

μμ μ

:

ActionScript 3.0 (II)



μ μ

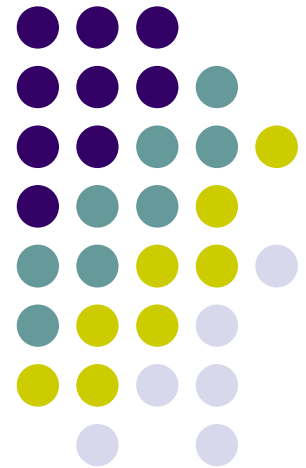
μ

μ

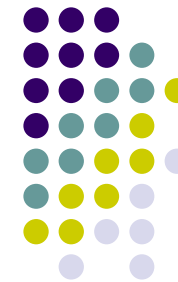
&

.

μ

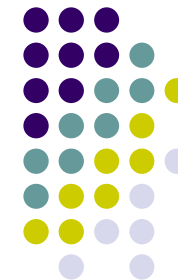


(functions)



- ActionScript 3.0 μ μ
 - (function statement)
`function traceParameter(aParam:String)`
`{`
`trace(aParam);`
`}`
`traceParameter("hello"); // hello`
 - (function expression)
`var traceParameter:Function = function (aParam:String)`
`{`
`trace(aParam);`
`};`
`traceParameter("hello"); // hello`

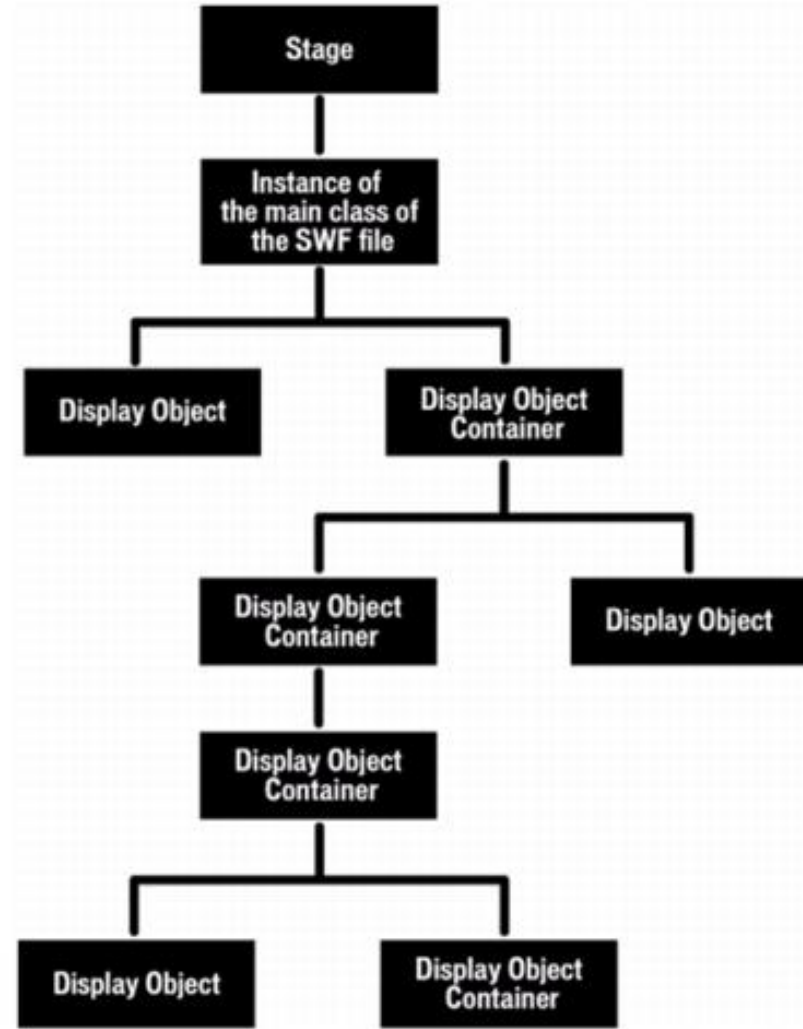
(functions)



