6 -da matn maydonini vizual ravishda yaratishda, xuddi statik matn (static text) va dinamik matn (dynamic text) kabi, Flash dasturining asboblari panelida joylashgan Text vositasidan foydalaniladi. Input matn maydoni turi va nusxa nomi Xususiyatlar(Properties Panel) panelida tayinlanadi.

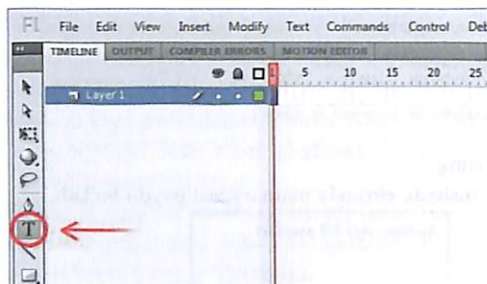
Bundan tashqari, xuddi shu panelda quyidagilarni belgilashingiz mumkin:

- foydalanuvchi ushbu maydonga matn kiritishda ko'radigan shrift turi, o'lchami va rangi
- matn maydoni atrofida 1 piksel qalinlikdagi qora konturni ko'rsatish kerakmi (agar bu parametrlar belgilanmagan bo'lsa, chegara ko'rinmas bo'ladi)
- ko'p qatorli matn maydonini yaratish, belgilar sonini (raqamlar va harflar) belgilash imkonini beruvchi paragrafni formatlash uchun sozlashlarni amalga oshirish.

HTML-dagi matn kiritish maydonlariga o'xshab, Flash CS5 matn maydonini turi bo'yicha statik matn(static text) bo'lgan qulay yorliq (label) bilan ta'minlashingiz mumkin.

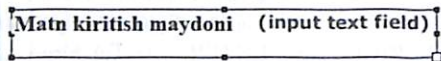
## Flash CS5 da matn maydonini yaratish

Text vositasini bosing.



9.4.1-rasm. Flash CS5 da matn maydonini yaratish

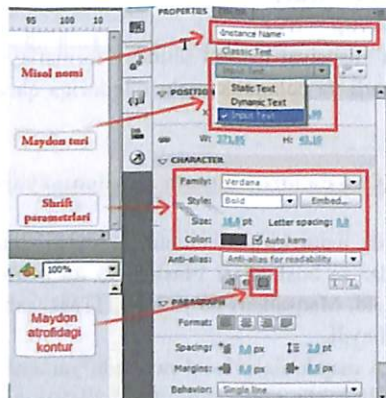
Sahnaning(stage) istalgan joyini bosing va matn maydonini chizish uchun torting.



Xususiyatlar panelida matn kiritish maydoni(Input) uchun maydon turini tanlang.

Shrift turini, shrift hajmini, uning rangini tanlang.

Matn oynasi atrofidagi chegarani ko'rsatish uchun chegarani ko'rsatish (ShowBorder) tugmasini bosing.



9.4.2-rasm. Matn formatini o'zgartirish uchun xususiyatlar paneli

**Ctrl + Enter** ni bosing . Flash Player Flash CS5 matn maydonini ko'rsatadi.

Ushbu maydonni bosing va matnni kiriting.

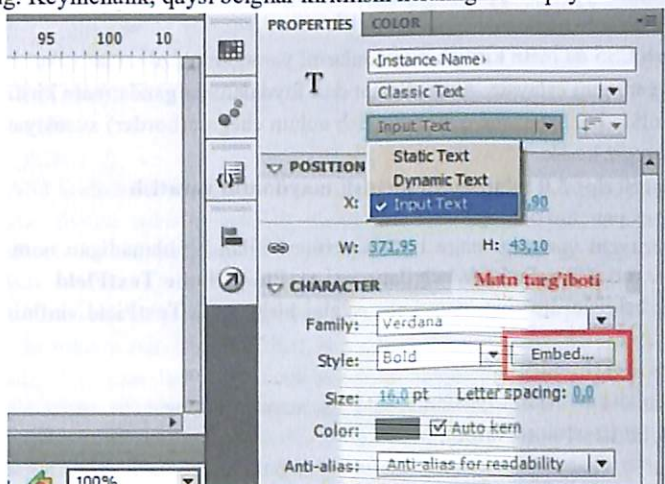
Kiritgan matn matn maydonida ko'rsatiladi.

### Qo'shimcha ma'lumot

Film/o'yinda statik matndan foydalanilganda Flash avtomatik ravishda shriftlarni natijada paydo bo'lgan swf fayliga joylashtiradi, lekin matn kiritishda buni qilmaydi. Shuning uchun, agar loyihangizda biron bir nostandart shriftdan foydalanmoqchi bo'lsangiz (Arial,

Times New Roman yoki Verdanadan tashqari), Flashga bunday shriftni chop etilgan swf fayliga joylashtirishingiz kerakligini to'g'ridan-to'g'ri aytishingiz kerak.

Buni amalga oshirish uchun xususiyatlar panelidagi "Embed"(amalga oshirish) tugmasini bosing. Keyinchalik, qaysi belgilar kiritilishi kerakligini aniqlaysiz.



9.4.3-rasm. Flash da swf fayliga shriftni joylashtirish uchun "Embed" tugmasidan foydalanish

Dinamik matn maydonlari (dynamic text field) uchun tanlangan matn shriftining barcha belgilarini, matn kiritish maydonlari uchun esa (input text field) faqat klaviaturadan kiritilishi mumkin bo'lgan belgilarni amalga oshirish mantiqan to'g'ri keladi: harflar, raqamlar, tinish belgilari. Ko'rinib turibdiki, swf fayliga o'rnatilgan shriftlar yakuniy fayl hajmini oshiradi, shuning uchun iloji bo'lsa, bunday joylashtirishdan qochish kerak.

## 5. TextFieldType.INPUT, TextFieldType.DYNAMIC TextField AS3 sinf nusxalari

Matn kiritish maydonlari (input text field), shuningdek dinamik matn maydonlari (dynamic text field) TextField sinfining nusxalari. Shuning uchun ular ActionScript bilan yaratiladi.

Harakatlar ketma-ketligi quyidagicha:

- o'zgaruvchi e'lon qilinadi va unga nom beriladi
- o'zgaruvchi uchun ma'lumotlar turi TextField ga o'rnatiladi
- teng belgisini qo'yning va new kalit so'zini kiriting
- TextField sinfining konstruktori chaqiriladi
- addChild usulidan foydalanib, nusxa sahnaga qo'shiladi
- x va y xususiyatlariga kerakli qiymatlarni, shuningdek, kengliklarni(width)

belgilash orqali yaratilgan namuna sahnada to'g'ri joyga joylashtiriladi.

Masalan :

```
var txtUserName:TextField = new TextField();  
addChild(txtUserName);
```

Ammo, agar yuqoridagi kodni qo'llasangiz, dinamik matn maydonini (dynamic text field) olasiz. Dinamik matn va kiritish maydoni o'rtasidagi farq type xususiyati qiymatida!

type xususiyatining qiymati

**TextFieldType.INPUT**

yoki

**TextFieldType.DYNAMIC**

ham bo'lishi mumkin.

Flash CS5 da matn kiritish maydonlarini yaratganingizda, ular sukut bo'yicha belgilanmaganligini eslaysiz. ActionScript dan foydalanayotganda matn kiritish maydoni nusxasi atrofida konturni true qilib belgilash uchun chegara(border) xususiyatidan ham foydalanishingiz kerak.

### ActionScript 3.0 bilan matn kiritish maydonini yaratish

Actions panelini oching.

O'zgaruvchi yarating, unga txt qisqartmasi bilan boshlanadigan nom bering va uning ma'lumotlar turini TextField ga belgilang **var txtPhotoName:TextField**

Teng belgisini qo'ying; new kalit so'zini kiriting va TextField sinfining konstruktorini chaqiring

```
= new TextField();
```

```
addChild() usulidan foydalanib, sahnaga nusxa qo'shing
```

```
addChild(txtPhotoName);
```

Turini matn maydon input sifatida belgilang

```
txtPhotoName.type = TextFieldType.INPUT;
```

Kerakli x, y, width va border qiymatlarini

```
o'rnatib txtPhotoName.x = 150;
```

```
txtPhotoName.y = 150;
```

```
txtPhotoName.width = 250;
```

```
txtPhotoName.border = true;
```

**Ctrl + Enter** ni bosib. Flash Player ishga tushadi va sahnada belgilangan joyda ekranda matn kiritish maydoni paydo bo'ladi.

Ushbu kiritish maydonini bosib. Istalgan qiymatni kiriting.

### Qo'shimcha ma'lumot

ActionScript kodidan foydalangan holda matn kiritish maydonining yaratilgan nusxasiga kerakli shrift, shrift rangi va shrift hajmini belgilash uchun siz TextFormat sinfidan foydalanishingiz kerak. Uning xususiyatlari buni amalga oshirishga imkon beradi. Bu sinf bilan ishlashni keyingi darslarda ko'rib chiqamiz.

ActionScript kodi bilan statik matn yarata olmaganingiz uchun, kiritish matn maydonlari(input text field) uchun tushuntirish yorliqlarini yaratish uchun dinamik matn maydonidan(dynamic text field) foydalanishingiz mumkin. Uning nusxasini yaratganingizdan so'ng, uni x, y va kenglik xususiyatlari bilan to'g'ri joyga joylashtirasiz. Matn xususiyatidan foydalanib kerakli izohli yozuv yoki yorliqni yaratasiz.

## 6. TextInput AS3, change, link, scroll - TextField hodisalari

ActionScript 3 dagi boshqa sinflar kabi, foydalanuvchi matn maydoni TextField bilan o'zaro aloqada bo'lganda javob berishga imkon beruvchi, u bilan bog'langan voqealarga ega. click standart bosish hodisalariga qo'shimcha ravishda, TextField yana 4 ta maxsus hodisani o'z ichiga oladi:

- TextInput

- change
- link
- scroll

Ushbu hodisalarning har biri TextEvent sinfidan tegishli o'zgarmlar (constant) bilan ifodalanadi:

- TEXT\_INPUT
- CHANGE
- LINK
- SCROLL

**textInput AS3** hodisasi foydalanuvchi kiritish maydoniga matn kiritganda ishga tushadi. Matni klaviatura, ovoqli matnga uzatish moslamasi yordamida yoki nusxa ko'chirish operatsiyasi natijasida kiritish mumkin. Kompyuter klaviaturasida har gal tugma bosilganda hodisa hosil bo'ladi.

**change** hodisasi matn maydoni (input) mazmuni o'zgariganda chaqiriladi. Misol uchun, agar foydalanuvchi matnni maydonga kiritrsa, u holda har bir tugmani bosgandan so'ng, bu hodisa boshlanadi. Bu, masalan, real vaqt rejimida foydalanuvchi tomonidan kiritilgan ma'lumotlarni bir matn maydonidan ikkinchisiga nusxalash uchun ishlatilishi mumkin. JavaScript kabi boshqa dasturlash tillarida bu hodisa matn kiritish maydoni fokus rejimidan chiqqandan keyingina ishga tushadi. ActionScript-da, bu hodisa kompyuter klaviaturasida har bir tugma bosilganda boshlanadi.

Matnli giperhavola bosilganda **link** hodisasi ishga tushadi. Bunday holda, matn maydonining o'zi HTML formatida bo'lishi kerak. Buni qanday qilishni keyingi darslarda ko'rib chiqamiz.

**scroll** hodisasi foydalanuvchi tomonidan chaqiriladi. Aslida, bu hodisa aylantirish tugagandan so'ng ishga tushiriladi, shuning uchun uni faqat aylantirish tugallangandan keyin ishlatish kerak. Misol uchun, sizning saytingizda foydalanuvchi yuklab olishi uchun hujjat - shartnoma yoki shartnoma mavjud. "Yuklab olish" tugmasini faqat foydalanuvchi shartnomani/kelishuvni oxirigacha aylantirib ko'rgandan keyingina faollashtirish uchun aylantirish hodisasidan foydalanishingiz mumkin. Bu foydalanuvchini hech bo'lmaganda, agar o'qimagan bo'lsa, keyingi harakatlarni amalga oshirishdan oldin butun matnni ko'rib chiqishni taklif qiladi.

