

Flash Tutorial Part I

This tutorial is intended to give you a basic overview of how you can use Flash for web-based projects; it doesn't contain extensive step-by-step instructions and is therefore not a standalone tutorial (always have the online help files open when working with Flash). This document covers the fundamentals of creating a basic Flash website. We won't have time to go into depth on many of the specific tasks presented, so I recommend using the help files and the wealth of online materials to broaden your understanding of what we cover here.

A. OVERVIEW

1. Flash file formats

Flash files are called movies. The editable format in Flash is .fla. Flash movies are compressed as .swf files; these compressed files can be published as .html. The published html file requires the .swf file to work; the html file simply loads the .swf as an html in the browser. The .fla doesn't need to be uploaded to the server, but always keep a copy to make further modifications. Flash files (.fla) consist of layers and a timeline. The timeline consists of frames—which are, in the simplest form, time steps in the animation. Frames can be used to create animations as well as to create multiple states of the movie—Flash can be used for much more than just animation. We will explore both of these uses of the timeline later. Unlike HTML documents, all elements are embedded in the movie; therefore you don't have to move images around with the flash movie. The compression algorithms of Flash are extremely efficient; that said, I still recommend editing images in Photoshop first if you need to minimize file size as much as possible.

2. ActionScript

ActionScript is the programming language used in Flash. Both JavaScript and ActionScript are based on the same scripting language, ECMAScript, and are therefore similar in many ways. ActionScript has evolved from a simple scripting language in Flash 4 to today's ActionScript in Flash 10 that is a sophisticated programming language capable of pretty much anything you can imagine doing in the Flash environment. ActionScript 3.0 (AS3) was released in 2006 and was such a fundamental change in the language that many people still have not made the switch. I am one of them and this tutorial will therefore only use ActionScript 2.0 (AS2). The main reason that people have been slow to switch is that the AS3 online community is limited—at the moment—and can't offer support on many issues. On the other hand, AS2 has a huge online inventory of extensive, well written tutorials and an online community that can support pretty much any project idea or problem that you have. AS3 is more similar to JavaScript than AS2; therefore it is easier to learn for those that know JavaScript, C# or C++. AS3 is an object orient programming language and is definitely an improvement on AS2, as it offers a more efficient coding environment and is in conjunction with code-writing standards for Java and C++. For anyone planning on becoming an expert in Flash, learning AS3 would be the best option and the online community is growing. On the other hand, AS2 is a better choice for those that would like to use Flash to create dynamic online material without having to master the programming language, since it is rare that you have a problem that someone else hasn't already come across and published a solution for it online. I find that the resources of the AS2 online community outweigh the benefits of AS3 at the moment, but this will obviously change with time.

3. Drawing

The anatomy of vector graphics in Flash is similar to other drawing programs like AI, Corel, Freehand, etc. Vector graphics—called Shapes—are made up of lines and anchors. In Flash, lines are called curves. The unique part of Flash vector graphics is in how you can modify them. There are two selection tools: the Selection and Subselection tools. The Selection tool allows you to grab objects by clicking on them (click and release). It also allows you to move elements of objects and entire objects by click and drag. Use click and drag to modify curves

and to move anchor points or entire objects. In many ways vector graphics in Flash are simpler than in AI. Unlike in AI, you don't have to select end points to join artwork. You can simply join objects by moving them together; you can divide an object simply by selecting part of it (with the Selection tool) and moving that part. Another important difference from AI is that with vector objects in Flash, the outline and the fill are treated as separate elements of an object. The Subselection tool allows you to select the path of an object, similar to the Direct Selection tool in AI. The Subselection tool allows you to move anchors and the Bézier handles ("whisker") to adjust the shape of a curve.

