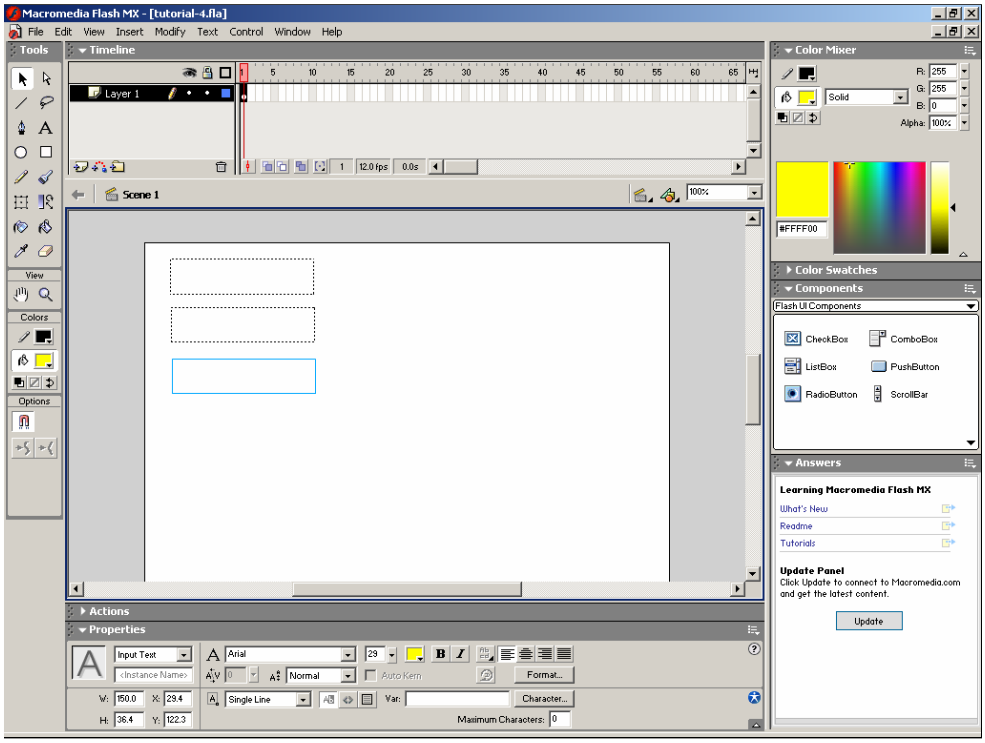
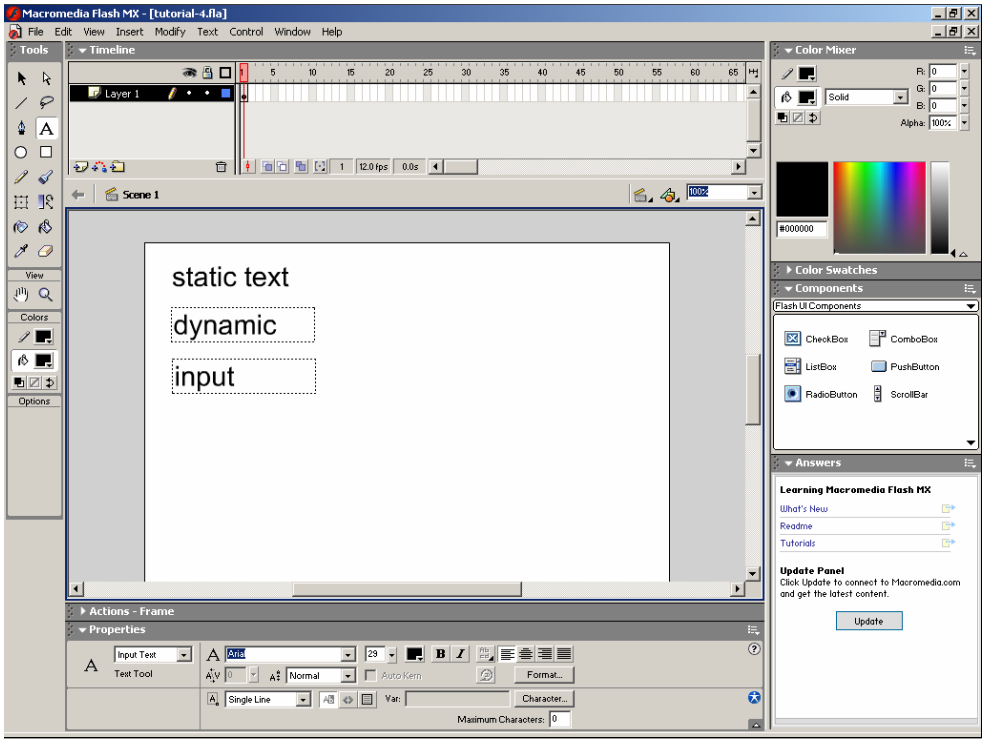
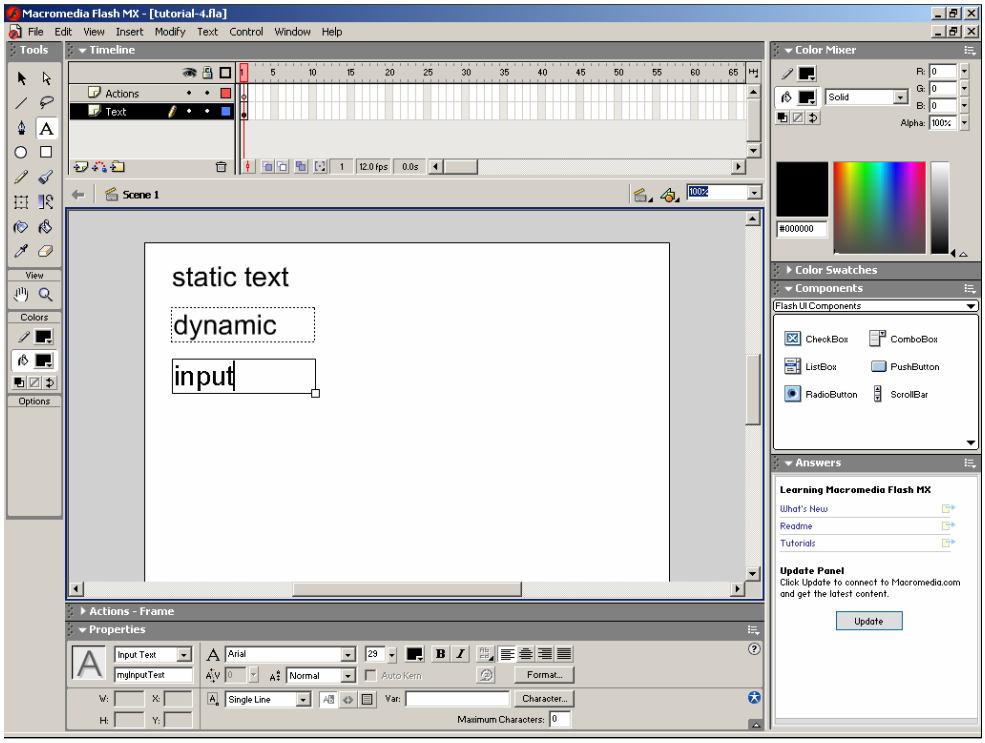


