Creating an Interactive Electronic Storybook with Flash MX

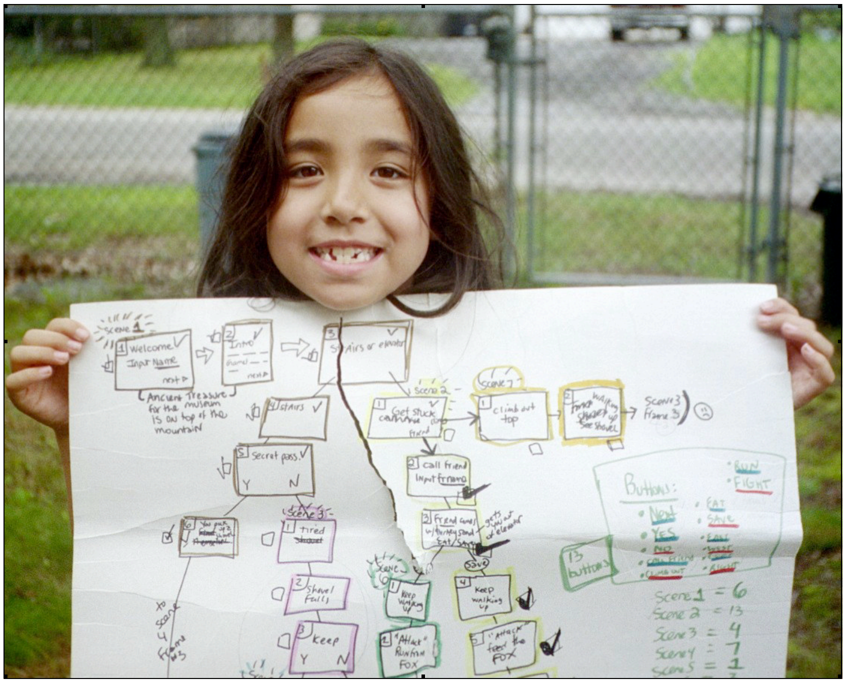
Multimedia Creation for Kids

Lesson 1: Setting Up Your Project

In this lesson you will be storyboarding your entire project. It is important to know exactly where you are going with your project before you start to sit down and design and code it! You will make a list of all the buttons you will need, and you will know how many "scenes" your finished project will have as well as what each and every frame will contain.

Get a great big piece of oak tag and draw out the frames for your entire story. Each sequence (or section) of possible outcomes should be considered a scene. You can use colored highlighters to outline boxes for frames within the same scene. Some frames sill branch out based on 2 choices (for example "yes" and "no"). You can use arrows labeled with "yes" and "no" to lead to new boxes (frames).

Here is a sample student storyboard:

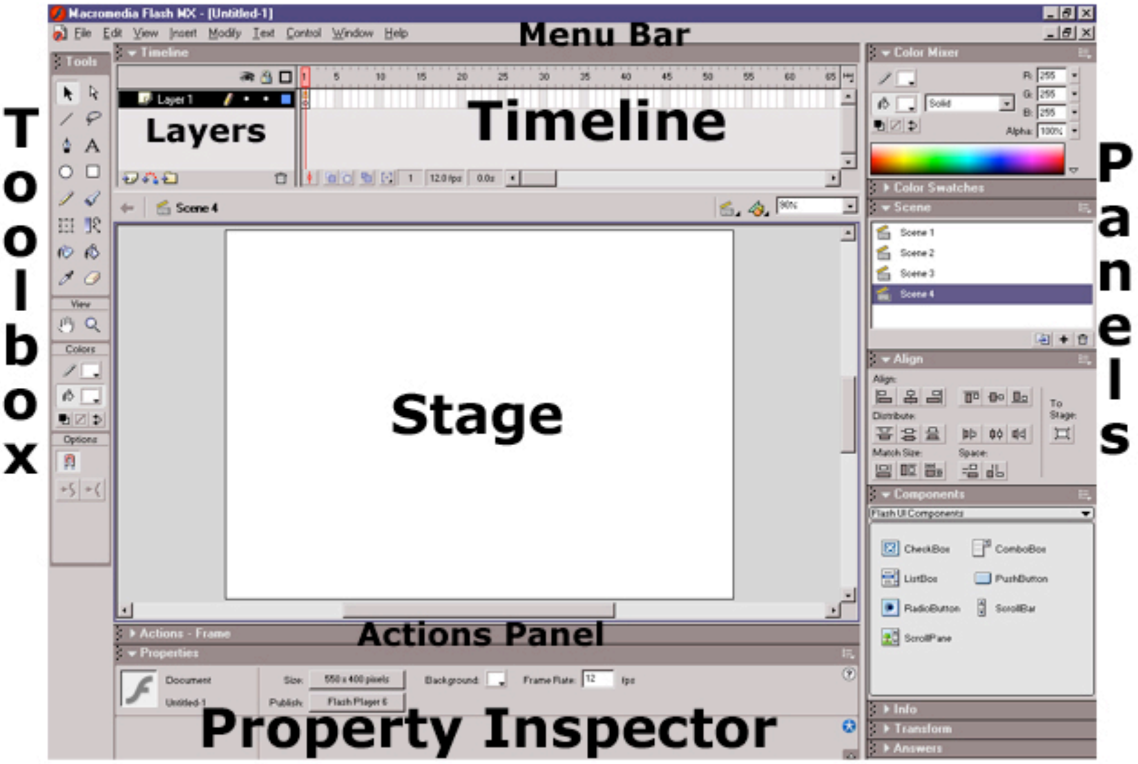


Lesson 2: Intro to Flash Animation

In this lesson you will be introduced to the Flash interface and Flash animations. You will learn how to create text & graphics in Flash and to make an animation with the help of the "Onion Skin" feature.