- μ
function doubleNum(baseNum:int):int {
 return (baseNum * 2);
 trace("after return"); // This trace statement will not be executed.
}
- μ
function getNameAndVersion():String
{
 function getVersion():String
 {
 return "9";
 }
 function getProductName():String
 {
 return "Flash Player";
 }
 return (getProductName() + " " + getVersion());
}
trace(getNameAndVersion()); // Flash Player 9

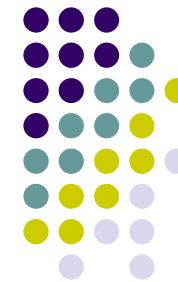


μ



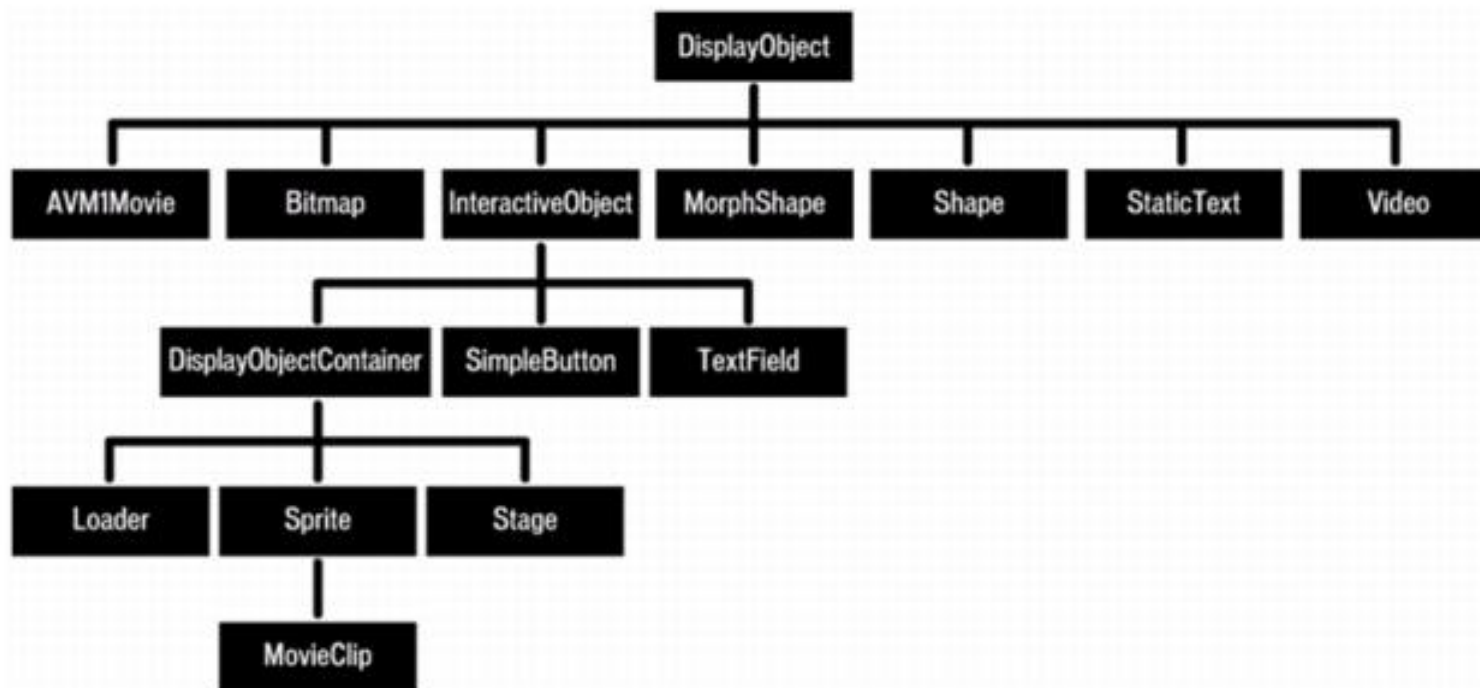
•
μ
μ
μ

μμ μ



μ

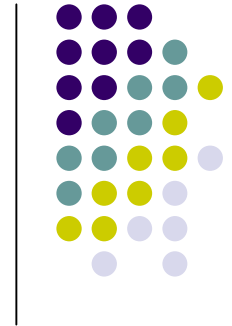
ActionScript 3.0



μμ μ

μ

μ



- μ μ μ μ (. . μ)
- μ μ display list
- (child) μ μ .
- μ :