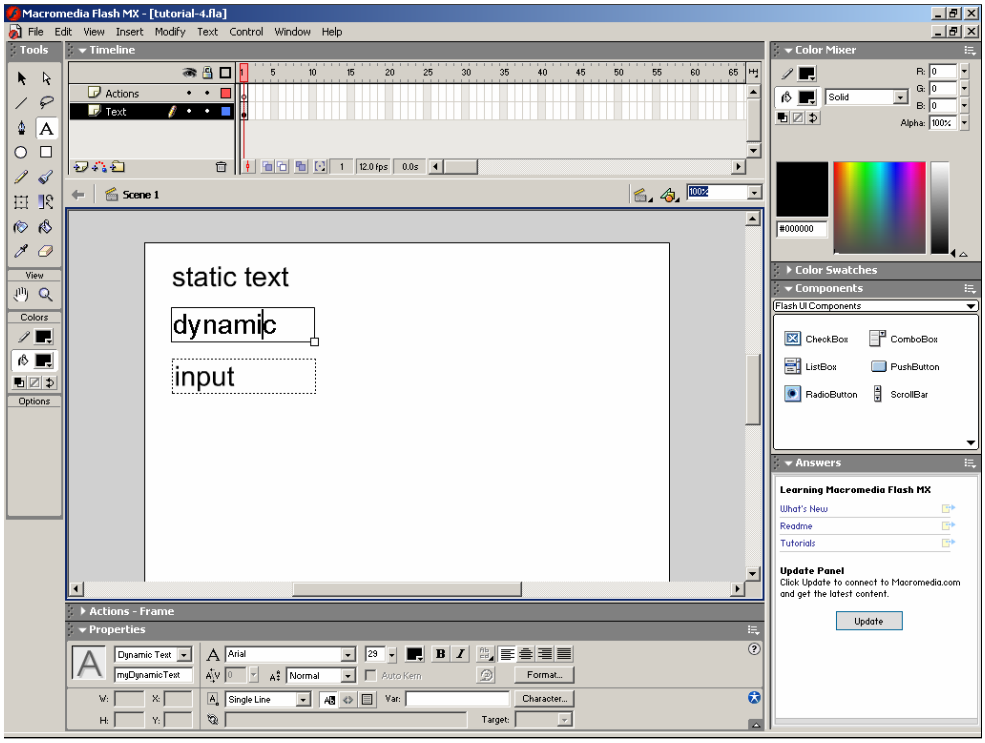
We will start here by creating three text fields.



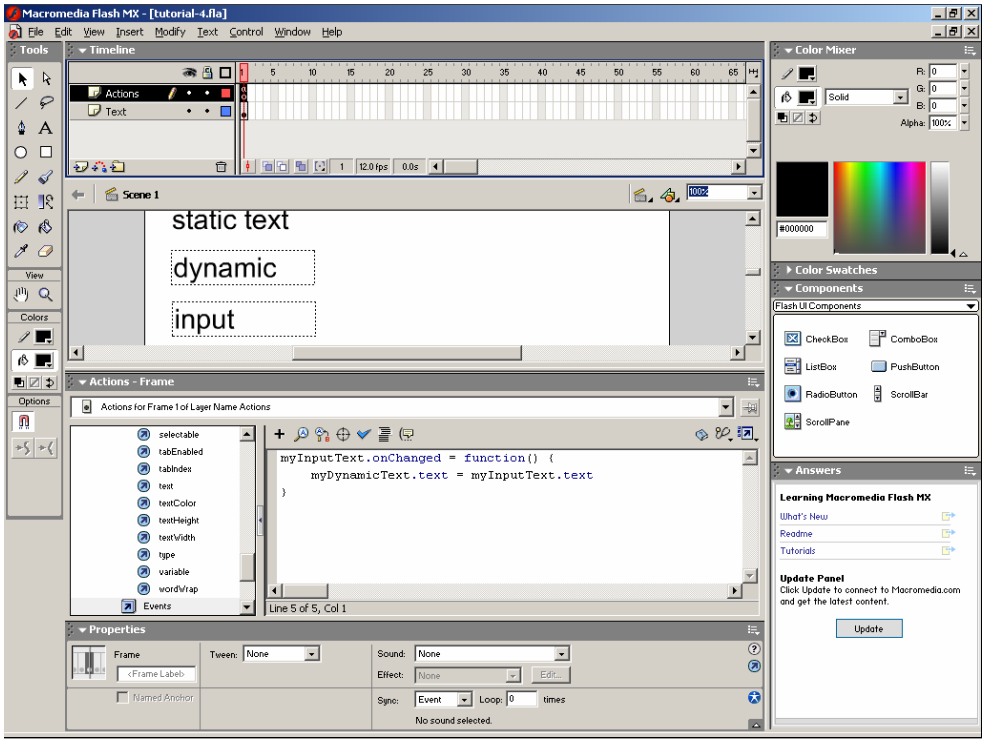
There are three types of text fields – static, dynamic, and input. We have used static up until this point, which allows us to display text. A dynamic text field allows us to change the text that is displayed there while our animation is running (like we did with Label.Caption = “abcd” in Visual Basic). An input text field works like a textbox did in VB, allowing the user to input text.



In order to interact with these text fields programmatically, we need to give them names. While VB automatically assigns default names (like Label1, Label2, etc), Flash does not. Elements that you do not interact with programmatically do not need to be named. We will name our input text field “myInputText”.

