

Flash Movie Essentials

What we'll cover in this chapter:

- *The **Flash stage**: where you create your movies in Flash.*
- *The **Property inspector**: a central store of tool and object properties guaranteed to make authoring in Flash easier.*
- *Manipulating the size of the stage, and changing your movie's overall background with the Property inspector.*
- *The **timeline**: controls the playback of your movies.*
- ***Frames**: these are essential for creating and arranging the content of your movie. You can think of a frame as representing a moment in time.*
- ***Animation**: making content move.*
- ***Layers**: allow you to add depth to your movie, and help you keep track of complex content.*
- ***Scenes**: create separate scenes that contain distinct chunks of our movie.*

Macromedia Flash MX is really two things; firstly, it's the standard file format for delivering interactive, visually-rich content and animation on the web – this is the SWF file format we talked about in the Introduction; and secondly, it's the actual authoring environment that lets you create and publish those SWF files. The Flash program you install on your computer is the gateway to state-of-the-art web content.

In this chapter we're going to introduce you to the authoring environment – the Flash MX interface – and take you through the essentials of creating visual content in Flash and making it move. In doing this, we'll start building up a picture of the main components of a Flash movie, and see how they fit together.

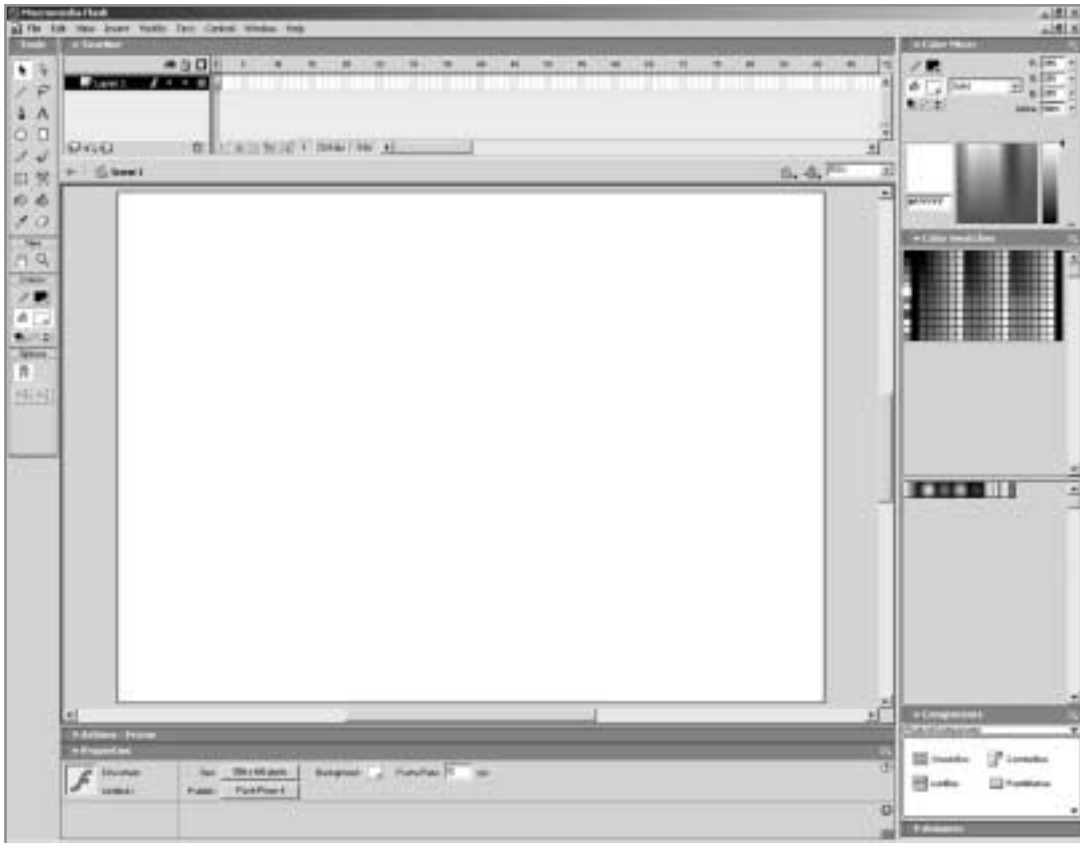
Taking time to understand the core elements at the heart of the Flash movie will pay off later – you'll have a firm grasp of the foundations, and you'll be able to build effectively on these as you learn.

So...let's begin by looking at the first thing that almost everybody wants to do when they open up their copy of Flash – create a movie and make interesting things happen on the screen.

The authoring environment

If you don't already have Flash MX up and running, start it up now.

When you first open up Flash, you're presented with an array of screen elements: icons, menus, content preparation areas, toolbars, panels, and status bars:



Don't worry if your screen doesn't look exactly like ours when you start up Flash: if you've already opened Flash up before and played around with it you'll doubtless have altered the setup in some way.

This is the feature-rich authoring environment that lets you create your Flash movies and export them so that they can be published on the web and accessed by the adoring multitude. It's Flash's studio, workshop and test track combined.

