Лекция # 2

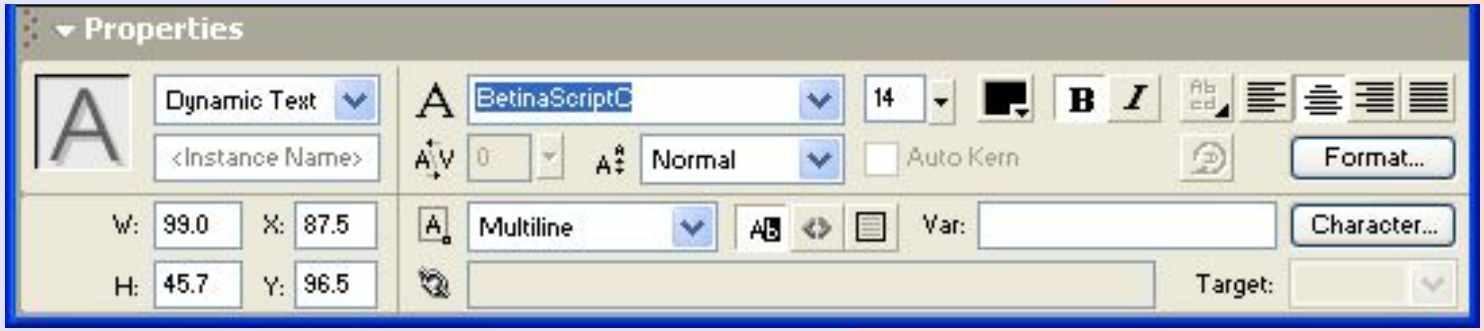
Основы анимации на основе Flash-технологий

Динамический текст

Single Line (Однострочное);

Multiline (Многострочное);

Multiline no wrap (Многострочное без переносов);