(Stage)

```
import flash.display.*;
import flash.text.TextField;
var myText:TextField = new TextField();
myText.text = "Welcome to class";
this.addChild(myText);
```

- μ μ μ μ μ Flash
- μ μ ActionScript μ μ μ μ

```
myText.text = "Hello from Real Player";
myText.x=100;
```

μμ μ

μ

μ



•

μ

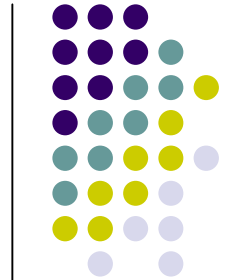
μ

```
tf1:TextField = new TextField();  
tf2:TextField = new TextField();  
tf1.name = "Welcome to";  
tf2.name = "UOP";  
  
container1:Sprite = new Sprite();  
container2:Sprite = new Sprite();  
  
container1.addChild(tf1);  
container1.addChild(tf2);  
container1.removeChild(tf1);  
container2.addChild(tf1);  
  
trace(container1.numChildren); // 1  
trace(container1.getChildAt(0).name); // UOP  
trace(container2.numChildren); // 1  
trace(container2.getChildAt(0).name); // Welcome to
```

μμ μ

μ

animation



- `μ : μ Sprite μ circle , μ (μ alpha) μ .`

```
import flash.display.Sprite;
import flash.events.Event;
import flash.events.MouseEvent;
```

```
// draw a circle and add it to the display list
var circle:Sprite = new Sprite();
circle.graphics.beginFill(0x990000);
circle.graphics.drawCircle(50, 50, 50);
circle.graphics.endFill();
addChild(circle);
```

```
// When this animation starts, this function is called every frame.
// The change made by this function (updated to the screen every
// frame) is what causes the animation to occur.
```



μ animation

```
function fadeCircle(event:Event):void
{
    circle.alpha -= .05;

    if (circle.alpha <= 0)
    {
        circle.removeEventListener(Event.ENTER_FRAME, fadeCircle);
    }
}

function startAnimation(event:MouseEvent):void
{
    circle.addEventListener(Event.ENTER_FRAME, fadeCircle);
}

circle.addEventListener(MouseEvent.CLICK, startAnimation);
```

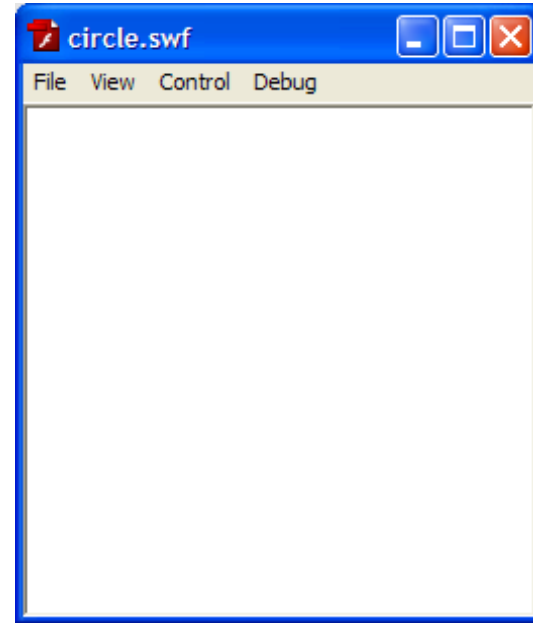
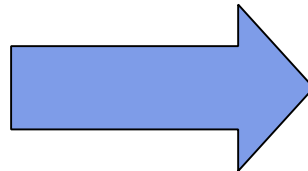
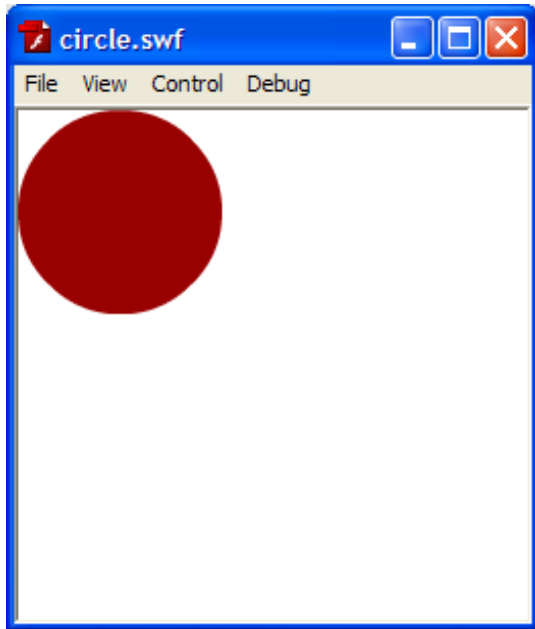