If you've never used a Macromedia product before, you might be intimidated by the unfamiliar interface the first time you open Flash – don't be: before long you'll be navigating the interface with ease. And there's an added bonus to learning the Flash MX interface: Macromedia have a common interface across their software, so once you're familiar with Flash, you'll have no trouble finding your way around other Macromedia programs such as Fireworks or Dreamweaver.

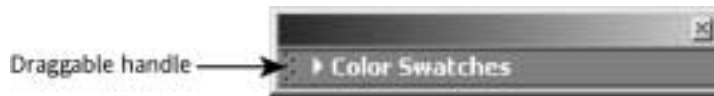
There's a tremendous amount of detail and power tucked away in the Flash MX interface, and at first it can seem a little daunting if you're new to the software. To avoid the sense of clutter, and in order to 'turn down the volume' a little, let's clear some of the elements out of the way: that way, we can concentrate on the bare essentials for a moment (but don't worry, you won't be missing a thing – we'll be explaining all of the core features during the course of this book).

Configuring the authoring environment

1. First, let's minimize any palettes on screen. Minimize all of the palettes – called **panels** – on the right-hand side of the previous screenshot. You can do this by clicking your mouse on the white text at the top of each panel until all the panels are stacked up:

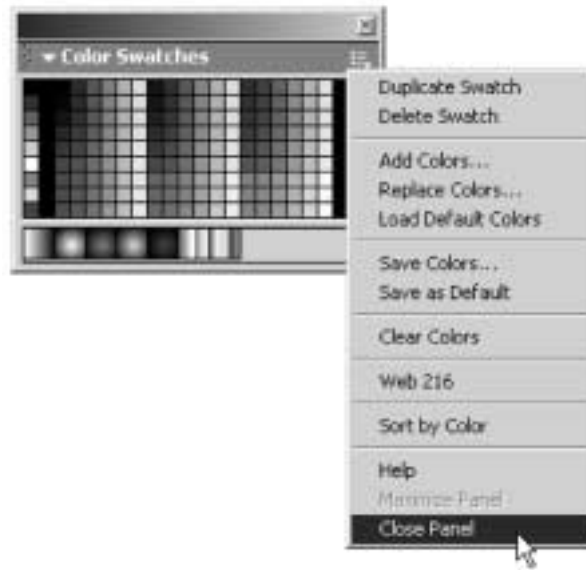


Floating panels are used to help you modify and manipulate the content of your Flash movie once you've created that content. This content could be graphic images, pieces of animation, text or any number of things – but you can use the panels (and menu options) to alter their characteristics and the way that they behave. All the panels are dockable; you can drag them around the screen and dock them to other panels if you choose. This lets you customize the authoring environment. Flash also provides some default layouts for the interface with the panels set up for a specific designer/developer interface (**Window > Panel Sets**) and you can even save your own layout (**Window > Save Panel Layout...**).



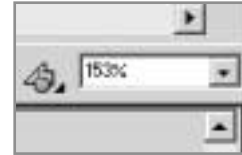
You can think of any element you create in your movie (such as pictures or text) as a discrete object: each of these objects has its own attributes, such as color, transparency and size, and you can use the panels to change these attributes. Additionally, changing panel settings can also alter the way that an object behaves. We'll be looking at all of these aspects as we progress through the book. At the moment though, we don't actually have any content to work with, so let's move these panels out of the way for reasons of clarity:

2. Click on each of the panels' 'close' boxes, or choose the **Window > Close All Panels** menu option. Alternatively, pressing F4 will also close all panels, including the Tools panel. We'll come back to these panels later. Individual panels can be closed by right-clicking on the gray bar and choosing the **Close Panel** option, or choosing this option from the drop-down menu when the panel is maximized:

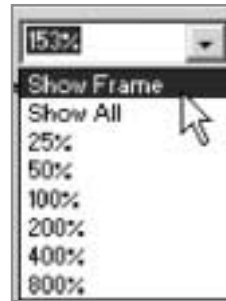


Next, we want to make sure that we can see all of the area where we're going to create the visual content in our Flash movie:

