Creating Timeline Based Animation

In this unit, you will build an animation in Flash and use best practices in development for mobile phones.

Adobe

Objectives

After completing this unit, you should be able to:

- ►Build an animation in Flash using motion tweening.
- ►Control the Timeline using ActionScript.
- ►Build an animation using frame-by-frame animation.
- ►Implement best practices for animation.

Creating Animations

- A frame is a snapshot of visual content which the Flash player displays at a given point in time.
- Animation refers to graphics that appear to move and transform over a period of time.
  - Animation is more or less an illusion created by displaying the contents of different frames very quickly. If the content in each frame has a slightly different appearance or position than the contents of the frame before it, the eye will think that the graphic is physically moving or changing shape.

Note: Animated documents are frequently referred to as movies..

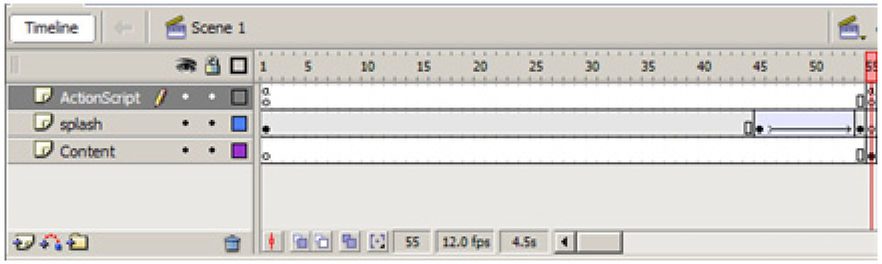
- The way in which you animate objects in Flash is to change the state of the object using Frames and Keyframes in the Timeline.
- There are several ways you can animate within Flash:
  - Create the starting and ending states of an animation and let Flash create the in between states using a motion tween.
  - Manually create a frame-by-frame animation, creating every state of the animation.

Note: There is a third type of animation, shape tweening, where you define the beginning and ending of the animation and Flash draws the animation by morphing from the starting to the ending shape. However, this type of animation is processor-intensive and not recommended for use in mobile applications.

Introducing the Timeline

The Timeline provides structure to a Flash document. The following is true about Timelines:

- ▶The Timeline is divided into frames.
- ▶The playhead determines what currently appears on the Stage in the FLA or on screen in the SWF.
- ▶By default, the playhead starts at Frame 1 and moves forward through frames until the document's last frame.
- ▶By default, the playhead continues to loop through the frames.
- ▶Layers are also part of the Timeline. Each layer has its own Timeline.



The Timeline

Creating Motion Tweens

- Instead of creating a unique drawing for each frame on a Timeline, you visually represent how a graphic should appear at the animation starting and ending points.
  - Applying a motion tween will fill in all the frames between the starting and ending points. The process of drawing the in between steps is referred to as *tweening*. Motion tweening is an efficient way to create animations.
  - A motion tween animation is also referred to as a *motion tween transition*.

Motion tween elements

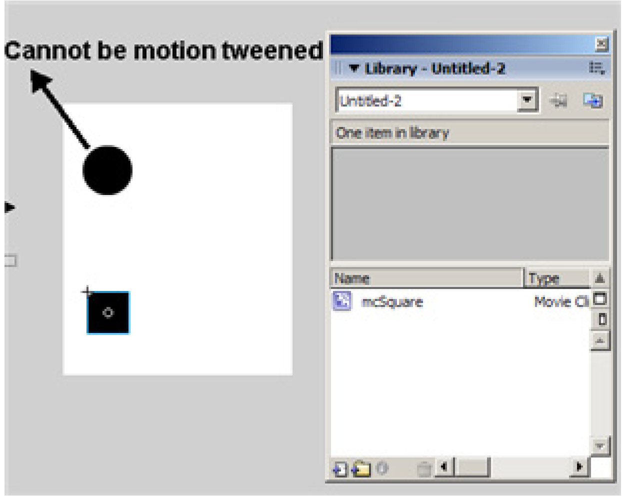
To create motion tween animations in Flash, you'll typically use:

- ►Symbols
- ►Keyframes
- ►Motion tweening

You will always use keyframes when you are producing animations on the Timeline, since keyframes are the only frames that can be edited. In the creation of an animation you might also be required to add frames to the animation.

Using symbols

- ▸Only symbols can be animated using a motion tween.
- ▸Typically, you'll animate a graphic or movie clip symbol by placing and modifying instances of it on different keyframes that you insert on a Timeline.