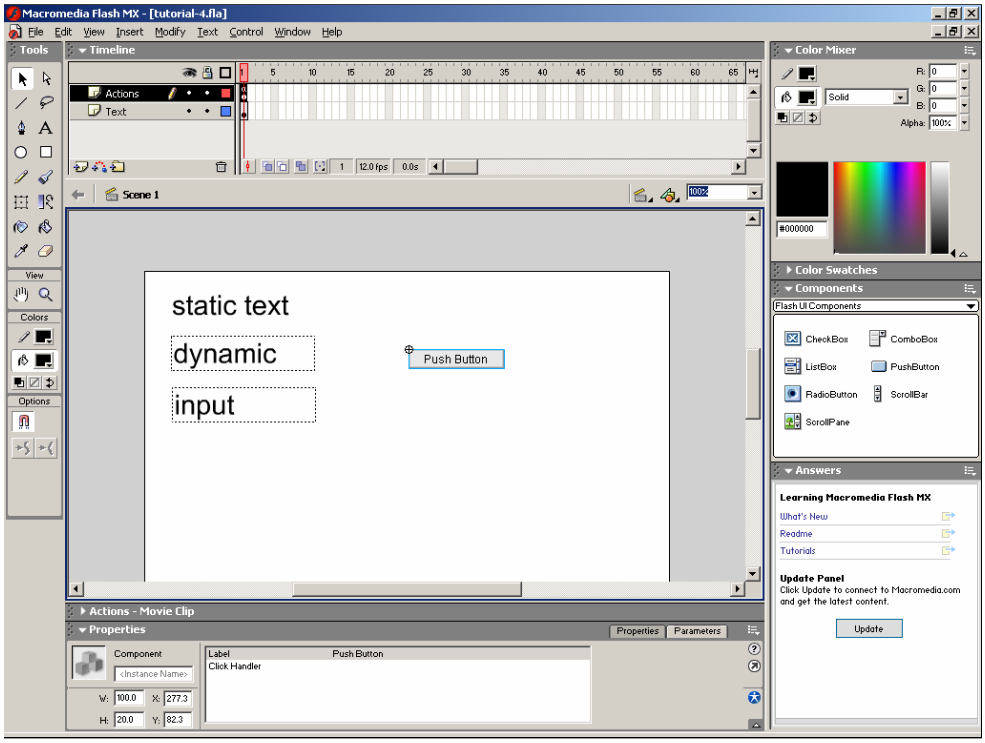


Similarly, we'll named the dynamic text field "myDynamicText". We will also create an actions layer.

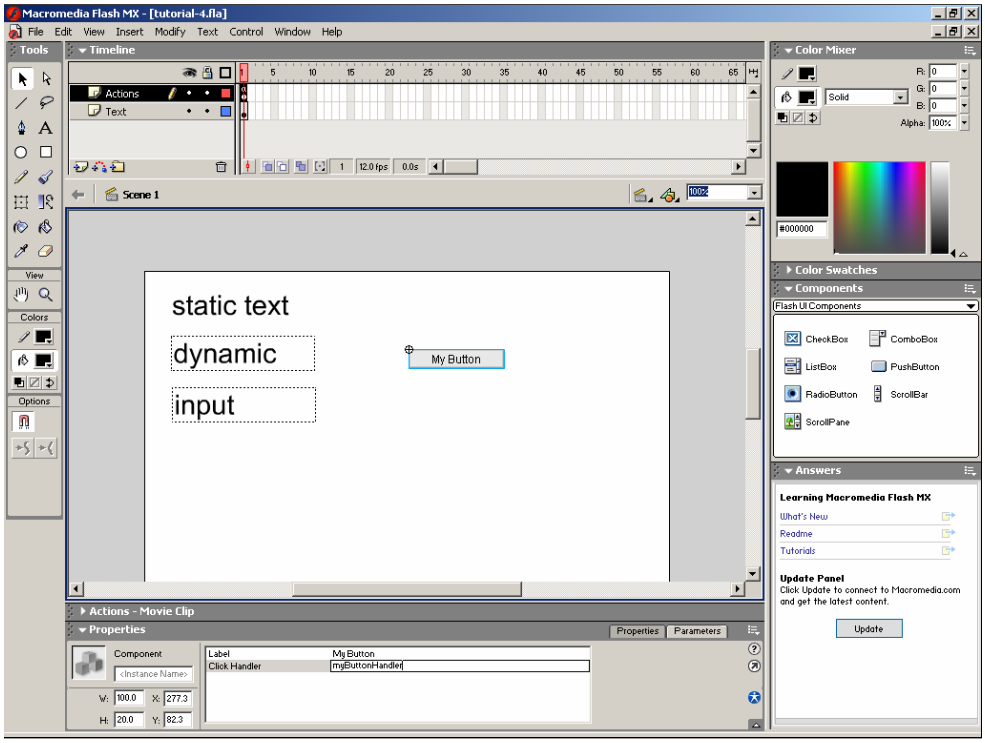


We will now add code that, whenever the user types something into our input text field, copies that value into the dynamic text field.

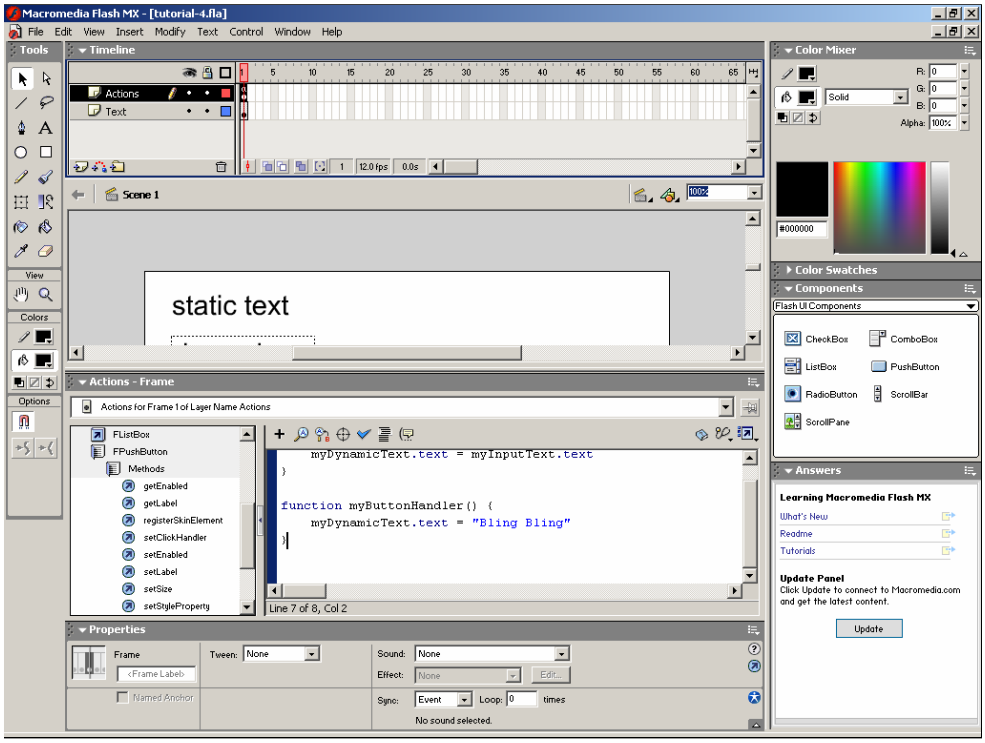
The code shown here will do this. There is an onChanged property on the input text. We set this equal to a function that copies the text property of our input text to the text property of our dynamic text.



