

Controlling Your Animations

Directions

Your job is to read and follow the directions below in order for you to learn about ActionScripts. Once you have read and finished the practice activities, you need to answer the Quiz questions. Complete the assignment and turn-in the finish assignment to the G:/Drive > Inbox as **period_lastname_actionscript**.

Use you time wisely and be ready to show me what you have learned!

Introduction

In our last lesson, you learned a lot about how to create some simple Flash animations using the frame-by-frame method as well as the tween method. But what good is movement if you can't control it?

You probably noticed that your animations in the last lesson would repeat over and over again when you tested them using the Test Movie command. This is fine for some animations, but what if you've created a transition to take you from one part of a site to another? You wouldn't want that to repeat. You'd want it to begin and end appropriately.

Or maybe you have an animated character that talks to your audience and introduces them to a product on your Web site. You don't want the character to repeat its pitch over and over until the person leaves the page, do you?

Luckily, there's a solution. Flash comes with a rather powerful scripting language called *ActionScript* that's built right in. In this lesson, you'll be learning some basic ActionScript that will allow you to control your animations and prevent unwanted looping.

Before we begin, please move the files from the G:/Drive to your H:/Drive. You'll need these files for this lesson.

Frame Actions

ActionScript is a very powerful scripting language. It provides you with thousands of commands, but it doesn't have to be difficult to work with. By default, Flash's Timeline will loop over and over when you publish an animation or test it using Test Movie (CTRL + ENTER or COMMAND + ENTER on Mac).

You've seen this in past examples, but let's open a file together so I can show you how this works.

Open the file named *TravelingBird.fla* from the lesson files you moved from the G:/Drive to your H:/Drive.

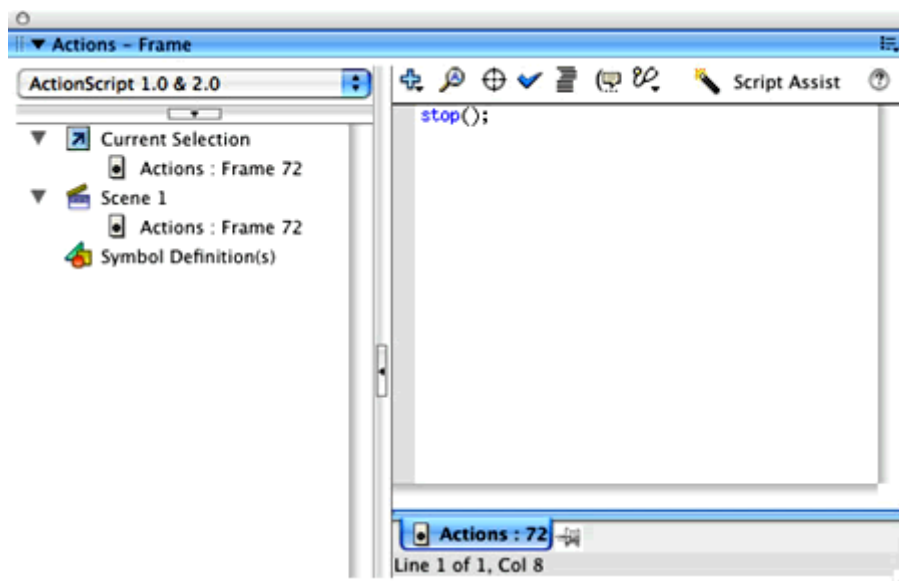
What we have here is a simple animation using a motion tween to make Gary (the bird) appears as if he's traveling along the paved path.

If you just press ENTER, the Timeline will play inside Flash and stop at the end. That's great, but it only does this because it's playing the Timeline. It's not showing you an accurate depiction of what will happen. To really see what the final document will look like, you have to test the movie by pressing CTRL + ENTER or COMMAND + ENTER on Mac.

Right now, Gary gets to the end of his little journey and then he suddenly pops back to the start. This must be very frustrating for Gary, so let's help him out a bit by telling Flash we want the animation to stop once Gary's done walking.

We're going to stop the looping of the Timeline by using a *frame action*. I'll explain exactly what that is in a minute. But it's actually easier to show you than to tell you; so follow these steps to add your first action, and we'll talk about what's happening when you're finished:

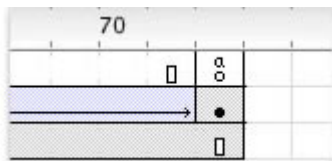
1. Save the *TravelingBird.fla* file and give it a new name so we can make some changes to it without losing the original. Let's name it *TravelingBird2.fla*.
2. Now create a new layer on the Timeline (click the **Insert Layer** button or use **Insert > Timeline > Layer**). Name the new layer *Actions*.
3. Select the last frame (72) on the new Actions layer and press F6 to insert a new keyframe.
4. With the new keyframe on frame 72 selected, open the Actions panel (**Window > Actions**). You'll turn to this panel whenever you want to add an action.