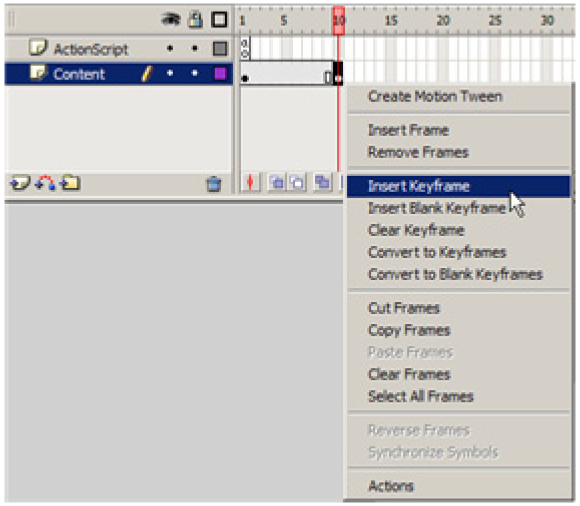
Only symbols can be motion tweened

Using symbols (continued)

- You can only tween instances of the same symbol; the symbol instances must be the only assets in the animating keyframes throughout the life of the animation.
  - For instance, you cannot have a symbol and a text asset on the same keyframe in the same layer. The text asset must occupy a keyframe in a different layer if you want it to be visible.

Inserting keyframes

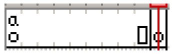
- ►A keyframe is a type of frame that you use to add symbols and other assets to a layer's timeline.
  - Once an asset is added to a keyframe on the Timeline, it is available on subsequent frames. You can insert other keyframes along a layer's Timeline to modify the asset and add motion to the frames between two keyframes. Insert keyframes using one of the following methods:
  - Select the frame, right click and choosing Insert Keyframe.
  - From the Menu Bar, select Insert > Timeline > Keyframe



inserting a keyframe

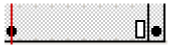
Understanding keyframe display information

- ►A hollow circle displays in a keyframe when there is no content in it.
- ►A keyframe containing nothing in it is referred to as a blank keyframe. For example, there is no content in any of the frames shown below:



hollow circles indicate no content

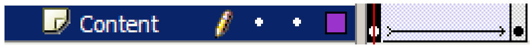
- ►A solid circle displays in a keyframe when there is content in it.
  - ▪ For example, in the layer below, you can see a solid dot in both the first and the last frames indicating that there is content in both frames.



solid circles indicate content

Understanding keyframe display information

▸The Timeline now shows a motion tween between the starting and ending points of the animation using an arrow as shown below.



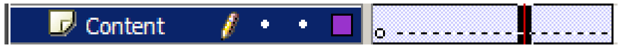
Timeline with motion tween

▸When creating a motion tween animation use the Stage to add or modify a symbol instance on keyframes at the start and end of the motion tween transition. Changes that you can affect to objects include:

- Scale, rotate, and skew the instance between the start and end keyframes.
- Change alpha, brightness, and tint over time.

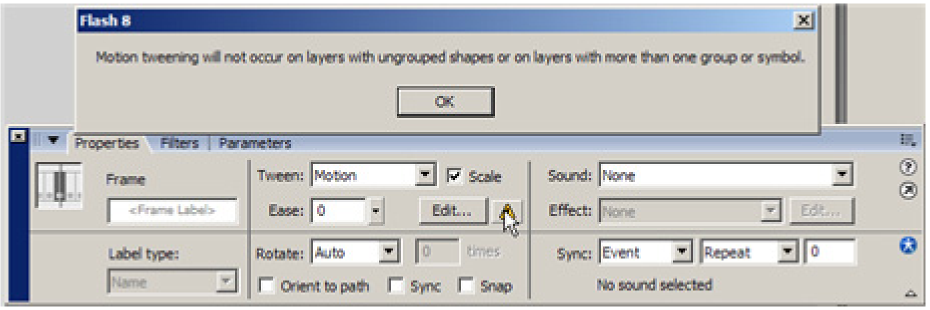
Understanding motion tweening errors

- When your motion tween is unable to be successfully created, Flash will indicate an error with a dotted line in the Timeline in the layer the animation occurs in, as in the image below.



dotted line indicates error

- If you apply motion tweening to something other than a symbol instance the Timeline and Property inspector display errors.
- A motion tween error displays an Alert icon in the Property inspector. When you select the icon, the error message below appears.