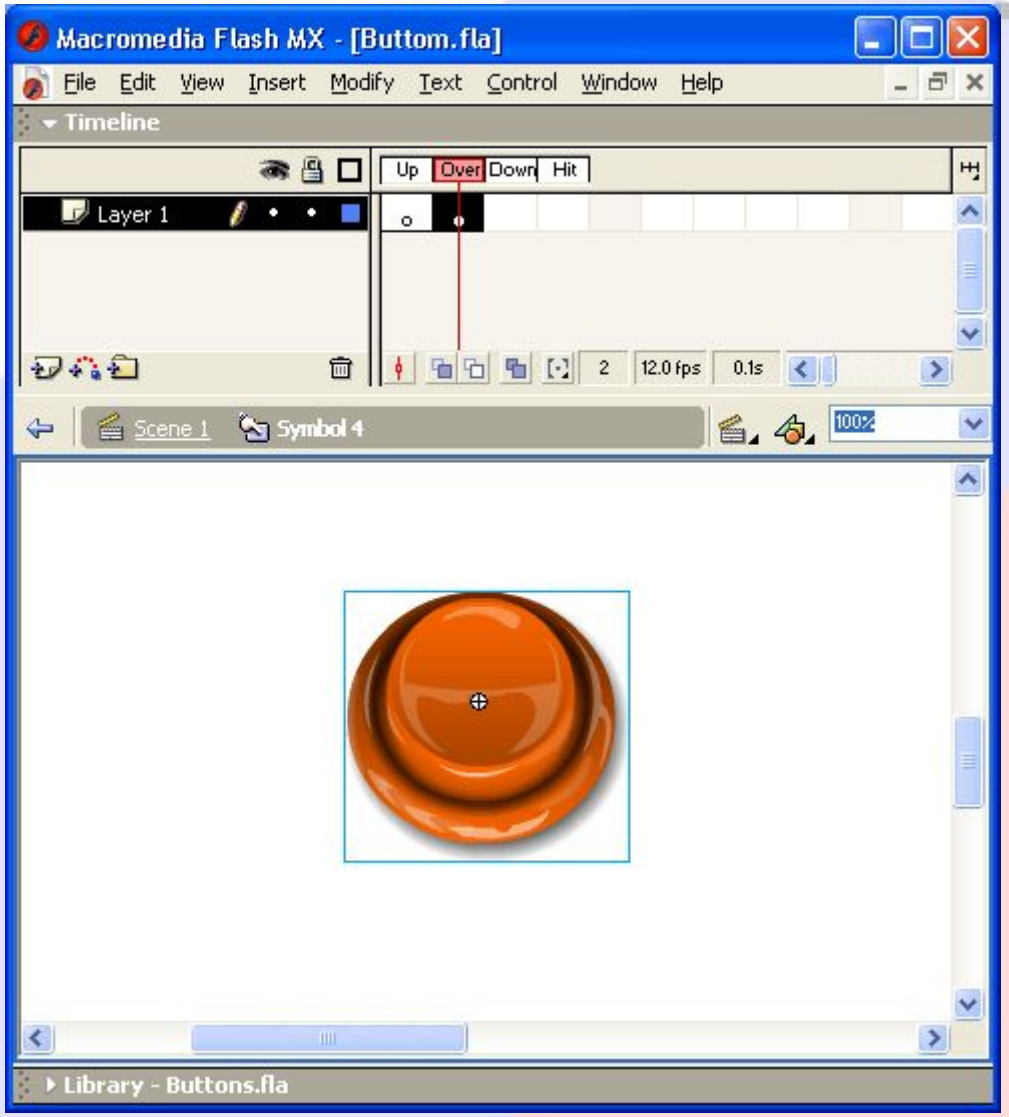


Типы символов

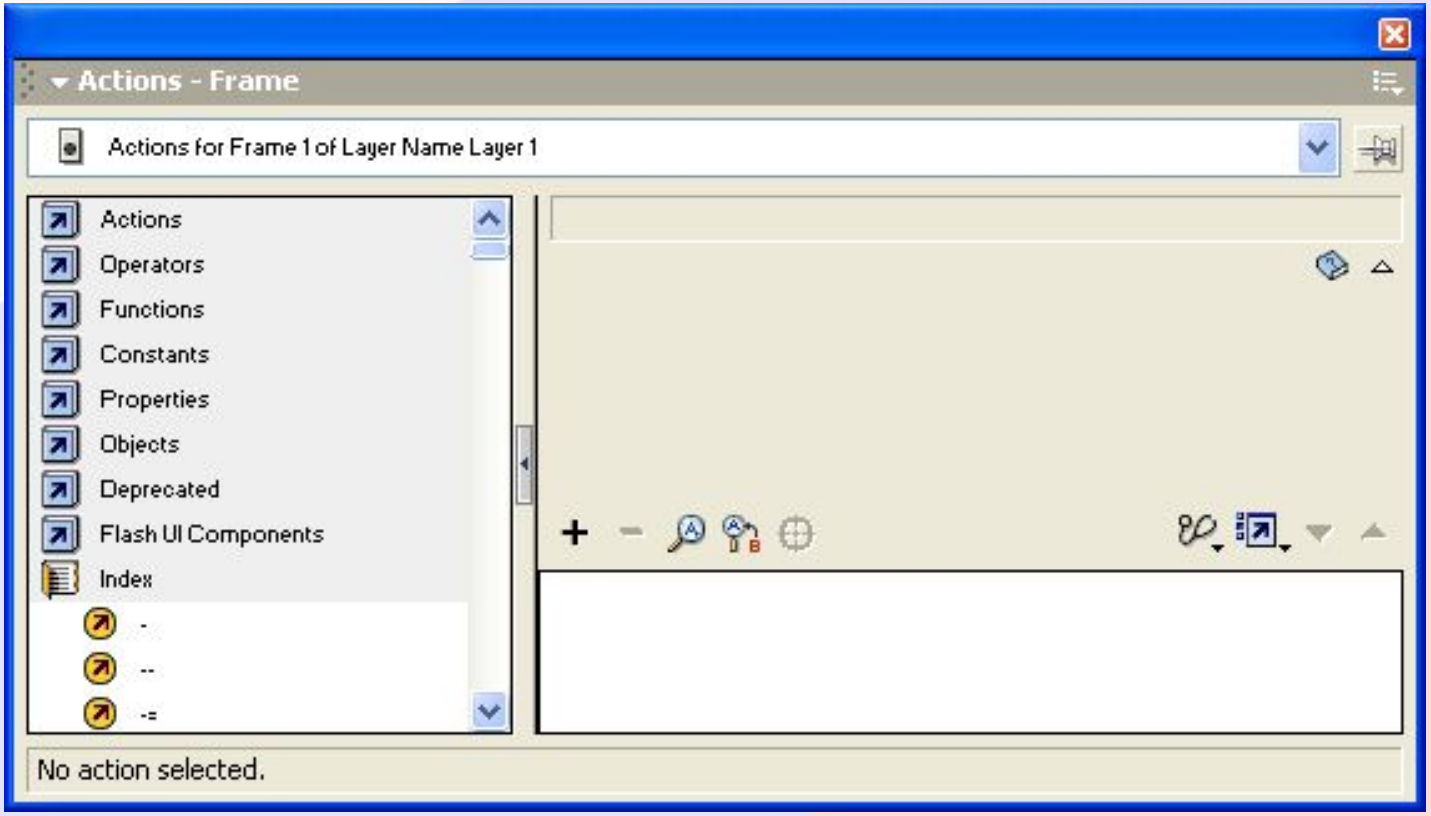
Графический символ (*Graphic symbol*)
Символ-кнопка (*Button symbol*)
Символ-клип (*Movie clip symbol*)