The Actions panel showing a stop action

On the right side of the Actions panel you'll see the area where you'll actually enter the actions (the *script pane*). While there are a couple of ways to get an action into the pane, we're going to just type ours in for now.

5. Type the following exactly as you see it here into the script pane: `stop();`
6. You can tell you've added your action to the keyframe on 72 because Flash now shows a tiny *an* above that frame.



A keyframe with a frame action on it shows an *a* at the top of the frame

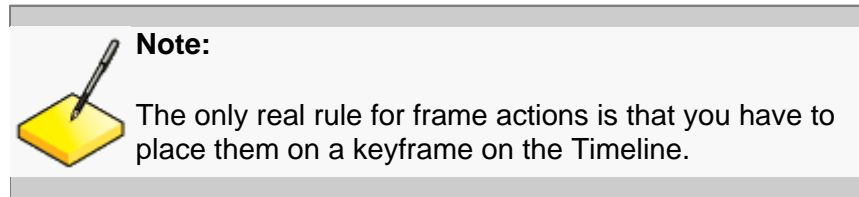
7. Finally, press CTRL + ENTER or COMMAND + ENTER on a Mac to test your movie and make sure the animation plays once and stops.

Well, now the animation works the way we want it to, but a lot just happened that you haven't seen before, so let me explain each step.

First, we created a new layer, but we purposely left it empty of any artwork. You don't have to do this, but it's a good habit to get into. The same thing goes for naming the layer *Actions*. Flash doesn't care

what you name your layers—I just suggested it so you'll see the connection between that layer and the action.

The new action is known as a *frame action* because we placed it on a frame—or, more specifically, on a keyframe.



The reason I recommend you create a new empty layer, name it *Actions*, and place your frame action on it is because it will help you keep your Timeline organized. It also makes it easier to keep your actions separate from your artwork.

For example: Let's say you had put the stop action on the final keyframe of the *Gary's Walk* layer and then wanted to move or remove that keyframe. You'd be moving the stop action, too. By putting all of your actions on their own layers, you can move actions around independently without interfering with the animations.

Speaking of the stop action, let's take a closer look at what you typed in the script pane. In ActionScript, the code to stop a Timeline from playing is simply `stop();` Right now, I'm telling you the code for that action, but soon we'll explore how you can look up other actions on your own or have Flash paste the code in for you (sometimes).

Another thing to keep in mind regarding frame actions is that they run right when the playhead gets to them on the Timeline. Since the playhead moves from left to right on the Timeline, the stop action we inserted only executes when the Timeline reaches the final frame (72). At that point, Flash knows you want to stop playback. Otherwise, it would jump back to frame 1 and start playing all over again (looping).

Let me show you what happens when you move the stop action to another frame.

1. Click the keyframe on frame 72 of the Actions layer to select it.
2. Now click that selected frame again and drag it (without letting go of the button) to frame 40 of the Actions layer.
3. Finally, press CTRL + ENTER or COMMAND + ENTER on a Mac to test your movie.

As you can see, simply moving the stop action to frame 40 prevents the rest of the Timeline from playing at all. Now the animation stops even earlier.

To give you a mini-exercise before we move on to Chapter 3, move the stop action to frame 1. You'll see that you can prevent another default behavior of the Flash Timeline. Normally a Timeline will play automatically, but with the stop action on frame 1, you can start your animation in a paused or stopped mode. Just press ENTER to view the animation past the stop action in the first frame. This can be useful for testing, but the general user won't be able to do this with your published content.

Move the stop action back to frame 40, save your work, and then close the file before moving on to the next chapter. This has been pretty easy so far, but feel free to check your work against *TravelingBird_Final.fla* from the lesson files if you have any questions.

Frame Labels and More Actions

Now you've conquered two of Flash's default behaviors by placing a single `stop();` action on the right frame of the Timeline. As you can imagine, there are more actions to cover.

Before we look into these other actions, let's create a problem that only an action can solve. Start by opening the file *AnimatedLetters.fla* from the lesson files. Save a copy, naming the new file *AnimatedLetters2.fla*, to preserve the original.