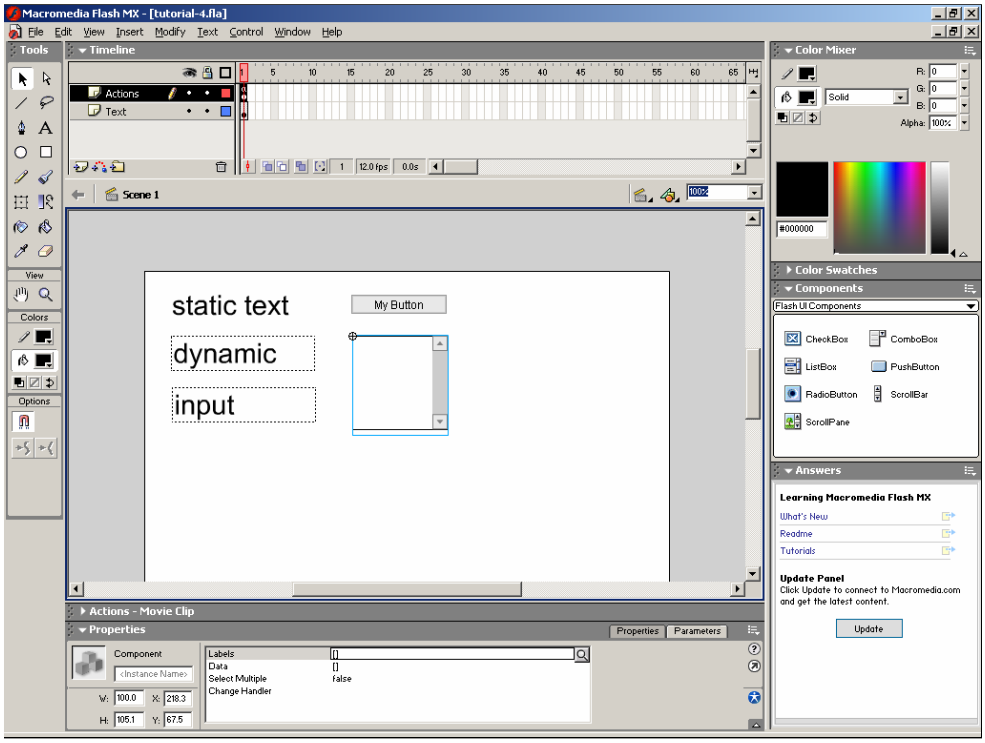
We can also use ActionScript to interact with the default components that Flash provides. These components are symbols, like the fancy button we created before. They provide a functionality, but you can also open them up and edit them. I really strongly recommend against trying this, but knowing that this is how they are defined is helpful because it means you treat them just like your other symbols. Drag a pushbutton onto the stage.



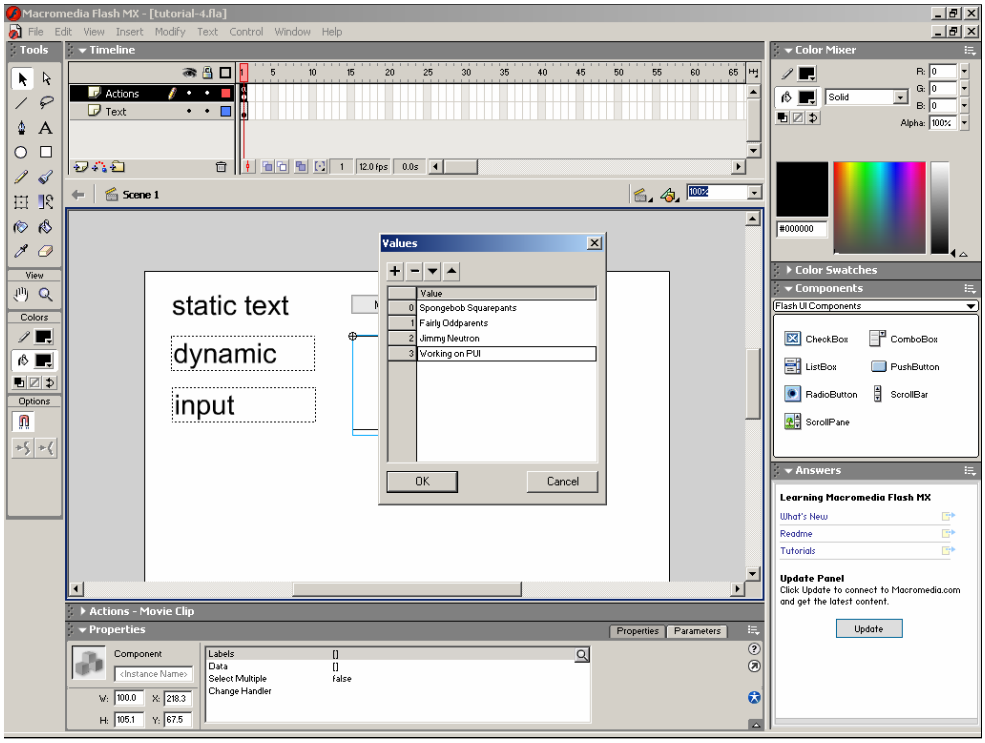
If we select it, the properties pain shows the properties “Label” and “ClickHandler”. We’ll give it the label “My Button”. The click handler property is where we specify what to do when this is clicked (this is different from how text fields work, we’re illustrating both techniques here). Here we specify the name of a function that should be called (we’ll use “myButtonHandler”).