Преобразование в символ существующего объекта

Создание кнопок



Панель Actions



Классы

Movie (Объекты фильма);

Core (Объекты ядра);

Client/Server (Объекты клиент-серверной среды исполнения);

Authoring (Объекты среды разработки).

Объекты класса *Movie*

Button (Кнопка)
MovieClip (Клип)
Mouse (Мышь
Key (Клавиатура
Color (Цвет)
Sound (Звук)
Stage (Стол)
TextField (Текстовое поле)
TextFormat (Формат текста)

Объекты класса *Core*

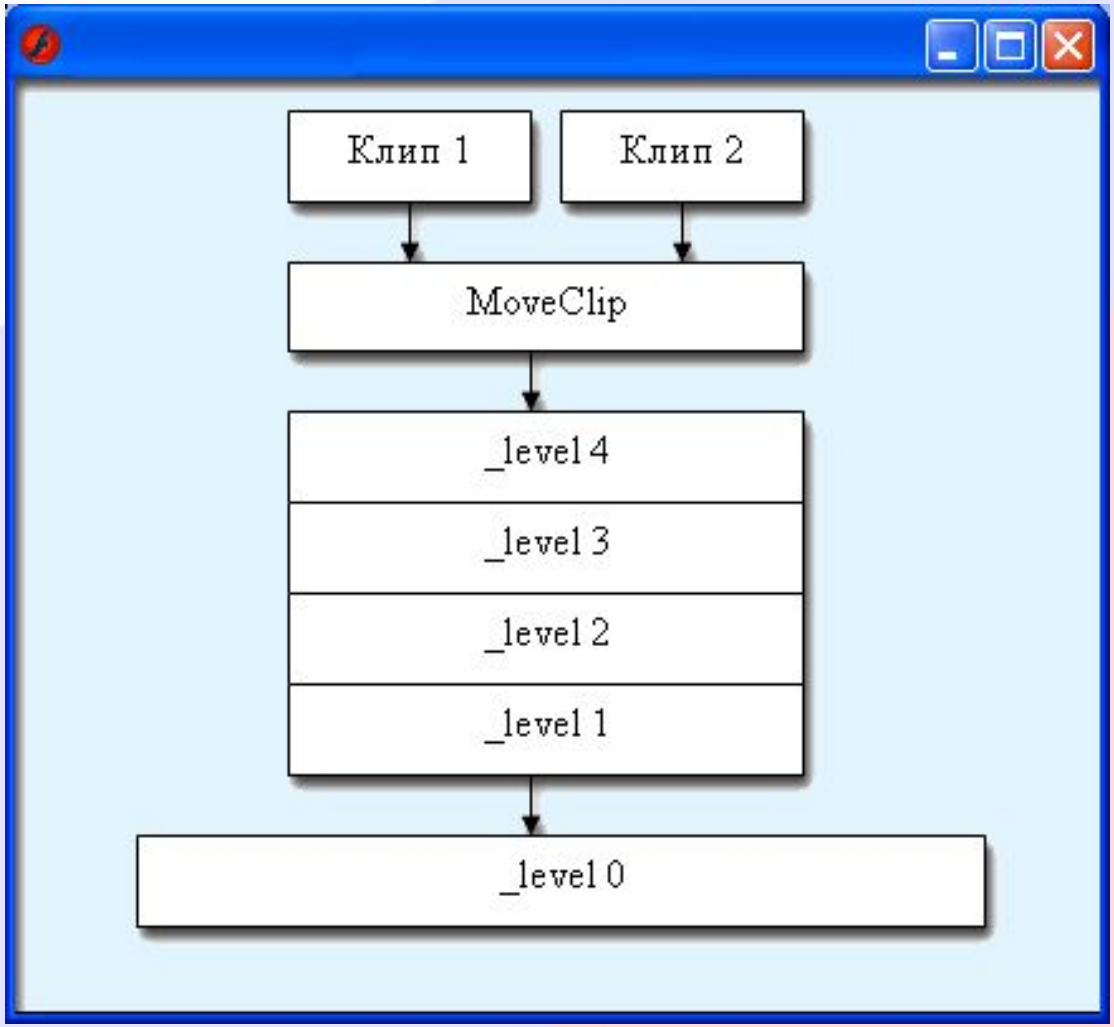
Math (Математика)