## 7. Matn hodisalari bilan ishlash

Flash dasturida yoki ActionScript kodidan foydalanib kiritish matn maydonini yarating ( "Flash CS5 da matn kiritish maydonini yaratish" va " ActionScript 3.0 bilan matn kiritish maydonini yaratish" darslariga qarang ). Masalan :

```
var txtInputText:TextField = new TextField();
addChild(txtInputText);
txtInputText.type = TextFieldType.INPUT;
txtInputText.x = 150;
txtInputText.y = 150;
txtInputText.width = 250;
txtInputText.height = 50;
txtInputText.border = true;
```



Flash yoki ActionScript kodi bilan dinamik matn maydoni (dynamic text field) yaratish (qarang: Flash CS5 da dinamik matn yaratish va ActionScript 3.0 bilan dinamik matn yaratish darslari). Masalan : `var txtDisplayText:TextField = new TextField(); addChild(txtDisplayText);`

```
txtDisplayText.type = TextFieldType.DYNAMIC;
txtDisplayText.x = 250;
txtDisplayText.y = 150;
txtDisplayText.width = 250;
txtDisplayText.border = true;
```

Kerakli hodisani belgilash orqali voqea tinglovchisini qo'shing. Masalan :

```
txtInputText.addEventListener(TextEvent.TEXT_INPUT, textHandler);
```

Ishlov beruvchi hodisalarni yaratish. Masalan:

```
function textHandler(event:TextEvent):void
```

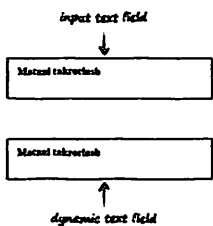
Hodisa sodir bo'lgandan keyin qayta ishlash uchun ko'rsatmalar yozadigan

figurali qavslar qo'ying {

```
txtDisplayText.text = txtInputText.text;
}
```

Ctrl + Enter ni bosish. Flash Player ishga tushadi.

Matn kiritish maydoniga (input) matn kiritish. Natijada kiritgan matn dinamik matn maydonida (dynamic) takrorlanadi.



9.7.1-rasm. Matn hodisalari bilan ishlash

### Qo'shimcha ma'lumot

Har gal klaviatura tugmasi bosilganda change va textInput AS3 hodisalari ishga tushadi.

Shubhasiz, ba'zi hollarda, bu, masalan, foydalanuvchi juda ko'p matnli ma'lumotlarni kiritganda, vaqt birligi uchun juda katta ma'lumot oqimiga olib kelishi mumkin. Yana bir misol, agar server bilan bog'lanish uchun Flashdan foydalansangiz.

Ushbu muammoning oldini olish uchun click hodisasidan foydalaning. Ya'ni, foydalanuvchi kerakli ma'lumotlarni kiritgandan so'ng, u tugmani bosadi, bu esa kiritilgan barcha ma'lumotlarni bir vaqtning o'zida uzatish uchun hodisani ishga tushiradi.

Shuni ta'kidlash kerakki, foydalanuvchi dinamik matn maydoniga matn ma'lumotlarini kiritmaydi yoki o'zgartir olmaydi, lekin uni tanlash va nusxalash imkoniyatiga ega. Garchi bu amalni tanlash mumkin selectable bo'lgan xususiyatni "false" ga o'rnatish orqali ham o'chirib qo'yish mumkin.

### O'tkazish funksiyasi.

Agar ma'lumotlar butun ekranga mos kelmasa, ScrollBar komponentidan foydalangan holda foydalanuvchi ma'lumotlarning ko'rsatilgan qismini boshqarishi mumkin. O'tkazish

paneli to'rt qismdan iborat: ikkita o'q tugmasi, trek va slayder. Slayderning joylashuvi va tugmalarning ko'rinishi aylantirish panelining joriy holatiga bog'liq. O'tkazish panelining ekran holatini hisoblash uchun to'rtta parametr qo'llaniladi: diapazonning minimal qiymati; diapazonning maksimal qiymati; diapazon qiymatlari orasida bo'lishi kerak bo'lgan joriy pozitsiya; va diapazondan kichik yoki teng bo'lishi va bir vaqtning o'zida ko'rsatilishi mumkin bo'lgan diapazondagi elementlar sonini aks ettiruvchi ko'rish oynasining o'lchami.

### Umumiy xususiyatlar

9.7.1-jadval. Umumiy xususiyatlar jadvali.

Xususiyatlar	Xususiyat ta'rifa
direction : String	O'tkazish paneli gorizontal yoki vertikal ekanligini ko'rsatadigan qiymatni oladi yoki o'rnatadi. Ruxsat etilgan ScrollBarDirection.HORIZONTAL va ScrollBarDirection.VERTICAL. Standart qiymat ScrollBarDirection.VERTICAL.
enabled : Boolean	O'tkazish paneli yoqilganligini ko'rsatadigan mantiqiy qiymatni oladi yoki o'rnatadi. Standart qiymat true.
height : Number	Komponentning balandligini oladi yoki o'rnatadi (piksellarda).
lineScrollSize : Number	aylantirish paneli slayderi bosilganda sahifani aylantirish qiymatini belgilaydi yoki o'rnatadi. PageScrollSize qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi bosqichma-bosqich o'lchanadi. Agar bu qiymat 0 bo'lsa, pageSize xossasining qiymati ishlatiladi. Standart qiymat 0.
maxScrollPosition : Number	Maksimal aylantirish holatiga mos keladigan raqamni oladi yoki o'rnatadi. scrollPosition qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi nisbiy pozitsiyani aks ettiradi . Bu xususiyat aylantirish panelini o'z ichiga olgan komponent tomonidan o'rnatiladi va odatda nolga o'rnatiladi. Odatda, bu xususiyat komponentning pastki cheti va kontentning pastki cheti orasidagi piksellar sonini belgilaydi, lekin u ko'pincha aylantirish harakatini o'zgartiradigan boshqa qiymatga o'rnatiladi. Misol uchun, TextArea komponenti bu xususiyatni matn maydonining maxScrollH qiymatiga
	o'rnatadi, bu esa matn satrini aylantirish paneli yordamida aylantirish kerakligini belgilaydi. Standart qiymat 0.
minScrollPosition : Number	Minimal aylantirish pozitsiyasiga mos keladigan raqamni oladi yoki o'rnatadi. ScrollPosition qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi nisbiy pozitsiyani aks ettiradi . Bu xususiyat aylantirish panelini o'z ichiga olgan komponent tomonidan o'rnatiladi va odatda nolga o'rnatiladi. Standart qiymat 0.

pageScrollSize : Number	O'tkazish paneli slayderi bosilganda sahifani aylantirish qiymatini oladi yoki o'rnatadi. PageScrollSize qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi bosqichma-bosqich o'lchanadi. Agar bu qiymat 0 bo'lsa, pageSize xususiyatining qiymati ishlatiladi. Standart qiymat 0.
pageSize : Number	Har bir sahifadagi satrlar sonini oladi yoki o'rnatadi. LineScrollSize qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi bosqichma-bosqich o'lchanadi. Agar bu xususiyat 0 ga o'rnatilgan bo'lsa, aylantirish paneli ishlamaydi. Standart qiymat 10.
scrollPosition: Number	Joriy aylantirish holatini oladi yoki o'rnatadi va slayder o'rni yangilaydi. ScrollPosition qiymati minScrollPosition va maxScrollPosition qiymatlari orasidagi nisbiy pozitsiyani aks ettiradi. Standart qiymat 0.
width : Number	Komponentning kengligini oladi yoki o'rnatadi (piksellarda).
ScrollBar ()	ScrollBar komponentining yangi nusxasini yaratadi.
getStyleDefinition():Objekt	Joriy komponent uchun standart uslublar xaritasini qaytaradi.
setScrollProperties (pageSize: Number , minScrollPosition: Number , maxScrollPosition: Number , pageScrollSize: Number = 0): void	ScrollBar komponenti uchun ko'rish oynasining diapazoni va hajmini o'rnatadi.
setSize(width:Number, height:Number):void	Komponentning belgilangan balandligi va kengligini o'rnatadi.

## 2. Nazorat savollari



1. TextField dinamik matn maydonlarining nusxalari qaysi sinfga kiradi?
2. TextField nechta maxsus hodisalarni o'z ichiga oladi?
3. **textInput AS3** hodisasi qachon yonadi?
4. **change** hodisasi qachon yonadi?
5. **link** hodisasi qachon yonadi?
6. **scroll** hodisasi qachon yonadi ?
7. O'tkazish paneli enabled komponentining yoqilgan xususiyati nimani ko'rsatadi?
8. **PageSize** nimani qaytaradi?
9. **minScrollPosition** usuli nimani belgilaydi?
10. **pageScrollSize** usuli nimani belgilaydi?
11. Matnni dinamik maydondan nusxalashni qanday taqiqlash mumkin?

## 2.10. Klipli-keyslarga ishlov berish haqida umumiy ma'lumot

### Reja:

1. Dinamik kliplar nusxalarini yaratish.
2. Film - belgisini bog'lash jarayoni.
3. Bo'sh film belgisining yangi nusxasini dinamik ravishda yaratish.
4. ActionScript bilan dinamik chizish.
5. Nazorat savollari.

**Kalit so'zlar:** *tugmalar yaratish, teskari aloqa, MoveClip, attachMovie, obyekt massivlari, ActionScript, attachment, export, bog'lash, identifier, converter, symbol, iframe, paint, klipni takrorlash, out-of-the-box, Canvas, icon, controller, Windows, holder, inline funksiyalar, sichqoncha, chaqiruv, linearization, radial, ratio, matrix, obyekt, swapDepth, swapping, dragging, DROP, clear, raqamlash, print, swapping, dragging.*

### 1. Dinamik kliplar nusxalarini yaratish

Film-belgini dinamik ravishda yaratish uchun MovieClip obyektining uchta usuli qo'llaniladi:

- **duplicateMovieClip()**: Bu usul sahnada mavjud film-belgisi nusxasining dublikatini (yangi nusxasini) yaratadi.

- **attachMovie()**: Bu usul sahnadagi kutubxonada kino belgisining yangi nusxasini yaratadi.

- **createEmptyMovieClip()** : Bu usul bo'sh film-belgisi nusxasini yaratadi, ya'ni ma'lumotlar yoki grafiklarni o'z ichiga olmaydi.

Ushbu usullarning har birini ushbu darsdagi mashqlar jarayonida qo'llaymiz.

#### **duplicateMovieClip()**

Bu usul bilan Skriptlarni takrorlanishlar bilan avtomatlashtirish bo'limida allaqachon tanishgan edik, ammo biz buni tafsilotlarga kirmasdan qisqacha qildik.