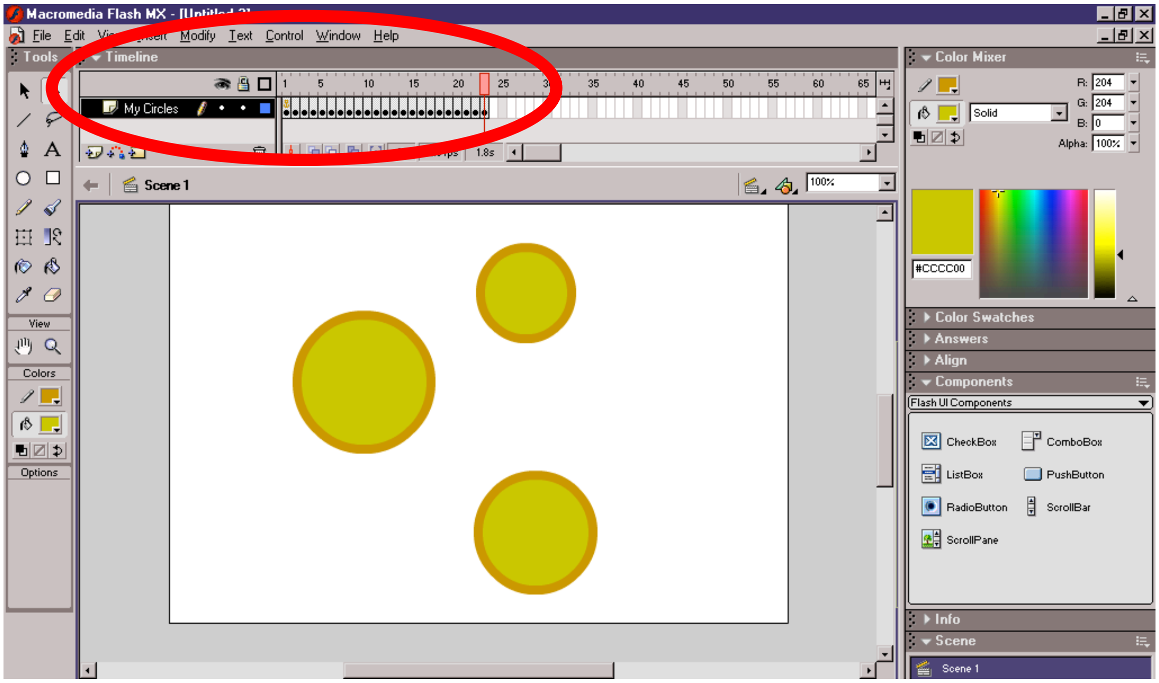
The **Flash MX Interface** is composed of:

- **Menu Bar** (dropdown menus for File, Edit, View, Insert, Modify, Text, Control, and Window)
- **Toolbox** (all the artistic tools you need, plus the selection and view tools)
- **Timeline** (insert the frames and of your animation here, and organize them into **Layers**)
- **Panels** (specialized for various functions such as alignment, color mixing, adding components, etc)
- **Stage** (the area that will display your movie)



The **stage** is your workspace. It looks like a big white sheet of paper. (You can change its size and color buy going to **Modify > Movie**). Elements that are on stage will appear on “screen” in your movie.

A real movie is just a really long filmstrip with thousands of individual frames. Playing the movie in a projector makes these frames appear on a screen at such a fast rate that viewers can see motion. In flash, the **timeline** is a series of empty slots that can hold frames. A new flash file opens with one empty frame for you to work with. You add more frames and keyframes as you see fit.

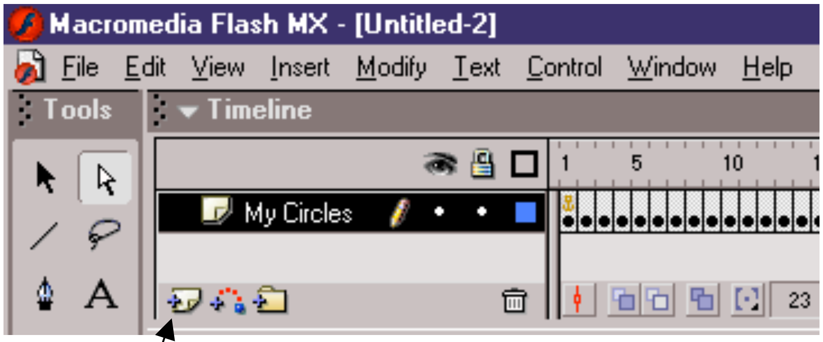


View **sample1.fla**.   Flash is smart! You can use *tweening** to make tings move. See what happens before & after a shape tween. (Use **Control > Test Movie** or just **“scrub”** the playhead in the timeline.)

**Traditional cell animation: Lead Animator makes keyframes & the inbetween artists make the frames in between.*

MS Word makes documents (.doc), illustrator makes files (.ai), pictures are mostly jpegs (.jpg) and web graphics are mostly gifs (.gif). As the flash animator you save the file as .fla.   When you want someone to see it, you export it as a .swf, .exe, or as a web page.

The timeline can have as many **layers** as you want. Think of layers as overhead transparencies stacked on top of one another. The top layer is always on top of the stack and will block parts of layers underneath it.


Macromedia Flash MX - [Untitled-2]
File Edit View Insert Modify Text Control Window Help
Tools
Timeline
1 5 10
My Circles
23

Add New Layer button

Remember, when starting a new file, with the exception of the 1st frame, the timeline is EMPTY. The "slots" you see are holders for frames. There are 3 ways to put a frame in an empty slot, depending on what you want to do:

- **F5** (**Insert > Frame**) means you want to keep the **same frame** up there.
- **F6** (**Insert > Keyframe**) means you want to insert a COPY of the prior frame so that you can alter a little. (For example, to make a cat wink. You would not redraw the cat; you would just change its eye.)
- **F7** (**Insert > Blank Keyframe**) means you want to fill the empty slot with a blank new frame.

*Tip: Use **SHIFT+F5** to remove frames from the timeline!*

MyCat.fla is an example of what we will make. It is time for you to make a new Flash file. You can so this by using **File > New**.

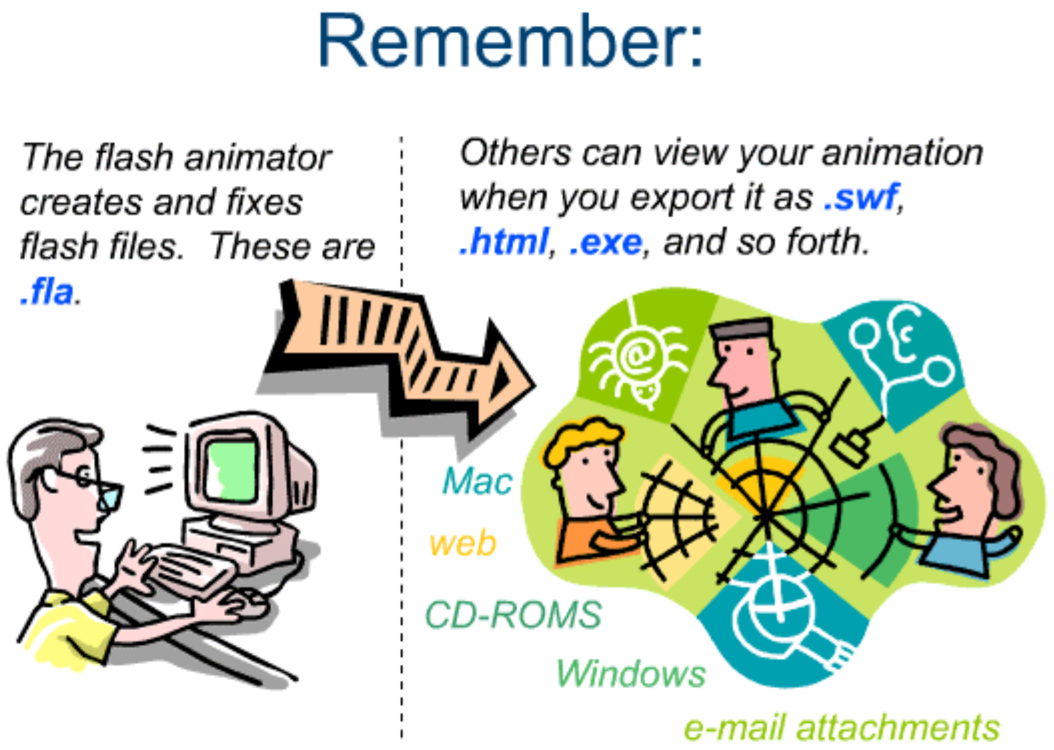
1. Make 2 layers named *Graphics* & *Text*. Draw a cat in the 1st frame of the Graphics layer. You can use the **brush** tool from the toolbox. In the timeline, keep pressing **F5** to keep your cat up there for 8 frames. Now, in frame 9, use **F6** to make a copy of this cat. **Select** & delete the dot and draw a "U" with the **pencil** tool. Use **F5** to keep this up there for 8 frames. Now use **F7** in frame 17 to insert a blank keyframe. Draw random lines ("poof"). Use **F5** to keep the "poof" visible through Frame 20.