4. Images

Raster images (referred to generically as bitmaps in Flash) can be resized and transformed (scale, rotate and skew) in Flash (Modify>Transform). You can also adjust the compression settings like in Photoshop (on an object in the Library, right click>Properties). Other than that, raster editing is limited. There is a trace option however; this is similar to the *Live Trace* in AI (Modify>Bitmap>Trace Bitmap). Import images to the stage (the term stage refers to your document in Flash) through the file menu (File>Import>Import to Stage; or Ctrl + R).

5. Timeline

The timeline is made up of frames just like in any animation. There are two types of frames: keyframes and frames. Keyframes indicate a new instance or modification of objects; a frame is just a time step in the timeline and a continuation of a keyframe. Adding a Keyframe allows you to change the contents of that frame in the timeline. Keyframes are indicated by a black dot (the dot will be open if the keyframe is empty); frames will not have an icon in the timeline except when they indicate the end of a movie or precede a keyframe in which case they will have an upright rectangle. Both types of frames will be grey if they contain objects and white if they are empty. Layers are independent of each other, but are bound together in the movie by the timeline. Think of the timeline window as rows (layers) and columns (frames). The rate of the timeline can be changed (frames per second—fps). Keyframes with a little "a" contain ActionScript (this will be explained below). Flash movies contain a main timeline called the parent timeline and symbols contain their own timeline which is synchronized to the parent timeline in some cases and independent of it in others (discussed in more detail below).

6. Objects and symbols

There are three basic types of objects in Flash: Shapes (vector), bitmaps (raster) and text. Objects are static elements (with the exception of input and dynamic text) that cannot be animated or controlled by ActionScript. Objects must be converted to symbols for this (Modify>Convert to Symbol or F8). This actually wraps the object inside of a symbol, rather than converting it; the object still exists inside the symbol and can be edited as a static object. There are three types of symbols: Graphics, Buttons and Movie Clips. Graphics are often used to create animations using tweens. There are limited options with Graphics as they cannot be controlled by ActionScript. Buttons and Movie Clips, on the other hand, can be controlled by ActionScript and with tweens. Buttons are exactly that, buttons; however they can be used in ways other simply click and release buttons. For example, buttons can be used to create rollover graphics and text, to control the parameters of a movie, or to dynamically control a Movie Clip. Movie Clips (MCs) are dynamic objects that can have tweens applied to them and can be controlled by ActionScript. We will get to this later. Symbols contain their own set of layers along with their own timeline. All symbols that you create and any bitmap images that you import will be stored in the Library (Window>Library). You can add multiple instances of a symbol by dragging them from the Library, and you can edit all instances of a symbol by double clicking on it in the Library.

7. The Stage and hierarchy of symbols

The design area of the document is called the stage. The stage contains all visible objects and symbols in the Flash movie. This main level of the stage is called Scene 1 and contains what is called the parent timeline; this is the main timeline for the movie. Graphics, Buttons and Movie Clips can be thought of as sub-scenes within Scene 1 of the movie. Think of it as a hierarchy of symbols; Scene 1 contains symbols (Graphics, Buttons and Movie Clips), and those symbols can contain other symbols, which can contain other symbols and so forth. The current level of this hierarchy is shown right above the design window, and will always start with Scene 1.

8. Graphics

Graphics are useful for complex animations that don't require ActionScript. The main benefit of Graphics is that their timeline is synchronized with the parent timeline; Buttons and Movie Clips have timelines that are separate from the parent timeline. Actions that control the parent timeline will also control the timeline of Graphics, but not the timelines of Buttons or MCs.

9. Buttons

A button can be created from any graphic on the stage (Modify>Convert to Symbol; or F8). Choose 'Button' and always name all symbols you create. Double click on the button to enter into the sub-scene of the button. You will see a timeline that has four frames: Up, Over, Down, Hit. Push F6 to populate each of these states. Here you can change how the button will look when it is up, when the mouse is over it and when it is clicked on. The Hit frame controls the clickable area of the button. If your button is just text, draw a rectangle around the text in the Hit frame to create a clickable area. The Hit frame is invisible. With the *up*, *over*, and *down* states you create the visual effects of the button. What it does must be controlled with ActionScript.