Once you have the file open, use the **Test Movie** command to see what the animation looks like.

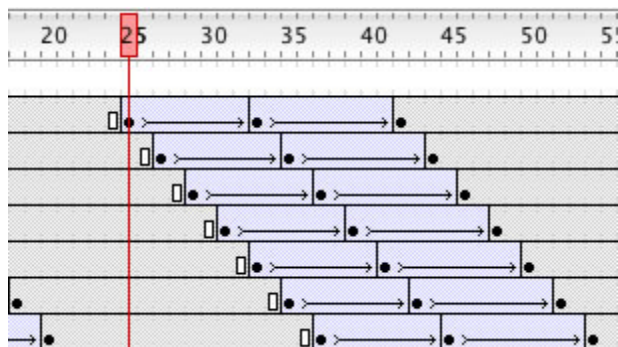
This time, you're faced with a problem that's similar to the one in the last chapter: The *Welcome* letters keep flying in, just like Gary the bird kept starting his walk over again. But this time, there's an exception. It's not enough to just have the whole animation play once and then stop. This time, you'll want to have the first part of the animation (when the letters fly in) play only once and the second part repeat over and over (when the letters jump).

To do that, you'll need to use a different kind of action. Here's how:

1. Create a new layer and name it *Actions* as you did last time.
2. We know we want the whole thing to play through once before we start looping the second part of the Timeline, so let's add a keyframe (F6) at the very end of the Actions layer (frame 67). This is where we'll add our frame action.
3. With the keyframe on 67 selected, open the Actions panel (**Window > Actions**).
4. In the script pane, type this new action: `gotoAndPlay(25);`
5. Use **Test Movie** to see your new action in action!

You can see that this simple action does exactly what we intended. Let's look a little closer at the action to find out what's going on.

When you add `gotoAndPlay(25);` as an action, you're basically telling Flash to go through the whole Timeline, but when the playhead gets to frame 67, go back to frame 25 and play the Timeline over again from there. If you look at frame 25 of the Timeline, you'll see that the second part of the animation starts exactly on that frame. Of course, that's why we chose to jump to frame 25.

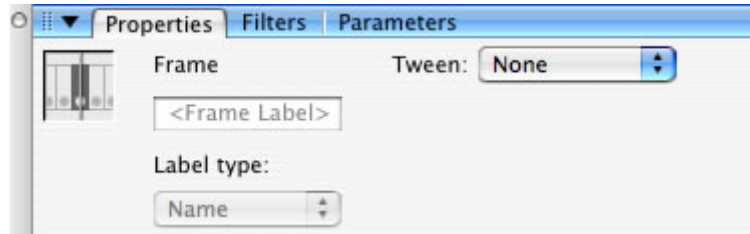


Frame 25 of the Timeline where the second animation begins

Using the goto command like this is perfectly effective, but let's imagine you're making a lot of changes to an animation. You might not always want to jump specifically to frame 25. You might want to move it around, starting it at a couple of different frames to see how each looks. If you've written frame 25 specifically in the action frame, you'll have to change the frame number if you ever need to move it.

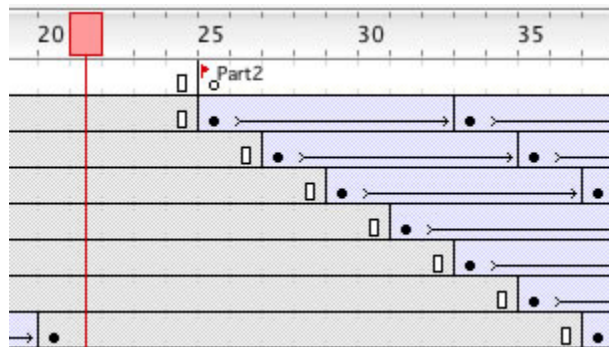
That's not exactly difficult, but there's another way that might be easier. You can use something called a *frame label* to create a kind of bookmark on the Timeline. Let's try it:

1. To create a new frame label, you have to have a keyframe to add it to. So let's add one on the Actions layer at frame 25. Select frame 25 of the Actions layer and press F6 to insert a new keyframe.
2. Look at the Property inspector, and you'll see a text field that says *Frame Label* in it. This is where you'll type your label.



The Frame Label field in the Property inspector

3. You should always name your labels something that indicates what they're for, so let's name this one *Part2*.
4. Finally, press ENTER to set the label. You'll see it appear in the Timeline once you do.
5. Go ahead and test your movie with CTRL + ENTER or COMMAND + ENTER on Mac to see it work. It should be the same as when we pointed to frame 25.

