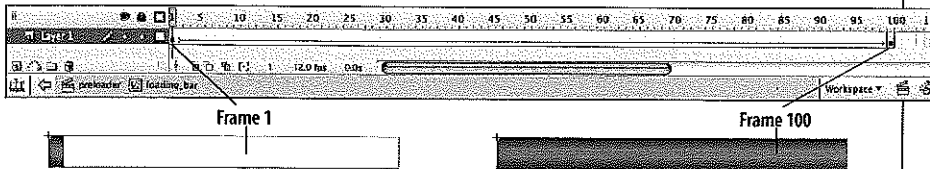


## Animated preloader

In this exercise, you will build a preloader which has a number of variables in it; one to determine the size of the file; another to determine the amount of data already downloaded; and a third which stores that value as a percentage. Work with the provided file 'preloader.fla'. Open the file and you will see that this is a basic text presentation screen (see right).

- 1 Create a new *Scene* in which to store the preloader graphics and script; from the *Scene* window (*Window > Other panels > Scene*) click the *Add Scene* button at the bottom. Double click to rename the scene "preloader" and drag it upwards so that it sits above Scene 1 (see right).  
**Note:** Never use *Scenes* for anything other than preloaders and linear animations!
- 2 In this 'preloader' scene, make a rectangle with an outline and fill. Select the fill only and convert it into a *Movie clip* (see right).
- 3 Name the instance of this *Movie clip* 'loading\_mc'.
- 4 Double-click to edit this *Movie clip*. Make a **100 frame** (1 frame for each percent) shape tween which tweens the bar from very short to full size (see below). Make sure that on both ends of the tween, you have set the transformation point at the left-hand side of the shape (see right).



- 5 Back on the main *Timeline* add any other graphics you think your audience might need to see (eg the word "loading").
- 6 Create a new 'actions' layer and open the *Actions* window in the first keyframe. Type in the script shown below. This is a complex script, but you should be able to recognise some elements of it:

`stop();` ——— Stops the movie looping between the preloader and Scene 1

`addEventListener(Event.ENTER_FRAME, loading);` ———

'Listens' for the Playhead to enter the frame and triggers the 'loading' function. 'ENTER\_FRAME' happens 12 times per second (the movie's frame rate) so the function is running almost constantly.

```
function loading(event:Event) {
    var bytestotal = stage.loaderInfo.bytesTotal;
    var bytesloaded = stage.loaderInfo.bytesLoaded;

    var percent = Math.round(bytesloaded*100/bytestotal);
    loading_mc.gotoAndPlay(percent);

    if (bytesloaded >= bytestotal) {
        nextScene();
        removeEventListener(Event.ENTER_FRAME, loading);
    }
}
```

Variables to determine how big the movie is and how many bytes have been downloaded.

Based on previous variables, work out a new variable 'percent' and tell loading\_mc Movie clip to go to that numbered frame.

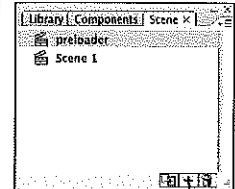
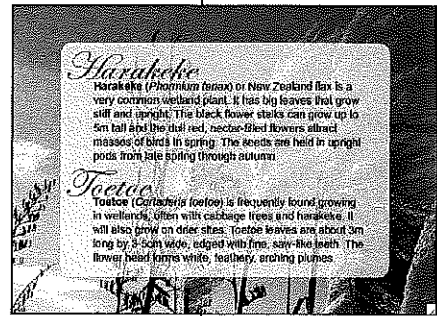
Evaluate whether the whole movie is loaded. If it is, go to the next scene.

Remove the listener so that the function stops.

- 7 The last thing you will need to do is to go back to the main Scene (**Scene 1**) and add an action on the first frame to **stop** the *Timeline*, otherwise the movie will try to loop between Scene 1 and the preloader scene.
- 8 Test your movie. Flash Player has a useful tool for simulating download speeds, which will allow you to test your preloader properly. Once you are in Flash Player, choose *View > Simulate Download* (or press **⌘/Ctrl-Enter** again). You should see your loading bar increasing in size until the whole movie is loaded, at which point it flicks on to the main Scene.

### tip Preloaders

A preloader ensures that the whole movie is loaded before it plays the content, ensuring that the content behaves correctly. You should add a preloader to your movie if the .swf file is bigger than 100kb.



Add scene



Move transform point (dot) to here.

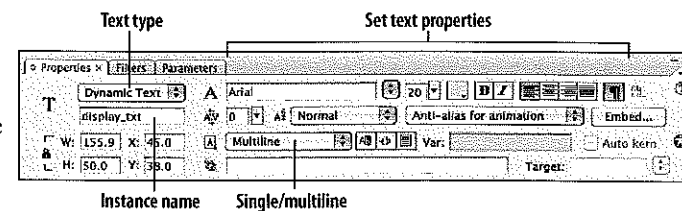
## Other useful ActionScript techniques

### Setting dynamic text

Previously you have made both *Static Text* and *Dynamic Text*. Dynamic text boxes can be populated by ActionScript.