2. In 2nd layer (*Text* Layer): Type "My cat" use **F5** to make it stay up there for 20 frames. Alternatively, you can click in frame 20 & then press **F5**.

3. Watch your movie: **Control > Test Movie**.

Cool shortcut for adding frames: Click and drag down all cells in frame 30. Press **F5**.

*Note: Whenever you make and test a movie, Flash automatically makes and saves a .swf file of your movie. If you use **Control > Test Movie** again, this .swf file is updated automatically. Remember, there is a difference between the .fla file you create as the flash animator & the .swf file.*



Remember:

The flash animator creates and fixes flash files. These are .fla.

Others can view your animation when you export it as .swf, .html, .exe, and so forth.

CREATING TEXT

When you click the Text Tool in the toolbox, the **Property Inspector** (**Window > Properties**) contains everything and anything you want to change about your text (font, color, size, and more!) Try it!

DRAWING TOOLS

Now you will explore the following tools: Line, Pen, Pencil, Oval, Rectangle, Brush, Ink bottle, Paint Bucket.

Working with basic shapes (try these):

- ☐ **Oval** verses shift+oval
- ☐ **Rectangle** verses shift+rectangle
- ☐ Rounded Corners (for rectangle)
- ☐ **Ink bottle** (stroke) & **Paint Bucket** (fill)
- ☐ Overlapping shapes with & without creating a group (**Modify > Group**)
- ☐ Selecting to change **stroke** & **fill**

*Tip: When making/using graphics, use **ALIGN** (cntrl K) to easily arrange your images. (You can "align to stage" or align objects to each other.)*

Pencil:

- ☐ Straighten
- ☐ Smooth
- ☐ Ink
- ☐ Use the Properties Panel (**Window > Properties**) to change stroke to dotted, and more!

*Tip: Press **TAB** to hide all panels on your workspace so you have room to work. Press **TAB** again when you need them!*

Brush:

- ☐ Size
- ☐ shape

Color Mixer:

Open the color mixer by going to **Window > Color Mixer**. From here you can apply linear (in a line) and radial (in a circle) gradients to any filled shape you make. You can also fill a shape with a picture (called a bitmap). Try applying different kinds of fills with the color mixer.

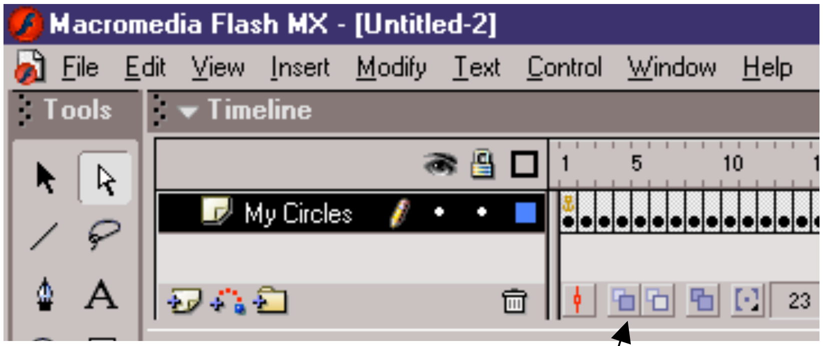
Color mixture fills:

- ☐ None

- □ Solid
- □ Linear
- □ Radial
- □ Bitmap

TRACING WITH THE ONION SKIN FEATURE

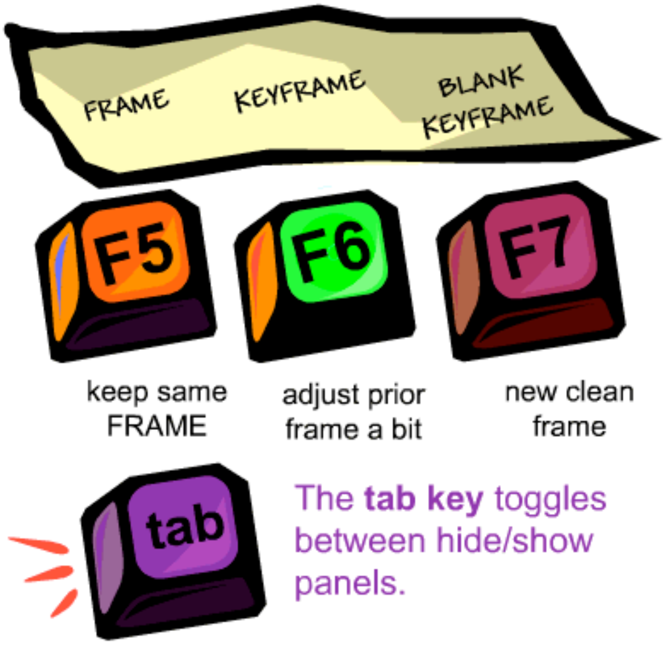
Write your name. In the timeline, hit **F7** (blank keyframe) to make a new blank frame. Use the **onion skin** to "see through" to the first frame. This allows you to trace your name!



Onion Skin button

HELPFUL KEYS

Tip: Put a skinny sticky note above the F5, F6, and F7 keys on your keyboard to remember the three different ways to add frames…



The **tab key** toggles between hide/show panels.

Lesson 3: Symbols & Scenes

In this lesson you will learn about using symbols in flash. You will create a fun symbol: A button! You will also learn how to organize your Flash movie into sections called scenes.

Working with Flash involves

SYMBOLS

- Graphics
- Movies
- Buttons

Any object you plan on using more than once should be converted to a symbol. This will help keep your file size down.