Flash 8

Motion tweening will not occur on layers with ungrouped shapes or on layers with more than one group or symbol.

Frame <Frame Label>
Label type: Name
Tween: Motion ☑ Scale
Ease: 0
Rotate: Auto 0 times
□ Orient to path □ Sync □ Snap
Sound: None
Effect: None
Sync: Event Repeat 0
No sound selected

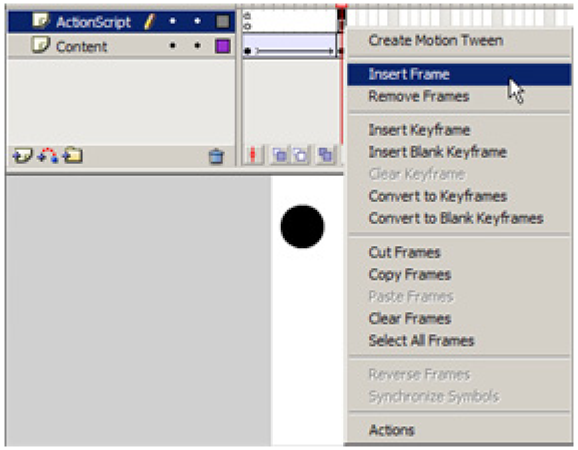
motion tween alert

Inserting frames

- An animation can extend over many frames.
- When your document contains other layers, you may need to extend their Timelines so that their content is visible over the same duration as the layers with animation.
  - Extend a layer's Timeline by inserting a frame at the same frame location as the end of the animating layers.
  - The frame will serve to extend the visibility of the asset without duplicating the asset itself. Unlike keyframes, you cannot edit the content of a normal frame.

Inserting Frames (continued)

- ▶Select a frame on a layer and select Insert > Timeline > Frame to extend a layer's Timeline.
- ▶You can select a non-keyframe frame and it will automatically select the content of the keyframe before that frame.



extending the ActionScript layer

- ▶Frames are indicated with a rectangle. Frames with content are shaded grey. Empty frames are not shaded.

Controlling the Timeline

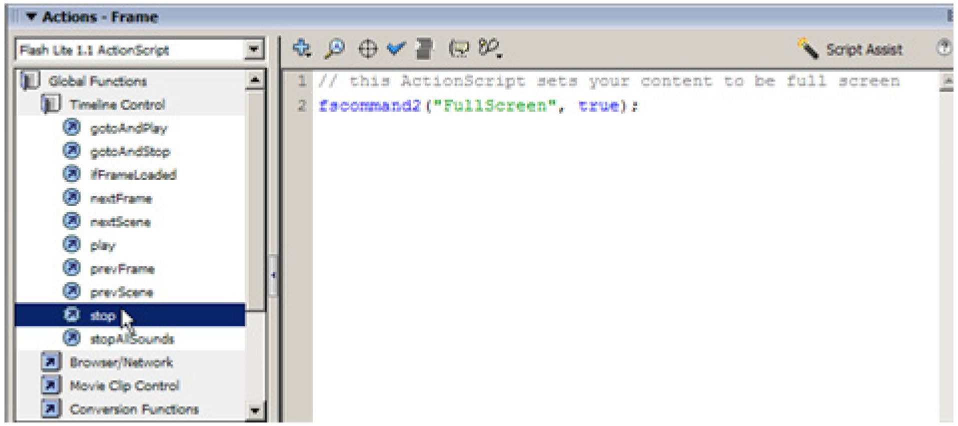
A Flash document by default:

- ▶Starts at Frame 1 and plays sequentially to the last frame.
- ▶The playhead automatically returns to Frame 1 when it reaches the end of the Timeline.

You can control the Timeline playhead using actions to stop it or send it to a specific Frame. You control the playhead using ActionScript.

Using the Actions toolbox

- You use the Actions toolbox to add actions to your document:
- The Actions toolbox arranges actions in categories by type. You can open the Actions panel by selecting Window > Actions or using the keyboard shortcut F9. Select an action type within the toolbox, expand it to view and select commands and add them to the Script window.
- In the following example, Global Functions is expanded to view commands that control movie playback. You can select the stop action by double-clicking on it and it gets added to the Script window.



```
// this ActionScript sets your content to be full screen
fscommand2("FullScreen", true);
```

The Actions toolbox

Using a stop action

Stop an animation from looping by placing the `stop()` action in a keyframe at the end of a document. This tells the movie to stop playing when it gets to that keyframe's location.

