

μμ μ

:

ActionScript 3.0



μ μ

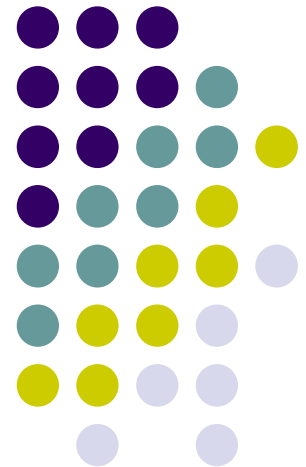
μ

μ

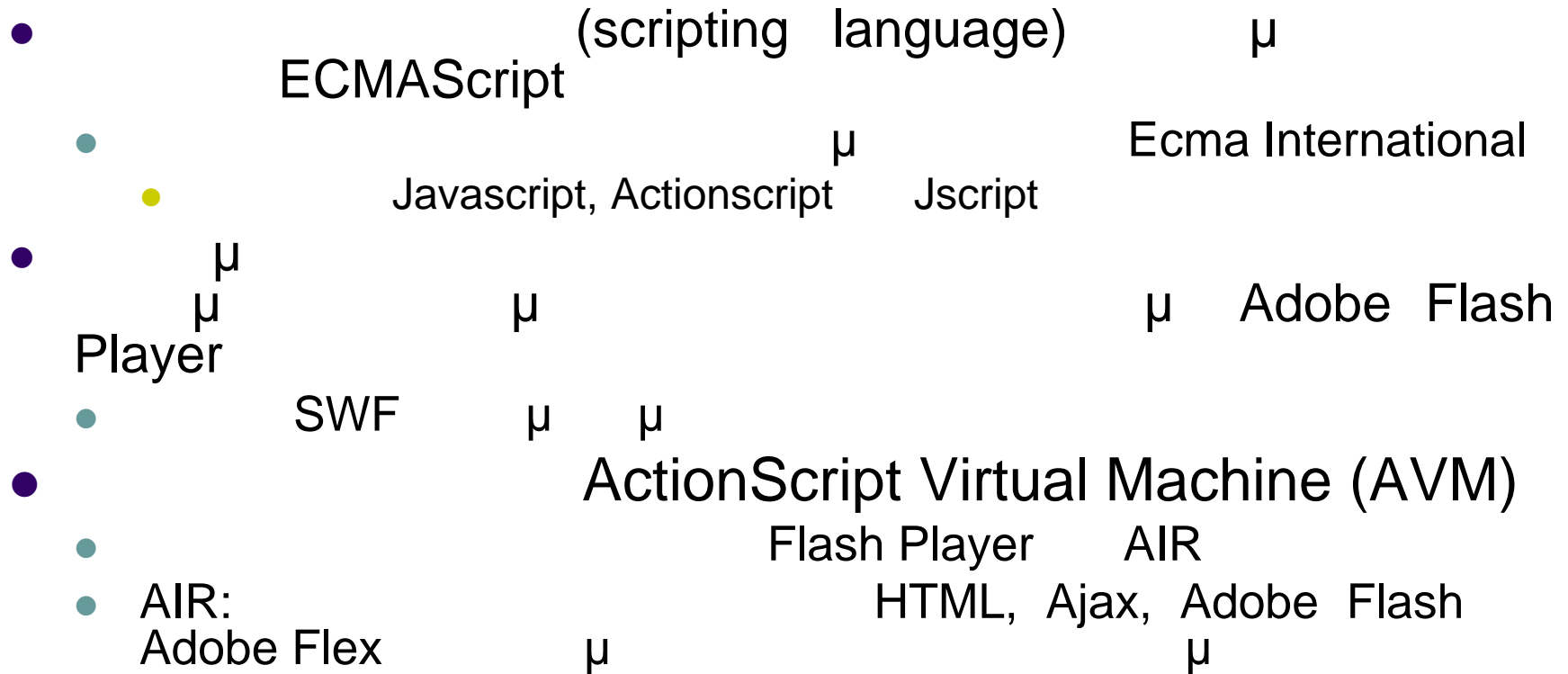
&

.

