



ACTIONSCRIPT ESSENTIALS

INTRODUCTION TO ACTIONSCRIPT

Why should I learn how to use Actionscript?

Actionscript is the programming language used in Macromedia Flash to create interactivity and dynamic movies, games and applications. You can still create great Flash projects without Actionscript, however, if you want your files to be interactive with user input, you will need to learn some actionscript.

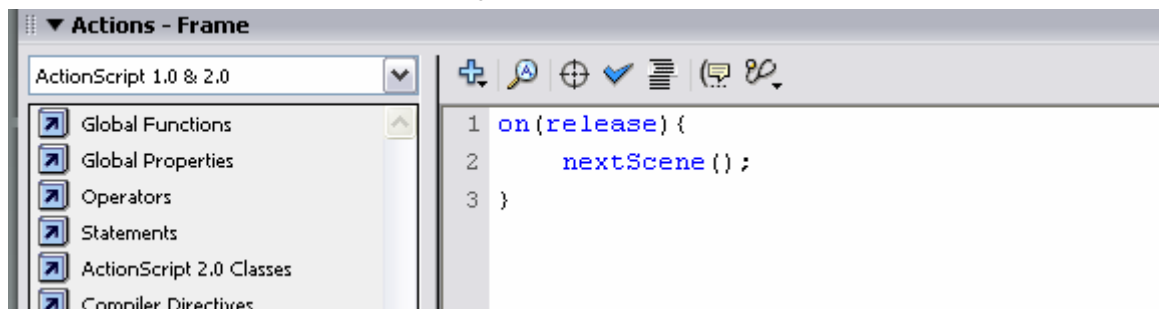


What scripting language is Actionscript based on?

Actionscript is based on ECMA Script – a standardized version of the Javascript language. Macromedia deliberately chose this script so that people familiar with javascript would feel comfortable with the structure of actionscript.

How do I add Actionscript to my flash files?

Actionscript can be added through the Actions panel. The easiest way to open this panel is to click on F9. Actionscript can be entered on to keyframes of the timeline or can be attached to buttons and movie clip symbols.



How do I write Actionscript?

This handout will provide you with an overview of the syntax of Actionscript and an understanding of terms such as variables, strings, expressions, if then statements, conditions and loops. Using this knowledge, you will be able to start using Actionscript at an introductory level.

To learn more, we recommend that you:

1. Work through some other online tutorials that are available on this site and hundreds of other sites on the web. Check out the list of sites on the final page of this tutorial. These are a great place to start.
2. Open existing flash files to look at what script has been used to make them interactive. Ask friends for their base files or look for .fla files you can download on the web. Some sites such as www.flashcomponents.com allow you to download hundreds of their .fla files for a small fee. Flash Kit at www.flashkit.com contains hundreds of base files that you can download for free.
3. Join discussion forums or online communities where you can talk to like-minded people about Flash. There are many available and you generally get a response to posted questions within a day or two.





THE SYNTAX OF ACTIONSCRIPT UNDERSTANDING THE CONVENTIONS OF THE LANGUAGE

The Evaluate Operator

; All statements end with a semicolon. In Actionscript, the semicolon is referred to as the "evaluate operator". The ; signals to the computer that the line of script before it is complete and that the computer should run this script.

Text Values

Text values are surrounded by quotes. This text is called a string.

e.g. `variableone = "Text" ;`

Numeric Values

Numeric values are not surrounded by quotes. They simply appear as shown below.

e.g. `variableone = 1`

Expressions

Expressions are references to another variable. Just like numeric values, expressions are not surrounded by quotes. They simply appear as shown below.

e.g. `variableone = variabletwo`

Statement Blocks

Single lines all end with the evaluate operator or semi colon. Multiple lines of script are called statement blocks. They are separated by these curly braces { } .

Parent & Child Objects

Child objects are separated by dots in parent.child.grandchild order.

e.g. `_root.movieclipname._x= 10;`

The Trace Function

The trace function is a tool that you can use to test scripts whilst you are writing them. This is a special window called the Output Window when you launch Test Movie mode. e.g.

```
trace ("text");  
trace (myvariable*10)
```

Comments

When writing code or script of any kind, it is a good idea to use comments. Comments are non-executing lines of code. Programmers use comments to notate parts of their script for future reference.

You can make comments throughout your script for future reference. This really helps when you want to use the same script again in a future project.

Comments can be single-line or multi-line as shown below.