Number (Число)

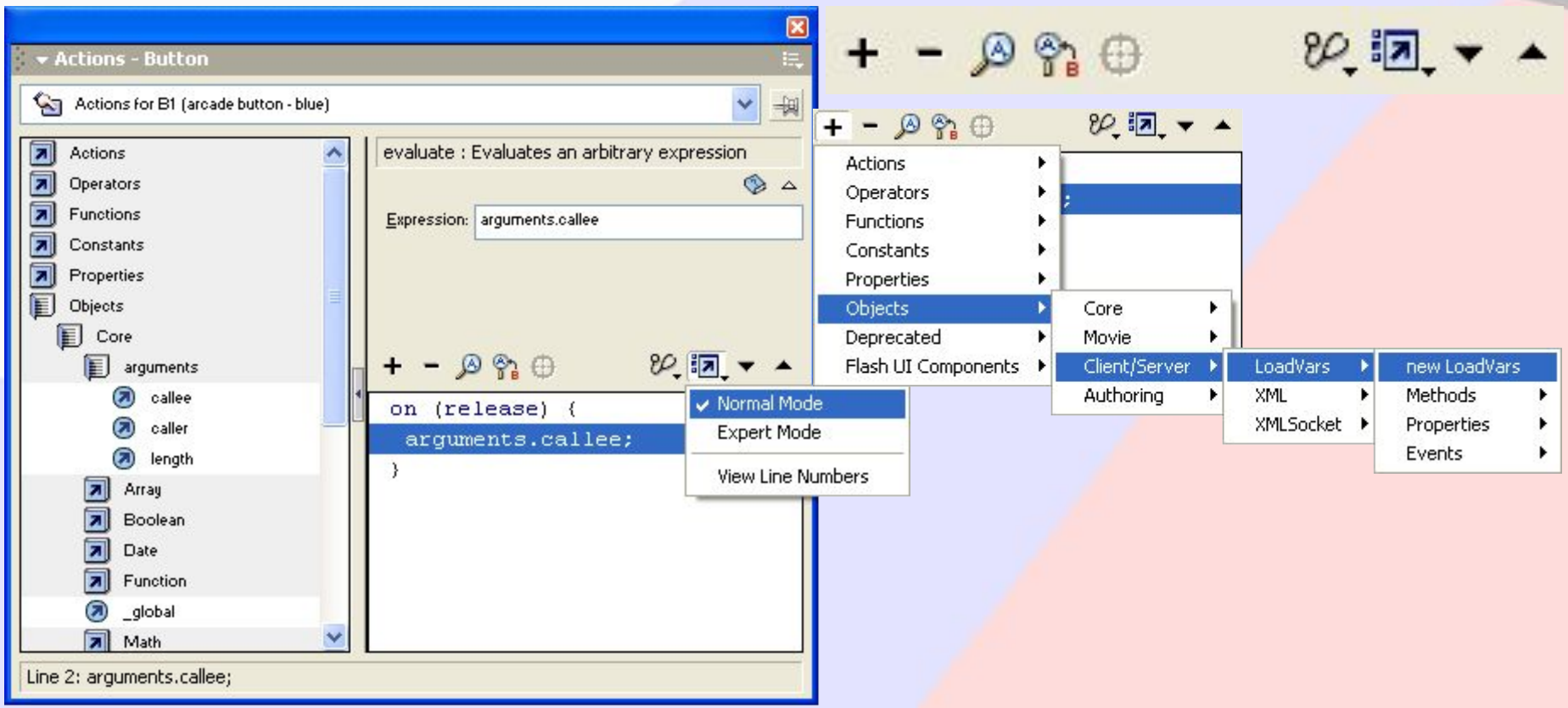
String (Строка)

Date (Дата)

