

Buttons

A button must be a symbol – but not a graphic symbol or movie. Must be a button!

Four States

up, down, over, *hit

Actions

rollover, click, release

ACTIONSCRIPT

ASSIGNMENT PART 1

Create a button with different up, down, and over states

Functions

Functions need names

Parameters, including inputs, go in parentheses ()

Parentheses are NOT optional

End in a semi-colon

Examples:

```
stop();
gotoAndStop(2);
addEventListener(MouseEvent.CLICK, whateverFunction);
function clickMyButton(evt:MouseEvent) {
    gotoAndStop(3);
}
```

NOTE: clickMyButton is just a function name. It can be anything. The (2) and (3) are frame numbers

ASSIGNMENT PART 2

Find a picture online. Import it into your library. Place the picture on frame #15. On frame #1 (in a new layer) create action script that tells the movie to stop.

Instances and Comments

Another name for an instance is an occurrence.

These are assigned in the properties window.

To make buttons do work, they MUST have an instance name.

In your action script, you can add comments that don't work. They start with // Example:

```
// This is what a comment looks like. Anything on that line is ignored. Note the slashes.
```

ASSIGNMENT PART 3

Give your button an instance name. (I called mine ButtonFace). Also, in your action script, add your name as a comment. Put it on a separate line with two slashes in front

Event Listeners

To make your button do work, there are two parts. First, the computer has to “listen” for an event, such as a mouse click. Second, when the computer “hears” the click, it must perform an action. This is known as calling a function.

Example event listener:

```
// This is an event listener. Note that this comment is not important
ButtonFace.addEventListener(MouseEvent.CLICK, theProgramThing);
```

Translation. To my button which has the name “ButtonFace”, listen for when it's clicked. When it is clicked, run the program called “theProgramThing”.

Example function:

```
// This is a function. Adding this code helps you keep everything organized.
Function theProgramThing(evt:MouseEvent) {
    gotoAndStop(11);
}
```

Translation: Run a little program called “theProgramThing” that's triggered by a mouse event. When this happens, go to frame 11 and stop.

By the way, you really don't need the extra spaces around the brackets {...} But when you do this, it's easier to proof-read your function. Also, watch the spacing, capitalization, punctuation, and types of brackets. It really matters in action script.

Fun Fact: When Flash recognizes a piece of code, it turns blue. That's a good sign, usually.

ASSIGNMENT PART 4

Add an event listener to your action script. Make sure it's listening for the instance name you called your button. Use the above code. Then add the function. It should take the user to frame 15 where you put your picture. TEST YOUR MOVIE and make sure it works.

NEXT and PREVIOUS

Two more pieces of code that are useful are nextFrame(); and prevFrame(); These will take you to the next frame or previous frame. These will be very handy later. We won't use it now.

ASSIGNMENT PART 5

Create a separate button on frame 15. You will need a separate event listener and function on frame 15. Make a button there to take you back to frame #1. I recommend that its function look something similar to this:

```
// this is the function
function whatEver(evt:MouseEvent){
    gotoAndStop(1);
}
```

and

```
// this is the event listener. Notice you had to add an instance name of SecondButton
SecondButton.addEventListener(MouseEvent.CLICK, whatEver);
```