#### To make a dynamic text box

- 1 Draw a text box in the usual way. Make sure that its text type is set to *Dynamic Text* (see below). You will see a dotted border around dynamic text.
- 2 Even though you may not be typing anything into the text box, use the regular formatting tools (size, font, colour etc) to format the text. Any text put into this box will take on those properties.
- 3 Set the box to *Multiline* if you think the text might stretch to more than one line (see right).
- 4 Always give your *Dynamic Text* an instance name—this is how ActionScript will ‘talk’ to it (see right).



#### To fill a dynamic text box

There are two useful ActionScripts to populate a *Dynamic Text* box:

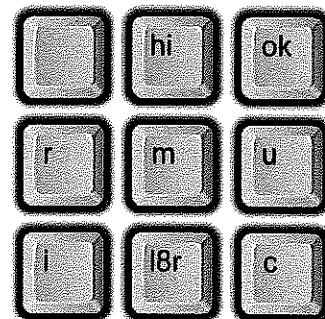
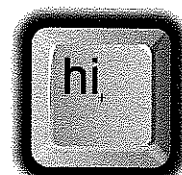
- `textBox_txt.text = "Your text here";`
- `textBox_txt.appendText("Your text here");`

The first one puts the text into the named box, wiping out any text that was there before, while the second adds the text to any text that was in the named box already.

### 4.4 Dynamic text

In this exercise you will build a ‘text machine’, where the person using the Flash movie will determine what text is shown.

- 1 In a new document, open *Window > Common Libraries > Buttons*. Open the ‘classic buttons’ folder then open the ‘Key Buttons’ folder inside that. Find the ‘key – labelled button’ symbol (see right) and drag and drop it onto the *Stage*.
- 2 This button needs editing. Double-click it, to enter symbol editing mode, and edit the text box to read ‘hi’ instead of ‘A’ (see right). With the text box still selected, change its type from *Dynamic* to *Static*. If you don’t do this you will get lots of output errors! You will also need to edit the text in the *Down* state of the button.
- 3 Go back to the main *Stage*. Duplicate this button eight more times. Do this by *Control/right-clicking* the symbol in the *Library* panel and choosing *Duplicate* from the drop-list.
- 4 Drag and drop one instance of each button symbol on the *Stage* to make a keypad (see right). Give each one an instance name: **hi\_btn; ok\_btn; r\_btn; m\_btn; u\_btn; i\_btn; l8r\_btn; c\_btn; space\_btn**.
- 5 Edit the text in each button’s symbol to match the button name. For the ‘space’ button, just delete the text.
- 6 Add a text box above the keypad. In the *Property Inspector*, check that it is set to *Dynamic Text* (see top). Change *Single line* to *Multiline*. Give it an instance name of ‘display\_txt’.
- 7 Create a new ‘actions’ layer, and open the *Actions* window on the first keyframe.



### 4.4 Dynamic text cont...

- 8 Add listener-function pairs which add the appropriate text to ‘display\_txt’ when the buttons are clicked:  

```
hi_btn.addEventListener(MouseEvent.CLICK, hiText);  
function hiText(event){  
    display_txt.appendText("hi");  
}
```
- 9 For the ‘space’ button, either ask ActionScript to add a space (“ ”) or an underscore (“\_”) (see below).

```
space_btn.addEventListener(MouseEvent.CLICK, spaceText);  
function spaceText(event){  
    display_txt.appendText("_");  
}  
  
hi_btn.addEventListener(MouseEvent.CLICK, hiText);  
function hiText(event){  
    display_txt.appendText("hi");  
}  
  
ok_btn.addEventListener(MouseEvent.CLICK, okText);  
function okText(event){  
    display_txt.appendText("ok");  
}
```

- 10 Add your own ‘skin’ with simple graphics on another layer.

**Exercise extension:** See if you can add a new button with a ‘reset’ function. Use the following action inside the function.  
`display_txt.text = "";`



### Drag and drop

Another relatively simple piece of ActionScript is the action to drag and drop a *Movie clip*.

#### To drag and drop a Movie Clip

- 1 Create a new movie with a *Movie clip* instance on the *Stage*.
- 2 Give the *Movie clip* an instance name.
- 3 Add an actions layer, and an action which says:  

```
movieClip_mc.addEventListener(MouseEvent.MOUSE_DOWN, drag);  
function drag(event){  
    movieClip_mc.startDrag();  
}
```

The listener is waiting for the *MOUSE\_DOWN* event—one you haven’t used before, but which is distinct from *MOUSE\_UP* which you will use in the next step.
- 4 Test the movie and you’ll find that you will be able to ‘pick up’ the *Movie clip*, but you won’t be able to ‘let go’ of it.
- 5 Add to the script:  

```
movieClip_mc.addEventListener(MouseEvent.MOUSE_UP, drop);  
function drop(event){  
    movieClip_mc.stopDrag();  
}
```
- 6 Don’t forget to add another piece of script to make the hand cursor appear!  
`movieClip_mc.buttonMode = true;`