Flash has Common Libraries (play buttons, standard sounds, etc) that you can use no matter what flash project you are working on. **Window > Common Libraries >** (choices).

Each flash file you make builds its own **Library**. If you make a graphic, it will be in your Library for you to use over and over again (**instances**). Get your library on screen by hitting **cntrl+L** or **F11** or by using **Window > Library**.

*Tip: If you like a library that you made in a previous flash file, you can import the library. **Use File > Open as Library**.*

CREATING A BUTTON

Buttons have 4 states:

- **Up** (what the button looks like when it is just sitting there)
- **Over** (ex: when cursor is in the "hit area" the button may light up or hum)
- **Down** (ex: when the user clicks a button the text turns a different color)
- **Hit** (the area that will elicit the action of the button)

Use **Insert > New Symbol > Button** to start from scratch
or **Insert > Convert to Symbol > Button** for an object already on the stage.

Tips:

- You can create buttons with sound. Flash comes with a library of sounds or else you can use **File > Import** to bring in your own sounds (these will end up in the library of the current Flash file). To add sound to different states of a button, create a new layer (called "Sound") within the button timeline. Make sure you have 4 different frames for each state. Click the frame in the sound layer that you want to add sound to. Drag the sound from the library onto the stage.

  Note: If you already designed a button with 4 states (up-over-down-hit), when you make a new layer, "helpful" flash will put an empty "frame" in across all four frames. Press the control key on the keyboard and bring your cursor to the right edge of this big "frame". The cursor changes into a double arrow – this let's you resize the "frame" so that it takes up only 1 frame instead of 4. Then use F7 to make 3 new frames.

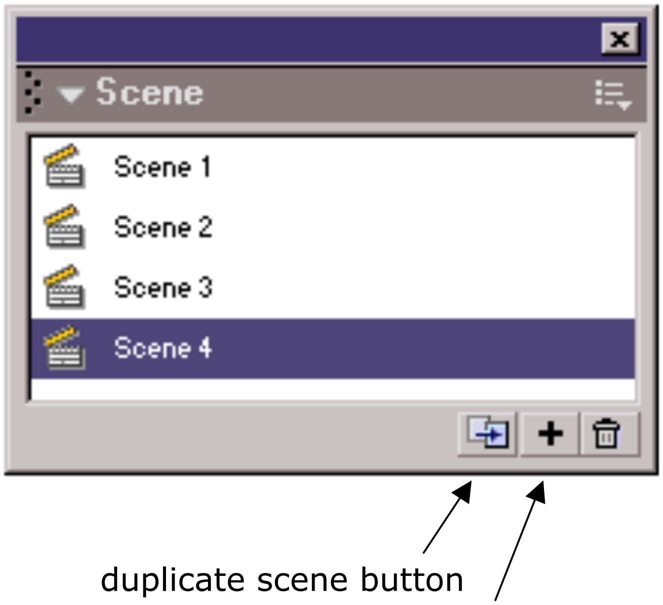
- Creating uniform buttons: *Right-click* on a button in the library & pick **Duplicate** to make a copy of the button. You can then alter (and rename) this copy instead of starting from scratch!

WORKING WITH SCENES

Scenes are a great way to organize the different sections of your project. For example: Home, About Me, Contact Info, etc. Tip: You can duplicate a scene & then modify it as a way to save time & have a consistent look to your web site.

To make a new scene you can:

1. Use **Insert > Scene**
2. With the Scene Panel open (Window > Scene) click the **+** button at the bottom right of the panel
3. Click the **Duplicate Scene** button to copy a scene you already made & then alter it.

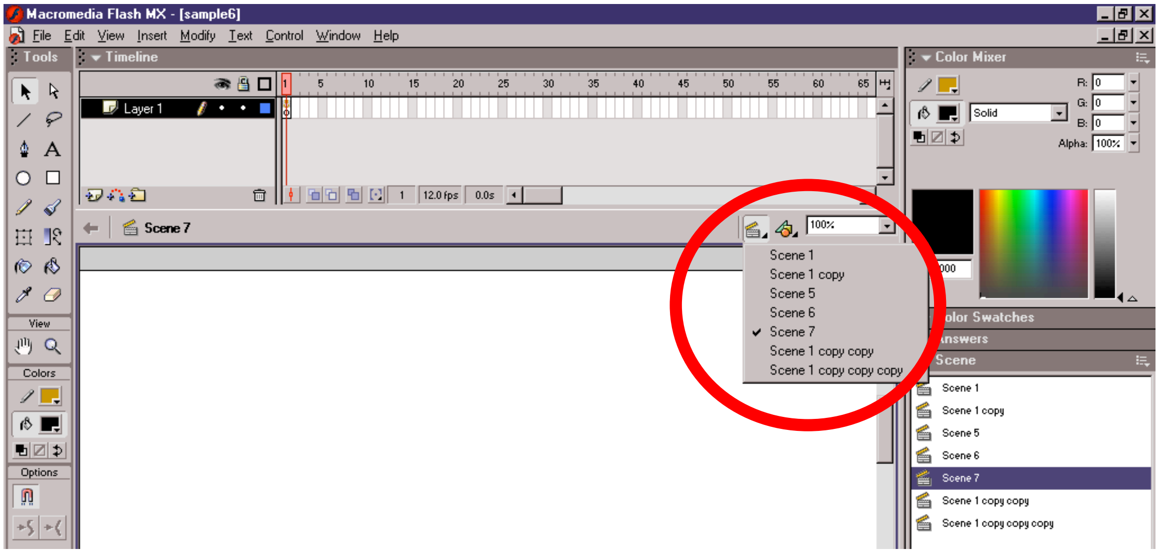


duplicate scene button

new scene button

SWITCHING BEWTWEEN SCENES

You can switch scenes by clicking the name of the scene you want in the Scene Panel (shown above). You can also click the little scene button located at the top right part of the stage area…


Macromedia Flash MX - [sample6]
File
Edit
View
Insert
Modify
Text
Control
Window
Help
Tools
Timeline
Color Mixer
Layer 1
12.0 fps
0.0s
Scene 7
100%
Scene 1
Scene 1 copy
Scene 5
Scene 6
Scene 7
Scene 1 copy copy
Scene 1 copy copy copy
Solid
R:
G:
B:
Alpha:
View
Colors
Options
olor Swatches
nswers
Scene
Scene 1
Scene 1 copy
Scene 5
Scene 6
Scene 7
Scene 1 copy copy
Scene 1 copy copy copy

Lesson 4: Intro to ActionScript

In this lesson you will be introduced to coding in ActionScript. Your goal is to look at the code here to get an idea of what it LOOKS like. You do not have to memorize anything. Are you ready to take a peek?

ActionScript is the code used in Flash. You type it directly into the **Actions Panel** (**Window > Actions**).