Single-Line Comments

```
// Comment
```

Multi-Line Comments

```
/* Comment s that are multi-line should  
be surrounded by these. */
```





VARIABLES & EXPRESSIONS

Variables and expressions are an essential part of a scripting language. They provide a program with the ability to remember things and to store information.

Variables

Variables are containers for holding information.

They are used to hold values.

e.g. you can set up input text boxes that the user enters information into. The responses that the user enters becomes the value of the variables.

For example:

```
Name = "Anna"; //string literal
Age = 15; //numeric literal
Location = "Cairns";
```

Variables can be accessed across scenes and timelines. This means that you can have variables within movieclip symbols that feed information to variables that are on the stage. If you are in a movieclip and want to refer to a variable on the stage, you need to precede your variable name with `_root`. e.g.

```
_root.user = _root.movieclipname.name;
```

// this line informs the flash player that the variable named user should take on the value of the variable named name, which is located in the movieclipname symbol on the stage.

Variables can have constant values or can refer to another variable.

e.g.

```
name = Paul;
user = name; // this statement tells the Flash player that the value of user is Paul.
```

Expressions

Expressions are used to calculate values. They are like phrases or part sentences that provide us with the means to convert our thoughts about how we want a program to work into a script that our computer can process to make it do what we want.

For example, you may want to create a Flash file that appears to interact with the user by using their name when asking them to complete a task. You would ask the user to enter their name into an input text box and then would create dynamic text boxes throughout your file that you assign a variable name to. You then create an expression like the one below to create the text for that dynamic text box.

```
Welcome = "Hello " + Name + "!";
```

If the user had entered the name Peter into the input text box called name, the expression above would make the words **Hello Peter!** appear in the dynamic text box assigned the variable name Welcome.





CONDITIONAL STATEMENTS & LOOPS

Loops and conditional statements are used in scripting to add repetition and decision making to scripts. This increases the flexibility of a script and makes it more than simply an unchanging set of instructions.

If Statements

If Statements are the most common conditional statement. "If" statements are used to perform a certain action but only if a certain condition is 'true'.

e.g.

```
if (answera = australia) {  
    result= "Correct";  
}
```

If Then Else Statements

If then else statements are also commonly used in Flash Actionscripting. An example is shown below. This script is for a drag and drop game where the movieclip being dragged will snap back to it's original screen location if not dropped in the correct place. The correct place in this case, is over a shape named shapetarget. I have made comments throughout this script so that you can see what parts of the script perform what task.

```
// when mc is pressed , drag it  
    on(press) {  
        startDrag(this);  
    }  
// when mc is released stop dragging  
    on(release) {  
        stopDrag();  
        // if the right shape , leave it there  
        if (this._droptarget == "/shapetarget") {  
            this._x = _root.shapetarget) ._x;  
            this._y = _root.shapetarget) ._y;  
        }  
        // if wrong shape,  
        else{  
            // take it back to original position  
            this._x = 59.5;  
            this._y = 280.3;  
        }  
    }  
}
```

Loops

Ifs only perform a check once and then continue on through the program. To repeat a set of statements you need to add a loop. . Loops are invaluable as they can save you having to type out commands manually when you can give a base case and loop. The most common type of loop is called the 'for' loop. An example of a loop is shown below. This script produces 100 copies of the movie clip on the stage.

```
for (x=0; x<50; x++) {  
    duplicateMovieClip ("targetMC", "dupMC"+x, x);  
}
```





TOP ACTIONSCRIPTING TUTORIAL SITES

The previous pages have provided you with a brief overview of the actionscript language. The following websites provide hundreds of great tutorials on how to use actionscript to create interactive movies, files and games.

Developing Webs Actionscripting Tutorials

<http://www.developingwebs.net/flash/actionscriptindex.php>

Web Wasp Flash Tutorials

<http://www.webwasp.co.uk>

Actionscript.org

<http://actionscript.org/tutorials.shtml>

Flashkit Actionscripting Tutorials

<http://www.flashkit.com/tutorials/Actionscripting/>

Flashbax Flash MX Tutorials

http://www.flashbax.com/flash_mx_tutorials.htm

The Actionscript Toolbox

<http://www.actionscript-toolbox.com/>

Flash MX Actionscript Tutorials at Video-Animation.com

http://www.video-animation.com/mx2k_00.shtml

These are just a few of the hundreds of great websites available on the web. If you are looking for some script to do a particular task, we recommend that you do a detailed search in Google. We have found that this is usually a very effective way of finding tutorials / script that we need for a specific purpose.