Иерархия временных диаграмм, загруженных в Flash-плеер



Режим Normal Mode



Go To

Play и *Stop*

Toggle High Quality

Stop All Sounds

Get URL

FSCommand

Load Movie и *Unload Movie*

Tell Target

Типы событий для кнопки

Press

Release

Release Outsid

Key Press

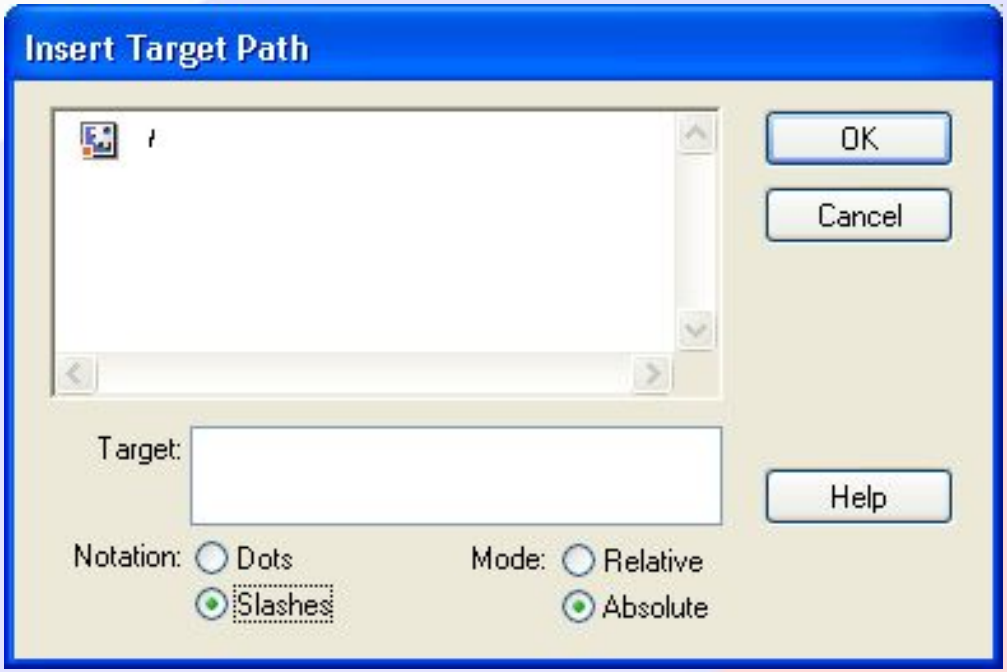
Roll Over

Roll Out

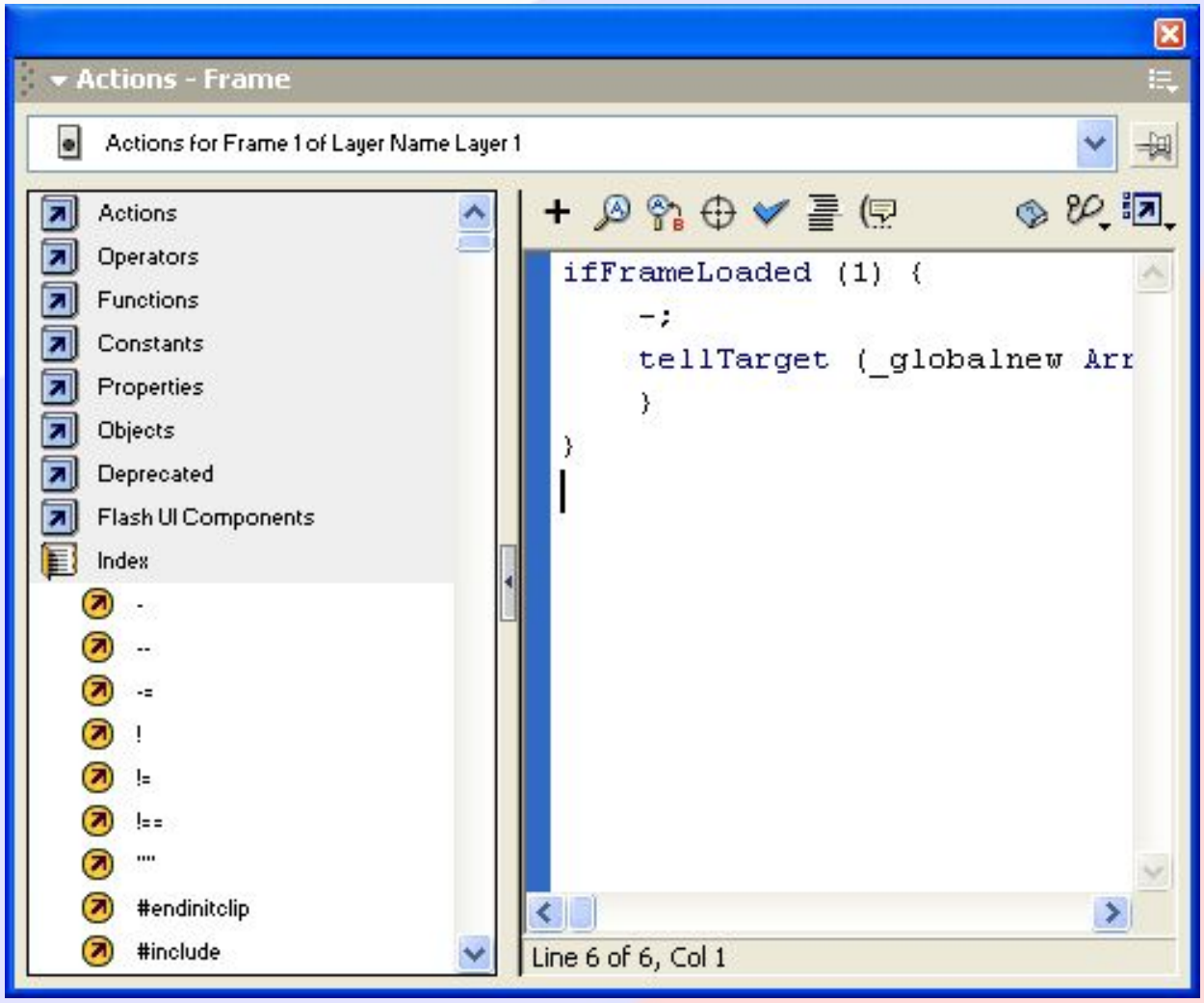
Drag Over

Drag Out

Формат окна Insert a target path



Режим Expert Mode



Переход по заданному URL

getURL(...)
url,
_self
_ blank
_parent
_top ,
Don't send (Не отправлять)
Send using GET (Отправить, используя GET)
Send using POST (Отправить, используя POST)

Основные действия с Movie Clip и свойства MC

play()
stop()
gotoAndPlay()
gotoAndStop() :
_x, _y
_xscale, _yscale
_width, _height
_rotation
_alpha
_visible

Пути

this
_parent
_root

leaf.play();
_parent.tree.leaf.stop();
_root.banner._visible = false;

GetProperty()

```
on (release) {
    _root.xpos = getProperty ( _target, _x );
    _root.ypos = getProperty ( _target, _y );
    _root.xscale = getProperty ( _target, _xscale );
    _root.yscale = getProperty ( _target, _yscale );
    _root.alpha = getProperty ( _target, _alpha );
    _root.rotation = getProperty ( _target, _rotation );
}
```

```
on (release) {
    _root.xpos = this._x;
    _root.ypos = this._y
    _root.xscale = this._xscale;
    _root.yscale = this._yscale;
    _root.alpha = this._alpha;