There's a lot of places you can put your code. These are the 2 main places where you will be putting your code:

- in the timeline (on a frame)
- on a button.

The basic structure of code can be in the format of:

```
who.when = function() {
       stuff;
       stuff;
       stuff;
}
```

Where the curly brackets **{ }** tell Flash "Do the stuff in here" and the semicolon **;** separates the stuff. The following is **FAKE** code.

```
Kira.onPointFinger = function() {
            handsOnHead();
             gotoAndStop(“hallway”);
            turnInvisible();
}
```

☞ What do you think will happen when I point my finger at Kira?

Notice the use of the period **.** in ActionScript. This is part of the famous **dot syntax** (because a period looks like a dot). Here the dot is used to separate the "who" from the "when".

Also notice the use of parentheses **()** in the code. Sometimes they are empty like:

```
stop();
```

If you click a frame in the timeline and type **`stop();`** in the Actions Panel, Flash knows exactly what to do. But other times, you need to provide Flash more specific information about what you want to accomplish.

Whenever you tell flash to **`gotoAndPlay`** another frame or scene, or to **`gotoAndStop`** at a certain frame or scene, you need a way to tell flash exactly where to go. Three examples are below:

```
gotoAndPlay(1);

gotoAndStop("CoolFrame");

gotoAndStop("Scene1",1);
```

👓 *Even though you are not trying to start a fight, the technical term for the stuff inside the parenthesis is called **arguments**. In flash, arguments are helpful! If you don't like the idea of calling the stuff in parenthesis arguments, you can call them **parameters**!*

The code below is from the e-Yearbook. This code is attached to the shiny blue button next to the word **Roster**. When the user clicks the button, we want Flash to take us to the scene called Roster and we want that scene to play because it contains 3 frames for the mousetrail effect.

```
on(release) {
    gotoAndPlay("Roster", 1);
}
```

Another basic structure of code can also be in the format of:

```
who.property = value;
```

For example, to make something invisible, or partly transparent in flash, you just adjust the property called alpha. Flash doesn't understand the word **alpha**, you have to type **_alpha** as below:

```
alien._alpha = 30;
```

Remember the dot syntax? Here, the dot is separating the "who" from a particular "property" of that who.

Let's say you have a ZAP button with the instance name zapButt and a movie clip called alien1. Look at the following code:

```
zapButt.onRelease = function() {
    alien1._alpha=50;
}
```

☞ What do you think happens to the alien with instance name alien1?

Let's say you have another button with the instance name megazapButt. Look at the code below:

```
megazapButt.onRelease = function() {
    alien1._alpha=10;
    alien2._rotation=90;
    alien3._visible=false;
}
```

☞ What do you think happens to the aliens named alien1, alien2, and alien3?

☞ Sneaky question alert! What type of flash code was used above?

a.

```
who.when = function() {
            stuff;
            stuff;
            stuff;
         }
```

b.

```
who.property = value;
```

c. both a and b

d. none of the above

As a flash developer you get to decide where you want to put your code. Note that above the two buttons with instance names of zapButt and megazapButt were controlled from the timeline. For the e-Yearbook, the code was attached directly to the Roster button. It is always your choice, but as you get more advanced, remember that if you controllyour buttons from the timeline, you will be able to control more stuff and your code will be easy to find.

Another cool part about coding in ActionScript is the way you can make a note to yourself within the code. Flash will ignore anything you type after typing **//** in the script.

```
//change this later and add in aliens 4, 5, and 6
//Right now I want my Indian Food
megazapButt.onRelease = function() {
    alien1._alpha=10;
    alien2._rotation=90;
    alien3._visible=false;
}
```

📖 ***Advice from a flash developer:*** *There are so many books about Flash. It takes a while to get used to writing your own code. Even after many months (okay, over a year) coding in flash, I still look up code before I type it. I have started a Flash Journal of all my favorite code and all of the solutions I have come up with while using*

Flash to make stuff for myself and for others. I keep track of where I got good code in case I need to look up more information later. Never worry about the price of a book. If it is filled with great information, it is worth the money. Never be shy about highlighting and marking up a great book. I usually use sticky notes for the first few chapters just in case I decide want to return it. After I realize I am going to keep it, out comes the highlighter! Don't be scared to start out in Expert Mode in the Actions Panel. If you liked Lesson 4, then that is exactly where you belong!

Lesson 5: Scripting Your Buttons

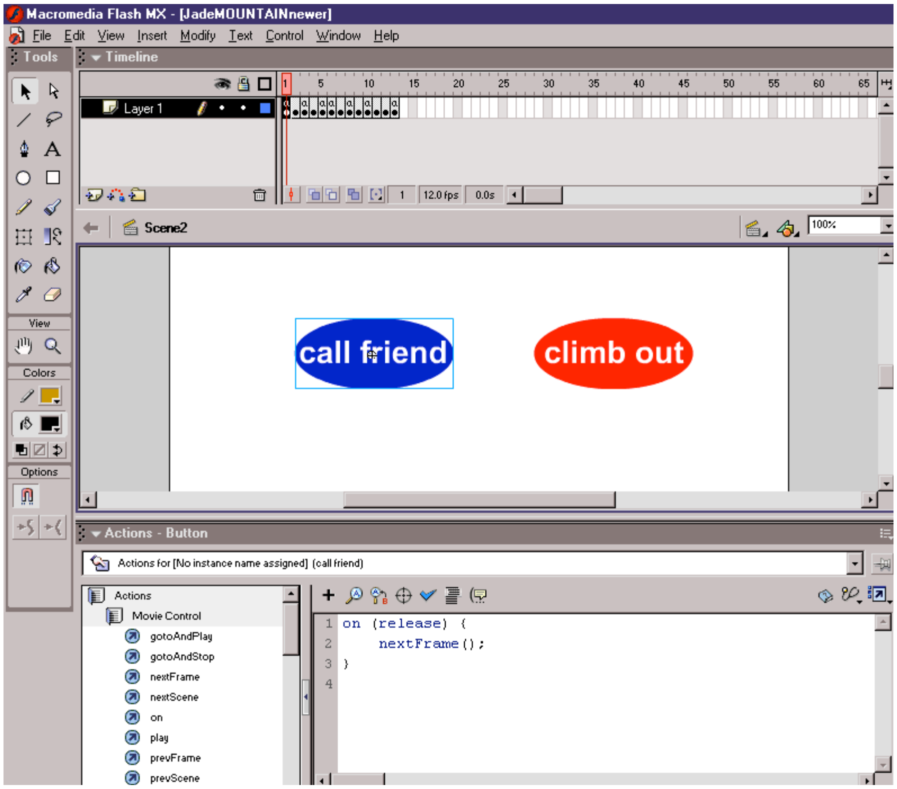
In this lesson you will be looking at how to make buttons take you to the next frame, or to a particular page in a particular scene.