μ



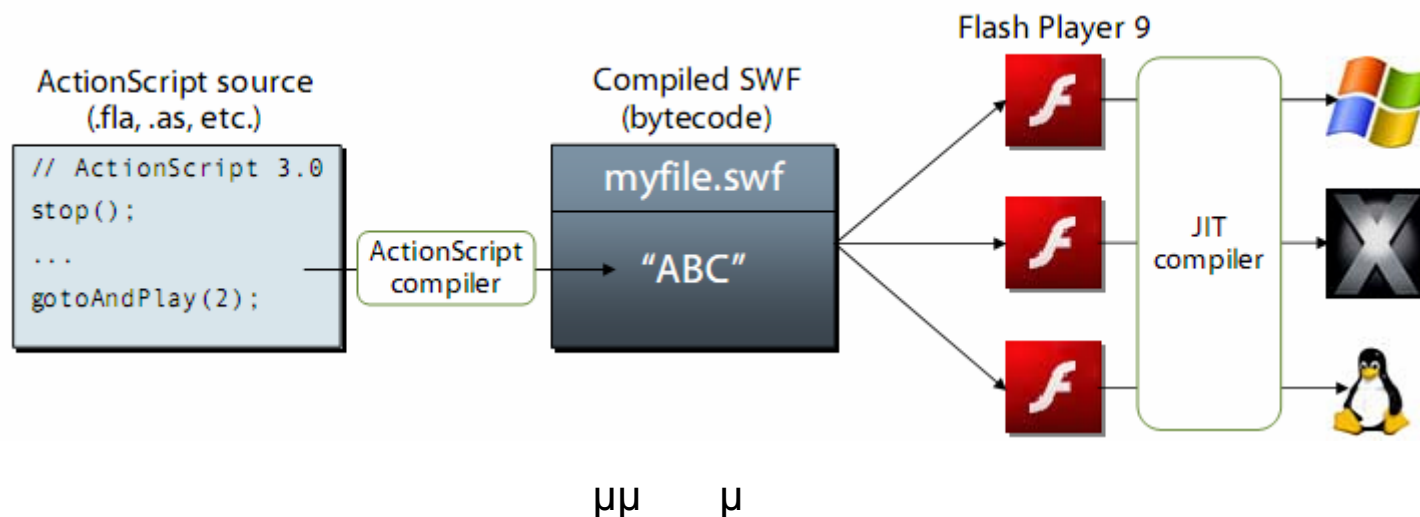
ActionScript



ActionScript



- ActionScript μ (bytecode)
 - (compiler) ActionScript μ
 - Adobe Flash, Adobe Flex Builder μ
 - Adobe Flex SDK μ
 - SWF files μ
 - Flash Player μ
 - ActionScript Virtual Machine μ



μ



- μ μ μ : Flash 4 (1999)
- ActionScript 1
 - Flash 5 (2000)
- ActionScript 2
 - Flash MX 2004 (Flash Player 7): (2003)
- ActionScript 3
 - Flex 2
 - Flash CS3 Professional: (2006-7)



μ

- ActionScript μ
- (Properties)
 - . . μ
- (Methods)
 - . . μ μ
- (Events)
 - . .

μ



- μ μ
- μ μ
- x μ μ 230

`square.x = 230;`

- μ square μ μ triangle

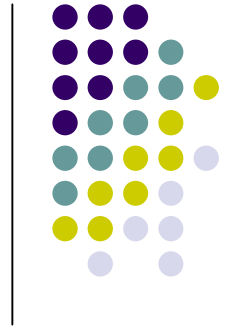
`square.rotation = triangle.rotation;`

- μ square μ μ

`square.scaleX = 2.5;`

μμ μ

μ



- μ
- μ
- μ MovieClip μ μ shortFilm
`shortFilm.play();`
- μ MovieClip μ μ shortFilm μ (
)
- μ MovieClip μ μ shortFilm μ
(Frame) 1 μ .
`shortFilm.gotoAndStop(1);`
- μ μ
`var numericData:Number = 9;`
`var textData:String = numericData.toString();`
- μ μ μ (μ calculatorDisplay
TextField)
- μ μ
`var numericData:Number = 9;`
`var textData:String = numericData.toString();`

ActionScript



- events
ActionScript
- events
- ..
- (Adobe Flash Player)
- ActionScript

ActionScript



ActionScript



μ

:

μ

μ



..

μ



:



μ

μ

μ



..

,

μ

μ



(response)

:

μ

μ

μ



..