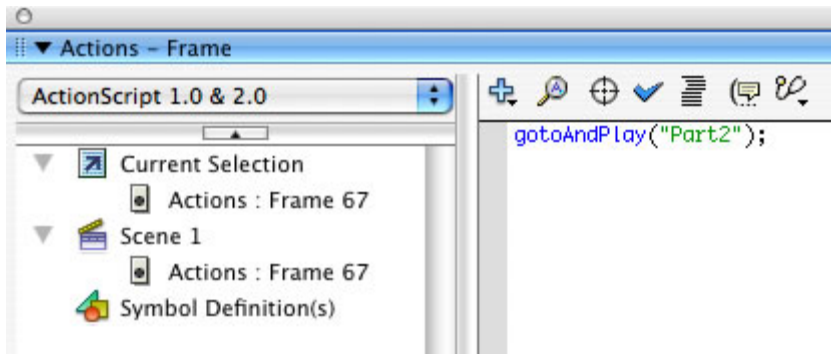


The Timeline showing the *Part2* label on frame 25

Now that you have your label in place, it can act like a bookmark for the goto action you used before. The only difference now is you need to tell Flash to jump to the label instead of going to frame 25. To do that, click frame 67 (the one with the goto action), select the number 25, and press DELETE.

When you want a goto action to point to a frame number, all you have to do when typing the frame action is put the number between the parentheses. But when you want a goto action to point to a frame label, you have to put the name of the label in double quotes. So now type "*Part2*" just like you see it here.

When you're finished, use Test Movie to see how it looks.



The goto action jumping to a frame label

At first you may be thinking, "Okay, but it works just the same, so what's the point?" You're right. Since our label is on frame 25, it works just like it's supposed to. The advantage is that now you can move the keyframe with the label on it anywhere on the Timeline, and the goto action will automatically find it and go to it. And it's easier to remember to move a label that's visible on the Timeline than to change an action that you can't always see.

The goto action is proving to be pretty flexible, but it doesn't end here. Right now we're telling the playhead to go to this frame or label and play from there, but we can also say "go to this frame or label and just stop."

Test your movie one last time to watch what happens. You can check your work against the lesson file *AnimatedLetters_Final fla*.

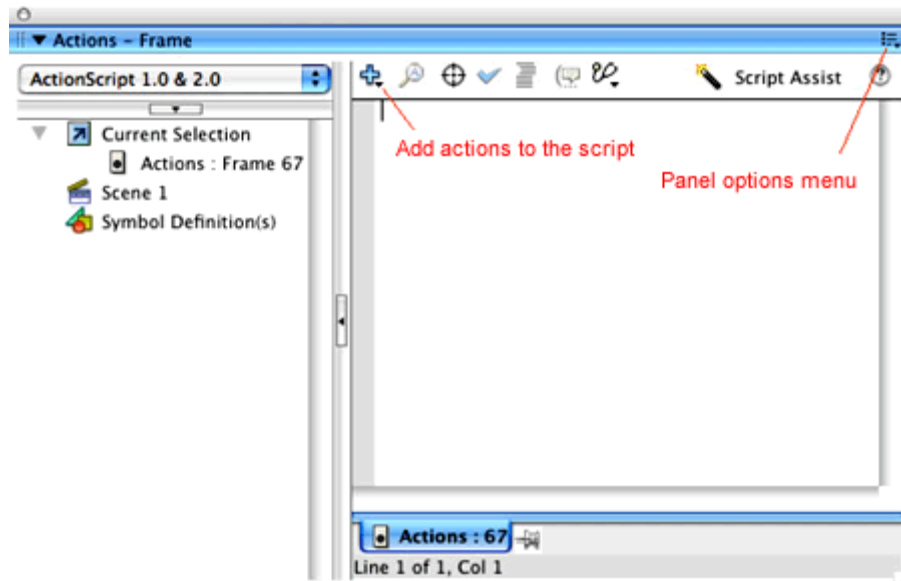
As a final test, let's move the Part2 frame label from frame 25 to frame 35 of the Actions layer. Do this by just moving the keyframe itself. Now select the keyframe at the end of the Actions layer that has the goto action on it. Open the Actions panel if it's not already open and change the `gotoAndPlay("Part2");` to `gotoAndStop("Part2");`. With those changes made, the animation now completes once and then stops half-way through the letters moving (some are up, some are down) instead of looping the animation. Yep, it's that easy.

The final file won't show this last change we made to the goto action, but it's an easy change to make if you want to do it on your own.