For your interactive Flash Story, you will be adding code directly to your buttons.

GOING TO THE NEXT FRAME

Select the button by clicking it once. In the actions panel type:

```
on (release) {
     nextFrame();
}
```



Actions for [No instance name assigned] (call friend)

```
on (release) {
     nextFrame();
}
```

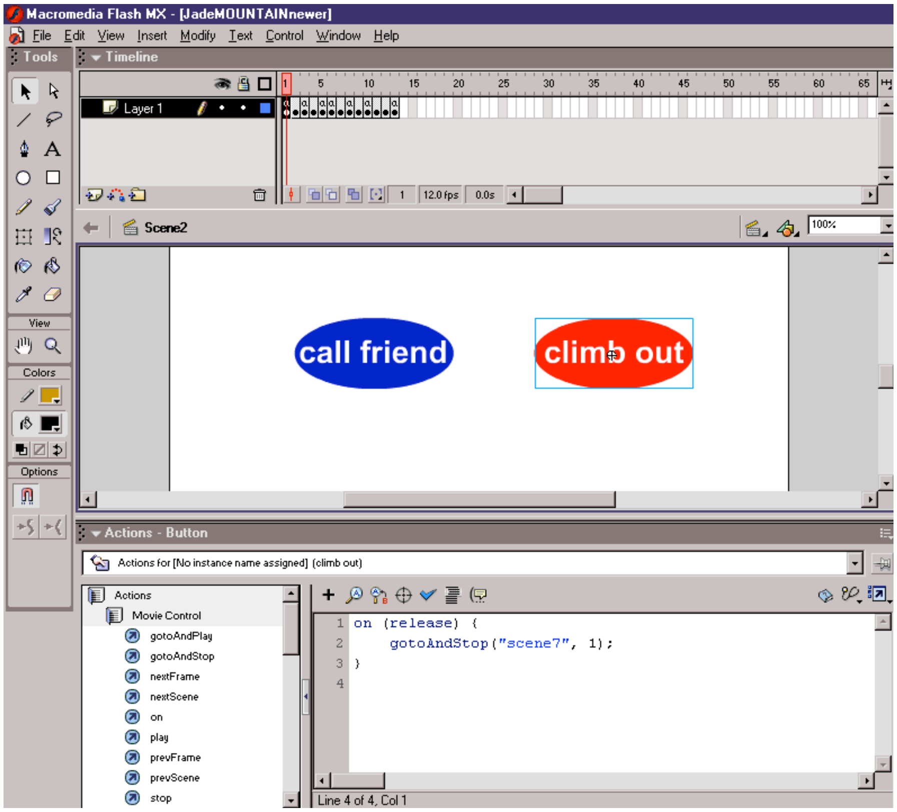
In the above example, calling a friend takes the user to the next frame of the current timeline (Scene2).

GOING TO A FRAME IN ANOTHER SCENE

Select the button by clicking it once. In the actions panel type:

```
on (release) {
     gotoAndStop("_____", ___ );
}
```

Where the first blank is the name of the scene, and the second blank ins the number of the frame you want.



Actions for [No instance name assigned] (climb out)

```
on (release) {
    gotoAndStop("scene7", 1);
}
```

Line 4 of 4, Col 1

In the above example, the code on the Climb Out button is:

```
on (release) {
     gotoAndStop("scene7", 1);
}
```

☞ Where do you think this button takes the user to?

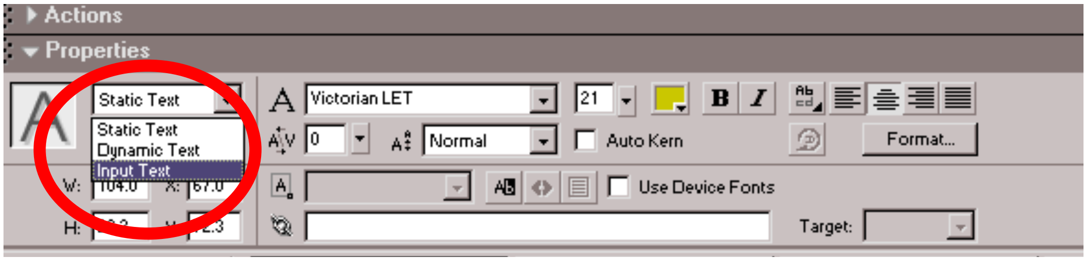
Lesson 6: Personalizing the Story

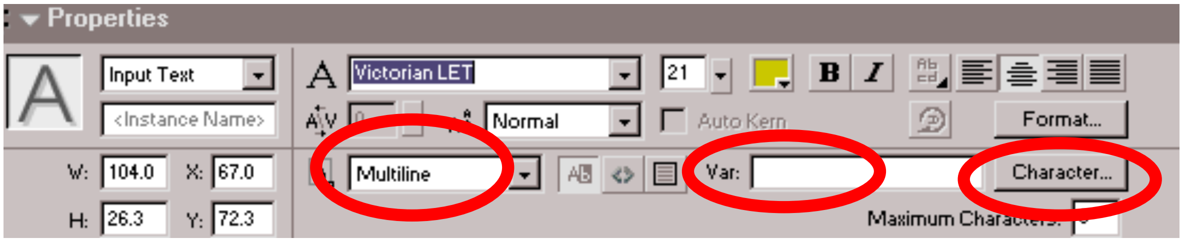
In this lesson you will learn about two new special types of text: one that lets a user give information to Flash (INPUT TEXT), and one that lets Flash give back personalized information (DYNAMIC TEXT).

Inputting Text

From the toolbox, pick the text tool. In the properties panel, you need to adjust a few settings:

- Change the dropdown menu from Static Text to **Input Text**
- The text inputted by your box will be regarded as a variable if you type a name for that variable in the field called **var:**
- Click the next dropdown menu and pick **Multiline**.
- Click the **Character...** button to make sure that all characters are allowed.