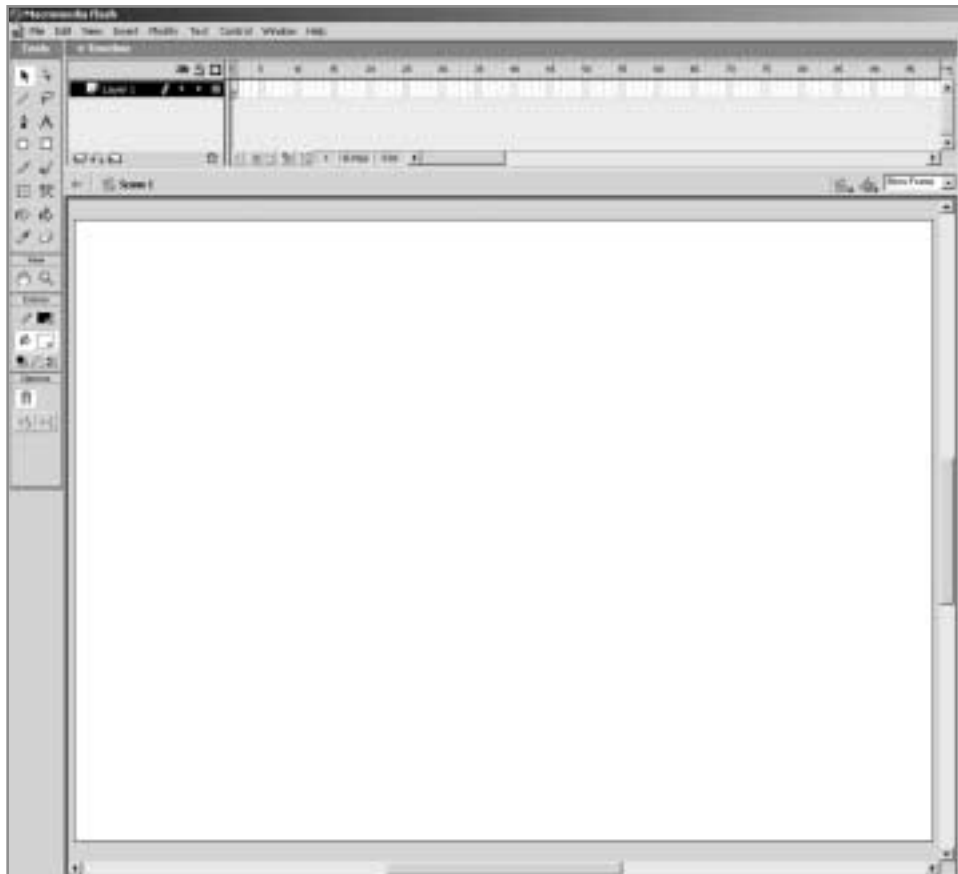
3. Click on the **Magnification** drop-down list box near the top right of the screen below the timeline...



4. ...and choose the **Show Frame** option:



Notice that the white area in the center of the screen is now visible in its entirety:



This white area is called the **stage**, and it's where all the action in your movie takes place. The gray area around the stage is called the **work area**. Let's talk about these different areas a little.

The stage

The stage can be likened to what a movie director can see in the viewfinder of a movie camera: what you see in the viewfinder is what will appear in the movie that the viewer watches in the theater. In the motion picture world, you can have action taking place on the film set (stage), and you can have actors waiting off stage, ready to make their entry and come into shot.

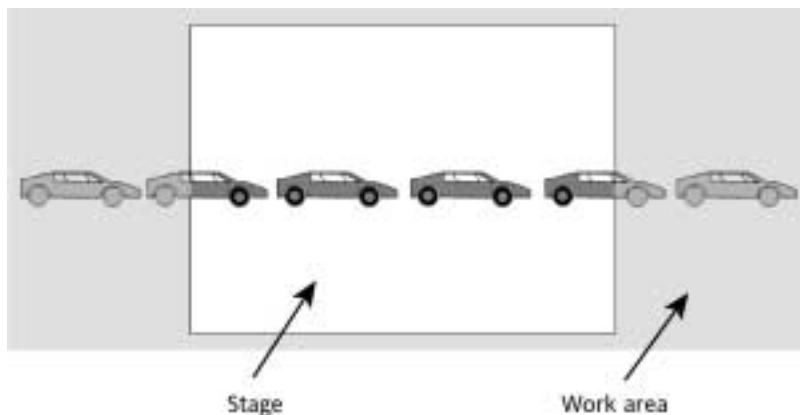
At different times in the movie, different people and objects will be visible on the stage and, consequently, 'in shot'. The stage in Flash works on the same principle: at any given point in your Flash movie, the things that are on the stage are what the viewer will see when the movie is rendered in their browser. Another thing to consider is that the movie set can be much larger than the camera's field of view, and the camera can move around and seek out previously hidden corners.

If you want the end user to see something in the Flash movie that plays in their browser, then that something has to be visible on the stage area when you create the movie. This also means that movie content can move onto the stage from the 'wings': for example, an animated actor could enter stage left, walk across the stage, and exit stage right. In the Flash authoring environment, any visual element that moves beyond the boundaries of the stage winds up in the **work area**.

The work area

The work area surrounds the stage. You can place content in the work area, but only content that actually appears on (or moves across) the stage will show up in the finished movie that the user watches. So, when you're designing your movie, you need to think about whether the visual elements it contains will spring into existence directly on the stage, or whether they're going to wait in the wings and then move onto the stage at some point.

Another example of this would be a car that starts its journey in the work area to the left of the stage, moves across the stage (and into the viewer's sight), and then accelerates off into the work area on the right:



While the content is on the stage, the viewer sees it in the browser. When it's 'in the wings', it's invisible to the viewer.