10. Movie Clips

Movie Clips are created the same way as buttons, select an object and push F8. Choose Movie Clip and name it. Double click on the MC to edit its contents. As mentioned previously, MCs have their own set of layers and their own timeline. You will notice that the object you converted into the MC is now in a layer inside the MC and remains in its original form (shape, bitmap or text). You can create animations with MCs using tweens and ActionScript. You can put MCs and buttons within MCs to create sub-layers of complex actions.

11. Tweens

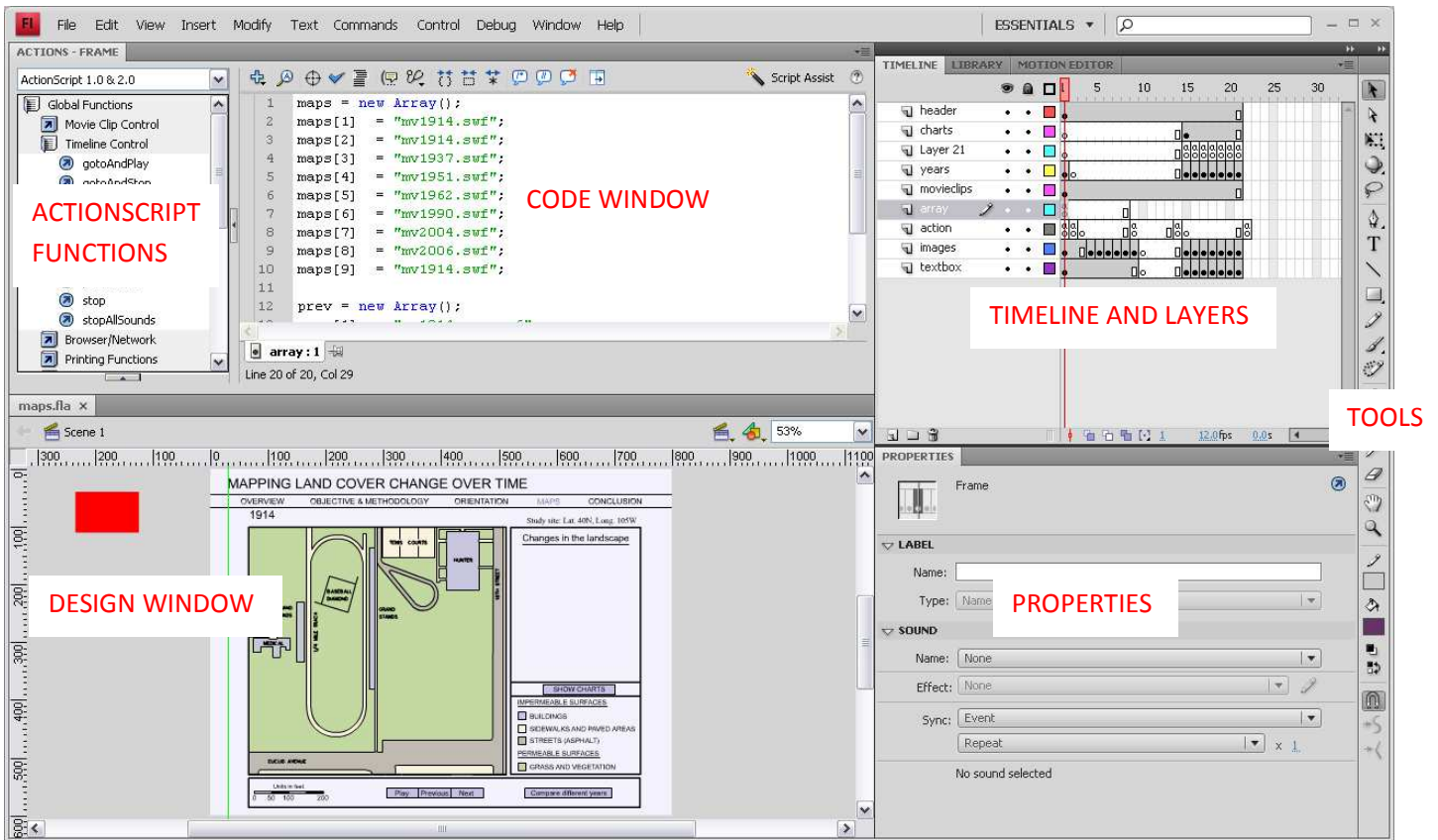
The word "tween" comes from traditional animation. Tweening is the process of filling in the frames between two different keyframes to create an animation; it's also called inbetweening. There are three types of tweens: Motion, Shape and Classic. Motion tweens can be used to control the movement of one and only one object. Shape tweens can be used to morph one object into another. Classic tweens are the most complex and allow more flexibility. Refer to the Flash Help files for step-by-step instructions of how to implement tweens.