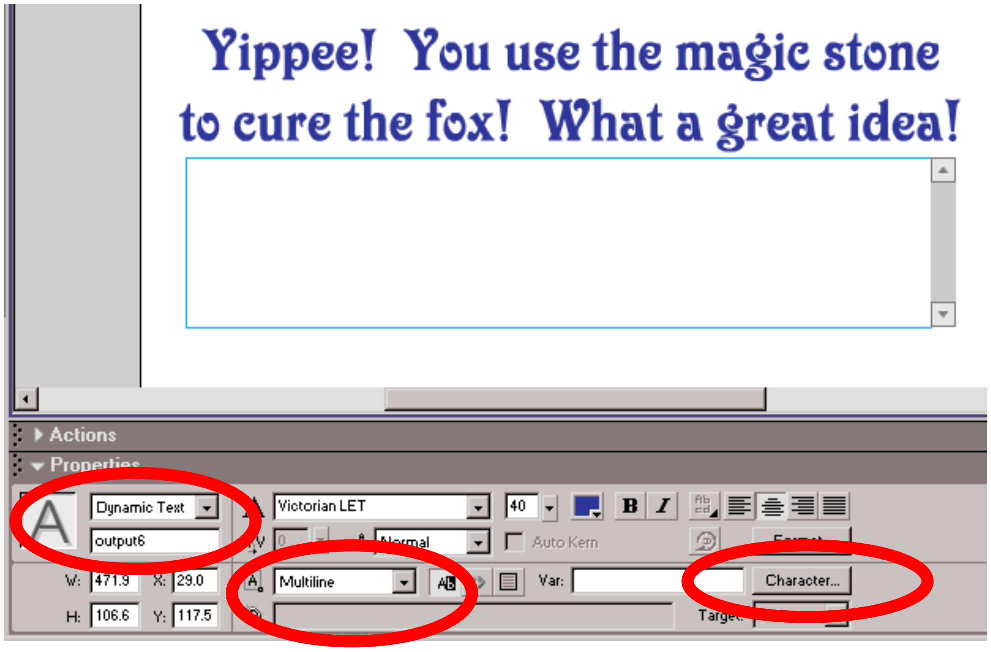


W: 104.0 X: 67.0
H: 26.3 Y: 72.3

Outputting Dynamic Text

From the toolbox, pick the text tool. In the properties panel, you need to adjust a few settings:

- Change the dropdown menu from Static Text to **Dynamic Text**
- Give your text box an **instance name**. Here, the instance name is output6.
- Click the next dropdown menu and pick **Multiline**.
- Click the **Character…** button to make sure that all characters are allowed.



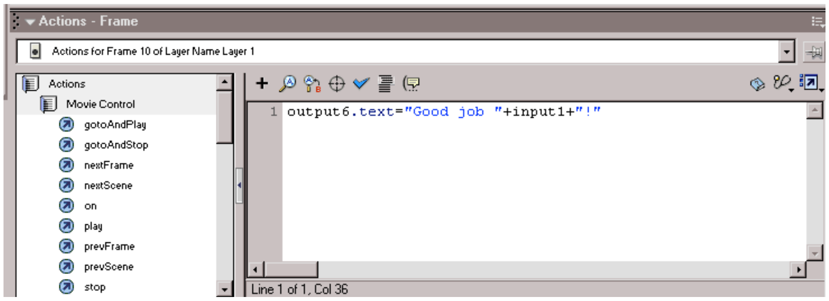
Yippee! You use the magic stone to cure the fox! What a great idea!

Dynamic Text
output6
Victorian LET
40
Normal
Auto Kern
Format...
W: 471.9 X: 29.0
H: 106.6 Y: 117.5
Multiline
Var:
Character...
Target:

Now it is time to puts some actions on the Frame. Click the frame in the timeline. In the Actions panel, type:

```
_________.text =
```

…where the blank will be the name of the text box (here we used output6).



Actions for Frame 10 of Layer Name Layer 1

```
output6.text="Good job "+input1+"!"
```

Line 1 of 1, Col 36

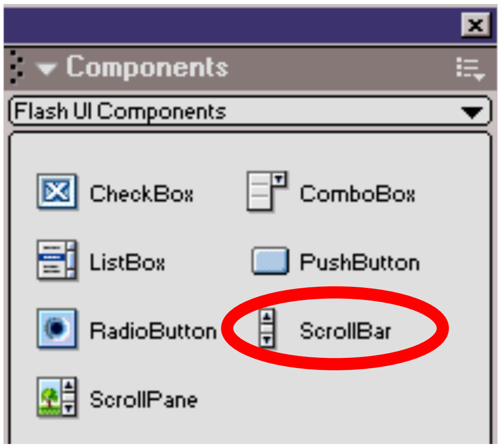
In the example, the code on the frame is:

```
output6.text="Good job "+input1+"!"
```

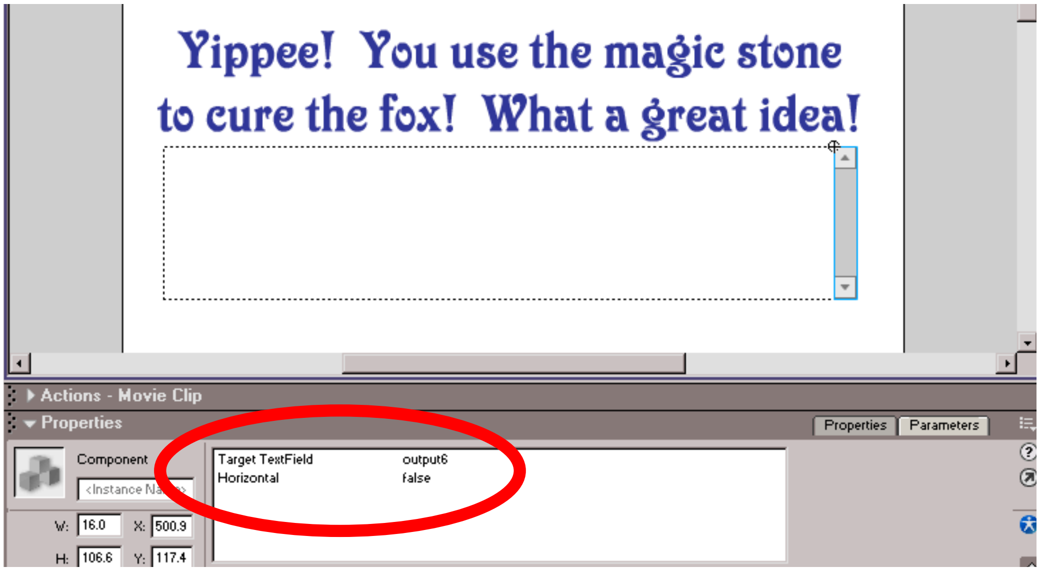
Note that this code is in the form of **`who.property = value;`** where the "who" is the textbox and the property is the actual text in the box. (Believe it or not, the text box has a bunch of properties, so you need to tell flash that you want to tell the text box what **text** it should contain. That is why we has to type **`output6.text=`** and not just **`output6 =`**.)

The Scrollbar Component

If your Components Panel is not open, go to **Window > Components** or press Cntrl+F7 on the keyboard.



Now that your Dynamic Text box has an instance name, and while it is still selected, drag an instance of the ScrollBar component into the text box just by the right edge of the box.



Target TextField output6
Horizontal false

In the above example, Flash figured out that it should couple the scrollbar with output6.