μ

animation



μ



μμ μ

ActionScript



- actionscript
 - μ «Actions»
 - .as
 - (μ μ) μ ActionScript
- (import)
 - μ μ **include**
include "my_file.as";

ActionScript



- μ
- μ μ Flash
- μ TextField (Dynamic Text)
- μ exfile.as
- `var myText:String = "Welcome to ";`
- Actions
- `include "exfile.as"`
- `myTextField.text = myText + "Web Programming";`

Welcome to Web Programming



μ

- ActionsScript μ μ μ μ μ μ μ
- μ μ μ ActionScript
- 1. μ Flash Flex Builder μ μ (. .
- Deramweaver) μ μ (. . notepad)
- 2. μ μ μ public
- μ public class Car
- {
- }



μ

• 3. μ μ μ **ActionScript**

3.

```
μ package  
μ {  
package mypackage  
{  
    public class Car  
    {  
    }  
}  
}
```

4.

```
μ μ μ var  
μ
```

- public: μ
- private: μ

```
public var plate:String = "AB1345";  
public var numSeats:Number = 4;  
public var dateConstruction:Date;
```

μμ μ



μ

μ

- μ μ (instances) μ
var toyota:Car = new Car();
var mazda:Car = new Car();

- μ



```
public function Car (plateStr:String, dateCon:Number)
{
    plate = plateStr;
    dateConstruction = dateCon;
}
```



```
μ μ
var toyota:Car = new Car(" 1345",new Date(2008, 3, 16));
var mazda:Car = new Car("NI5342",new Date(2009, 1, 26));
```

μ

μ

—



μ

μ

μ

μ

- **TextWithImage.as**

```
package  
{  
    import flash.display.Sprite;  
    import flash.text.*;  
  
    public class TextWithImage extends Sprite  
    {  
        private var myTextBox:TextField = new TextField();  
        private var myText:String = "Hello from Web";  
  
        public function TextWithImage()  
        {  
            addChild(myTextBox);  
            myTextBox.text = myText;  
        }  
    }  
}
```

μμ μ

μ

μ

-

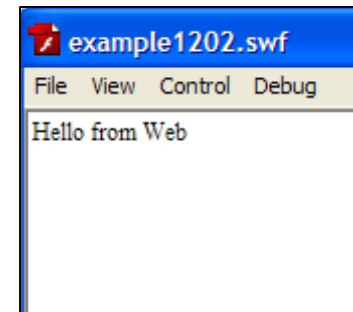
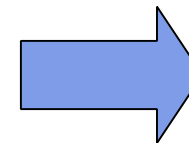
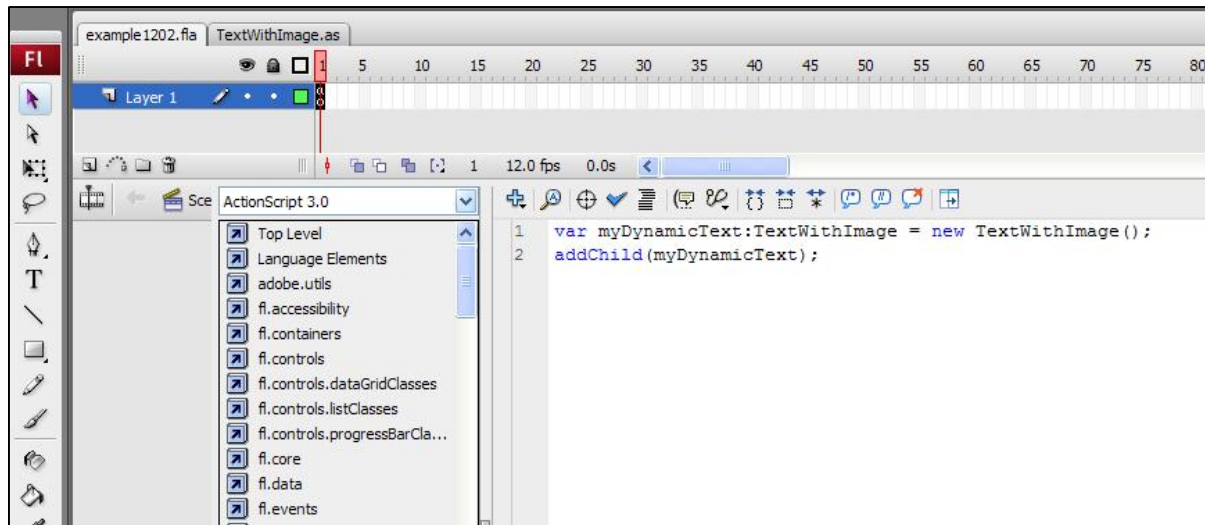