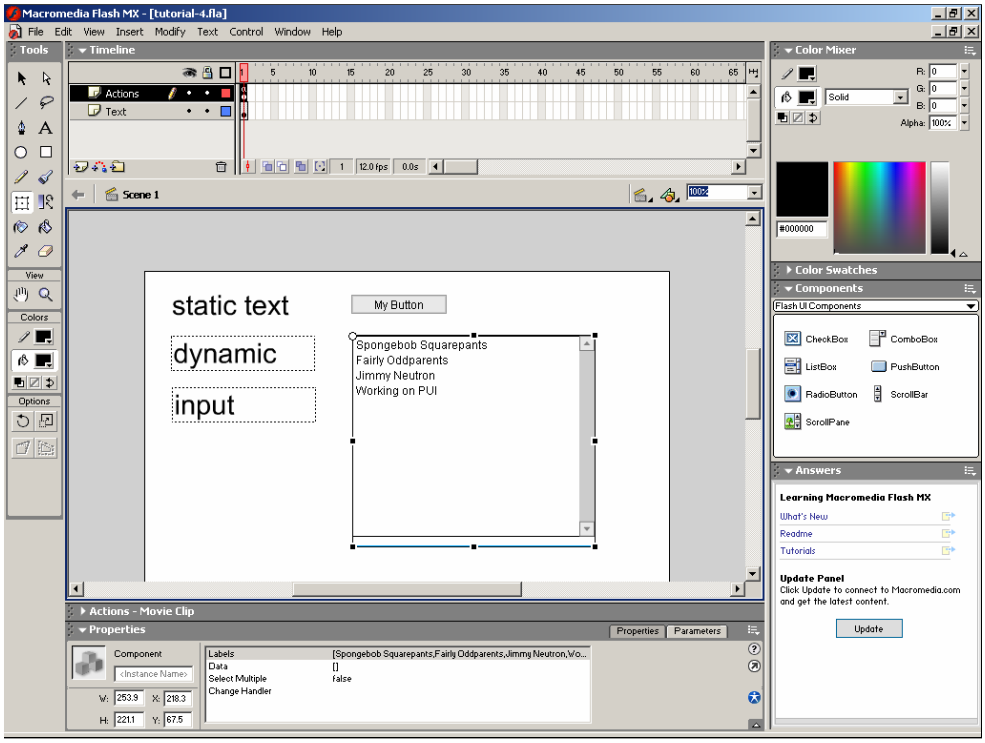
Now we define a function myButtonHandler. We'll just assign a value into myDynamicText.text.



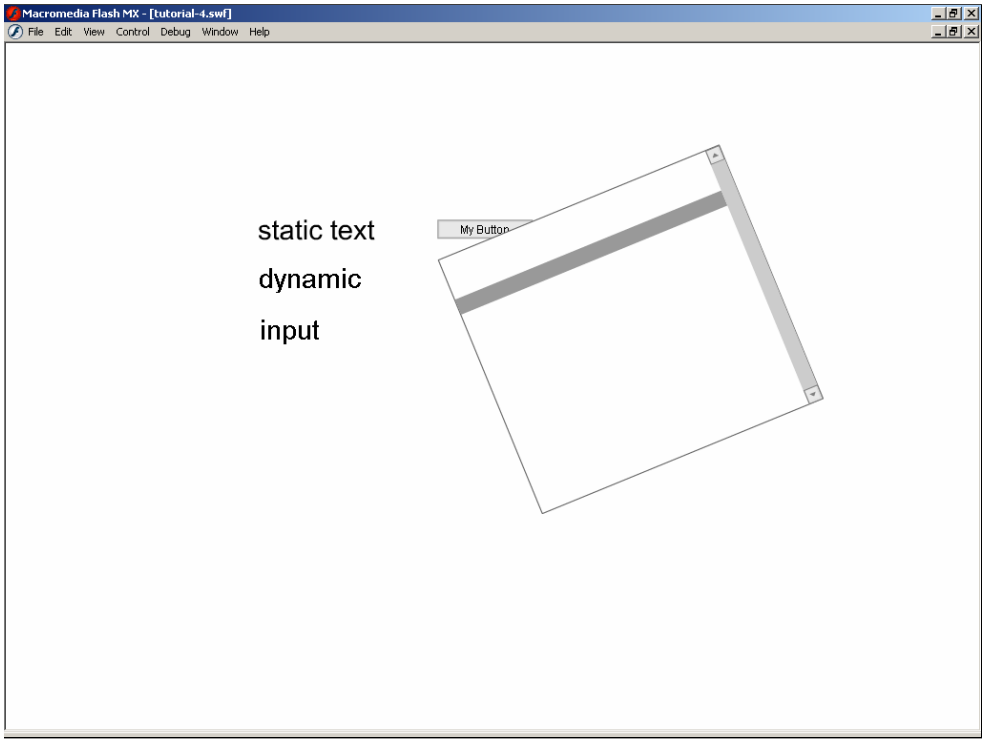
We can use the listbox component in a very similar way.



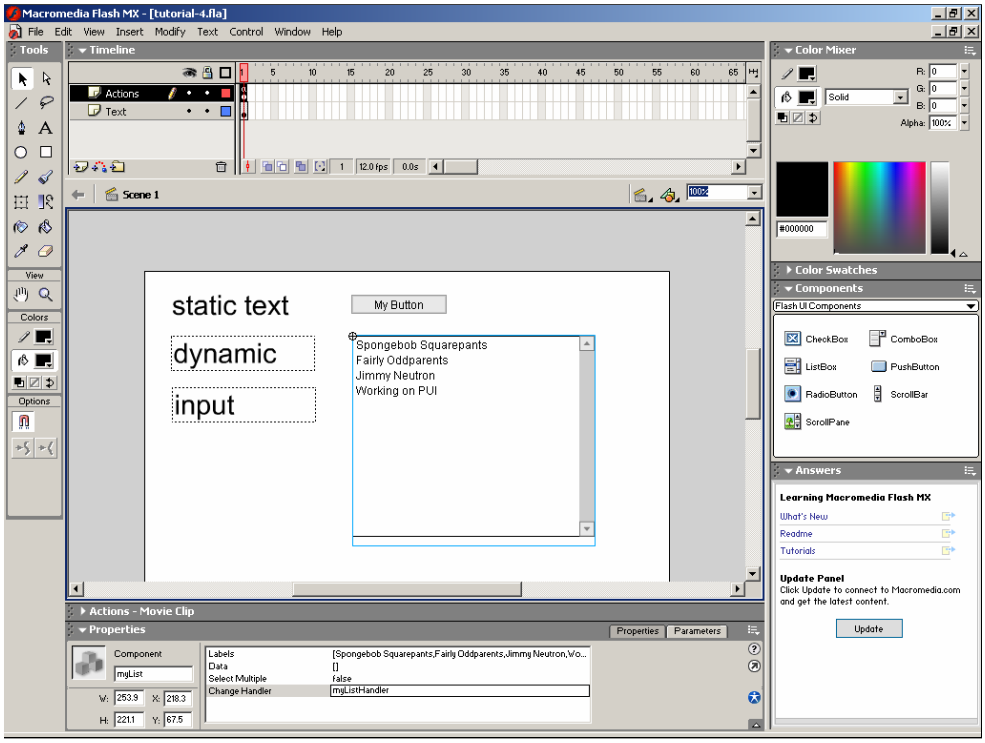
Its “Labels” property is where we can specify a list of items to display in the listbox.