12. Flash Movies as a basic web site

Flash can be used to create simple web pages with links between them just like in html. Once you create a few Flash movies you can link them together with ActionScript. You only need to learn a few basic commands to use buttons for navigation. Flash movies have a hierarchy of levels starting at level 0; this is a front to back hierarchy, just like in any drawing package; but these levels are dynamic, not static. The first movie you load will load at level 0; you can load other movies on top of that movie at levels 1, 2, etc. Movies have transparent backgrounds, so if you want to load a movie in the place of another, load it into level 0 and it will replace the original movie.

B. INTRODUCTION

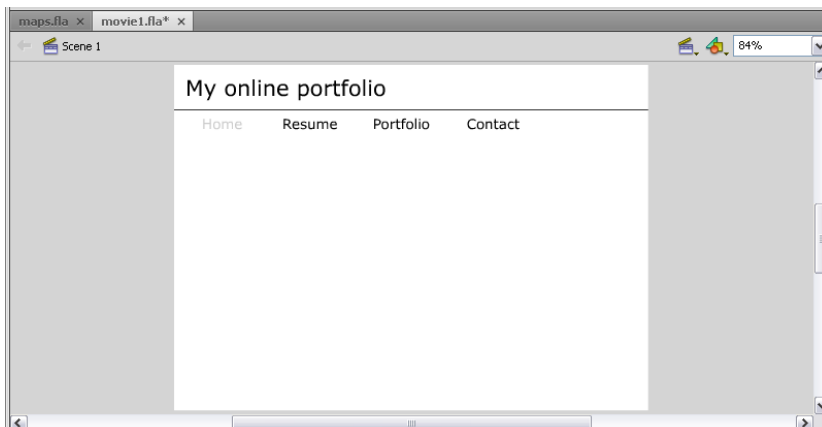
First, let's take a look at the interface:



Useful keyboard shortcuts: view at 100% → Ctrl+ 1; fit view to window → Ctrl + 2; Import to Stage → Ctrl + R; Convert to Symbol → F8; Insert frame → F5; Insert keyframe → F6; Test movie → Ctrl + Enter; Show/hide code window → F9; Open online help → F1; Paste in place → Ctrl + Shift + V; Selection tool → V; Subselection tool → A; Transform tool → Q; Eyedropper → I; Text tool → T; Zoom tool → M; momentarily use Hand tool → hold down the Space bar; Most standard Windows shortcuts also work, such as Save, Select All, Undo, Redo, Cut, Copy, Paste, etc.

1. Create the structure of a basic web page

In this exercise we will layout the structure for a basic online portfolio. Start with a new ActionScript 2.0 document. Add a title bar and some text that we will make into links to other pages in your portfolio. Start with something like this:



Save this page as index.flas. Then, go to Control>Test Movie (or push Ctrl + Enter). This will test the movie as a .swf file (and will automatically save a .swf into the same folder as the .fla). Now use Sava As to create a copy of the index.flas document; name it resume.flas, and make appropriate changes. Test resume.flas (Ctrl + Enter).