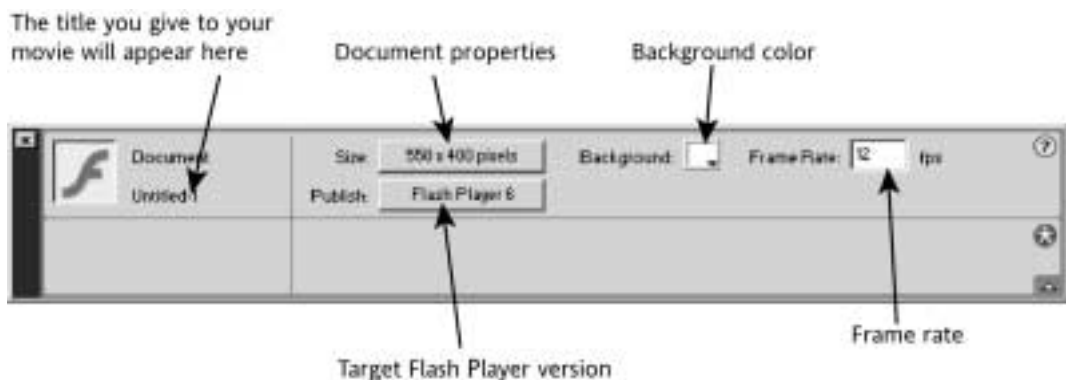
What we've done so far is use the Magnification box to change our view of the stage **in the authoring environment**. Any changes we make by zooming in and out and making the stage look bigger or smaller in the authoring environment will **not** be applied to the finished movie seen by the end user. These magnification changes are just to help us see things more clearly when we're creating and modifying our movie.

To alter the size and proportions of our finished movie once it's displayed in the user's browser, we need to change the **properties** of the stage itself. Flash MX has a new resource that will enable us to do that easily. It's called the **Property inspector** and we'll take a quick look at that now.

The Property inspector

The Property inspector is a new feature in Flash MX and was introduced to make using Flash a whole lot easier. With the Property inspector, we can easily manipulate all of our movie's contents from one place. We'll be using it frequently throughout this book and it'll come to be your best friend when creating movies in Flash MX.

By default, the Property inspector is positioned at the bottom-center of the screen, although you might have closed it just now if you chose to close all panels with the **Window > Close All Panels** menu option or if you pressed F4. If you don't already have it open, you can access it with the **Window > Properties** menu option, or by right-clicking on the stage and selecting **Properties** from the context-sensitive menu. Alternatively, you can press CTRL+F3.



We don't have any content on our stage yet, so this is how you'll first see the Property inspector.

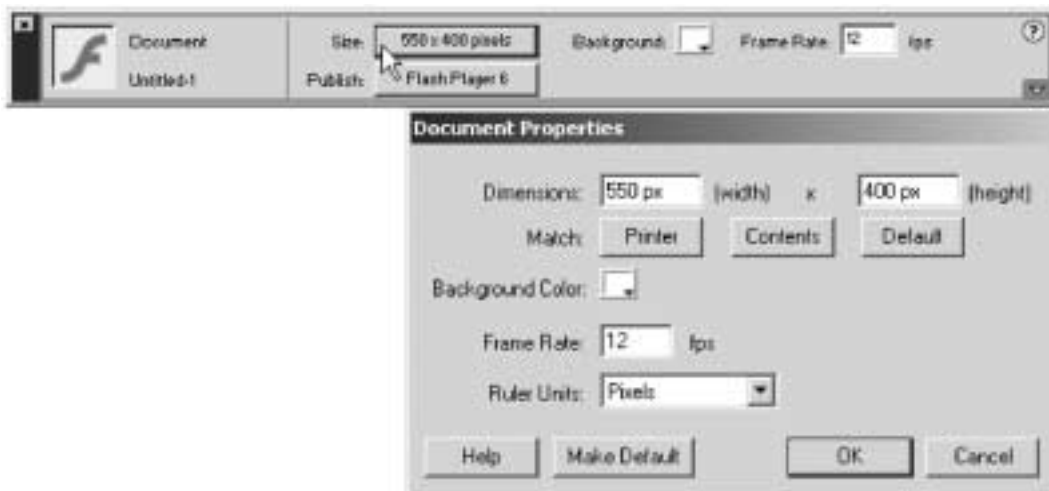
The Property inspector is split into two parts: upper and lower sections. We'll see in the next chapter that when using items from the Tools panel, the upper section has properties related specifically to that tool, while the lower section is related to an actual item you have selected in your work area. The lower section can be opened and closed using the arrow in the bottom-right corner.

Let's move on to see how we can use the Property inspector even though we have no content on our stage yet.

Changing the size of the stage

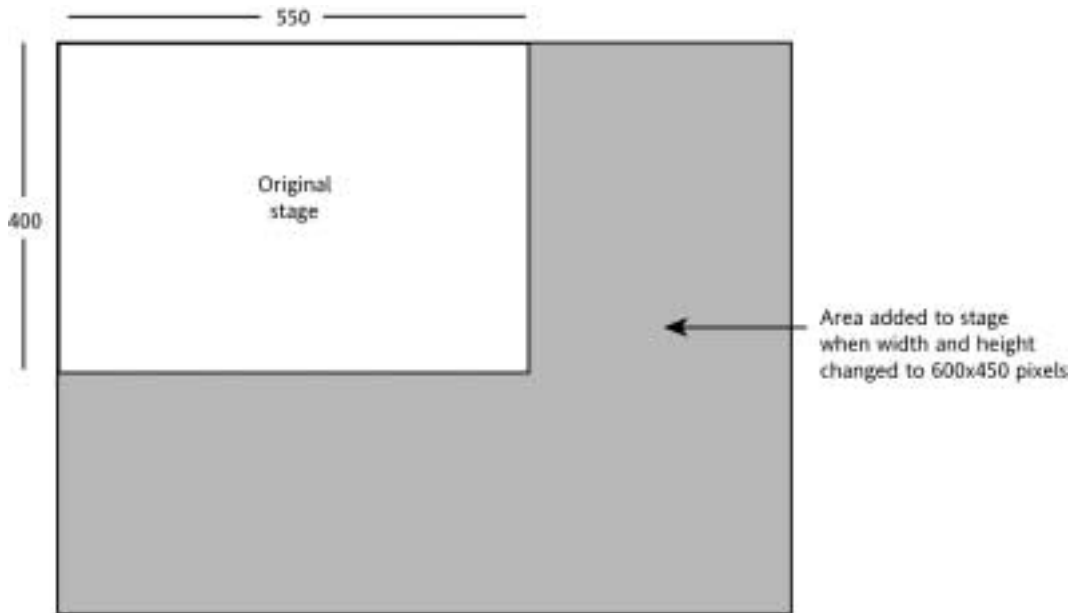
When you're planning your Flash movie, you should consider how much browser real estate you want the finished movie to take up – you'll need to decide what size you want the movie's window to be, based on factors such as what kind of content you're displaying, what else will appear alongside the movie in the host page, and so on. When you've made that size decision, you can alter the size of the stage to match your plan.