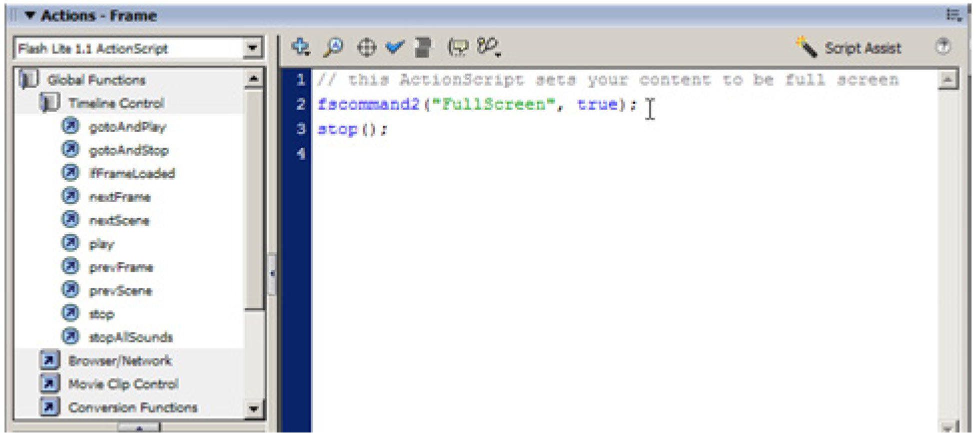
- ▶If you are adding ActionScript to the main timeline, an ActionScript layer has been created for you in the mobile template.
- ▶If you are adding ActionScript to the Timeline of a Movie Clip, you will need to create your ActionScript layer.

To add a `stop()` action to stop an animation from looping:

1. If necessary, create a new layer at the top of the Timeline panel, for the Timeline to be stopped, named `ActionScript`.
2. Select the ActionScript layer and insert a keyframe at the end of the movie.
3. Select that keyframe, select the Actions Toolbox in the Actions panel and expand Global Functions and then Timeline Control.
4. Double-click `stop()` action in the Toolbox to add it to the Script window.

Using a stop action

The Script window below contains a `stop()` action. Later in this course we will refer to an "action" by the more accurate term "function".



```
// this ActionScript sets your content to be full screen
fscommand2("FullScreen", true);
stop();
```

Adding a stop action

Walkthrough 1: Creating an Animation

In this walkthrough, you will perform the following tasks:

- ►Set up keyframes.
- ►Create a motion tween.
- ►Control the animation via ActionScript.

Creating Frame-by-Frame Animations

- ▸Frame-by-frame animation is another way to create animation in Flash.
- ▸The main benefit of frame-by-frame animation is that it gives you complete control over your animations, which means that you can make both complex and subtle changes to them.
- ▸The downside to this method is that creating individual animations in each frame can be time consuming and tedious. It can also result in very large files if symbols are not used. To understand frame-by-frame animation, you must use the Timeline.

Frame by frame animation and symbols

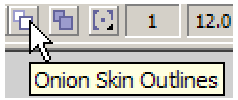
- Although you can modify many attributes of a symbol when you create an instance of it, you cannot change the basic shape of the symbol the instance is based upon.
- While you can create frame -by-frame animations of both symbol- and nonsymbol-based elements, you cannot create a frame-by-frame animation of a symbol-based element if you need to change its basic shape in the course of the animation.

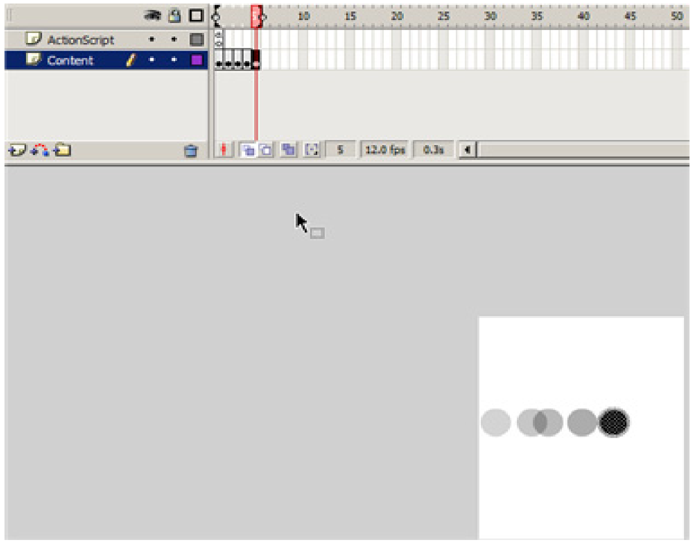
Onion skinning

▸The Onion Skinning feature shows an outline of the content in a range of frames.