μ

μ

μμ

μ

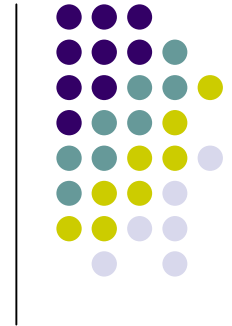
ActionScript



- μ
 - : eventSource
 - : eventObject:EventType
 - (response) : {.....}

```
function eventResponse(eventObject:EventType):void
{
    // Actions performed in response to the event go here.
}
eventSource.addEventListener(EventType.EVENT_NAME,
    eventResponse);
```

μ



clip

- μ μ (playButton)
`this.stop();`
`function playMovie(event:MouseEvent):void`
`{`
`this.play();`
`}`

`playButton.addEventListener(MouseEvent.CLICK, playMovie);`

- μ μ μ (entryText) μ
μ μ (outputText) μ
`function updateOutput(event:TextEvent):void`
`{`
`var pressedKey:String = event.text;`
`outputText.text = "You typed: " + pressedKey;`
`}`

`entryText.addEventListener(TextEvent.TEXT_INPUT, updateOutput);`

μμ μ

μ



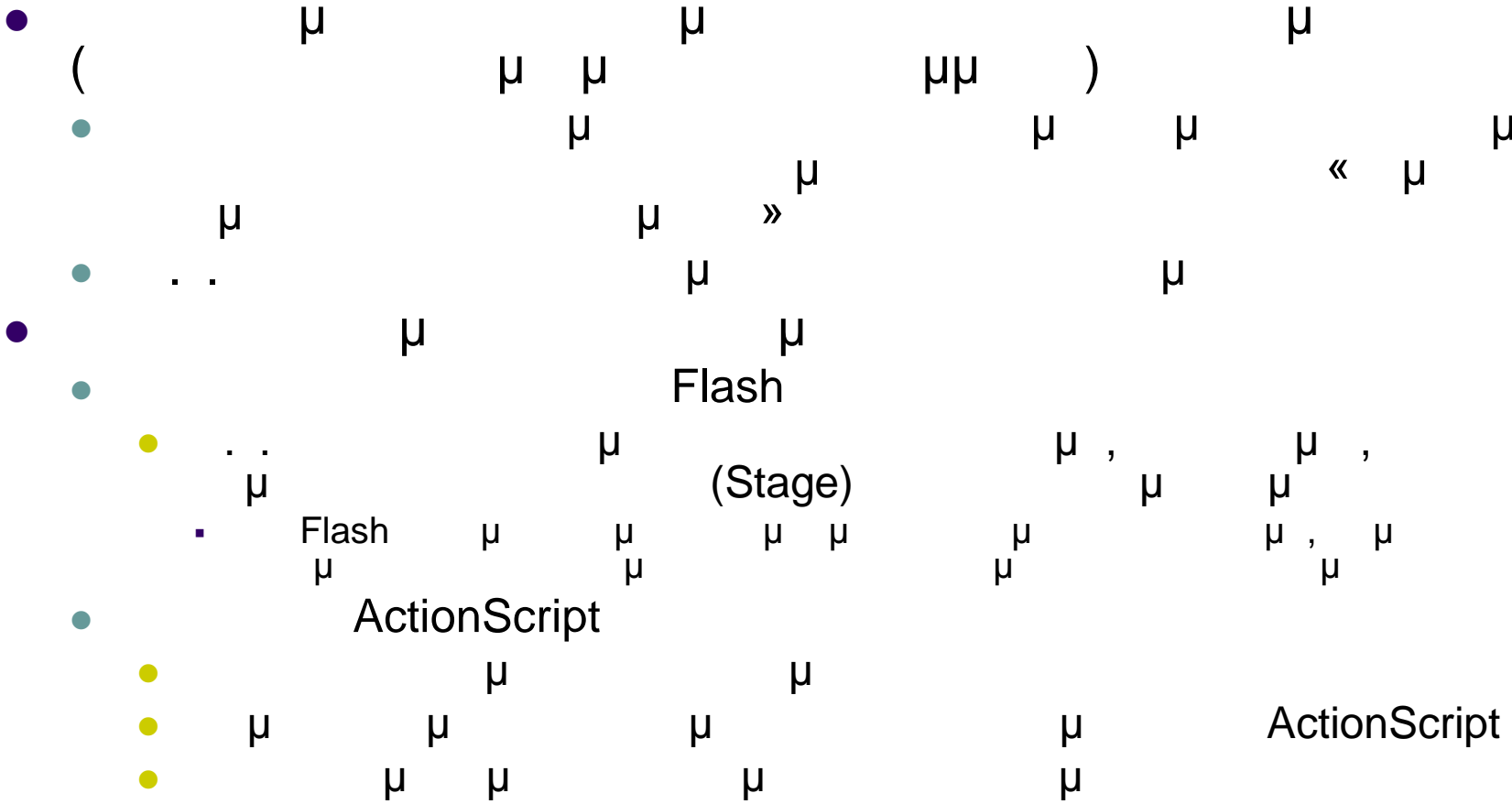
- ```
function gotoUOPSite(event:MouseEvent):void
{
 var adobeURL:URLRequest = new
 URLRequest("http://www.uop.gr/");
 navigateToURL(adobeURL);
}

linkButton.addEventListener(MouseEvent.CLICK, gotoUOPSite);
```