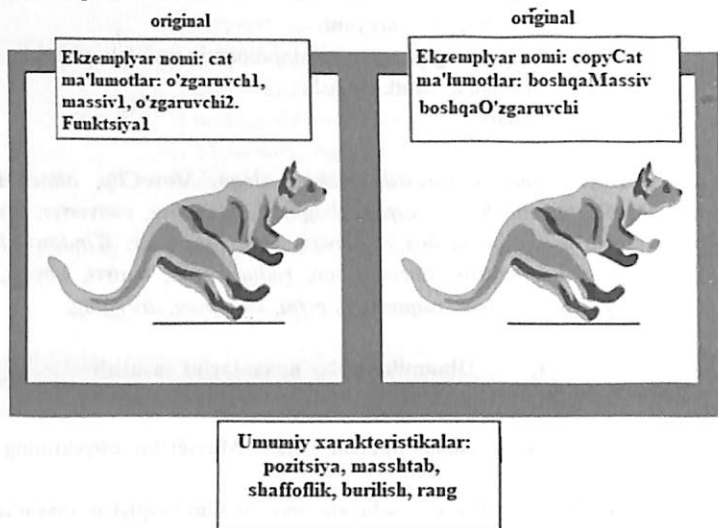
duplicateMovieClip() usuli Flashga sahnada film-belgi nusxasining dublikatini yaratishni buyuradi va unga yangi nom beradi. Agar klipning ko'rsatilgan nusxasi sahnada bo'lmasa (u oldingi yoki keyingi kadrlardan birida bo'lsa), Flash uni takrorlay olmaydi. Yana bir cheklov shundaki, dublikat faqat asl nusxa bilan bir xil montaj stolida yaratilishi mumkin. Har qanday montaj stolida klip nusxasini dinamik ravishda yaratish uchun **attachMovie()** usulidan foydalaning.

Yaratilgan dublikat asl nusxaning takrorlanish vaqtidagi barcha fizik xususiyatlarini meros qilib oladi.

Ikki nusxadagi film belgisi asl nusxaning quyidagi xususiyatlarini meros qilib oladi:

- Joylashuv
- Masshtab
- Shaffoqlik
- Burilish burchagi
- Rang

- Klip nusxasiga birlashtirilgan voqea ishlov beruvchilari Ikki nusxadagi film belgisi meros qilib olmaydi:
- O'zgaruvchilar, massivlar, obyektlar
- Ism
- "Ko'rinish"
- Joriy kadr raqami



10.1.1-rasm. Flashda kliplarni nusxalash ushun `duplicateMovieClip()` usuli

Dublikat nusxa, hatto dublyaj qilinganda asl klip boshqa freymda bo'lgan bo'lsa ham, ikkidan o'ynay boshlaydi.

Film belgisining nusxasini ko'paytirish sintaksisi:

**`myClip.duplicateMovieClip(nom, chuqurlik, obyekt);`**

Ushbu ActionScript qatori ko'paytiriladigan film-belgi nomidan (manzil bilan) boshlanadi, so'ngra MovieClip obyektining duplicateMovieClip() usulini chaqiradi. Natijada, yangi nusxa yaratiladi, uning nomi chuqurlikdagi parametr nomining qiymati bo'ladi. Obyekt parametri ixtiyoriy. Misol: **`name = "ball2"`**;

**`depth = 100;`**

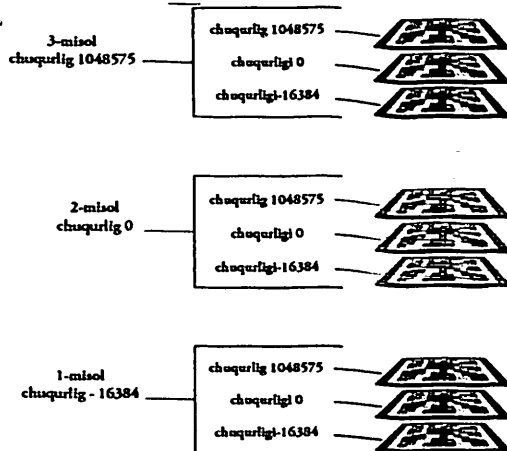
**`ball.duplicateMovieClip(name, depth);`**

Ball ning dublikat klipi nusxasini yaratadi; dublikat nusxa ball2 deb nomlanadi va 100 chuqurlikda joylashgan.

Bu yerda chuqurlik (**depth**) haqida gapirganda, klip nusxalarining montaj stolidagi joylashuv tartibini nazarda tutamiz. Flashda, agar ikkita klip nusxasi bir-biriga mos tushsa, yuqoridagisi kattaroq chuqurlik qiymatiga ega bo'ladi. Film belgisining har bir nusxasi sahnadagi boshqa obyektlarga nisbatan uning o'rnini belgilaydigan noyob chuqurlik qiymatiga ega. Ko'paytirishda yangi klip nusxasiga har qanday ijobiy raqam bo'lgan chuqurlik qiymati tayinlanadi. Bu raqam qanchalik katta bo'lsa, "sahna ustidagi" nusxa shunchalik yuqori bo'ladi. Loyihani ishlab chiqish jarayonida xronologiyaga joylashtirgan film belgilarining barcha nusxalariga avtomatik ravishda chuqurlik berilishini, chuqurlik qiymatlari -16384 manfiy raqamdan boshlanadiganligini tushunmagan bo'lishingiz mumkin. Bu shuni anglatadiki, 1

chuqurlikda joylashtirilgan dinamik ravishda yaratilgan misol sahnaga qo'lda joylashtirilgan har qanday narsani qoplaydi.

Filmidagi har bir montaj stolining chuqurlik diapazoni -16384 dan 1048575 gacha, barcha chuqurliklarga filmdagi asosiy montaj stoliga nisbatan havola qilingan. Boshqacha qilib aytadigan bo'lsak, ma'lum "1-nusxa" o'z montaj stolida -16384 dan 1048575 gacha bo'lgan chuqurlikda joylashgan kichik nusxalarni o'z ichiga olishi mumkin. Ammo agar asosiy montaj stolida "1-nusxa" "2-nusxa" ostida joylashgan bo'lsa, har qanday, hatto eng yuqorisi ham "1-nusxa" ning asosiy nusxasi "2-nusxa" ning eng past bola nusxasidan past bo'ladi.



10.1.2.-rasm. Flashda kliplarni nusxalash ushbu `duplicateMovieClip()` usuli

Har bir chuqurlikda istalgan vaqtda film-belgining faqat bitta nusxasi bo'lishi mumkin. Klip nusxasini boshqa nusxa mavjud bo'lgan chuqurlikka ko'paytirishga urinish bu ikkinchi nusxani yo'q qiladi.

Shunday qilib, har bir montaj stolida 1064960 chuqurlik darajasi mavjud (-16384 dan 1048575 gacha). Shu bilan birga, turli xil o'rnatish stollarining chuqurliklari hech qanday tarzda bir-biri bilan kesishmaydi.

`duplicateMovieClip()` usulining uchinchi parametri, obyekt parametri ixtiyoriy, ammo foydasiz emas. Ushbu parametr sifatida ko'rsatilgan obyektning xususiyatlarini yangi dublikat klipning ma'lumotlar a'zolariga aylantiradi. Agar parametr o'tkazib yuborilsa, u e'tiborga olinmaydi.

Oldingi misolni kengaytiramiz: `myObject`

```

= new Object();
myObject.ballColor =
"red"; name = "ball2";
depth = 100;

```

`ball.duplicateMovieClip(name, depth, myObject);` ball2 ning yangi dublikat

nusxasi `myObject` obyektining barcha xususiyatlarini o'z ichiga

oladi. Bunday holda, yangi misolda "qizil" qiymatiga ega `ballColor` o'zgaruvchisi yaratiladi.

Agar o'zgaruvchilarni asl nusxadan dublikatga ko'chirmoqchi bo'lsangiz, uchinchi parametr sifatida asl nusxa nomidan foydalaniladi. Masalan: `ball.duplicateMovieClip(name, depth, ball);`

**attachMovie()**

**attachMovie()** usulidan foydalanib, kutubxonadan film belgisini olishingiz va uning nusxasini dinamik ravishda film sahnasida mavjud bo'lgan istalgan montaj stoliga "qo'shishingiz" (**attach**) mumkin. Misol u qo'shilgan filmning mazmuniga qo'shiladi. Biriktirilgan klip u biriktirilgan montaj stolining bolasiga aylanadi. U ota-onada bajarilgan barcha grafik o'zgarishlarni (o'lchamini o'zgartirish, shaffoflik, aylantirish va boshqalar) meros qilib oladi, shu bilan birga ma'lumotlarga, ko'rinishga, joriy kadrqa va boshqalarga nisbatan mustaqillikni saqlaydi.



## 10.2.1.-rasm. film belgisini bog'lashda attachMovie() usuli

**duplicateMovieClip()** usuli o'rtasidagi farq nima? Yuqorida aytib o'tilganidek, **attachMovie()** kutubxonadan klip nusxasini filmdagi har qanday montaj stoliga biriktiradi. Ilova qilingan klip kutubxonadan kelganligi sababli (va kutubxonada filmdagi barcha kliplarni mavjud), nusxa biriktirilgan klip biriktirish vaqtida sahnada bo'lishi shart emas. **duplicateMovieClip()** usuli bilan buning aksi bo'ladi: dunlikat klip nusxasi sahnada bo'lishi kerak. Bundan tashqari, dublikat faqat asl nusxa bilan bir xil montaj stolida yaratiladi. Yana bir farq quyidagicha. Agar dublikat nusxada voqea ishlov beruvchilari biriktirilgan bo'lsa ( **data**, **enterFrame**, **mouseDown** va boshqalar), u holda dublikat ularni avtomatik ravishda meros qilib oladi. Bu biriktirilgan klip bilan sodir bo'lmaydi. Albatta, biriktirilgan klipga hodisa ishlov beruvchilarini ham qo'shish usullari mavjud, ammo jarayon takrorlashdagi kabi oddiy emas.

Oddiy qilib aytganda, biriktirish har qanday montaj stolini boshqa har qanday montaj stoliga qo'shish imkonini beradi. Ko'paytirish faqat asl nusxasi joylashgan montaj stolida klip nusxasining aniq nusxalarini yaratishga imkon beradi.



Birlashtirishga misol



Takrorlashga misol

## 10.2.2.-rasm. attachMovie() usuli

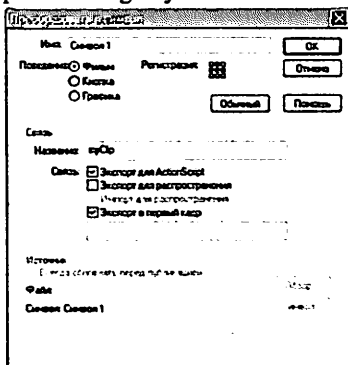
attackMovie() usulini qo'llash muayyan shartlarni bajarishni talab qiladi. Birinchi shart - birlashtirilgan kutubxonadagi kliplar to'g'ri tayyorlangan bo'lishi kerak. Ushbu tayyorgarlik bog'lanish deb ataladi. Bog'lanish kutubxonadagi bog'lanadigan kliplarni aniqlash jarayonini tavsiflash uchun eng yaxshi atama bo'lmasligi mumkin. Ammo gap shundaki, bu atama (va bu jarayon) kutubxonani almashish bilan ham bog'liq - bu yerda bitta SWF faylining kutubxona elementlari boshqa SWF fayliga ulanishi mumkin (ya'ni, umumiy bo'ladi). Qo'shish uchun (shuningdek, ikkita SWF fayli o'rtasida almashish uchun) mavjud bo'lgan filmlar belgisini belgilash uchun ularga nomlar - identifikatorlar berilishi kerak. Macromedia kutubxonasidagi film belgilariga identifikatorlarni tayinlash bilan bog'liq har qanday jarayon bog'lanish sifatida qaraladi.

Film belgisini bog'lanishning bosqichma- bosqich jarayoni quyidagicha ko'rinadi:

1. Kutubxonani oching.
2. Sizni qiziqirgan klipni ustiga sichqonchani o'ng tugmasini bosing.
3. Kontekst menyusidan Bog'lanish(Linkage) ni tanlang.
4. ActionScript uchun o'rnatishni belgilash katagi Eksport (Export for ActionScript).

5. Ism (Identifikator) maydoniga nom kiriting (bu identifikator ActionScript da kutubxonadagi klipga havola sifatida xizmat qiladi).

Simvolga aylantirish (Convert to Symbol) dialog oynasidagi Kengaytirilgan (Advanced) tugmasini bosish orqali film-belgini yaratishda havolani o'rnatishingiz mumkin.