- For example, if you create an animation of a ball moving across the stage, you can see where the ball is in each frame by using the OnionSkinning feature. You can turn onion skinning on and off by clicking on the onion skinning icon at the bottom of the Timeline, as shown here.

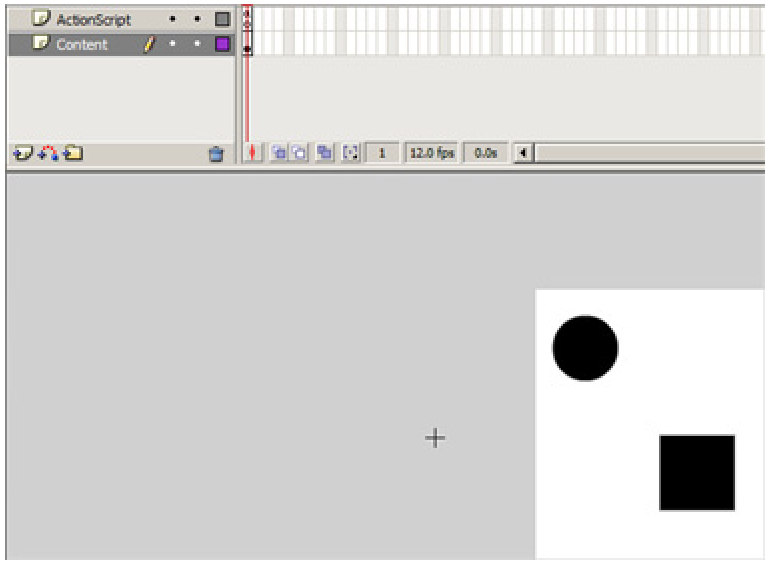
The Onion Skinning icon





Using distribute to layers

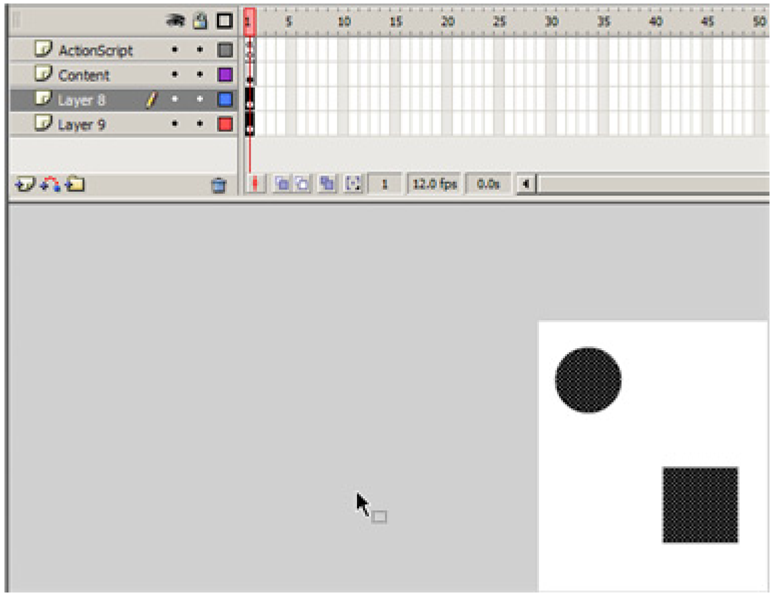
- You can use the Distribute to Layers command to help place elements on their own layer. In the following example, using Modify > Timeline > Distribute to Layers puts the circle and square on separate layers.



two assets on one layer

Using distribute to layers

The separate layers are shown in the Timeline.