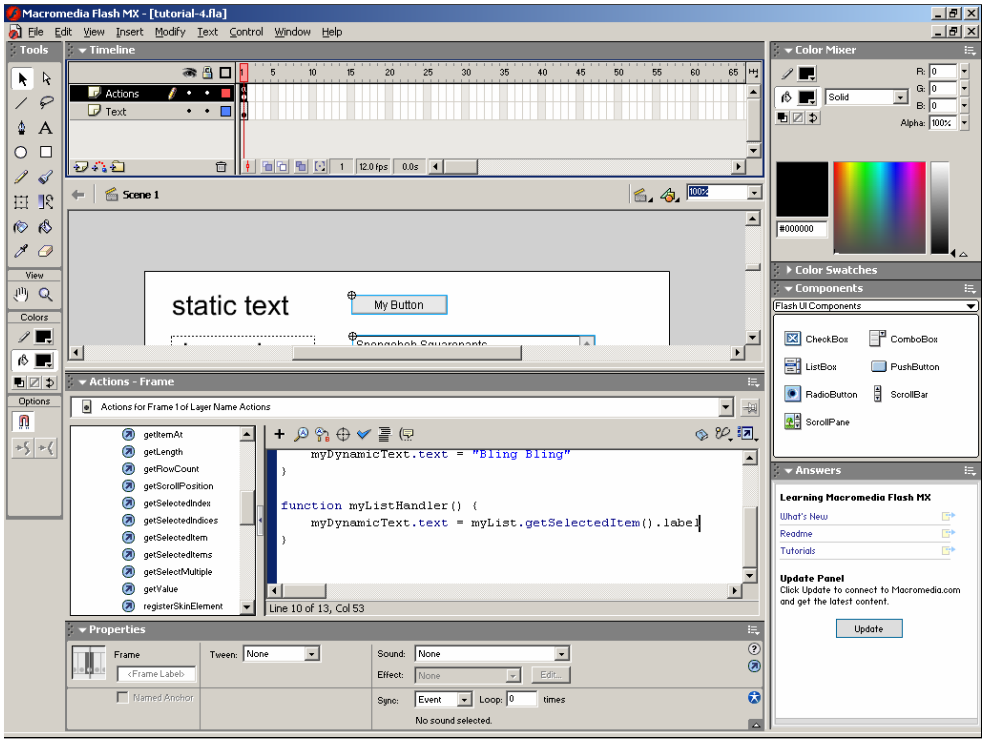
We can then use the transform tool to expand the listbox to the size that we want it (remember what we said about treating them just like any other symbol).



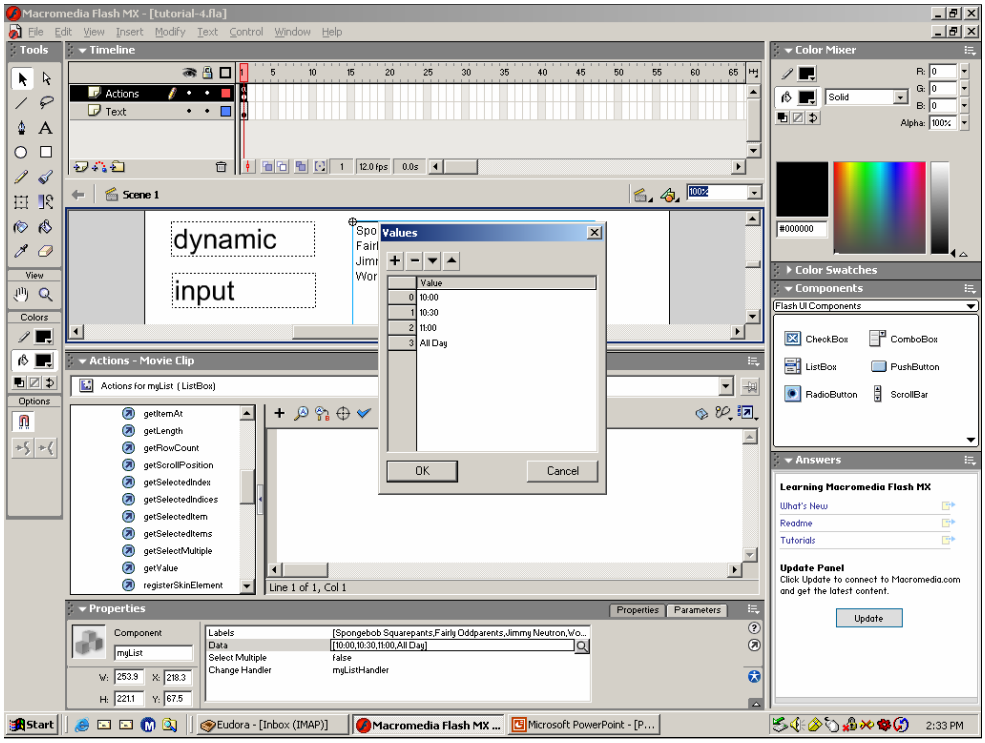
But they don't work perfectly like other symbols. If you try to rotate the listbox, your text won't display anymore.