10.2.3.-rasm. Film-belgini yaratishda havolani o'rnatish

Ulanishda yana bir variantga e'tibor bering - "Birinchi kadrda eksport qilish" (Export in first frame), dastlabki qiymatida bo'yicha yoqilgan. Ushbu parametr film ko'rilganda bog'langan (yoki birlashtirilgan) film belgisi qaysi nuqtada yuklanishini aniqlaydi. Belgilansa, klip asosiy filmning 1-kadridan oldin yuklanadi va shuning uchun uning mazmuni uni birlashtirishga urinishdan oldin (masalan, 1-kadrda) mavjud bo'ladi. Katta hajmdagi kliplardan foydalanilganda

(masalan, boshqa ko'plab film belgilari yoki ovozi yozuvlarni o'z ichiga olgan), Flash film boshida "sekinlashishi" mumkin. Xavotir olmang, u faqat bog'langan barcha xarakterli filmlarning mazmunini yuklab oladi. Agar "Birinchi kadrda eksport qilish" katagi belgilanmagan bo'lsa, birinchi kadrlardan oldin film- belgi yuklanmaydi. Bunday holda, uni birlashtirishga urinishdan oldin bunday klipning nusxasini sahnaga qo'yishingiz kerak. Film belgisi istalgan kadrda qayta ishlangandan so'ng (filmida paydo bo'ladi), uni istalgan vaqtda va



istalgan joyda biriktirish mumkin. Misol uchun, agar 25-kadrda klipni biriktiruvchi harakat mavjud bo'lsa, u holda ushbu klipning nusxasi 24-kadrda yoki undan oldin filmda paydo bo'lishi kerak yoki `attachMovie()` usuli muvaffaqiyatsiz bo'ladi. 24-kadrda so'ng ushbu klipning nusxalarini xohlaganingizcha ko'p marta biriktirishingiz mumkin, chunki u allaqachon xotiraga yuklangan. Kutubxonadan film belgisini montaj stoliga qo'shish sintaksisi:

```
path.attachMovie(id, yangi nom, chuqurlik, obyekt);
```

Masalan :

```
_root.wall.attachMovie("paint", "paint2", 10);
```

Film belgisining `_root.wall` nusxasiga "paint" identifikatori bilan kutubxonadagi film belgisini biriktiradi. Yangi biriktirilgan nusxa `paint2` deb nomlanadi va 10 chuqurlikda joylashtiriladi. Biriktirilgan klip o'zi biriktirilgan nusxaning bolasiga aylanadi, shuning uchun unga yo'l `_root.wall.paint2` bo'ladi. Sintaksis misoliga e'tibor bering (birinchi qator): bu yerda yaratilgan klipni obyekt xususiyatlari va o'zgaruvchilari bilan to'ldirish imkonini beruvchi variant ham mavjud. `duplicateMovieClip()` da bo'lgani kabi, bu parametr ixtiyoriy va bizning misolimizda (ikkinchi qator) o'tkazib yuborilgan.

Klip hodisalarini biriktirilgan (yoki dublikat qilingan) klipga dinamik ravishda belgilashning eng oson usuli, ushbu misolda bo'lgani kabi, biriktirishni (dublikat lashtirishni) amalga oshiradigan amaldan so'ng hodisa ishlov beruvchilarini aniqlashdir:

```
_root.attachMovie("box", "dynamicBox", 1);  
dynamicBox.onEnterFrame = function(){  
dynamicBox._rotation += 15;  
}
```

Ushbu misolda, film belgisi qutisining nusxasi (kutubxonadagi uning nomi) dinamik ravishda asosiy montaj stoliga biriktirilgan. Biriktirilgan misol `dynamicBox` deb nomlanadi. Qo'shish amaldan keyingi qatorlar yangi nusxaga klip hodisasi ishlov beruvchisini tayinlaydi.

### 3. Bo'sh film belgisining yangi nusxasini dinamik ravishda yaratish

#### `createEmptyMovieClip()`

`createEmptyMovieClip()` usuli dinamik ravishda (istalgan montaj stolida) bo'sh filmklipning yangi nusxasini yaratishga imkon beradi. Bu nima uchun kerak bo'lishi mumkin? Mana bir nechta misollar.

- Ushbu usul yordamida yaratilgan bo'sh klipga boshqa kliplarning nusxalarini biriktirishingiz mumkin, masalan, ro'yxatlar va menyularni dinamik ravishda yaratish. Agar menyu elementlari bunday "tayanch" klipga biriktirilgan bo'lsa, har bir elementni alohida ko'chirish o'rniga butun menyuni ko'chirishingiz mumkin.

- Bo'sh klipni yaratganingizdan so'ng, unga ovozi yozuv yoki rasmni yuklashingiz mumkin.

- Kliplarning bo'sh nusxalari Flashning tasvirlash xususiyatlari bilan yaratilgan chiziq, to'ldirish va gradient plombalarni saqlash uchun ajoyib joy yaratadi. Bo'sh film belgisini yaratish sintaksisi quyidagicha: `path.createEmptyMovieClip(nom, chuqurlik)`; Masalan:

```
_root.createEmptyMovieClip("box", 1);
```

Bu yerda birinchi parametr yangi bo'sh klip misoliga beriladigan nomdir; ikkinchi parametr - bu misol joylashtiriladigan chuqurlik. Agar ushbu amalni sinab ko'rmoqchi bo'lsangiz, hech qanday ko'rinadigan natijani topa olmaydiz - axir, yaratilgan nusxa butunlay

bo'sh. Ushbu qo'llanmada chiziq'larni dinamik ravishda chizish uchun createEmptyMovieClip() usulidan foydalanildi.

#### 4. ActionScript bilan dinamik chizish

ActionScript -dan film o'ynalayotganda dinamik ravishda chizish uchun foydalanish mumkin. Bu imkoniyat MovieClip obyektining chizish usullari bilan ta'minlanadi.

Chizish usullari quyidagilarga imkon beradi:

- Joriy "chizish kursori" dan belgilangan nuqtaga chiziq torting
- "Chizish kursorini" boshqa joyga o'tkazing (chiziq chizmasdan)
- Montaj stoli uchun chiziq uslubini belgilang
- Shaklni rang bilan to'ldiring
- Shaklni gradient bilan to'ldiring
- Ikki nuqta orasidagi chiziq egri chiziqli
- Chizilgan chiziq'lar va plombalarni montaj stolidan olib tashlang.

Ushbu bo'limda chiziq'lar chizishni, "chizish kursorini" siljitishni, chiziq uslubini o'rnatishni va film belgisi misolidan chizmalarini tozalashni o'rganasiz. Keyingi bo'limda muntazam va gradient to'ldirishga to'xtalamiz. curveTo() usuli (egri chiziq'larni dinamik ravishda chizish imkonini beradi) bu yerda tavsiflanmaydi, lekin ushbu dars oxirida Flash -da dinamik chizmachilik asoslarini yetarlicha o'zlashtirgan bo'lasiz, shuning uchun ushbu usulni ilovalaringizda hech qanday muammosiz ishlatishingiz mumkin.

##### lineStyle()

Doskaga biron bir chiziq chizishdan oldin, ushbu montaj stoli uchun chiziq uslubini o'rnatishingiz kerak. Bu Flashda belgilashingiz kerakligini anglatadi:

- Chiziq qalinligi
- Chiziq rangi
- Chiziq shaffofligi Sintaksis bu: **path.lineStyle(qalinlik, rang, shaffoflik)**

Chiziq qalinligining qiymati 0 dan 255 gacha bo'lgan butun son bo'lishi kerak (0 qiymati eng nozik chiziqqa mos keladi). Chiziq rangi o'n oltilik kod sifatida ko'rsatilgan. Shaffoflik 0 dan (to'liq shaffof) 100 (to'liq shaffof) gacha bo'lgan raqamdir. Quyidagi misolni ko'rib chiqaylik:

```
_root.myClip.lineStyle(10, 0x009900, 100);
```

myClip uchun chiziq uslubini o'rnatadi: ushbu klipda chizilgan barcha chiziq'lar yashil, noaniq va 10 qalinlikda bo'ladi.

##### moveTo()

Film belgisining har bir nusxasi chizma kursorining joylashuvi, lineTo() usuli ishlatilsa, chiziq boshlanadigan koordinata kabi xususiyatga ega (keyingi kichik bo'limga qarang).

Boshqacha qilib aytganda, bu keyingi qatorning boshlang'ich nuqtasidir. "Chizish kursori" nusxasi yaratilganda  $x=0$ ,  $y=0$  ga o'rnatiladi. Biroq, chizilgan kursorni istalgan vaqtda moveTo() usuli yordamida ko'chirish mumkin. moveTo() dan foydalanish sintaksisi quyidagicha: **path.moveTo(x, y);**

Chizish kursorining yangi joylashuvi uchun x va y koordinatalarini belgilash kifoya.

Masalan:

```
_root.myClip.lineStyle(10.0x009900.100);
```

```
_root.myClip.moveTo(100,100);
```

Ushbu skript chiziq uslubini o'rnatadi va keyin chizilgan kursorni harakatga keltiradi.  
**lineTo()**

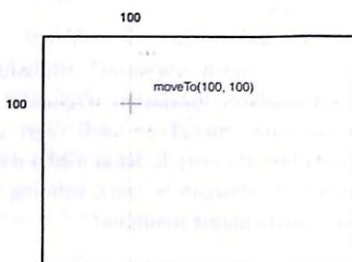
MovieClip obyektining lineTo() usuli ko'rsatilgan ish stolida ( joriy montaj stolining lineStyle formatida) chizish kursorining joriy holatidan usul chaqirilganda ko'rsatilgan nuqttagacha chiziq chizadi.

lineTo() sintaksisi oddiy: **myClip.lineTo(x,y)**; x va y parametrlari, taxmin qilganingizdek, chizilgan chiziqning oxirgi nuqtasini aniqlaydi.

Chizilgan kursorning joriy holati chiziqning oxirgi nuqtasiga o'tkaziladi.

Mana tasvirlagan barcha usullardan foydalangan holda chiziq chizish misoli:

```
_root.createEmptyMovieClip("canvas",1);  
_root.canvas.lineStyle(2,0x009900,100);  
_root.canvas.moveTo(100,100);
```

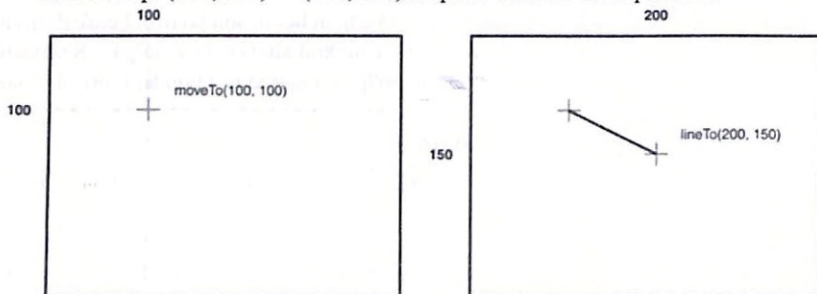


10.4.1.-rasm. Chizish kursorining joylashuvi uchun x va y koordinatalarini belgilashda moveTo()

usuli

```
_root.canvas.lineTo(200,150);
```

Ushbu skript (100, 100) va (200, 150) nuqtalar o'rtasida chiziq chizadi.



10.4.2.-rasm. Barcha usullardan foydalangan holda chiziq chizish

## 5. Nazorat savollari



1. Dinamik kliplar nusxalari yaratish usullari necha xil?
2. duplicateMovieClip() usuli ta'rifi
3. attachMovie() usuli ta'rifi

4. createEmptyMovieClip() usuli ta'rifi
5. Bo'sh film belgisining yangi nusxasini dinamik ravishda yaratish
6. createEmptyMovieClip() usuli qanday ishlaydi
7. ActionScript bilan dinamik chizish 8. moveTo() usuli haqida nimalarni bilasiz?
9. lineTo() usuli haqida nimalarni bilasiz?
10. lineStyle() usuli haqida nimalarni bilasiz?