μ



Actions

```
var myDynamicText:TextWithImage = new TextWithImage();  
addChild(myDynamicText);
```



μ μ

Drop

Drag-and-



- drag-and-drop

```
instanceName.addEventListener(MouseEvent.CLICK, mouseDown)
function mouseDown(event:MouseEvent):void
{
    instanceName.startDrag();
    //rest of the code
}
```

```
instanceName.addEventListener(MouseEvent.CLICK, mouseReleased);
function mouseReleased(event:MouseEvent):void
{
    instanceName.stopDrag();
    //rest of the code
}
```

Drag-and-



Drop

- μ

```
import flash.display.Sprite;  
import flash.events.MouseEvent;
```

```
var circle:Sprite = new Sprite();  
circle.graphics.beginFill(0xFFCC00);  
circle.graphics.drawCircle(40, 40, 40);
```

```
var target1:Sprite = new Sprite();  
target1.graphics.beginFill(0xCCFF00);  
target1.graphics.drawRect(0, 0, 100, 100);  
target1.name = "Upper Sector";
```

```
var target2:Sprite = new Sprite();  
target2.graphics.beginFill(0xCCFF00);  
target2.graphics.drawRect(0, 200, 100, 100);  
target2.name = "Lower Sector";
```



Drag-and-

Drop

```
addChild(target1);  
addChild(target2);  
addChild(circle);
```

```
circle.addEventListener(MouseEvent.CLICK, mouseDown)
```

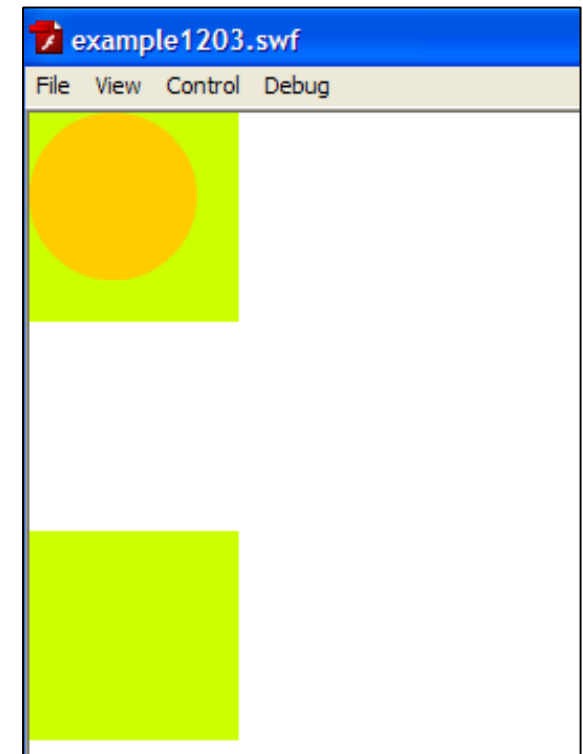
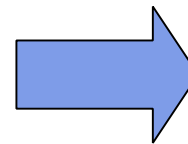
```
function mouseDown(event:MouseEvent):void
```

```
{  
    circle.startDrag();  
}
```

```
circle.addEventListener(MouseEvent.CLICK, mouseReleased);
```

```
function mouseReleased(event:MouseEvent):void
```

```
{  
    circle.stopDrag();  
    if (circle.dropTarget!=null)  
        trace(circle.dropTarget.name);  
    else  
        trace("White Sector");  
}
```