    _root.rotation = this._rotation;
}
```

SetProperty()

```
on (release) {
    setProperty ("_level0.mouse", _x, xpos);
}

on (release) {
    setProperty ("_level0.mouse", _y, ypos);
}
```

```
on (release) {
setProperty ("_level0.mouse", _xscale, xscale);
}

on (release) {
setProperty ("_level0.mouse", _yscale, yscale);
}
```

```
on (release) {
setProperty ("_level0.mouse", _alpha, alpha);
}

on (release) {
setProperty ("_level0.mouse", _rotation, rotation);
}
```

```
on (release) {
setProperty ("_level0.mouse", _visibility, visibility);
}
```

```
setProperty ("_level0.mouse", _visibility, visibility);
_root.mouse._visibility = visibility;
_level0.mouse._visibility = visibility;
```

duplicateMovieClip()

```
on (release) {
duplicateMovieClip ("box", "box2", 1);
}
```

```
on (release) {
duplicateMovieClip("box", "box2", 1);
setProperty ("box2", _x, "200");
}

on (release) {
duplicateMovieClip("box", "box2", 1);
_root.box2._x=200;
}
```

```
on (release) {
count = 1;
while (count<20) {
_root.box.duplicateMovieClip("boxx"+count, count);
_root["boxx"+count]._x = random(550);
_root["boxx"+count]._y = random(150);
_root["boxx"+count]._xscale = random(150);
_root["boxx"+count]._yscale = random(150);
_root["boxx"+count]._alpha = random(100);
count += 1;
}
}
```

Создание меню через MC.nextFrame()

```
on (release) {
  if (!a) {
    _root.menu2.nextFrame();
  } else {
    _root.menu2.prevFrame();
  }
a=1-a;
}
```