μ

μ





μ

μ

• μ μ μ ActionScript

• μ μ μ μ

```
var firstNumber:Number = 23.14;
var myNegativeInteger:int = -23;
var secondUint:uint = 13;
var firstName:String = "The tallest house";
var isReady:Boolean = true;
var student:XML = <student>
 <firstName>Nick</firstName>
 <lastName>Papas</lastName>
</student>;
```

• μ μ «new», μ μ

```
var raceCar:MovieClip = new MovieClip();
var birthday>Date = new Date(2001, 5, 6);
```

μμ μ



- $\mu$ 
  - $\mu$ 

```
var i:int;
i = 20;
```
  - $\mu$ 

```
var i:int = 20;
```
  - $\mu$ 

```
var numArray:Array = ["zero", "one", "two"];
var customItem:CustomClass = new CustomClass();
var a:int, b:int, c:int;
var d:int = 10, e:int = 20, f:int = 30;
```





μ

μ

- μ μ μ μ
- μ μ μ
- (global): μ μ μ μ μ μ μ μ
- (local): μ μ μ μ μ μ μ μ
- ‘ μ

```

var strGlobal:String = "Global";
function scopeTest()
{
 trace(strGlobal); // Global
}
scopeTest();
trace(strGlobal); // Global

```



μ

μ

μ

μ

:

```
function localScope()
{
 var strLocal:String = "local";
}
localScope();
trace(strLocal); // error because strLocal is not defined globally
```

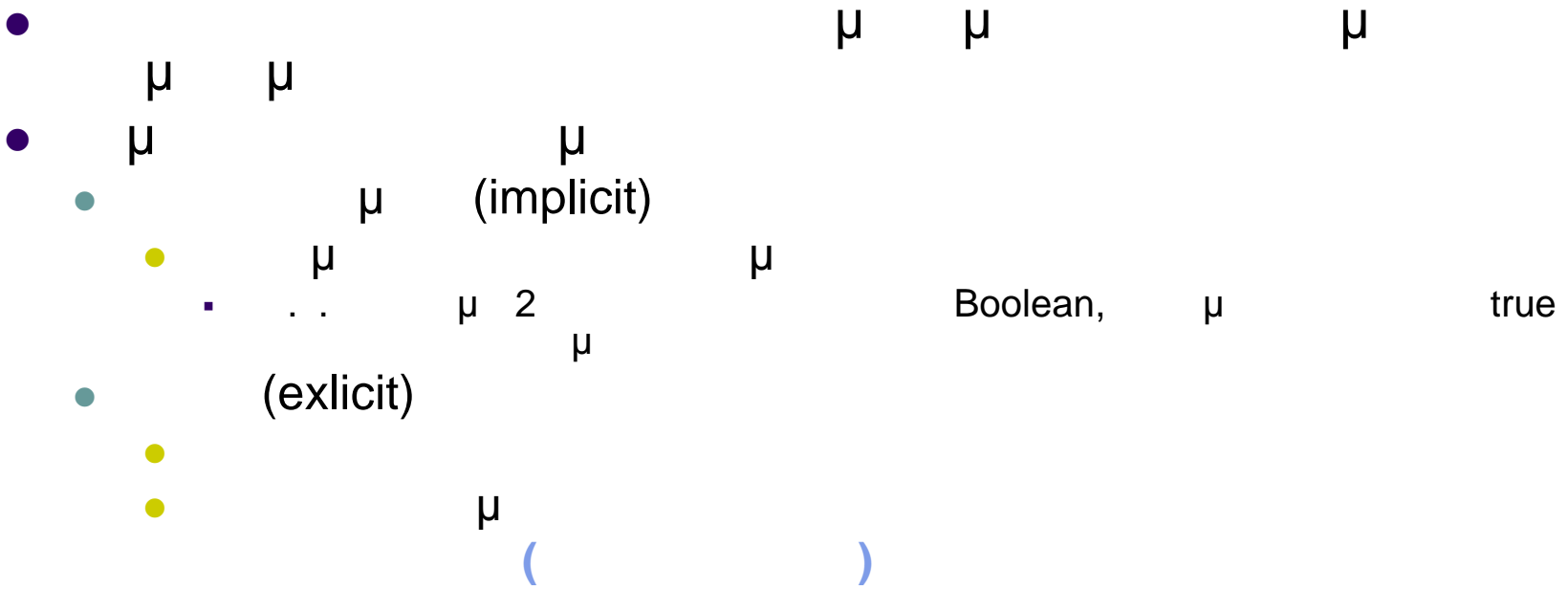
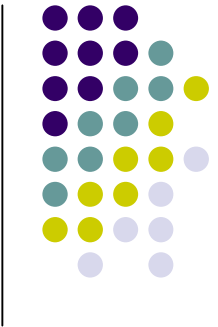
μ

