

Actionscript 3.0


Fundamentals

Basics

- Execution Order :: Top-to-Bottom, Left-to-Right
- Semi-colon (;) marks the end of a line, instruction.
- // single line comment
- /* multiple line comment */
- { marks a block of code }

Variables

- Variables :: containers for information
- Creating a variable ::
 - `var myVariable: Number = 1;`
 - Keyword that is only used once when first creating a variable
 - Variable name
One word
Descriptive and unique
 - Initial data stored in the variable
 - Defines what type of data the variable will hold



Data Types

Table 6-3. Variable types

Data type	Example	Description
Number	4.5	Any number, including floating point values (decimals)
int	-5	Any integer or whole number
uint	1	Unsigned integer or any nonnegative whole number
String	"hello"	Text or a string of characters
Boolean	true	True or false
Array	[2, 9, 17]	More than one value in a single variable
Object	myObject	The basic structure of every ActionScript entity, but also a custom form that can store multiple values as an alternative to Array

Shortcuts

Shortcut arithmetic operators combine two tasks into one operator. The following are standard operators:

`x++` means `x = x + 1` (add 1 to current value)
`x--` means `x = x - 1` (subtract 1 from current value)
`x += n` means `x = x + n` (add value on right of equals sign)
`x -= n` means `x = x - n` (subtract value on right of equals sign)
`x *= n` means `x = x * n` (multiply by value on right of equals sign)
`x /= n` means `x = x / n` (divide by value on right of equals sign)

Comparison

Comparison operators are typically used in conditional (`if`) statements for comparing values. The following examples include equal to, not equal to, greater than, greater than or equal to, less than, and less than or equal to.

```
if (x == 1) { }
if (x != 1) { }
if (x > 1) { }
if (x >= 1) { }
if (x < 1) { }
if (x <= 1) { }
```

Logical Operators

Logical operators are also used in conditional statements. They group tests together using *and* (`&&`), or *or* (`||`) to create a new test. The combined test relies on either one or the other original tests passing (*or*) or both tests passing (*and*). Another logical operator tests for falsehood using *not* (`!`). You'll learn about conditionals later, but it's good to be able to recognize these operators:

```
if (x == 1 && y == 2) { }
if (x == 1 || y == 2) { }
if (!myClip.visible) { }
```

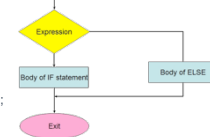
Conditionals

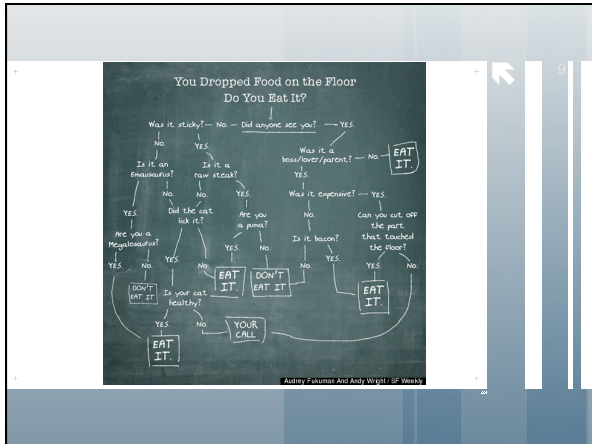
- A way to allow for branching in a program.

- One = assignment `x = x + 1;`

- Two == comparison `x == 1;`

```
if (weather == "raining") {
    trace("Wear a raincoat");
}
else {
    trace("leave raincoat at home");
}
```





Loops

- A way of repeating a set of instructions
- `i++` is the same as `i = i + 1;`

```

var N = 100;
for ( var i: Number = 0; i < N; i++ ) {
    trace("hello");
}
    
```

Functions

- A way of allowing code to be reused
- Defining a function ::


```

function myFunction() {
    trace("This is my Function");
}
            
```
- Using a Function ::


```

myFunction();
            
```

Events

- Events are things that happen.
- We use events to trigger scripts
- Example of an event
 - MouseEvent.CLICK
 - KeyboardEvent.KEY_DOWN
 - Event.ENTER_FRAME
 - TimerEvent.TIMER

Event Object: Objects always have the first letter in caps

Constant: a property that can't be altered or redefined
Constants are always all in Capitals

Event Listener

- Allows an object to listen for a specific event
- `addEventListener(EventType, whatToDo);`
- You can remove an event listener
- `removeEventListener(EventType, whatToDo);`

Event Handler

```
addEventListener(EventType, whatToDo);
```

- `whatToDo` is an event handler function
- Need to define this event handler that receives an argument of event type that will trigger the function

```
function whatToDo ( event: Event ) {
    //what you want to happen when the event occurs
}
```

Events in ActionScript 3

- using `addEventListener/removeEventListener`
- listening for event types stored in Event classes
- using a function (not an object) event handler
- defining event handler to receive an argument of the type Event

Some Last Thoughts

- You cannot add scripts to an object!
- `object.buttonMode = true; //cursor changes to hand cursor`