μ Point

- μ μ Point
- μ μ μ μ μ (x,y)
- μ μ Point

```
import flash.geom.*;  
var pt1:Point = new Point(30, 40); // x == 30; y == 40  
var pt2:Point = new Point();  
pt2.x = 30;  
pt2.y = 40;
```
- μ μ

```
import flash.geom.*;  
var pt1:Point = new Point(circle1.x, circle1.y);  
var pt2:Point = new Point(circle2.x, circle2.y);  
var distance:Number = Point.distance(pt1, pt2);
```



μ Point



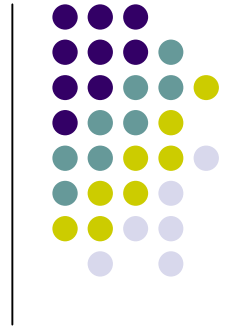
```
import flash.geom.*;
var stageCenter:Point = new Point();
stageCenter.x = this.stage.stageWidth / 2;
stageCenter.y = this.stage.stageHeight / 2;
var targetCenter:Point = new Point(target.x, target.y);
targetCenter = target.localToGlobal(targetCenter);
var distance:Number = Point.distance(stageCenter, targetCenter);
```



μ

```
import flash.geom.*;
var distance:Number = 100;
var angle:Number = 2 * Math.PI * (90 / 360);
var translatePoint:Point = Point.polar(distance, angle);
circle.x += translatePoint.x;
circle.y += translatePoint.y;
```


μ Rectangle



- μ Rectangle μ . μ Rectangle
- μ (x,y) ,
- μ μ Rectangle

```
var rx:Number = 0;
var ry:Number = 0;
var rwidth:Number = 100;
var rheight:Number = 50;
var rect1:Rectangle = new Rectangle(rx, ry, rwidth, rheight);
```
- ```
import flash.geom.Rectangle;
var x1:Number = 0;
var y1:Number = 0;
var width1:Number = 100;
var height1:Number = 50;
var rect1:Rectangle = new Rectangle(x1, y1, width1, height1);
trace(rect1) // (x=0, y=0, w=100, h=50)
rect1.x = 20;
rect1.y = 30;
trace(rect1); // (x=20, y=30, w=100, h=50)
```



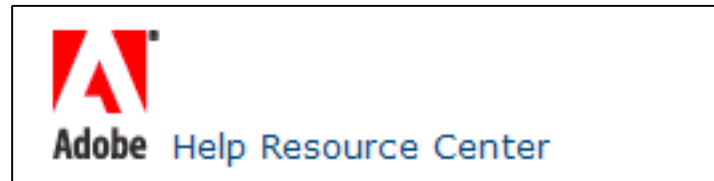
# μ Rectangle

- ```
import flash.geom.Rectangle;
var x1:Number = 0;
var y1:Number = 0;
var width1:Number = 100;
var height1:Number = 50;
var rect1:Rectangle = new Rectangle(x1, y1, width1, height1);
trace(rect1) // (x=0, y=0, w=100, h=50)
rect1.left = 20;
rect1.top = 30;
trace(rect1); // (x=20, y=30, w=80, h=20)
```

- ```
trace(rect1) // (x=0, y=0, w=100, h=50)
rect1.right = 60;
rect1.bottom = 20;
trace(rect1); // (x=0, y=0, w=60, h=20)
```



# ActionScript



[http://livedocs.adobe.com/flex/3/html/help.html?content=Part6\\_ProgAS\\_1.html](http://livedocs.adobe.com/flex/3/html/help.html?content=Part6_ProgAS_1.html)

**Flash Help and Support**

<http://www.adobe.com/support/flash/>