

# Buttons in ActionScript 3.0

Anatomy of a function

```
1  stop();
2
3  myButton2.addEventListener(MouseEvent.CLICK, goBack);
4
5
6  function goBack(evt:MouseEvent) {
7      gotoAndPlay(1);
8  }
```

**line 3** assigns an “event listener” to the button called myButton2. The “event” is a MouseEvent (this will respond to something the mouse does.) The next part tells the button WHAT MouseEvent to listen for. Here, it is told to listen for a mouse CLICK. The next part , goBack, points to a function called “goBack” that I have written to tell the button what to do when the mouse is clicked.

**line 6-8** is a function. I’ve named this function “goBack” and told the function that it will respond to a MouseEvent. (“evt” is a variable– a name that I have given this event that I can use other places in my code. For now, just follow my code.) WHAT we want the functions to do is gotoAndPlay a specific frame number. Here I’ve asked it to gotoAndPlay frame 1. Note that WHAT occurs is placed between the curly braces{ }.

---

## tutorial : HOW TO MAKE A BUTTON

### LAYER 1\_FRAME 1:

1-Make a shape on the stage.

2-Choose modify-convert to symbol and choose BUTTON.

3-name your button.

4-Insert a keyframe in the BUTTON EDITING MODE and make an over state (what the button looks like on mouseOver), a down state (what the button looks like when you click it) and a hit state (you won’t see this- but it needs to be there to define the area in which the viewer can click.

5- Go back to the main stage. In the properties toolbar you are going to give your button an instance name (I would give it the same name you previously did when you made the button + a number. Like greenBtn1) ORGANIZATION!

6-Go to your actions palette-located in the windows drop down menu.

7-In your actions panel you are going to type:

```
stop();  
myButton.addEventListener(MouseEvent.CLICK, toTwenty);  
function toTwenty (evt:MouseEvent){  
    gotoAndStop(20);  
}
```

8-Go back to your scene and make a keyframe on frame 20. Put something in that frame as an indicator that you have reached your destination!

9-Go back to your actions toolbar and type in:

```
stop();
```

Make sure that you are in the right frame. On the left will be a toolbar to let you know what frame you are in. Reminder: If the script is not accurate- it will not work! (this includes case sensitivity! mybutton is NOT the same as myButton.

You can add as many buttons as you would like and have them complete whatever functions you have written! Such as:

```
mybutton.addEventListener(MouseEvent.CLICK, toTwenty);  
redbutton.addEventListener(MouseEvent.CLICK, toTwenty);  
purplebutton.addEventListener(MouseEvent.CLICK, toTwenty);
```

All of these buttons would then execute the instructions in the toTwenty function.

Note: if we wanted to navigate to a scene instead of a frame, we would write

```
function goMichael(evt:MouseEvent){  
    gotoAndStop("one", "michael");  
}
```

"one" is a frame label name. "michael" is the name of the scene I want to go to. You must use frame labels when navigating from scene to scene. To apply a frame label, go to the frame in your timeline you want to label. Add an empty keyframe and give your frame a name in the properties bar. This is your frame label.

10-add 3 buttons and write functions that take them to different frames as well as different scenes. Become comfortable with understanding how buttons, event listeners and event handlers work.

---

This wonderful tutorial was made in collaboration with Suzanne Bartholemew.

THANKS SUZANNE!