

# Flash - Code Basics

This tutorial is an introduction to coding in Flash (ActionScript) (A) and how it can be used to accomplish different task in Flash.



# Flash - Code Basics

## Variables

Variables are used in Flash to hold information (text, numbers or list) or provide a Boolean value (true/false).

## Booleans

---

Only can be equal to true or false.

Define:

```
var recordOnOff:Boolean = false;
```

Use:

```
if (recordOnOff) {  
    recordOnOff = false;  
    record_dot.gotoAndPlay(1);  
} else {  
    record_dot.gotoAndPlay(2);  
    recordOnOff = true;  
}
```

## Numbers

---

There are three types of numbers in Flash: int (+/- integer), uint (+ only integer) and number (floating point).

Define:

```
var count:int = 0;  
var ucount:uint = 1;  
var num:Number = 1.05;
```

Use:

```
count ++ ; //adds 1 to the current value  
num = num + .5; // adds .5 to the current value
```

## Text

---

Text string has to be defined in quotes.

Define:

```
var string_txt:String = "video on";
```

Use:

```
if (string_txt == "video on") {  
    string_txt = "video off";  
} else {  
    string_txt = "video on";  
}
```

## List

---

List are used to store an array of numbers or strings.

Define:

```
var scores:Array = [100, 80, 60, 95, 100];  
var days:Array = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];
```

Use:

```
var count:int = 0;  
count = scores[0]; //equals to 80, first value is 0  
var string_txt:String = "";  
string_txt = days[3]; // value equals to "Wednesday"
```

# Flash - Code Basics

## “if” statements

The “if” statements are used to test a condition and then determine which outcome to follow.

### if/else

---

This “if” statement is a Boolean condition that only allows choice between two conditions.

Use:

```
if (recordOnOff) {  
    recordOnOff = false;  
    record_dot.gotoAndPlay(1);  
} else {  
    record_dot.gotoAndPlay(2);  
    recordOnOff = true;  
}
```

### if/if else

---

This “if” statement allows for multiply conditions. Here, it is three choices. If none match, then the statement is skipped.

Use:

```
Use:  
if (menu == “preference”) {  
    // do something if variable menu equals “preference”  
} else if (menu == “setup”) {  
    // do something if variable menu equals “setup”  
} else if (menu == “exit”) {  
    // do something if variable menu equals “exit”  
}
```

### if/if else

---

This “if” statement allows for multiply conditions. Here, it is two choices. If none match, then the statement is skipped.

Use:

```
if (num == 1) {  
    // do something if variable num equals 1  
} else if (num == 5) {  
    // do something if variable num equals 5  
}
```

# Flash - Code Basics

## Code to Create Buttons

Buttons in Flash follow a standard structure for setting up the code.

## Stage & Instance Setup

Select the button element on the stage (**A**) and input a name into the Instance Name slot in the Properties panel (**B**).

Here is the code for the button:

```
power_btn.addEventListener(MouseEvent.CLICK, fn_power_btn);
```

<u>power_btn</u>	<u>addEventListener</u>	<u>MouseEvent.CLICK</u>	<u>fn_power_btn</u>
Instance Name	Listener	Event	Function

Other possible events that can be added to a button include:

```
power_btn.addEventListener(MouseEvent.MOUSE_DOWN, fn_power_dwn);  
power_btn.addEventListener(MouseEvent.MOUSE_UP, fn_power_up);
```

To make the mouse pointer change on a rollover, add this code:

```
power_btn.buttonMode = true;
```

Any mouse listener event (click, down, up, etc) call a function. The structure of the function follows this format:

```
function fn_power_btn(event:MouseEvent):void  
{  
    gotoAndPlay(5);  
}
```

<u>function</u>	<u>fn_power_btn</u>	<u>(event:MouseEvent)</u>	<u>:void</u>
	Function	Event	

  

<u>gotoAndPlay(5);</u>	Custom Code
------------------------	-------------