You can view the stage's current dimensions and global characteristics by clicking on the **Size** button in the Property inspector. This opens up the **Document Properties** dialog box, from where you make global changes to the **properties** (or *characteristics*) that affect the whole movie:



*You can also bring up the **Document Properties** dialog box using the **Modify > Document** menu option or by right-clicking on the stage and selecting **Document Properties...** from the context-sensitive menu.*

You can see here that the default dimensions of the stage are 550 pixels wide by 400 pixels high. When you change the dimensions of your stage, Flash will always measure them from the top left-hand corner. For example, if you change the width of your movie to be 600 pixels and the height to be 450 pixels, Flash will simply add another 50 pixels to the right-hand side of the stage, and 50 to the bottom:



If your brain doesn't translate pixel-speak too easily, you can always change the units of measurement that Flash uses throughout the entire movie by picking a different option from the **Ruler Units** drop-down box:



Whichever option you choose here will be applied across this whole movie until you change it to be something different.

*Remember that when you set the **Dimensions** of the stage in this dialog box, you are directly affecting the size of the window that your Flash movie will be displayed in inside the user's browser. It's good practice to think about this before you start creating your visual content on the stage. You can always change the size of the stage as and when you like, but the more planning you do, the smoother your movie creation process is likely to be!*

Note that the **Match: Printer** and **Match: Contents** options will automatically change the size of the stage if you select them: **Match: Printer** will set the stage size to reflect the default paper size for your default printer, and **Match: Contents** will change the stage so that it's large enough to contain all of the content elements that you've created (even those that spill over into the work area outside the stage).

As you can see, there are a number of other global properties that you can change for the whole movie: we'll be covering all of the important ones as they come up in the course of building our example movies in this book. At this stage, let's just observe that the **Frame Rate** property influences the playback speed and smoothness of your movie. The default frame rate is 12 fps (**frames per second**), which will be fine for most of the movies that you'll produce.

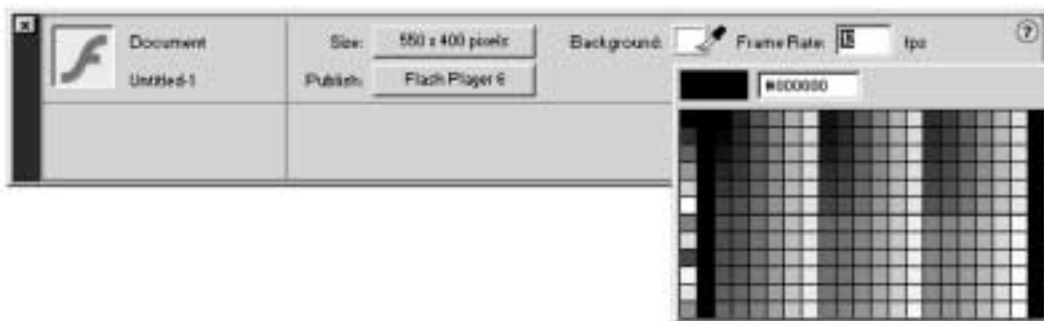
The next movie property we want to concentrate on here is the **Background Color** option. This is the option that you use to change the overall background color your movie has when it's rendered in the user's browser.

Changing the movie's background color

Probably one of the most frequently asked beginner questions in Flash is: 'how do I change the background color?'

Again, the background color is something you should probably think about when you're *planning* your movie. The kind of questions you might ask yourself are: What size will the display window be in the browser? Will the movie take up the whole display in the browser? Do I need to stick to a color scheme that matches the rest of the site that my Flash movie will appear in?

You can change the background color simply by clicking on the **Background Color** box in the Property inspector...



...and selecting a color of your choice. The only drawback is that the movie has to be the same color **throughout the entire movie**. You can't change the color in different parts of the movie.

Another thing you need to think about in this context is the background color of the web page that your movie will appear in. If your finished movie is embedded in an HTML page, the choice you make for the background color of the movie is important. By default, Flash will take the background movie color you specify here and use it as the background color of the HTML page that your movie appears in.