## 2.11. getTimer() funksiyasi. key obyektning asosiy usullari

### Reja:

1. FLASH da vaqtni orqaga hisoblashdan foydalanish
2. Joriy vaqt va sanani aniqlash
3. Vaqtni ortga hisoblash. getTimer() funksiyasi
4. Klaviatura bilan ishi. Key obyektning asosiy usullari
5. Nazorat savollari

**Kalit so'zlar:** *flesh, taymer, indikator, o'rnatilgan obyekt, animatsiya, ActionScript, setInterval, skript, funksiya, obyekt, parametr, raqamlash, ro'yxat, aniqlik, o'zgaruvchi, qiymat, massiv, aniq vaqt, clocked, hands-on, yorliq, joriy vaqt, sound, fayl, FPS, hodisalarni boshqarish, boshqarish, MovieClip, kursor, GET, phone, kadr, background, play, indicator, "clip", CLIPS, Actions, rewind, atribut, foiz, content, amount.*

### 1. FLASH da vaqtni orqaga hisoblashdan foydalanish

Flash loyihalarida vaqtni o'lchashning bir necha usullari mavjud. Eng keng tarqalganlari quyidagilar:

- **Date obyekt**i . Flashning o'rnatilgan obyekt; sanalar(kun, oy, yil) bilan ishlash imkonini beradi.

- **getTimer() funksiyasi**. Maxsus Flash funksiyasi; vaqtni millisekundlarda hisoblash imkonini beradi.

- **Kadrlar**. Kadr Flash - dagi eng keng tarqalgan "vaqt birligi" dir, chunki bu animatsiyalar, tovushlar, videolar va hokazolarni ijro etishning asosi bo'lgan kadrlar soni. Joriy kadrдан keyingi (yoki oldingi) ga o'tish loyihaning o'z vaqtida (oldinga yoki orqaga) rivojlanishini anglatadi.

Ushbu elementlardan foydalanib, loyihangizni quyidagi xususiyatlar bilan ta'minlashingiz mumkin:

- Foydalanuvchining harakatlariga qarab oldinga yoki orqaga o'ynatiladi
- Loyihaning joriy sana, vaqt yoki kadr raqamiga munosabati
- Yuklab olish jarayoni haqida ma'lumotni ko'rsatish (masalan, foizda)
- Va boshqalar.

**ActionScript** - da yana bir maxsus vosita mavjud - **setInterval()**, bu ma'lum bir interval bilan (millisekundlarda o'lchangan) har qanday funksiyani qayta-qayta chaqirish imkonini beradi.

Masalan, ushbu skriptni ko'rib chiqamiz:

```
function rotateClip() { myMovieClip._rotation += 10;
}
```

```
setInterval(rotateClip, 1500);
```

Birinchi uchta satr funksiyani tavsiflaydi. Keyin `setInterval()` amali bajariladi va `rotateClip()` funksiyasini har bir yarim soniyada chaqiradi (1000 millisekund = 1 soniya).

Agar chaqirilgan funksiya argument qiymatlarini o'tkazishi kerak bo'lsa, ular oddiygina `setInterval()` amaliga parametr sifatida qo'shiladi:

```
setInterval(updateMessageFunction, 2000, "Hello", arg2, arg3);
```

`setInterval()` amalini ushbu misoldagi kabi o'zgaruvchi nomini "tayinlash" orqali yoqish yoki o'chirish mumkin:

```
myVariable = setInterval(rotateClip, 1500);
```

Shunday qilib, `myVariable` nomi ushbu `setInterval()` amaliga tayinlangan. Endi `setInterval()` amalini bajarishni to'xtatish uchun quyidagi buyruqni ishlatish kifoya:

```
delete myVariable;
```

`setInterval()` amali istalgan vaqtda ishga tushirilishi mumkin, masalan, uning chaqiruvini biron bir funksiyaga kiritish orqali.

## 2. Joriy vaqt va sanani aniqlash

Biz hammamiz kalendardan foydalanamiz; ko'pchilik busiz qanday qilish mumkinligini hatto tasavvur ham qilmaydi - axir, kelajak uchun ishlaringizni rejalashtirishingiz kerak va hokazo. Flash da **kalendar sanasi** haqidagi ma'lumotlar ham foydali bo'lishi mumkin - uni foydalanuvchiga ko'rsatish yoki ma'lum bir kunda ba'zi muayyan harakatlarni bajarish yoki **taymer yaratish** ortga hisoblash va yana boshqalar.

**Flash -da kalendar sanasidan foydalanish uchun Date obyektini yaratish kerak:**

```
myDate = new Date(year, month, date);
```

Ushbu skript qatori yangi `Date` ning `myDate` nomli obyektini yaratadi. Qavslar ichidagi parametrlar ushbu obyektini ma'lum bir sana bilan bog'laydi. Masalan: `myDate = yangi sana (66, 6, 27);`

Ushbu skript yangi **obyektini yaratadi** 1966 yil 27 iyul bilan bog'liq sana. Birinchi parametr yilni, ikkinchisi oyni, uchinchisi esa kunni belgilaydi. Aytgancha, hayron bo'lmang, hech qanday xatolik yo'q, garchi iyul yettinchi oy bo'lsa-da va skriptda 6 deb yozilgan. Gap shundaki, **ActionScript -da**. Haftaning oylari va kunlari raqamlar bilan ko'rsatilgan, **raqamlash** noldan boshlanadi. Shunday qilib, yanvar - "nol" oy, fevral - "birinchi", mart - "ikkinchi" va shunga o'xshash "o'n birinchi" - dekabrgacha. Xuddi shunday, hafta nol kundan boshlanadi, yakshanba, keyin dushanba va hokazo.

**Eslatma.** Oylar va yillar raqamlari haqiqiy qiymatlarga ega. Shunday qilib, 66 1966 yilga to'g'ri keladi va 27 oying 27-kuniga to'g'ri keladi.

**Eslatma** Oylar va yillar soni haqiqiy qiymatlarga ega. Shunday qilib, 66 1996 yilga, 27 esa oying 27-kuniga to'g'ri keladi.

Oylar											
0	1	2	3	4	5	6	7	8	9	10	11
Yanvar	Fevral	Mart	April	May	Iyun	Iyul	Avgust	Sentyabr	Oktyabr	Noyabr	Dekabr
Hafta kunlari											
0	1	2	3	4	5	6					
Yakshanba	Dushanba	Seshanba	Chorshanba	Payshanba	Juma	Shanba					



**Maslahat.** Yaratilgan obyekt uchun parametrlar ro'yxati, agar ilovangiz bunday aniqlikka muhtoj bo'lsa, soat, daqiqa, soniya va millisekundlarni ko'rsatish orqali davom ettirilishi mumkin.

**Date** obyektini yaratish kerak bo'lsa, joriy sana va vaqtni o'z ichiga oladi (kompyuterining tizim soatiga ko'ra), obyekt konstruktori qavslarini bo'sh qoldirib, shunchaki parametrlarni ko'rsatmang. Masalan:

```
myDate = new Date();
```

**Date** Obyekt yaratilgandan keyin, undan tegishli usullardan foydalangan holda saqlangan sana haqidagi har qanday ma'lumotni olishingiz mumkin. Oldingi misolni, joriy tizim sanasini o'z ichiga olgan obyektning oling. Qaysi oy ekanligini bilish uchun quyidagi sintaksisdan foydalanamiz: `currentMonth = myDate.getMonth();`

Ushbu qatorni bajargandan so'ng, `currentMonth` o'zgaruvchi raqamli qiymatga ega bo'ladi, masalan, 5 bu iyun oyini bildiradi. Haftaning kunini bilish uchun boshqa usuldan foydalanamiz:

```
currentDay = myDate.getDay();
```

Ushbu qatorni bajargandan so'ng, `currentDay` o'zgaruvchiga raqamli qiymat beriladi, masalan, 4 payshanba.

**Eslatma.** Loyihada turli maqsadlar uchun bir nechta `Date` obyektlari bo'lishi mumkin.

### 3. Vaqtni ortga hisoblash. `getTimer()` funksiyasi

`getTimer()` funksiyasi film boshlanganidan beri o'tgan vaqtni (millisekundlarda) qaytaradi. Agar, masalan, ushbu funksiya chaqirilganda, film olti soniya davomida o'ynalayotgan bo'lsa, u holda skript `playBackTime = getTimer();`

`playBackTime` o'zgaruvchisini 6000 (har soniyada 1000 millisekund) ga o'rnatadi. Bu foydalanuvchi kompyuterining tizim soatiga asoslangan aniq vaqt; bu filmning kadrlar tezligiga, prosessor tezligiga ham, filmning bir xil kadrda o'ynalayotganiga yoki turishiga bog'liq emas. Funksiya alohida vaqt jadvalarini emas, balki butun film boshlanganidan beri o'tgan vaqtni qaytarishini ta'kidlaymiz.

Filmidagi ikkita voqea orasidagi vaqt oralig'ini hali ham o'lchash mumkin. Buning uchun kerakli vaqtda `getTimer()` funksiyasi tomonidan qaytarilgan qiymatni qandaydir o'zgaruvchiga tayinlang va funksiya tomonidan qaytarilgan yangi qiymatni ushbu o'zgaruvchi bilan solishtiring. Buni quyidagi misol bilan tushuntiramiz: // A tugmasi uchun skript `on (release) {`

```
    startTime = getTimer(); } //  
Skript uchun B tugmasi on  
(release) { nowTime =  
getTimer(); if (nowTime -  
startTime < 1000) {  
    message.text = " Siz tugmani juda tez bosdingiz";  
} else {  
    message.text = " Juda kech";  
}  
}
```

Bu yerda tugmalardan biri (A tugmasi) bosilsa va qo'yib yuborilsa, vaqtni belgilashning boshlang'ich nuqtasi o'rnatiladi. B tugmasi bosilsa va qo'yib yuborilsa, vaqt yana o'rnatiladi.

```
function rotateClip() { myMovieClip._rotation += 10;
}
```

```
setInterval (rotateClip, 1500);
```

Birinchi uchta satr funksiyani tavsiflaydi. Keyin `setInterval()` amali bajariladi va `rotateClip()` funksiyasini har bir yarim soniyada chaqiradi (1000 millisekund = 1 soniya).

Agar chaqirilgan funksiya argument qiymatlarini o'tkazishi kerak bo'lsa, ular oddiygina `setInterval()` amaliga parametr sifatida qo'shiladi:

```
setInterval (updateMessageFunction, 2000, "Hello", arg2, arg3);
```

`setInterval()` amalini ushbu misoldagi kabi o'zgaruvchi nomini "tayinlash" orqali yoqish yoki o'chirish mumkin:

```
myVariable = setInterval (rotateClip, 1500);
```

Shunday qilib, `myVariable` nomi ushbu `setInterval()` amaliga tayinlangan. Endi `setInterval()` amalini bajarishni to'xtatish uchun quyidagi buyruqni ishlatish kifoya:

```
delete myVariable;
```

`setInterval()` amali istalgan vaqtda ishga tushirilishi mumkin, masalan, uning chaqiruvini biron bir funksiyaga kiritish orqali.

## 2. Joriy vaqt va sanani aniqlash

Biz hammamiz kalendardan foydalanamiz; ko'pchilik busiz qanday qilish mumkinligini hatto tasavvur ham qilmaydi - axir, kelajak uchun ishlaringizni rejalashtirishingiz kerak va hokazo. Flash da **kalendar sanasi** haqidagi ma'lumotlar ham foydali bo'lishi mumkin - uni foydalanuvchiga ko'rsatish yoki ma'lum bir kunda ba'zi muayyan harakatlarni bajarish yoki **taymer yaratish** ortga hisoblash va yana boshqalar.