Objects distributed to Layers

Walkthrough 2: Creating a Frame-by-Frame Animation

In this walkthrough, you will perform the following tasks:

- Create a frame by frame animation with Text
- Make the word Flash move across the screen

Balancing Quality and Performance

To tune your animations for better performance for mobile devices, you can do the following:

- Use the `fscommand2(setQuality)` command
- Import a bitmap gradient

Using FSCommand2(setQuality)

The `SetQuality` command sets the quality of the rendering of the animation.

```
fscommand2( "SetQuality", “quality”)
```

The value of the quality argument can be:

- high
- medium
- low

When the quality is set to high, you are able to get the most out of your graphics. When the quality is set to medium or low, animations can be more efficient, but with less quality of the graphics.

The `SetQuality` command is executed immediately upon invocation. Some handset can only render animations at one level, if this function is not supported, a value of -1 is returned. .

Importing a bitmap gradient

- Gradients created in Flash can be extremely processor intensive because they are complex vectors.
- Animation is also processor intensive and if combined with a vector gradient it can take a lot of processing power.
- If you are going to use vectors and animation together, it is a best practice to create the gradients in an external tool and then import that bitmap into Flash. You can use the dithering features of a product such as Adobe Photoshop to avoid banding.

Walkthrough 3: Create a Best Practice Animation

In this walkthrough, you will perform the following tasks:

- Import a bitmap gradient to replace a vector.
- Set the quality of the animation.

Animation Best Practice: Close-up is better

For animations focus on close-ups, while avoiding panoramic views and pans.

- ►A lot of detail will be lost on the small screen, and as great as your animation may look on your desktop computer, if it can't be seen on the mobile phone the user will not appreciate it.

Animation Best Practice: Simpler is better

- The same principle holds true for the artwork itself-the simpler the better. Stylized cartoonish characters will come across much better than realistic looking ones; the details will be lost in translation. The more you'll try to squeeze onto the small screen, the more you'll lose. Instead, focus on what is really important and leave out space fillers.
- Do not have too many things going on at the same time.
- Avoid performance killers such as alpha transparencies and gradients. They create a better-looking animation, but they also affect performance. Besides, on a small phone display, much of the effect is lost anyway; it may even look worse.
- Simpler graphics will often look better and perform better on the small screen.

Animation Best Practice: Hand tune your animations

- Hand-tuning refers to the process of manually adjusting your animation keyframe by keyframe.
- A lot of motion in real life doesn't happen in even increments, nor does it follow an easing curve. Even if it does, it often is not perceived that way. You need to trick the human eye, so the animation appears where the eye expects it to be, not where it should be placed according to some mathematically correct calculations.

Animation Best Practice: Avoid overlapping tweens

- When Flash draws a region, it does so by defining a rectangular bounding box around the area.
- You can optimize this drawing by getting that rectangle as small as possible. This means that if you can, you should avoid overlapping tweens since Flash will see the merged area as a single rectangle resulting in a larger total region. Use Flash 8's 'show redraw region' feature within the player to optimize your tweens.

Animation Best Practice: Start your project with a frame rate of 10 to 15 fps

- Apart from making it easier for the Flash Lite player to keep up, a lower frame rate will also lead to a smaller file size. Remember, each keyframe of a symbol adds approximately 12 bytes or more to a SWF movie. This may not sound like much, but it can add up, especially in the case of animations.

Animation Best Practice: Place your artwork in symbols

- Optimize the symbols by selecting Modify > Shape > Optimize or by hand with the subselection tool. Remove unnecessary points, and any hidden shapes and symbols. This can be tedious work, but it will make your content look and perform better and it will reduce the SWF file size.

Animation Best Practice: Avoid pans and alpha fades

Short (five-frame) fades over a static background may work fine. If possible, work only with shapes and remove lines, including outlines around shapes. Lines are more complex to render for Flash, and thus slower. Create outlines using shapes instead if needed.

Summary

- ▸Animation generally refers to graphics that move and transform over a period of time.
- ▸The way in which you animate objects in Flash is to change the state of the object using frames and keyframes in the Timeline.
- ▸The Timeline provides structure for a Flash movie with frames over which objects can animate.
- ▸The only editable frame is a keyframe.
- ▸Manipulate objects in keyframes at the beginning and ending of an animation.
- ▸Use a motion tween to let Flash fill in the animation between the starting and ending points.
- ▸Insert frames in non-animating layers to extend their Timeline to match the animation.
- ▸Create a frame-by-frame animation to hand-tune the animation.

Adobe

- Specific to mobile devices, specify the quality of the animation using the `fscommand(setQuality)` command.
- To decrease processing requirements on the handset, use a bitmap instead of a vector to achieve the look of a gradient.