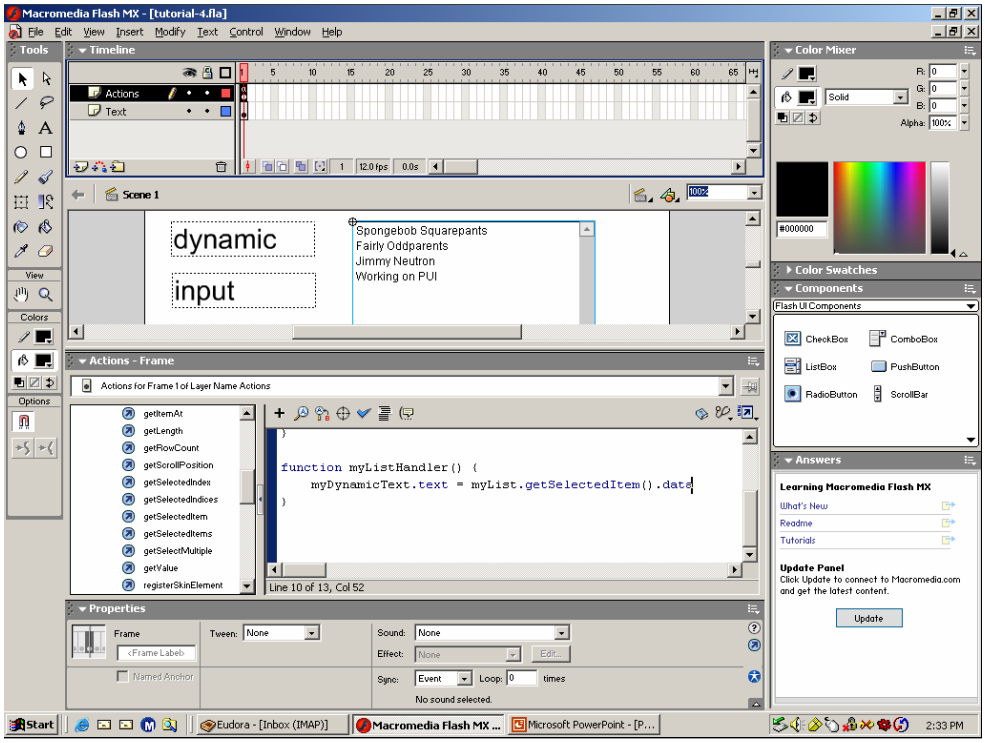
We'll name this list "myList", and give it a changeHandler of "myListHandler".



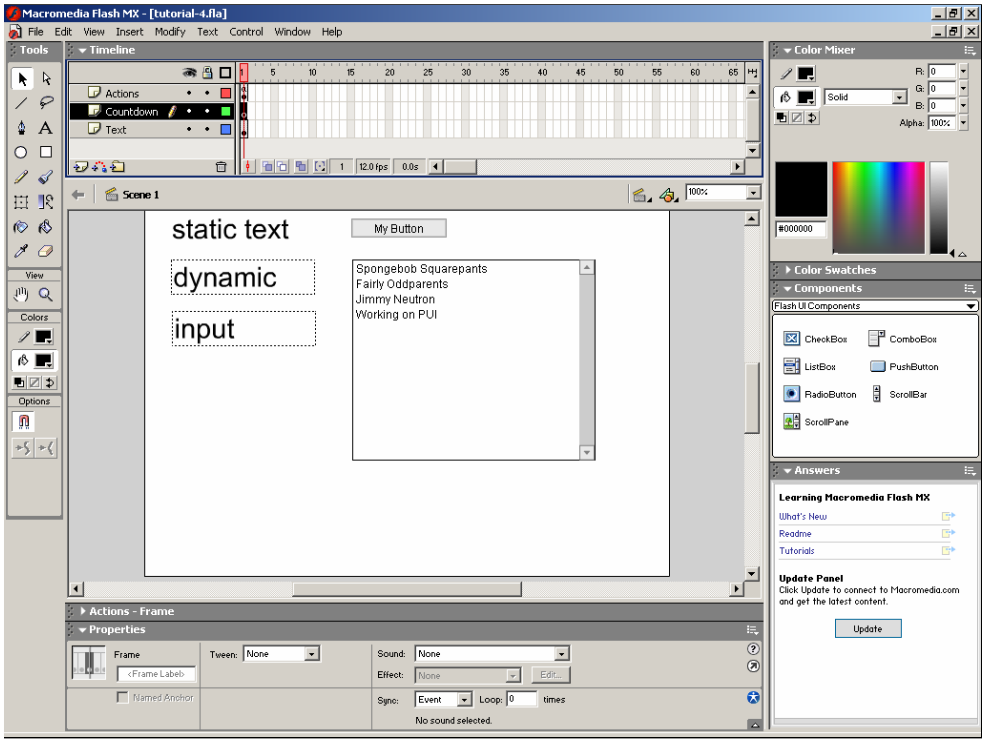
Our change handler for the list copies the label of the selected item into our dynamic text field.



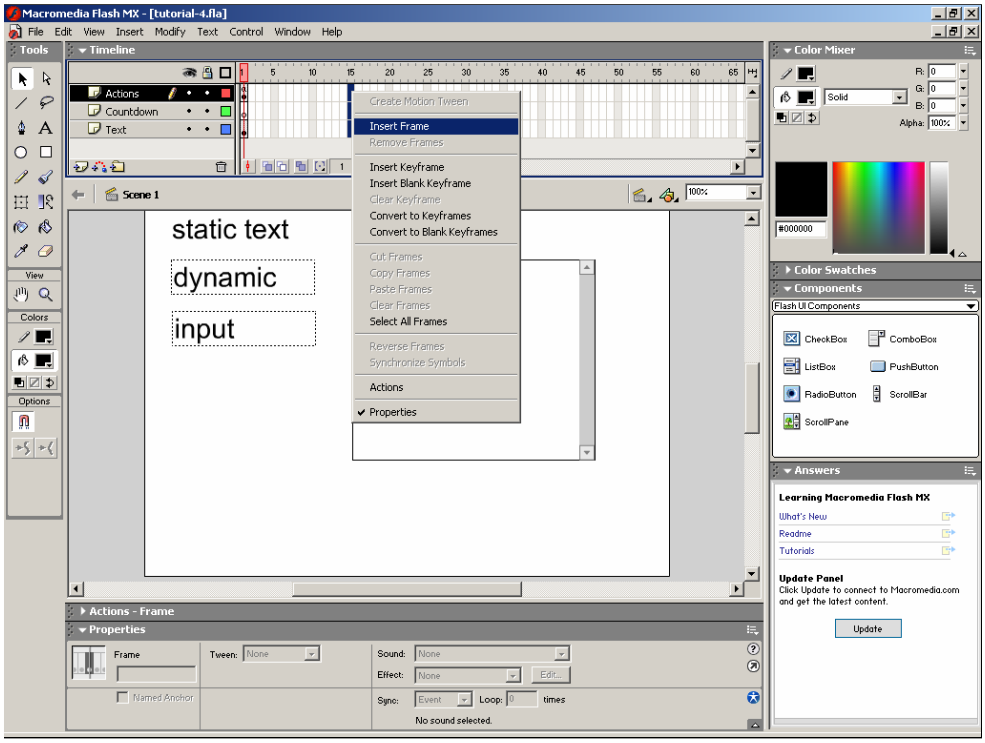
The listbox also supports a “Data” element associated with each label.