**Flash -da kalendar sanasidan foydalanish uchun Date obyektini yaratish kerak:**

```
myDate = new Date(year, month, date);
```

Ushbu skript qatori yangi `Date` ning `myDate` nomli obyektini yaratadi. Qavslar ichidagi parametrlar ushbu obyektini ma'lum bir sana bilan bog'laydi. Masalan: `myDate = yangi sana (66, 6, 27);`

Ushbu skript yangi **obyektini yaratadi** 1966 yil 27 iyul bilan bog'liq sana. Birinchi parametr yilni, ikkinchisi oyni, uchinchisi esa kunni belgilaydi. Aytgancha, hayron bo'lmang, hech qanday xatolik yo'q, garchi iyul yettinchi oy bo'lsa-da va skriptda 6 deb yozilgan. Gap shundaki, **ActionScript -da**. Haftaning oylari va kunlari raqamlar bilan ko'rsatilgan, **raqamlash** noldan boshlanadi. Shunday qilib, yanvar - "nol" oy, fevral - "birinchi", mart - "ikkinchi" va shunga o'xshash "o'n birinchi" - dekabrgacha. Xuddi shunday, hafta nol kundan boshlanadi, yakshanba, keyin dushanba va hokazo.

**Eslatma.** Oylar va yillar raqamlari haqiqiy qiymatlarga ega. Shunday qilib, 66 1966 yilga to'g'ri keladi va 27 oyning 27-kuniga to'g'ri keladi.

**Eslatma** Oylar va yillar soni haqiqiy qiymatlarga ega. Shunday qilib, 66 1996 yilga, 27 esa oyning 27-kuniga to'g'ri keladi.

Oylar											
0	1	2	3	4	5	6	7	8	9	10	11
Yanvar	Fevral	Mart	April	May	Iyun	Iyul	Avgust	Sentyabr	Oktyabr	Noyabr	Dekabr
Hafta kunlari											
0	1	2	3	4	5	6					
Yakshanba	Dushanba	Seshanba	Chorshanba	Payshanba	Juma	Shanba					

11.2.1.-rasm. Flashda kalendar sanasidan foydalanish uchun Date obyektini ishlatish natijasi



**Maslahat.** Yaratilgan obyekt uchun parametrlar ro'yxati, agar ilovangiz bunday aniqlikka muhtoj bo'lsa, soat, daqiqa, soniya va millisekundlarni ko'rsatish orqali davom ettirilishi mumkin.

**Date** obyektini yaratish kerak bo'lsa, joriy sana va vaqtni o'z ichiga oladi (kompyuterning tizim soatiga ko'ra), obyekt konstruktori qavslarini bo'sh qoldirib, shunchaki parametrlarni ko'rsatmang. Masalan:

```
myDate = new Date();
```

**Date** Obyekt yaratilgandan keyin, undan tegishli usullardan foydalangan holda saqlangan sana haqidagi har qanday ma'lumotni olishingiz mumkin. Oldingi misolni, joriy tizim sanasini o'z ichiga olgan obyektini oling. Qaysi oy ekanligini bilish uchun quyidagi sintaksisdan foydalanamiz: **currentMonth = myDate.getMonth();**

Ushbu qatorni bajargandan so'ng, **currentMonth** o'zgaruvchi raqamli qiymatga ega bo'ladi, masalan, 5 bu iyun oyini bildiradi. Haftaning kunini bilish uchun boshqa usuldan foydalanamiz:

```
currentDay = myDate.getDay();
```

Ushbu qatorni bajargandan so'ng, **currentDay** o'zgaruvchiga raqamli qiymat beriladi, masalan, 4 payshanba.

**Eslatma.** Loyihada turli maqsadlar uchun bir nechta Date obyektlari bo'lishi mumkin.

### 3. Vaqtni ortga hisoblash. **getTimer()** funksiyasi

**getTimer()** funksiyasi film boshlanganidan beri o'tgan vaqtni (millisekundlarda) qaytaradi. Agar, masalan, ushbu funksiya chaqirilganda, film olti soniya davomida o'ynalayotgan bo'lsa, u holda skript **playBackTime = getTimer();**

**playBackTime** o'zgaruvchisini 6000 (har soniyada 1000 millisekund) ga o'rnatadi. Bu foydalanuvchi kompyuterining tizim soatiga asoslangan aniq vaqt; bu filmning kadrlar tezligiga, protsessor tezligiga ham, filmning bir xil kadrda o'ynalayotganiga yoki turishiga bog'liq emas. Funksiya alohida vaqt jadvallarini emas, balki butun film boshlanganidan beri o'tgan vaqtni qaytarishini ta'kidlaymiz.

Filmidagi ikkita voqea orasidagi vaqt oralig'ini hali ham o'lchash mumkin. Buning uchun kerakli vaqtda **getTimer()** funksiyasi tomonidan qaytarilgan qiymatni qandaydir o'zgaruvchiga tayinlang va funksiya tomonidan qaytarilgan yangi qiymatni ushbu o'zgaruvchi bilan solishtiring. Buni quyidagi misol bilan tushuntiramiz: // A tugmasi uchun skript **on (release) {**

```
    startTime = getTimer(); } //  
Skript uchun B tugmasi on  
(release) { nowTime =  
getTimer(); if (nowTime -  
startTime < 1000) {  
    message.text = " Siz tugmani juda tez bosdingiz";  
    } else {  
    message.text = " Juda kech";  
    }  
}
```

Bu yerda tugmalardan biri (A tugmasi) bosilsa va qo'yib yuborilsa, vaqtni belgilashning boshlang'ich nuqtasi o'rnatiladi. B tugmasi bosilsa va qo'yib yuborilsa, vaqt yana o'rnatiladi.



Shartli gap bu ikki moment orasida bir soniyadan ko'p yoki kamroq vaqt o'tganligini aniqlaydi va tegishli harakatni bajaradi. Xuddi shunday, bitta tugma uchun ikki marta bosish bilan ishlov berishni amalga oshirishingiz mumkin - quyidagi skript faqat tugmani ikkinchi marta bosish birinchisidan yarim soniyadan kechiktirmasdan sodir bo'lsa, amallarni bajaradi:

```
on (release) {
    if (getTimer() - lastClick < 500) {
        // Amallar
    }
    lastClick = getTimer();
}
```

getTimer() funksiyasidan enterFrame hodisasi va shartli operatorlar bilan birgalikda foydalanib, ma'lum bir vaqtda ma'lum harakatlarni bajarish mexanizmini yaratishingiz mumkin – montaj stolidan qat'iy nazar va katta aniqlik bilan. Keling, ushbu misolni olaylik:

```
onClipEvent(enterFrame) {
    agar (getTimer() > 5000) {
        // Amallar
    } if (getTimer() > 10000) {
        // Amallar
    }
    if (getTimer() > 15000) {
        // Amallar
    }
}
```

Ushbu skript film ijro etilayotganda har besh soniyada bir amalni bajaradi - bu ko'pincha aniq vaqt talab qilinsa, borishning yagona yo'li.

Eslatma. getTimer() funksiyasini Functions bo'limidagi Harakatlar paneli asboblari ro'yxatidan topish mumkin.



11.3.1.-rasm. getTimer() funksiyasini joylashgan Functions bo'limi

### Montaj stolini bajarish tezligi va yo'nalishini boshqarish

Oddiy sharoitlarda filmning vaqt jadvali oldinga va film parametrlari sozlamalarida ko'rsatilgan fps (sekundagi kadrlar) qiymati bilan belgilangan tezlikda o'ynaydi. Biroq, ActionScript -dan foydalanib, montaj stolini ijro etish yo'nalishini va tezligini boshqarishingiz mumkin. Darhaqiqat, enterFrame hodisani boshqarishni birlashtirish orqali turli ActionScript elementlariga ega, loyiha montaj stolini deyarli mutlaq nazorat qilishingiz mumkin.

Bu yerda gaplashishimiz kerak bo'lgan birinchi narsa - `NextFrame()` va `prevFrame()` kabi `MovieClip` obyektining usullari. Ularning yordami bilan montaj stolini keyingi yoki oldingi kadrga ko'chirishingiz mumkin: `on (release) {`

```

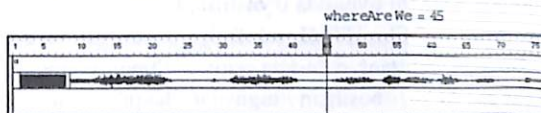
myMovieClip.nextFrame();
}
yoki
on (release) {
myMovieClip.prevFrame();
}

```

Montaj stolida oldinga yoki orqaga harakatlanadigan navigatsiya tugmachalarini olamiz, har bir tugmani bosish uchun bir kadr.

`_currentframe` kabi montaj stolining xususiyatini eslatib o'tmaslik mumkin emas (faqat o'qish uchun xususiyat). Uning qiymati hozirda o'ynash boshi joylashgan kadr raqamidir. Masalan, asosiy montaj stoli o'ynayapti deydik, keyin quyidagi skript `whereWeAre` o'zgaruvchisini joriy kadr raqamiga mos keladigan raqamli qiymatga o'rnatadi:

```
whereWeAre = _root._currentframe;
```



11.3.2.-rasm. Montaj stolini ko'rinishi.

Bu xususiyatni shartli operator bilan birga foydalanib, masalan, o'ynatish boshi kadrlar oralig'ida yoki yo'qligini aniqlashingiz va tegishli choralarni ko'rishingiz mumkin:

```

onClipEvent(enterFrame) {
if (_root._currentframe >= 50 && _root._currentframe <= 100){
// Ushbu amallarni bajaring
}
}

```

Ushbu skriptda shartli operatorning asosiy qismidagi harakatlar faqat asosiy montaj stolining o'ynatish boshi 50 va 100 kadrlar orasida bo'lsa bajariladi.

Agar `gotoAndPlay()` amali bilan birga `_currentframe` xususiyatidan foydalansangiz, montaj stolining yo'nalishini ham, tezligini ham boshqarishingiz mumkin. Keling, ushbu misolni olaylik:

```
_root.gotoAndPlay (_currentframe + 10);
```

Bu chiziq asosiy montaj stolining o'yin boshini joriy holatidan 10 kadr oldinga "sakrashga" olib keladi. Xuddi shu tarzda, montaj stolida 10 kadr orqaga o'tishingiz mumkin: `_root.gotoAndPlay(_currentframe - 10);`

#### 4. Klaviatura bilan ishlash. Key obyektning asosiy usullari

Ushbu bo'limda biz klaviatura yordamida Flash obyektlarini qanday boshqarishni ko'rib chiqamiz. `Key` obyektning asosiy usullari bilan ham tanishamiz. Bu obyekt hozir qaysi tugma bosilganligini va qaysi tugma oxirgi bosilganligini bildirish uchun ishlatiladi. Klaviaturada tugma bosilganda 3 xil holat yuzaga keladi. Bular `keyPress`, `keyDown` va `keyUp`

Flashning birinchi versiyalarida ham ba'zi tugmachalarni bosishga munosabat bildirish mumkin edi. Keyin bu imkoniyatning tashuvchisi keyPress tugmalarining maxsus hodisasi bo'ldi.

Bu belgilarning tugmachalarini, shuningdek, ba'zi xizmat kalitlarini "ushlash" imkonini berdi. Misol:

```
on(keyPress "A")
{ trace ("A Klaviatura bosildi");
}
```

KeyPress hodisasidan Flash MX 2004 da ham foydalanish mumkin. Biroq buni qilmaslik kerak, chunki Flash 5 klaviaturani bosish bilan ishlashga imkon beruvchi ancha kuchli elementlarni taqdim etgan. Ularning tashuvchisi maxsus o'rnatilgan Key obyektidir. Unda 2 ta hodisa ( `keyDown` va `keyUp` ), 4 ta usul va 18 ta xususiyat mavjud.