There are a number of different options that you can use when you're exporting your finished movie for publishing on the web. We'll talk about the built-in publishing features of Flash in more depth later in this book.

For now, let's start on a little movie project that we'll be working on over the next couple of chapters: this will be a simple movie that'll get you practicing your Flash skills – and it'll also give you the chance to start expressing your creativity in the Flash authoring environment.

The movie background at midnight

Our test movie is set at night. The cicadas are doing their thing, there's a cool breeze, the moon is up, and there's the faint aroma of fresh mushrooms rising from our garden mushroom patch. We're going to start creating that scene inside of Flash.

The background for our little movie scenario is going to be the night sky, so we need to choose a suitable background color to reflect that.

1. If you don't already have it open, go into the Property inspector (Window > Properties or press CTRL+F3). Knowing that mushrooms only grow while we're asleep, select a deep midnight blue from the Color box in the Property inspector:



2. Clicking on your chosen color will adjust the color of the stage accordingly. You'll notice that when you move the cursor over each color, its identity is displayed using a '#' followed by a **hexadecimal** value (made up of numbers and letters). These hexadecimal (hex) values are the same ones that people use when writing HTML web pages, and each combination of numbers and letters represent a unique, universally recognized color. This makes it very easy for you to match your background and host web page colors if you want to.

Saving global movie settings

The values you set for this particular movie in the Property inspector are saved automatically. If you want these settings to be applied to **all** of your Flash movies, go to the **Document Properties** box (**Modify > Document** or press CTRL+J) and click on the **Make Default** button: Flash will then assume that you want to use these settings whenever you subsequently create a new movie. You can obviously change these default settings as and when you need to.

Having set the global properties for our movie, the next step is to start creating some content. We're going to do that here by walking through a basic creation/modification/animation scenario in the context of our night time movie: this will familiarize you with some essential techniques and give you the chance to stretch those creative muscles.

In introducing you to the creation/modification/animation process, we need to talk about two absolutely critical features of the Flash authoring environment: the **timeline**, and **frames**.

The timeline

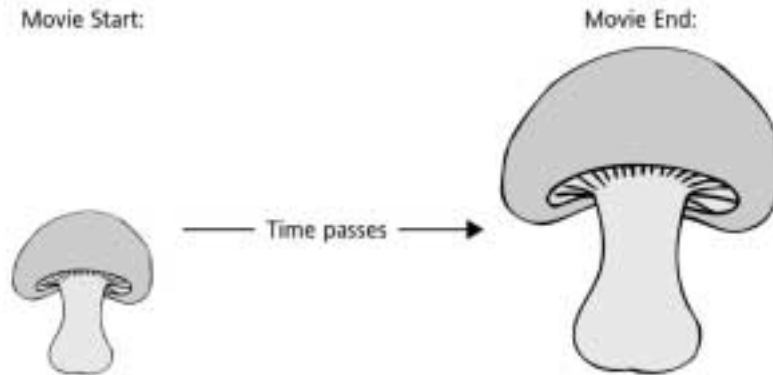
The **timeline** is one of the most important parts of the Flash interface, and one of the most important things to understand as you learn Flash.

When a web surfer visits your site and your movie starts to play, Flash 'interrogates' the timeline encoded in the SWF file and 'reads off' what should be displayed in the user's browser. The relationship between your movie's content and the timeline is absolutely vital in coordinating the end effect that the viewer sees.

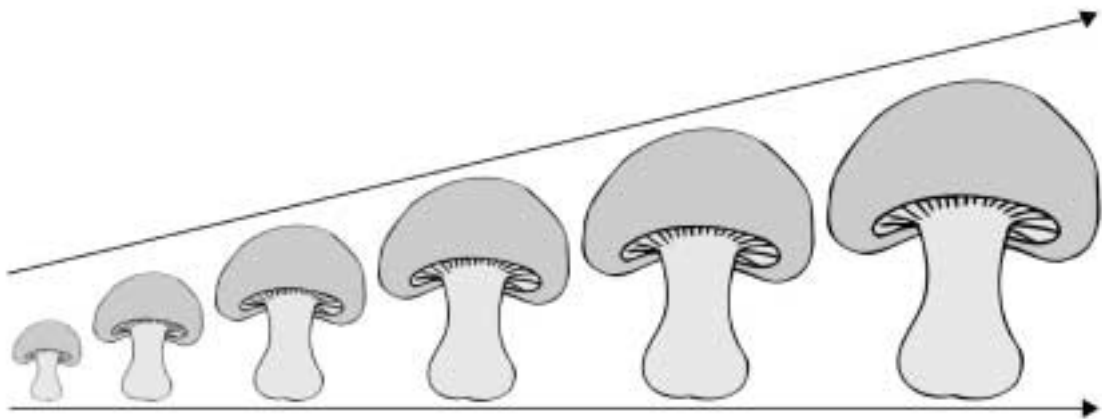
You've already seen that the Flash stage has height and width: the Flash movie also has another critical dimension – **time**. Although Flash movies can have a high degree of user interaction and change the way that they run based on choices that the user makes, the essential experience of seeing a Flash movie is that it **starts** when you open a web page or click on a link, and it **finishes** when the movie has run right through, or when you exit the web page that the Flash movie lives in.