The Actions Panel

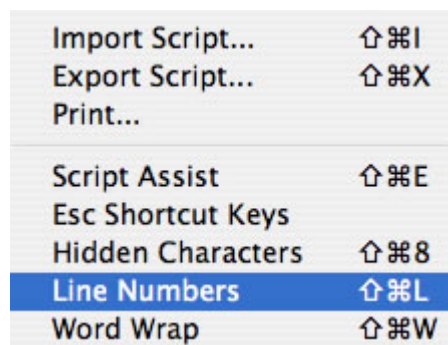
We've been typing all of the actions by hand so far, but it's not always necessary. Flash can help you avoid a lot of that typing, but it's important that you get comfortable with the typing method because when you turn to Flash to help, it's not always going to work perfectly.

To see what I mean, let's look closer at the Actions panel and what it has to offer.



The Actions panel


Let's start with a simple option that I've always found helpful. Click the Panel options menu to expand it and then put a check by Line Numbers. Right now, this won't help us much since we typically only have one-line scripts. But I always find it useful when I can find a line number at a glance.



The panel options menu

Moving on from that, there's another menu I'd like you to take a look at. I'm not sure that it has an actual name, so we'll call it the *Add Script* menu for the time being. When you click on this menu (the + button) you'll get a menu tree of actions that you can insert into the script pane.

There are dozens of actions listed in this menu, but this is only a fraction of the total number of ActionScript commands available. The really advanced stuff is only available when you type it in yourself. To keep us all focused on learning the basics we won't be getting into any of that in this course. Instead, let's look in the **Global Functions > Timeline Control** section of the Add actions menu. Here you'll see some familiar actions that you've already learned how to use.



```
gotoAndPlay
gotoAndStop
nextFrame
nextScene
play
prevFrame
prevScene
stop
stopAllSounds
```

Timeline Control menu

Choose **gotoAndPlay** from the list, and Flash will automatically insert the action into the script pane. The only problem is that it leaves out the frame number or label that belongs inside the parentheses. And even worse, except for the flashing cursor within the parentheses, Flash doesn't really let you know that anything's missing at all. So if you just picked the action from the list like this, you might not understand why nothing happens when you test your movie.

Aren't you glad I taught you how to type in your actions by hand first? Now you can use the Add Script menu to quickly add actions and then adjust them to work the right way. It's the best of both worlds.

Removing a Frame Action

We're about to call it a day after covering a lot of material. Using only two actions so far, you're now able to gain quite a bit of control over your animations. It's great to have this control, but sometimes you might want to remove actions and return the Timeline to normal.

To get rid of any action, all you have to do is select the keyframe where the action is and delete all of the text in the script pane. If you leave so much as a space behind, Flash will decide that an action still exists on that frame. So to thoroughly wipe an action off of a keyframe, follow these steps:

1. Select the keyframe with the action on it.
2. Open the Actions panel and place your cursor inside the script pane.
3. Press CTRL + A or COMMAND + A on Mac to select all the text in the panel.
4. Press BACKSPACE or DELETE to remove everything.

Conclusion

Frame actions give you a lot of control over the general playback of the Timeline. You can keep an animation from looping, or you can make it loop exactly how you want it to.

It's important to remember some of the techniques we talked about as well. Always having your actions on their own empty layers helps to keep your document's Timeline well organized and easy to change.

Then we talked about creating frame labels to add a type of bookmarking to the Timeline. Again, using labels instead of just jumping to frame numbers keeps things flexible and well organized, which is always a good thing when you're creating Flash animations.

You have a lot of options and a lot of flexibility when creating frame actions—the only real rule is that you have to place them on keyframes.

Quiz

1. What color is the flag indicator when you place a frame label on the timeline?
2. What indication does Flash give you that there's an action on a particular keyframe?
3. The main Timeline in Flash does what two things by default when you run the Test Movie command?
4. Which one of the following is a rule for both frame labels and frame actions?
5. When you use a goto action to jump to a frame label, what do you have to do differently from when you just jump to a frame number?

Assignment

In this assignment, you are going to practice using some of the actions you just learned. Open the file *L07_Assignment fla*. Take a look at the Timeline and run Test Movie to see what the animation looks like.

Right now, the tiki decides to stop and hop a little (probably from excitement) before he reaches his home. Then, when he gets to his home, he jumps back to the start and repeats the process.

Use the frame actions and frame labels to have the tiki skip the hopping altogether and stop when he gets to his home.

Good luck and have fun!