#### **Klaviatura hodisasi xususiyatlari**

Klaviatura tugmachalarini bosish va chiqarish faqat Flash playerga fokus bo'lsa yuboriladi. Aks holda, hozirda faol dastur ularni qabul qiladi.

Albatta, agar film alohida o'yinchi oynasida o'ynatilsa, bu hech qanday qiyinchiliklarga olib kelishi mumkin emas. Agar swf film HTML sahifasiga o'rnatilgan bo'lsa, qiyinchiliklar paydo bo'lishi mumkin. Bunday sahifani ochishda asosiy e'tibor o'yinchiga emas, balki brauzerga qaratiladi. Shunga ko'ra, bosilgan tugmalar haqidagi ma'lumotlar filmga yuborilmaydi.

Ta'riflangan muammoni hal qilishning ikki yo'li mavjud:

- Film o'ynatilishni boshlamasdan bosadigan maxsus tugmani yarating. Ushbu tugmani bosish orqali foydalanuvchi diqqatni brauzerdan playerga o'tkazadi. Ushbu yondashuv o'yinlar uchun maqbuldir.

- Shuningdek, JavaScript fokus() usuli yordamida filmga avtomatik ravishda fokuslashingiz mumkin.

Ushbu usul saytlar uchun maqbuldir.

Flash player, deyarli har qanday dastur kabi, o'zining "issiq" tugmachalari va kombinatsiyalariga ega.

Ular playerning asosiy parametrlarini nazorat qiladi va o'ynaladigan filmga buyruqlar beradi.

Masalan, + tugmasini bosib filmni to'liq ekran rejimiga o'tkazishingiz mumkin. Undan chiqish uchun <Esc> tugmasidan va hokazolardan foydalanishingiz kerak. Tezkor tugma yoki kombinatsiya bosilganda, ba'zi tugmalar faollashtirilganligi haqidagi ma'lumot filmga yuborilmaydi. Shunday qilib, agar tugmani bossangiz, `onKeyDown` hodisasi bo'lmaydi. + tugmasini bosish faqat pastga tugmani qayd qiladi.

Agar film klaviatura boshqaruvidan faol foydalansa, u holda "issiq" tugmalar va kombinatsiyalarning mavjudligi istalmagan bo'lishi mumkin (masalan, kalitni yopish o'yini yoki taqdimoti sifatida ishlatish qulay). Ularni qo'llab-quvvatlashni o'chirish uchun quyidagi qatorni kiritish kerak:

```
fscommand("trapallkeys", "true");
```

Afsuski, operatsion tizim va brauzerning "issiq" kombinatsiyalarini bloklash mumkin emas. Shunday qilib, + tugmachasini bosish orqali boshqa ochiq dastur oynasini faollashtirasiz.

<AH> +

o'yinchi oynasini yopadi. HTML sahifasiga o'rnatilgan filmda <Tab> tugmasini bosish undan fokusni olib tashlashi mumkin. Agar film brauzerda o'ynalayotgan bo'lsa, chertishni "tutib olish" umuman mumkin emas.

Ba'zi "issiq" kombinatsiyalarni o'chirib qo'yishning iloji yo'qligi sababli, loyihalarda maxsus tugmalardan umuman foydalanmaslik yaxshiroqdir. Bu, ayniqsa, tugmalar va <Tab>-ga tegishli. O'ziga xos "issiq" kombinatsiyalar va kalitlar ham Flash sinov muhitiga xosdir. Shuning uchun, masalan, <Tab> tugmasi yordamida klaviaturani aylanib o'tish yoki bosishni "tutib olish" mumkin emas. Qisqa klavishlar va yorliqlardan foydalanishni o'chirish uchun Control menyusidagi Klaviatura yorliqlarini o'chirish(Disable Keyboards Shortcuts) parametrini yoqing. Aytgancha, bu muammoni hal qilishning yagona yo'li, chunki fscommand() buyruqlari test muhitida ishlamaydi. **onKeyDown** va **onKeyUp** hodisalari

**onKeyDown** hodisasi foydalanuvchi klaviaturadagi tugmachani bosganda sodir bo'ladi. Shunga ko'ra, **onKeyUp** hodisasi kalit bo'shatilganda sodir bo'ladi. Obyekt ushbu hodisalar haqida xabarlarini qabul qilishi uchun u Key obyektining tinglovchilari ro'yxatiga qo'shilishi kerak. Buni **addListener()** usuli yordamida qilishingiz mumkin:

```
var obj:Object = {};  
Key.addListener(obj);  
// Har qanday tugmani bosganingizda, xabar Outputda paydo  
bo'ladi  obj.onKeyDown = function():Void { trace( "Tugma  
bosildi " );  
};
```

**Key** obyektning tinglovchilar ro'yxatidan olib tashlang **removeListener()** usulidan foydalanishingiz mumkin.

Flash 5 hodisa modeli ixtiyoriy obyektga hodisa generator yoki tinglovchisi bo'lishiga ruxsat bermadi. Keyin ular faqat kliplar va tugmachalarga birlashtirilgan. Klaviaturani bosish va chiqarish hodisalari sukut bo'yicha barcha kliplarga "tinglangan". Flash 5 uslubida ishlov beruvchilarni yaratib, hozir ham xuddi shunday kuzatishimiz mumkin:

```
onClipEvent(keyDown){           // Istalgan tugmani bosganda klip yo'qoladi  
this._visible=false;  
}
```

Shunday qilib, eski hodisa modeli bilan klaviatura hodisalari dastlab kliplar uchun mavjud.

Biroq, agar usul ishlov beruvchilardan foydalansangiz, u holda kliplar sukut bo'yicha **onKeyDown** va **onKeyUp** hodisalari uchun tinglovchilar emas; **this.createEmptyMovieClip("clip", 0);**

// Tugmachalarni bosish xabar bilan birga bo'lmaydi

```
clip.onKeyDown = function(): Void {  
trace (" Kalit  
bosildi!"); } }
```

Xuddi shu hodisa, lekin boshqa yozuv shaklida o'zini boshqacha tutishi hech bo'lmaganda g'alati ko'rinadi. Ayniqsa, Flash ishlab chiquvchilari ikkita hodisa modelini iloji boricha o'xshash qilishni xohlagan deb hisoblasangiz. Ba'zi tajribalar bilan kliplar hali ham **onKeyDown** va **onKeyUp** voqealarini "tinglashi" mumkinligini ko'rishingiz mumkin, ammo ularda fokus bo'lsa:

```
// Doira nomi bilan klip yaratish  
clip.focusEnabled = true; // Klipni tanlash mumkin bo'lsin
```

```

Selection.setFocus(clip); // Klipga e'tibor qarating
// Tugmani bosish klipni ikki baravar kattalashtirishga olib keladi
clip.onKeyDown = function(): Void {
this._xscale = this._yscale *=
2; }

```

Garchi **onKeyDown** va **onKeyUp**, bu amalda hech qanday foyda keltirishi dargumon. Va bu tanlangan klipni o'rab turgan zerikarli sariq kadr ham emas - uning ekranini osongina o'chirib, qo'yishingiz mumkin. Shunchaki, fokus, hatto u klip n ga kerakli vaqtda dasturiy jihatdan, **Selection** obyektining **setFocus()** usuli yordamida qo'yilgan bo'lsa ham, unga tayinlangan funksiyalarini bajarishga shunchaki vaqt topa olmasligi mumkin. Siqhqonchani eng kichik harakati bilan u yo'qoladi. Siqhqoncha tugmasi yoki bo'sh joy yoki kiritish tugmasi bosilsa ham xuddi shunday bo'ladi.

Kliplarda **onKeyDown** va **onKeyUp** hodisalarini faqat diqqat markazida bo'lgan juda kam hollarda bo'lganligi sababli, ular ushbu hodisalarining tinglovchilari emas deb taxmin qilish mumkin. Shuning uchun, agar klipni klaviatura hodisalarini olishni istasangiz, u **addListener()** usuli yordamida **Key** obyektining tinglovchilar ro'yxatiga aniq qo'shilishi kerak. Xuddi shu narsa tugmalar uchun ham amal qiladi (Flash MX 2004 da ularda **onKeyDown** va **onKeyUp** hodisalarini mavjud).

Bir qarashda, **onKeyDown** klaviatura hodisasi **onMouseDown** siqhqoncha hodisasiga juda o'xshaydi. Biroq, ular o'rtasida asosiy farq bor. Siqhqonchani chap tugmasi bosilganda **onMouseDown** hodisasi bir marta sodir bo'ladi. U zudlik bilan ozod qilinganmi yoki nisbatan uzoq vaqt ushlab turilganmi, muhim emas. **onKeyDown** hodisasi, tugmachani bosib ushlab turganda, kalit qo'yib yuborilgunga qadar ancha yuqori chastotada takrorlanadi. **onKeyDown** hodisalarini soni foydalanuvchining operatsion tizimi sozlamalari bilan belgilanadi va soniyada o'rtacha 30 dan 10 tagacha hodisa bo'lishi mumkin.

Agar bir vaqtning o'zida bir nechta tugmalar bosilsa, ularning har biri uchun alohida **onKeyDown** va **onKeyUp** hodisalarini sodir bo'ladi.

**onKeyDown** va **onKeyUp** hodisalaridan foydalanib, faqat ba'zi tugmalar bosilgan yoki qo'yib yuborilganligini bilishingiz mumkin. Qaysi tugma **Key** obyektining **getCode()** va **getAscii()** usullari yordamida aniqlanadi. Ularning tavsifi quyida keltirilgan.

**Bosilgan tugma kodini aniqlash. getCode() usuli.**

Klaviatura tugmalariga qanday murojaat qilish mumkinligini o'ylab ko'raylik. Ehtimol, eng aniq variant - ular kiritadigan belgilar. Misol uchun, agar tugmachada "A" harfi chizilgan bo'lsa, u holda bu kalitni ko'rsatish kerak. Biroq, bu yondashuv juda ko'p muammolarni keltirib chiqaradi. Belgilarga mos kelmaydigan xizmat kalitlarini qanday belgilash mumkin? Turli mamlakatlarda bir xil tugmalar turli belgilarni o'rnatishi mumkinligini qanday hisobga olish kerak? Agar tugma turli maketlarda turli belgilar kiritilsa nima bo'ladi? Raqamli klaviatura mavjudligini qanday hisobga olish kerak?

Shubhasiz, kalit tomonidan kiritilgan belgi bilan murojaat qilish universal emas. Har bir tugmachaga joylashuviga qarab standart raqam berish ancha yaxshi. U qanday belgini o'rnatadi va umuman o'rnatadi - bu ikkinchi darajali.

Windows-da har bir kalit uchun jismoniy joylashuviga qarab tayinlangan standart kod mavjud. Masalan, standart klaviaturada 0 dan 9 gacha raqamlari bo'lgan tugmalar 48 dan 57 gacha bo'lgan kodlar oralig'ini egallaydi. Qizig'i shundaki, A dan Z gacha bo'lgan kalit kodlari. va 0 dan 9 gacha bo'lgan raqamlar ASCII oilasi kodlarlarida egallagan pozitsiyalariga to'g'ri keladi.

ActionScript-da Key obyektining getCode() usuli yordamida oxirgi bosilgan tugma kodini aniqlashingiz mumkin. Masalan:

```
Key.addListener(this);  
// Bosilganda, Outputda xabar ko'rsatiladi  
this.onKeyDown = function():Void {  
if (Key.getCode() == 13) {  
    trace("Klaviatura bosildi");  
}  
}
```

Agar hech qanday tugma bosilmasa, getCode() usuli 0 ni qaytaradi. getCode() usuli onKeyDown va onKeyUp hodisalari bilan uzviy bog'langan. Haqiqatan ham, uni ishlovchilardan tashqarida ishlatish mantiqiy emas, chunki u qaysi tugma oxirgi marta bosilganligini ushbu hodisaning uzoqligi cheklanmagan holda ko'rsatadi. getCode() ning qaytish qiymati qaysi maket yoki registr faol bo'lganiga bog'liq emas.