What the user sees in their browser between the start and end points of the movie is determined by the **content** that you create in your movie, and by how you use the timeline to **organize** that content. The length of your movie's timeline will control how long the movie runs for, and how content changes in the movie over time. As the movie's author, you control all of this by using the Flash timeline in conjunction with the content you create.

To help you visualize this, picture a simple movie scenario: imagine a time-lapse nature movie that shows a mushroom growing in slow motion. At the *start* of the movie, you'd have the tiny head of a mushroom poking up from the soil, and at the *end* of the movie, you'd see a full-grown mushroom, standing tall above the grass:



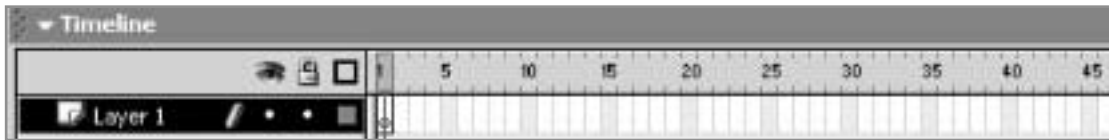
Clearly, your movie won't just consist of those two images, and your movie won't just jump from 'tiny stalk' to 'fully grown' instantaneously: in between the two points, time passes, and your mushroom moves through the intermediate stages of growth until it attains its full-grown state. In between the start and end-points of the time-lapse movie will be a whole series of images that represent the mushroom at the different stages of its development:



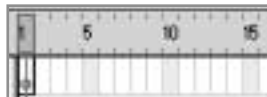
In the real world, mushrooms have a natural growth rate, which is determined by mushroom DNA or by divine guidance (depending on who you ask). In a Flash movie, you have to *imitate* the effect of time passing – and that's where the timeline comes in: to create the effect of time passing, you need a start point, an end point, and some space in between that represents the passage of time. The timeline lets you do this, and gives you complete control over the length of your movie, the speed that it plays at, and what you see on the screen at each point in time. How? By using **frames**.

Frames

Take a look at the timeline in the Flash authoring environment:



Now concentrate on the right part of the actual timeline:



You can see that the top part of the timeline (the all-gray part) is divided up into numbered segments, and that there are corresponding divisions in the next line down, consisting of groups of four white rectangles separated by single gray rectangles.

Each of these little boxes on the second line down represents a **frame**, and the numbers on the top line give us a frame number reference: frame 1 on the left, through to frame 45 (and beyond) on the right. Your movie can be 1 frame long, or it can be thousands of frames long. The length of your movie is determined by the highest number frame you create content for.

The (red) rectangle with the line coming out of it on the timeline is called the **playhead**.

The playhead

The playhead indicates where you currently are in your movie. You can click on the playhead and drag it back and forth along the timeline and the stage will change to show the contents in the selected frame. At the moment, of course, we don't have any images or objects on the stage, but as we start to add content you'll see how you can spool through your movie using the playhead.

The playhead lets us anticipate what'll be seen when the movie plays back in the browser: when we position the playhead over a frame, we see what'll be displayed on the screen at that point during playback.

Our movie begins in frame 1. By default, when it loads into the user's browser, our movie will start at frame 1 and play through the rest of the frames in sequence until the end point. Each successive frame on the timeline represents a moment in time, a moment that'll be played back when our movie runs.

At the moment, all of the frames in the timeline shown here are empty – that's because we haven't created any content yet. If we were to play back our night time movie now (by pressing `CTRL+F12`), we'd just see a block of color displayed in the browser window. This is Flash displaying the background color of the movie that we set using the Property inspector. To make our movie more interesting, we have to add content at various points along the timeline so that they can play back in the browser. This is where frames come in.

By placing content in frames at different points along the timeline, we imitate the passing of time, make animations, and generally make things happen in our movie: just like that time-lapsed mushroom movie we mentioned earlier, we can create start and end points and show all of the intermediate stages in

between. When the movie plays, it looks at each frame on the timeline and renders what it finds there in the user's browser. To achieve the effect of content appearing and disappearing, animating and morphing, Flash provides us with different types of frames that we can use.

To see the nature and effects of the different types of frame that Flash gives us, we first have to get some content onto the stage. So without further ado, it's time to start drawing.

Making mushrooms

What we're going to do here is create a mushroom that will live out its life under the midnight sky that we created as our background earlier.

1. First of all, click on the **Snap to Objects** button on the Tools panel:



*On the Mac, you'll need to use the **View > Snap to Objects** menu option.*

*If you don't see the Tools panel on your PC, use the **Window > Tools** menu option to display it. Alternatively, you can display the **Snap to Objects** button in your main toolbar at the top of the screen by selecting the **Window > Toolbars > Main** menu option. You can dock the toolbars in Flash MX the same as you can in other packages.*

The button will appear to be 'depressed' when you've turned the option on.

Turning on **Snap to Objects** invokes Flash's ability to help us make our drawings more precise: this option will automatically 'snap' the drawing cursor to certain points as Flash anticipates what you're trying to do. For example, when you turn snapping on and draw with the Oval tool, Flash will jump to a perfect circle when you get close to it, but with snapping off, it will let you draw the oval to whatever dimensions you want. Whether you use this snapping feature or not is entirely dependent on personal preference and what you're trying to achieve: if you're drawing perfect circles it makes sense to leave it on, but for most freehand drawings it's easier to leave it off. Experiment and see what works best for you in different circumstances.