μ

μ

:

```
var str1:String = "Global";
function scopeTest ()
{
 var str1:String = "Local";
 trace(str1); // Local
}
scopeTest();
trace(str1); // Global
```



```

μ Explicit μ :
var myBoolean:Boolean = true;
var myINT:int = int(myBoolean);
trace(myINT); // 1

```



- $\mu$ 
  - $\mu$ 

```
var quantityField:String = "3";
var quantity:int = quantityField; // compile time error
```
  - $\mu$ 

```
var quantityField:String = "3";
var quantity:int = int(quantityField); // Explicit conversion succeeds.
```
- $\mu \quad \mu$ 

```
var myBoolean:Boolean = true;
var myUINT:uint = uint(myBoolean);
var myINT:int = int(myBoolean);
var myNum:Number = Number(myBoolean);
trace(myUINT, myINT, myNum); // 1 1 1
trace(uint("5")); // 5
trace(uint("-5")); // 4294967291. It wraps around from MAX_VALUE
trace(uint(" 27 ")); // 27
trace(uint("3.7")); // 3
trace(int("3.7")); // 3
trace(Number("3.7")); // 3.7
```

# Case sensitive,

(.)



- Case sensitive

- ActionScript 3.0

- . .

- var number1:int = 10;
      - var Number1:int = 20;

μ μ

μ

- 

- 

(.)

(.)

μ

μ

μ

μ

μ

.

μ

μ

μ

.

μ

()

- 

μ :

- square.x = 230;
      - shortFilm.gotoAndStop(5);

μμ μ

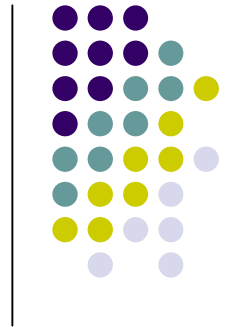


# μ (;),

- μ (;)
- μ (;) μ μ μ μ
- μ μ (;) compiler μμ

- μ μμ
- **// a single line comment**  
**var totalNumber: Number = 5; // a single line comment**

- μμ
- **/\* This is multiline comment that can span more than one line of code. \*/**



- **ActionScript 3.0**
- `trace(2 + 3 * 4); // 14`  
`trace( (2 + 3) * 4); // 20`
- `var a:int = 5;`  
`var b:int = 4;`  
`trace((a++, b--, a-b)); // 3`
- `trace("hello"); // hello`

μ μ

μ

μ μ μ μ (,) μ

μ μ

μ μ μ μ

μ μ μ



- **if..else**

```
if (x > 35)
{
 trace("x is > 35");
}
else
{
 trace("x is <= 35");
}
```
- **if..else if**

```
if (x > 0)
 trace("x is positive");
else if (x < 0)
 trace("x is negative");
else
 trace("x is 0");
```





- **switch**  

```
var currentDate:Date = new Date();
var dayNum:uint = currentDate.getDay();
switch(dayNum)
{
 case 0:
 trace("Sunday");
 break;
 case 6:
 trace("Saturday");
 break;
 default:
 trace("No DayOff ");
 break;
}
```



- **for**

```
var i:int;
for (i = 0; i < 5; i++)
{
 trace(i);
}
```

- **for..in**

- ```
var myObj:Object = {x:20, y:30};  
for (var i:String in myObj)  
{  
    trace(i + ": " + myObj[i]);  
}  
// output:  
// x: 20  
// y: 30
```

- ```
var myArray:Array = ["one", "two", "three"];
for (var i:String in myArray)
{
 trace(myArray[i]);
}
// output:
// one
// two
// three
```

μ



- **for each..in**  

```
var myXML:XML = <users>
 <fname>Jane</fname>
 <fname>Susan</fname>
 <fname>John</fname>
</users>;

for each (var item in myXML.fname)
{
 trace(item);
}
/* output
Jane
Susan
John
*/
```



- **while**

```
var i:int = 0;
while (i < 5)
{
 trace(i);
 i++;
}
```

- **do..while**

```
var i:int = 5;
do
{
 trace(i);
 i++;
} while (i < 5);
// output: 5
```