Masalan, lotincha "F" va ruscha "a" ning kiritilishi bitta kodga to'g'ri keladi - 70. Agar ma'lum bir tugmani bosishga emas, balki ma'lum bir belgi qo'shishga javob berishingiz kerak bo'lsa, unda getAscii() usulidan foydalanishingiz kerak.

getAscii() usuli bilan solishtirganda getCode() usuli muhim afzalliklarga ega. Shunday qilib, bu nafaqat belgi tugmachalarini, balki barcha xizmat ko'rsatish tugmachalarini ham bosishni aniqlash imkonini beradi. Undan asosiy yoki raqamli klaviaturada raqam yoki arifmetik operatsiya tugmasi faollashtirilganligini farqlash uchun foydalanish mumkin. getCode() usuli xalqaro: foydalanuvchi kompyuteri qaysi kod sahifasidan foydalanmasin, bir xil kod bir xil kalitga murojaat qiladi. Bundan tashqari, bu usul platformaga bog'liq; filmni Macintosh mashinalarida o'ynatganda kalit kodlari avtomatik ravishda ularning Windows ekvivalentiga tarjima qilinadi.

Windows virtual kalitlari sichqoncha tugmalarini ham o'z ichiga oladi (ular 1 va 2 kodlariga mos keladi). Biroq, getCode() usuli ular bosilganligini aniqlay olmaydi. Agar shunday ehtiyoj mavjud bo'lsa (masalan, sichqonchani o'ng tugmachasini bosish hodisasini yaratishingiz kerak), u holda isDown() usulidan foydalaning.

Klaviatura bosish tekshiruvchi. isDown() usuli

isDown() usuli ma'lum bir tugma bosilganligini tekshirish imkonini beradi. Parametr sifatida virtual kalit kodini oladi. Mantiqiy qiymat isDown() usuli bilan qaytariladi: true — tugma bosildi, false — u qo'yib yuborildi. Masalan:

```
this.onEnterFrame = function():Void {  
    // Output rejimida ushlab turilganda, har bir yangi kadrda xabar  
    yuboriladi if (Key.isDown(Key.ENTER)) {        trace("Enter  
    bosildi!");  
    }  
}
```

Bir qarashda isDown() usuli ortiqchadek tuyulishi mumkin. Masalan, getCode() va getAscii() usullari mavjud bo'lsa, nima uchun tugma bosilganligini tekshirish kerak. Haqiqatan ham, isDown() usuli juda tez-tez ishlatilmaydi. Biroq, u getCode() yoki getAscii() hal qila olmaydigan muammolarni hal qilishga imkon beruvchi noyob xususiyatlarga ega.

isDown() usulining eng muhim afzalligi shundaki, u bir vaqtning o'zida bir nechta tugmachalarni bosishni ro'yxatga olish imkoniyatiga ega. Va bu o'yinni rivojlantirish uchun juda muhim imkoniyat. Misol uchun, agar belgilar harakati o'q tugmalari yordamida

boshqarilsa, vertikal harakat tugmachasini va yon siljish tugmachasini bir vaqtning o'zida bosish diagonal siljishga olib kelishi kerak.

**isDown()** usuli yordamida amalga oshiradigan kod yozish juda oson.

Masalan: // Kichik doira shaklida klip yaratish va unga shar deb nom bering

```
this.onEnterFrame = function():Void {  
  if(Key.isDown(Key.LEFT)) {           // Chapga ko'rsatgich  
    ball._x -= 3;  
  if (Key.isDown(Key.RIGHT)) {       // O'ngga ko'rsatgich  
    ball._x += 3;  
  if (Key.isDown(Key.UP)) {         // Yuqoriga ko'rsatgich  
    ball._y -= 3;  
  if (Key.isDown(Key.DOWN)) {      // Pastga ko'rsatgich  
    ball._y += 3;  
}
```

**isDown ()** usulining ikkinchi afzalligi shundaki , u **onKeyDown** hodisasi va **getCode()** usulini birgalikda ishlatish orqali kutganingizdan ancha tez-tez tugmani bosib ushlab turish bilan bog'liq o'zgarishlarni amalga oshirish imkonini beradi. Bu o'yinda juda dinamik voqealar bo'lsa (masalan, kosmosdagi janglar) muhim bo'lishi mumkin.

**isDown()** usulining uchinchi muhim xususiyati bizga allaqachon tanish. Esingizda bo'lsa, undan foydalanib, nafaqat klaviatura tugmachalarini, balki sichqoncha tugmachalarini ham bosish orqali ro'yxatdan o'tishingiz mumkin. 2-kodli tugma bosilganligini tekshirish uchun **onEnterFrame** hodisa takrorlanishi yoki **setInterval()** funksiyasini tekshirish **ActionScript**da o'ng tugmani bosishni "ushlash" ning eng ishonchli usuli hisoblanadi.

## 5. Nazorat savollari

1. Flash loyihalarida vaqtni o'lchash uchun qanday usullardan foydalaniladi?
2. Flash loyihalarida vaqtni o'lchash usullari qanday maqsadlarda qo'llaniladi?
3. **ActionScript**dagi funksiyani ma'lum vaqt oralig'ida qayta-qayta chaqirish imkonini beruvchi maxsus vosita qanday nomlanadi ?
4. **ActionScript**da haftaning oylari va kunlari qanday ko'rsatiladi?
5. **ActionScript**da may oyi uchun nechchiga teng?
6. **ActionScript**da haftaning shanba kuniga qaysi raqam mos keladi?
7. **Date** obyekti nechta maydonga ega **ActionScript**da ?
8. Oyni **Date** obyektidan qanday chiqarish mumkin?
9. Qaysi funksiya film boshlanganidan beri o'tgan vaqtni qaytaradi?
10. **getTimer()** funksiyasi qaysi o'lchamda vaqtni qaytaradi?
11. Muayyan harakatlarni ma'lum bir vaqtda - vaqt jadvalidan qat'i nazar va katta aniqlik bilan bajarish mexanizmini qanday yaratish mumkin?
12. Montaj stolini keyingi yoki oldingi kadrغا o'tkazish uchun qanday usullardan foydalanish mumkin?

Ushbu bobda Actionscript dasturlash tilining relyatsion operatorlari, Actionscriptda oddiy va murakkab mantiqiy ifodalar qanday tuzilganligi, if-else va switch-case konstruksiyalarining tarmoqlanuvchi operatorlarini ko'rib chiqilgan. ActionScript-da harakatlar to'plamini bir necha marta takrorlash tsikllar yordamida tashkil etiladi. Skriptda bir xil harakatni (yoki harakatlar to'plamini) ko'p marta yozish o'rniga, uni bir marta yozishimiz mumkin, lekin uni kerakli takroriy sonni ko'rsatgan holda tsikl bayonotiga qo'yilishi kerakligi aytiladi. Bulardan tashqari ActionScript-da tsikllarning uchta turi mavjudligi va ulardan qanday foydalanish kerakligi ko'rsatib o'tilgan.

Ma'lumki, har bir harakat uchun reaksiya mavjud. Bu esa tabiatda atrofimiz va o'zimiz bilan har doim sodir bo'ladi. Biz natija yoki javob olish uchun qandaydir tugmachalarni bosamiz. Macromedia Flash da bu haqda shunday tushuntiriladi: har bir voqea harakatga ega.

Bobning davomida video va videolarning (flesh-filmlar) asosiy xossalari va ularning sozlamalari, shuningdek ularni dasturiy jihatdan ishlatish yoki o'zgartirish bilan tanishtiriladi. Matn maydonlarining turlarini va matn maydonlarini yaratishni, shuningdek, if, if ... else iboralari yordamida ularga kiritilgan ma'lumotlarni qayta ishlashni ko'rib chiqiladi.

Videolarda boshqa holatlarda shartli gaplardan foydalanish haqida ham aytilgan. Macromedia Flash-da yaratilgan film nafaqat foydalanuvchi harakatlariga (sichqonchaning harakati, tugmani bosish va h.k.) munosabat bildira oladi. Kinodagi voqealar ma'lum vaqtdan keyin yoki ma'lum miqdordagi kadrlardan keyin sodir bo'lishi mumkin - ya'ni foydalanuvchidan qat'i nazar. Bunday hodisalarga asoslangan elementlarni interaktiv elementlar bilan birlashtirib, siz yanada dinamik va qiziqarli dizaynlarni yaratishingiz mumkin. Ushbu bobda siz Flash-da vaqt va kadrlarni hisoblash qanday ishlashini bilib olasiz.



## Falsh texnologiyasida foydalaniladigan atamalar

A ActiveX ASP	B Ban list Black list	C CGI CSS CTR	D DHTML DNS Doorway page
F Flash FTP	G Gateway	H HTML HTTP	I IMAP
J Java JavaScript	P Pagerank Pay per click PERL PHP POP3	S SEO SERP SGML Site frequency SMTP Stemming	T Type-in traffic
U URL Usability	V VBS Visual Basic	W WAP W3C	X XM

## ADABIYOTLAR RO'YXATI

1. Грибов Д.Ye. Macromedia Flash 4. Интерактивная web-анимация. - М. ДМК. 2000. - 672 с.
2. Франклин Д., Паттон Б. Flash 4. Анимация в интернете. - СПб. Символ Плюс. 2000.464 с.
3. Милберн Кен, Крото Джон. Внутренний мир Flash 5 для дизайнеров. К: Днасфт. 2000. - 496 с.
4. Шпенников Д.Г. Создание образовательных ресурсов на основе использования технологий Macromedia FLASH. Санкт-Петербургский Государственный университет информационных технологий Механики и оптики. 2003. -11 с.
5. Слепченко Ксения. Самоучитель Adobe Flash CS3 2000. - 544 с.
6. Дронов В. Macromedia Flash MX. - СПб.: БХВ-Петербург, 2003. -848 с.:
7. Роберт Рейнхардт и Джой Лотт. Macromedia Flash MX ActionScript - Библия пользователя.- Диалектика, 2003.-1223 с.
8. Прохоров А. Изучаем Macromedia Flash [Электронный ресурс] // КомпьютерПресс. Уроки. URL: <https://compress.ru/article.aspx?id=12617> (дата обращения: 15.03.2022).
9. Изучаем Macromedia Flash. Уроки программирование [Электронный ресурс]. URL:<https://sangimarket.ru/iron/macromedia-flash-8-uroki-programmirovaniie-izuchaemmacromedia-flash-gladkaya-oropnaya/> (дата обращения: 20.01.2022).
10. Изучаем Macromedia Flash. Автоматизированное написание скрипта [Электронный ресурс]. URL: <https://cammeteo.ru/makromedia-flesh-8-uroki-po-animacii-dlyanachinayushchih-izuchaem-macromedia-flash/> (дата обращения: 20.01.2022).

## **“DASTURLASH III” fanidan o‘quv qo‘llanma**

**(O‘quv qo‘llanma Audiovizual texnologiyalar, telestudiya tizimlari va ilovalari yo‘nalishi bo‘yicha ta‘lim olayotgan talabalar uchun mo‘ljallangan.)**

**Bosishga ruxsat etildi. 24.05.2022 y.**

**Qog‘oz bichimi 60-84 <sup>1</sup>/<sub>16</sub>.**

**Times New Roman garniturasida terildi.**

**Ofset uslubida oq qog‘ozda chop etildi. Shartli bosma tabog‘i 11,3.**

**Buyurtma № 24-03. Adadi 100 nusxa.**

**“ZUXRA BARAKA BIZNES” MChJ bosmaxonasida chop etildi.**

**Toshkent shahri, Bunyodkor ko‘chasi 27 A-uy.**