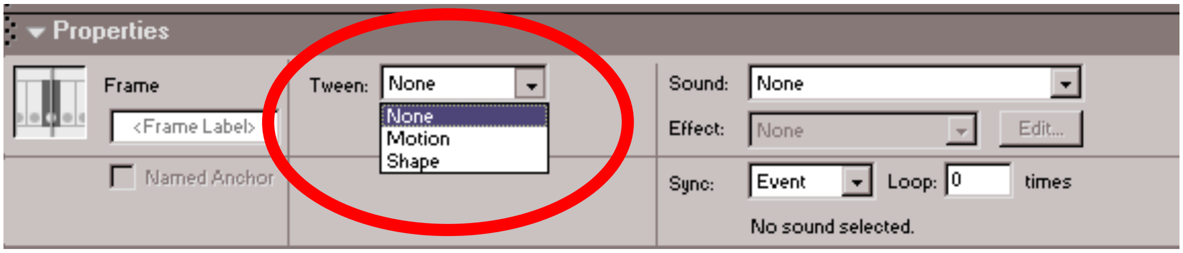
Lesson 7: Motion Tweening & Masks

In this lesson you will learn about motion tweening (making a symbol move) and masking (hiding part of your flash animation).

MOTION TWEENING

Make a new file (**File > New**). Make a star with the Pen Tool. Fix individual points with the **Subselect Tool** (white arrow in tool box). Next, convert your star to a **Graphic** symbol by using **Insert > Convert to Symbol** or pressing **F8**.

We will use **Motion Tweening** (set this in the **Properties Panel**).



Frame: <Frame Label>

□ Named Anchor

Tween: None

Sound: None

Effect: None

Sync: Event

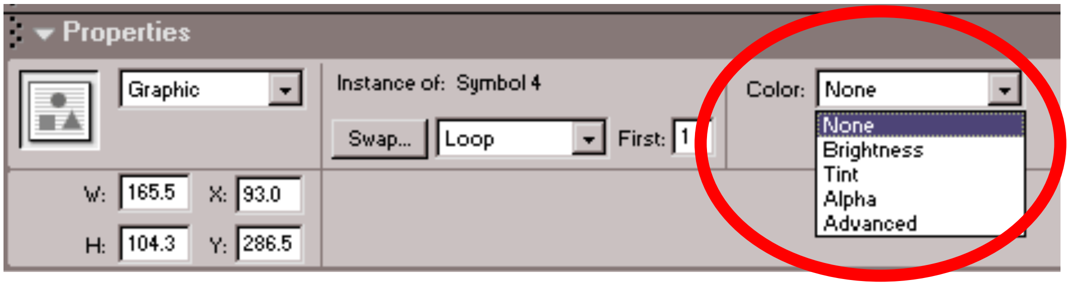
Loop: 0 times

No sound selected.

Because you converted your star to a symbol, you can drag copies (instances) of it out of your library. Create 4 layers and drag a star into Frame 1 of each layer. Create a 5th layer with any text you want.

- □ Layer 1: Star changes position. (F6 in Frame 30 & move)
- □ Layer 2: Star disappears. (F6 in Frame 30 & change alpha*)
- □ Layer 3: Star rotates. (F6 in Frame 30 & rotate)
- □ Layer 4: Star grows. (F6 in Frame 30 & scale)
- □ Layer 5: Text flies in and grows.

***Alpha** is used to tell how "see through" or transparent your symbol is. When your symbol is selected, you can use the Color dropdown menu in the Properties Panel to choose and set the Alpha value.



sample3.fla shows a finished version of this example.

MOTION GUIDES

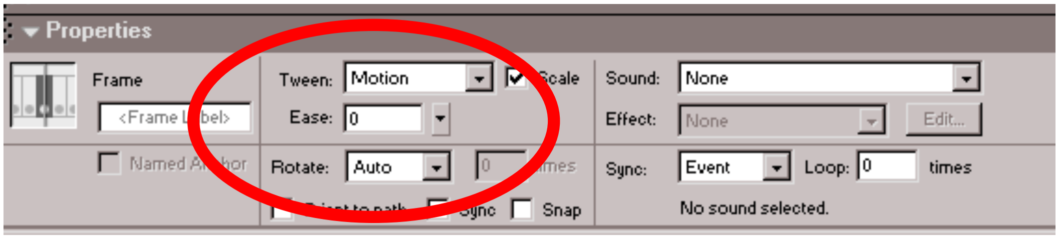
Instead of having a motion tween follow a straight line, you can define the path for your object (or text) to take.

1. In a new file, create a motion tween Layer 1.
2. Select Layer 1 and then choose **Insert > Motion Guide**.
3. Draw a curved line in the Guide Layer (use a fat pencil).
4. Make sure "**Snap to Object**" is selected in the **Options** section of your **Toolbox**.
5. In Layer 1/ frame 1 drag the instance of your object over to the beginning of the curve until you see a small circle appear. Let go & the object will snap to the curve. In the last frame, drag the instance of the object over to the end of the curve until the little circle appears.

Sample6.fla shows text that is guided by a curve.

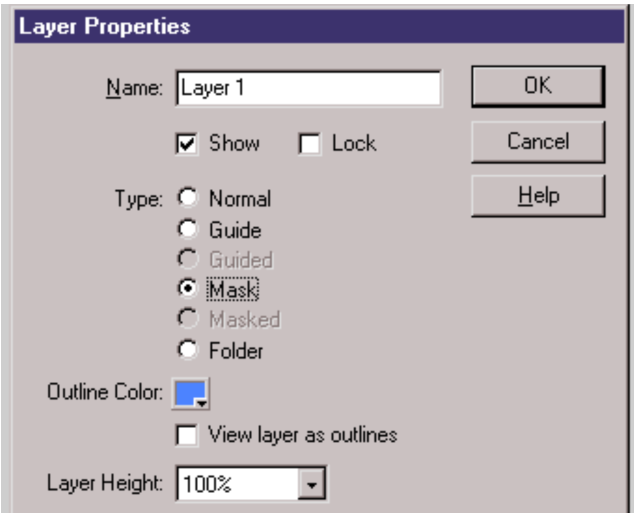
EASING

This is an option in the Frame Panel. **Easing in** (negative numbers) causes your tween to start slow & speed up. **Easing out** (positive numbers) starts fast & slows down.



MASKS

Look at **sample4.fla**. Here a circle is used as a **mask** on a photograph.   Go to **Modify > Layer** (or right-click on a layer) to get the layer properties dialog box:



Layer Properties

Name: Layer 1

☑ Show ☐ Lock

Type:
○ Normal
○ Guide
○ Guided
● Mask
○ Masked
○ Folder

Outline Color:

☐ View layer as outlines

Layer Height: 100%

Modify > Layer is used to designate the **mask** and the **masked layers** (and also the "Normal" and "Guide" Layers).

Open **Sample5.fla**. Select the "oval" layer. Use **Modify > Layer** and choose **Mask**. Select the "dancers" layer. Use **Modify > Layer** and choose **Masked**. Test your movie.