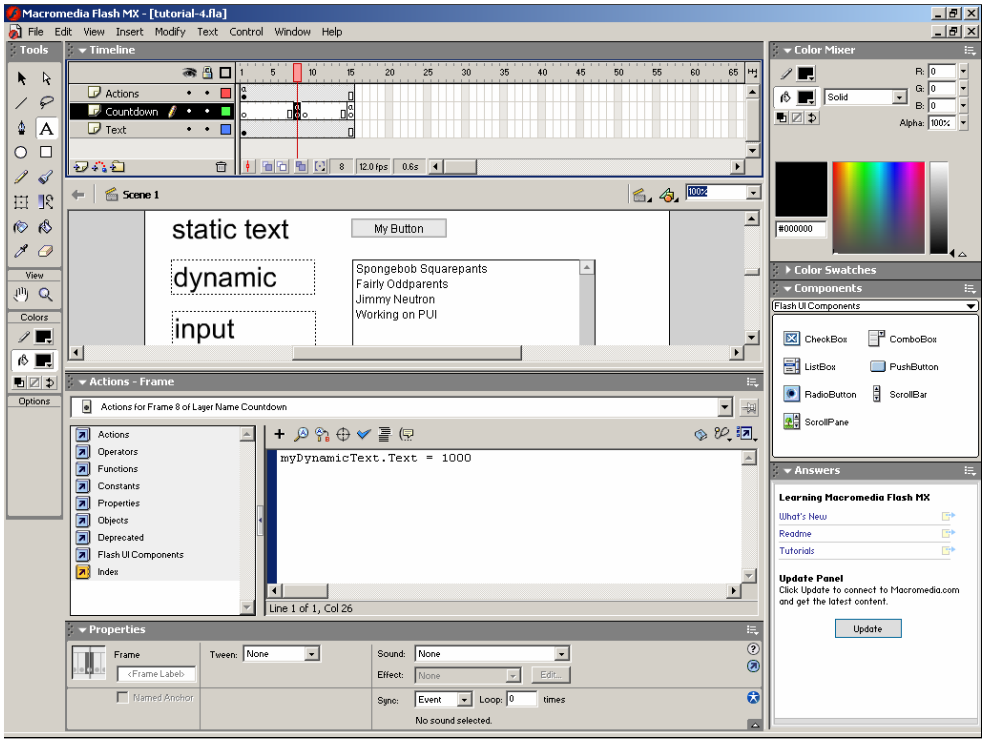
We can copy over the .data property instead of the .label property.



Finally, you can use a combination of ActionScript and the timeline to get the effects you desire. Create a layer called Countdown.

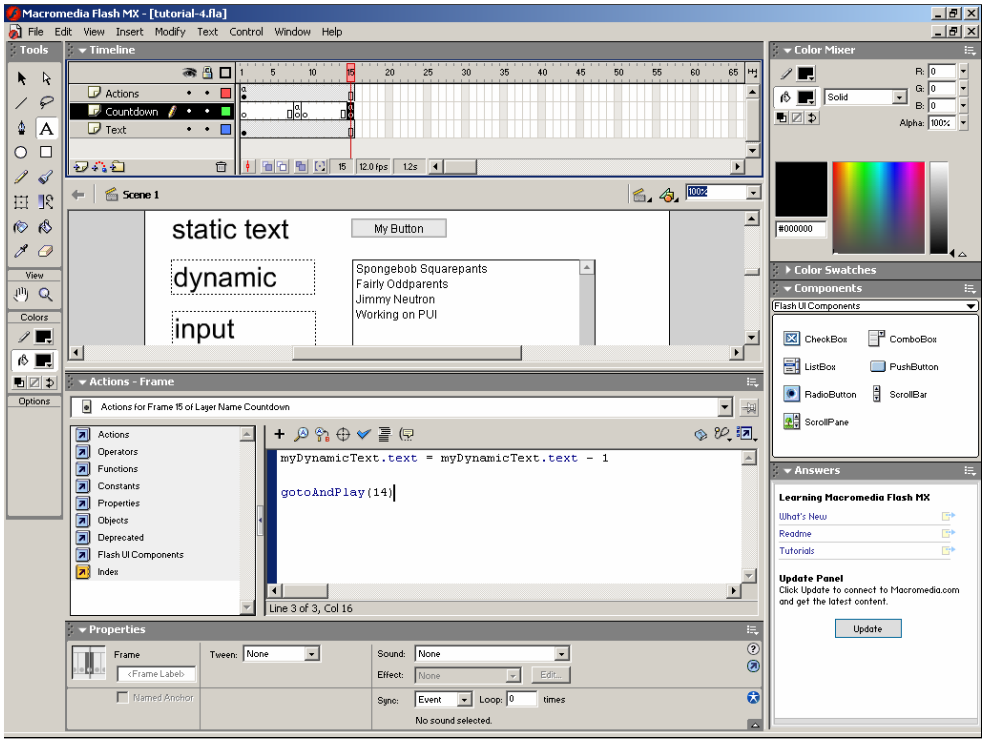


Select “Insert Frame” – not “Insert Keyframe” – for all three layers at frame 15.



Create a keyframe at frame 7, frame 8, and frame 15 in the countDown layer.

In frame 7, use ActionScript to set myDynamicText.text to “1000”.



In frame 15, subtract one from the value of myDynamicText.text. Then go back to frame 14. This will create a rapid countdown effect.

Note that while this work, it would not work if there were a keyframe in the layer containing our textbox. The settings in the keyframe would be applied each time we entered the frame, overriding the changes we had made in actionscript the last time we were in the frame. If you don't quite get that, you're probably fine. These things are not too likely to come up in the projects you'll do for class. Just want you to be aware of them.