This tutorial is a continuation of tutorial #1 - a very basic introduction to building a flash web site. The goal of this tutorial is to add more features and functionality to your flash web site. We will go over how to “grow” the home page via animation, add rollovers and functionality to the nav bar, and make sections with unique content for each of our sub pages.

setup

If you completed tutorial #1, your flash file should be all set up, ready for the addition of some interactivity.

Your timeline should look something like this:

![Timeline Image]

Your canvas should look something like this:
First, let's work on the transition from our intro to our home page. Right now the intro logo animation fades out and then the homepage frame appears instantly all at once. Let's animate this framework.

To begin, I need to make some space for the animation to occur at the beginning of my main scene. Therefore I am going to move over “home” and all key frames at frame # 1, and simply move them over to frame 30. Likewise, I will push the content over to frame 60 or so.

I will then decide which items I will animate and which I will not. Those that I will animate, I will create keyframes for them somewhere in the span of frames 1-30. For this tutorial, I am going to mask in the white lines, animate the logo sliding in from the left, and animate the grey nav color in from the top, in that order. The text on the nav will appear instantly on frame 30, but everything else will animate in.

You new timeline should looks something like this.

2. For the lines I will add a mask layer, and animate the mask. For the logo I will convert the logo to a symbol (INSERT > CONVERT TO SYMBOL) and name it g_logo. Then I will place it inside of my graphics folder in my library. For the nav back I will also animate a mask layer.
When I am done my timeline will look like this:

3. Since the content section of my home page is blank I need to add some content. If you remember, we had some content inside of our illustrator file, but it did not get imported because the layer was turned off when we exported out our swf from Illustrator. This is fine, because Illustrator does a poor job of exporting text blocks, therefore I need to do my text blocks from scratch inside of Flash. To do this I will use the text from “about us” page that we put on the layer entitled “txt_content”. I will slide this keyframe over to frame 30 where my “home” label is located, then fix the text box and the header to say “Welcome”.

When you are done your canvas should look like this and your movie should stop on this frame (labeled “home”):

4. Next, let’s iterate out our since content sections: section 1, 2, 3, 4, about us, and contact. I usually put them in frames right next to each other, but for this tutorial we will space them out by a few frames to that you can see their labels on the labels layer.
To do this you must make a label for each section, add a stop action for each section, and add some unique content for each section on the txt_content layer.

In order to further differentiate each sub layer, I am also going to include a background color for each sub page. An easy way to change the color of a background is to make it a graphic, then re-use it multiple times and change its color properties each time you use it.

To do this, you need to convert the green shape on the bg_body layer into a symbol (I named it g_subBack), then add keyframes of it for each section. For each section, select the g_subBack object on your canvas and change its color using the Color pull down menu inside your “Properties” palette and then hit the “Settings….” button.

When you are done, each of your sub pages should have a unique color background and content, and your new timeline should look something like this:

[Image of timeline with sections labeled]

and an example of a sub page may look something like this:
5. Next, let's add some functionality to the top navigation to make our site work. To do this, I will need to create a “buttons” layer – which I will place just below lines layer. I usually place my buttons layer just below my labels layer, but since I may want to add a rollover state that appears beneath the thin white lines of my design, I am going to place my buttons layer below the lines layer – this will make sense once we make our rollover button.

6. The goal for making buttons is to use a few graphical assets as possible and to utilize flash’s built-in button states. Therefore, let’s try to make one button that we can reuse for all of the top nav sections.

To begin, let’s make a red rectangle, with no stroke, and draw it right on top of our home top nav area. If you have snapping-to-objects turned on (VIEW > SNAP TO OBJECTS) this is super easy because of our already existing white lines around the top nav.

Once you draw your red shape, let’s convert it to a symbol, and this time name it “but_universal” and make it a button, not a graphic. Be sure to place it inside of your
buttons folder inside of your library.

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>_buttons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>but_universal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_graphics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_logo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_navBack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_subBack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_movieclips</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>txt_logo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Next, let’s clean up this button and give it a rollover state. To do this, I will double-click on it or open it up. On the timeline you will notice Flash’s 4 frame set-up for all buttons: Up, Over, Down, and Hit.

First, make a keyframe for each state. Then, delete the contents of the Up state (make it an empty keyframe). We do this for many reasons, but the most obvious is that since this is simply a button that will reside above our top nav graphics, it shouldn’t alter our “down” state. This has already been set up with our initial design, right? Instead, we need to make a rollover state (Over), a down state or the state of the button when you actually click it with your mouse (DOWN), and lastly we need to define what area of the button is clickable and what area is not – this is the HIT state. All this does is tell Flash what part of this button is “clickable”. You never actually see your hit state as a user, instead you sort of “feel it” – therefore, you can ignore the color of the state.

8. Next, give your over and down state new colors. For my design, I am choosing a very light grey (RGB: 204, 204, 204) and making it very transparent (25%) so that it will allow the text beneath it to show through. For the Down state I am keeping it red, but taking down the transparency to 50%.

When you back up into your main scene view, you’ll notice that the button is now a transparent green color on your canvas. This is a very good thing. This is Flash’s way of identifying that you have correctly created a button with an empty “Up” state. Flash likes this and therefore wants to make it easy for you to find it on your canvas – therefore every button with an empty “Up” state is treated as green for our convenience. Thank you, Macromedia. We love you.

9. After you dry your eye following that touching moment, test this rollover button out by testing your movie.

10. I will next need to iterate this button out for each of the sections of the top nav. The easiest way to do this is to copy-and-paste, or “duplicate”, the buttons across your canvas/nav bar. As you do this, you should resize each one to fit the space of the nav bar. Be sure to be exact (zoom in while doing this for accuracy) and be sure to not overlap your buttons at all.

Your canvas should then look like this:
11. Lastly, let’s add some functionality to our buttons. We need give each button the ability to jump from one “label” of our timeline to another – this is how our simple little flash web site will work.

To do this, we will select our second button (section 1) and then open up our actions panel. Since we are using an old version of Flash (MX), we will turn on our “Normal View” inside of our actions panel. You do this by clicking on the sub-menu icon in the upper right hand corner of the Actions panel:

12. Next, click on the plus (+) arrow and select ACTIONS > MOVIE CONTROL > ON.
13. This will place an on (release) {} action into out actions panel.

```
on (release) {
}
```

14. Next, we need to add a “goto” action by hitting the plus (+) arrow again and selecting ACTIONS > MOVIE CONTROL > GOTO.

This will add the line gotoAndPlay (); into the actions panel. We need to then change it to gotoAndStop(); and select the label that we want it to go to, in this case it is “section1”.

The easiest way to do this is via the normal view pull down menu. However, you should get used to coding this by hand using the “Expert View” settings.

The final code should look like this:

```
on (release) {
    gotoAndStop("section1");
}
```
16. Next, test your move to make sure that it works.

17. If all is well, then you need to iterate this out for each button. The easiest way to do this is simply to copy and paste the code/actions onto each button. You need to select each button on your canvas in order to see the actions settings for each button inside of the actions panel.

18. Lastly, test your movie. You have just completed tutorial #2 and you have just made a very simple flash web site.

In our next tutorial we will revisit the structure of our site and begin using movie clips so that we can add more advanced features to our site, such as transitions in between sections.