2. Create buttons

Now, convert the text you want to be links into buttons. Click on a text object (start with Résumé) and go to Modify>Convert to Symbol (or push F8). Choose "Button" and name it. Double click on your new button to enter into its contents. In the timeline populate the *over*, *down* and *hit* frames (simply by pushing F6 to populate each with a keyframe). Draw a polygon around the text in the *hit* frame to set the clickable area (see point 9 of Overview section for more details on the *hit* frame).

3. Link pages together

Back in the main scene (Click on "Scene 1" right above the stage window to return to the main level of the movie hierarchy), click on a button and then add the following in the code window:

```
on (press) {  
  
    loadMovieNum("resume.swf",0);  
  
}
```

- See Part C for annotated code of this command.

Now, in your Resume.flas, create a link back to your homepage. Test the movie to make sure you can link back and forth between Home and Resume. Next, create and link the remaining pages of the website. Finish off your portfolio by adding content to each page.

C. ACTIONSCRIPT

Basic ActionScript 2.0 commands: (all code is in *blue italics*, so you know what to copy and paste. // (double slash) will comment out text on the rest of that line so you can explain what you are doing (to comment out multiple lines begin with /* and end with */). This is helpful when someone else uses your code and to remind yourself what you were doing). ActionScript is added in the coding window. ActionScript 2.0 can be put in frames, buttons and movie clips.

1. **Frames:** you can control a movie by placing code simply in a frame. Make sure you don't have any objects selected. When you add code to a frame, you will see a little "a" in that frame in the timeline.

Note: Actions can only be put in keyframes (not regular frames).

stop(); // this will stop your movie at this frame.

or

gotoAndPlay(1); // this will go to frame 1 and play.

2. **Buttons:** You can use buttons to create links and to control the timeline.
 - o Create a link to another movie; make sure the button is selected.

on (press) {

loadMovieNum("myMovie.swf",0); / this command allows you to load a target movie and choose the level to load it into. Here are the parameters loadMovieNum(url,level,method): url → put the relative path (location) of the movie in quotes; level → choose level 0 or higher; method → this is for specifying an HTTP method (omit this parameter). */*

}

- o Create a link to a website (URL)

on (press) {

getURL("http://www.google.com",_blank); // put the URL in quotes; add a target if needed (_blank,_self,_top)

}

- o Go to a point along the timeline and stop

on (press) {

gotoAndStop(1);

}

3. **Movie Clips:** you can control MCs with ActionScript. Make an MC a click and drag object.

on (press) {

startDrag(this); / the parameter "this" refers the MC containing the code. You could have the startDrag command control a different MC. */*

}

on (release) {

```
stopDrag();
```

```
}
```

4. **Buttons that control MCs:** Make a button toggle an MC on and off.

- a) Create an MC and put “ball” in the *instance name* field of the properties window, and then put this code in the MC.

```
onClipEvent (load) {
```

```
    _root.viz = true; /* _root makes an action global, meaning that it affects and is affected by all parts of the movie. .viz is simply a Boolean variable that I created (you can name it anything that isn't a command in AS). */
```

```
}
```

- b) Create a button, and then put this code into it to control the MC called “ball”.

```
on (press) {
```

```
    if (viz){ // Here I'm using a simple if statement. This means: if "viz" is true, then do the following.
```

```
        _root.ball._visible = 0; /* Use _root to make it a global command. Add .ball to call the MC. Add the command _visible is a Boolean command (0 is false, 1 is true). You always link these commands together with dots (.). */
```

```
        viz = false; /* Set the variable I created to false, so that next time I click on the button it will do the "else" part of the statement. */
```

```
    }
```

```
    else { // If "viz" is false then do the following.
```

```
        _root.ball._visible = 1; // Make the ball visible
```

```
        viz = true; // Set the variable back to true.
```

```
    }
```

```
}
```