Next, we need to select the drawing tool that we'll use to draw our initial mushroom. All the drawing tools are accessed in the Flash **Tools panel**. We'll be detailing more aspects of the Tools panel and its contents in the next chapter, but we'll start making use of some of its features as we work through this example.

- The Tools panel is located (by default) on the left-hand side of the authoring environment. If you hold the mouse cursor over any of the tool buttons in the Tools panel, a tool tip will pop up with a description of the tool that the button represents:

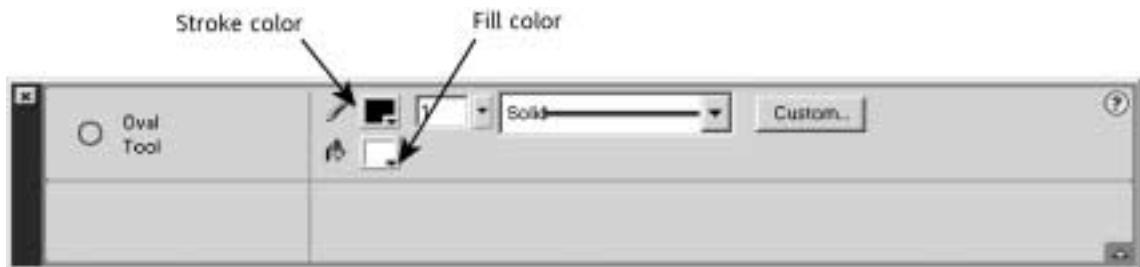


The letter in brackets next to the tool's name is the shortcut key for that particular tool. In this case, pressing O will select the Oval tool.

- Click on the Oval tool. Each of the tools will be covered fully in the next chapter, but we'll dip into them now as we set about creating some content.

Now we need to select a color for the outline of the shape that we're going to draw.

- If you're not already displaying it, bring up your Property inspector by pressing CTRL+F3. You'll notice that some aspects have changed from what we've previously seen inside the Property inspector. The Property inspector now reflects that we have the Oval tool selected:

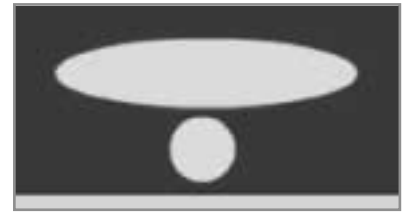


From here, you can pick a color for the outline (**Stroke color**) of the object that you're drawing. Similarly, the **Fill color** box allows you to select the color that you want to use to fill the area **inside** the outline. By default, many of the objects that you draw using tools from the Tools panel have an outline and a filled interior. You can control and modify these outlines and fills to a very fine degree, as you'll see as we progress through this book.

- Set the stroke color to a dark brown.
- Next, click on the Fill color box and set the fill color to a lighter, mushroom-like brown.
- Now for the actual drawing. At the bottom of the stage, about half way along, click and drag with the Oval tool to make a small stalk for your mushroom:



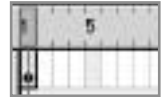
8. Now, using the same click and drag action, draw a flatter, larger oval just above the stalk. Make sure that there's a gap – a small one – between the two ovals:



This larger oval will be the cap of your baby mushroom.

9. If you look at the timeline now, you'll notice that the first frame has changed:

If you look closely at the second line down, you'll see that this frame is now shaded a darker gray, and that it contains a little black circle. Why is that?



Well, in a new movie like ours, Flash always assumes that we want the action to start in frame 1. So, when we started drawing on the stage, Flash thought:

'Aha, they're drawing an oval. This is the first piece of content that they're creating for this movie, so I'll put this drawing in frame 1 – that's where the movie will start, after all.'

So Flash has created the starting point for our movie, based on the drawing that we've made. This is the first fixed point in time for our movie – the first image in a sequence of images that we want to display changing over time. If you like, this is the first snapshot in a sequence that we want to show to the user.

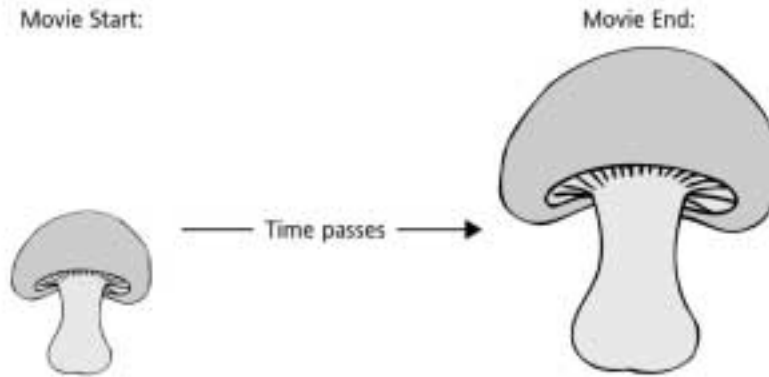
Flash uses a particular type of frame to store fixed points in time that hold visual (or other) content: this type of frame is called a **keyframe**.

Keyframes

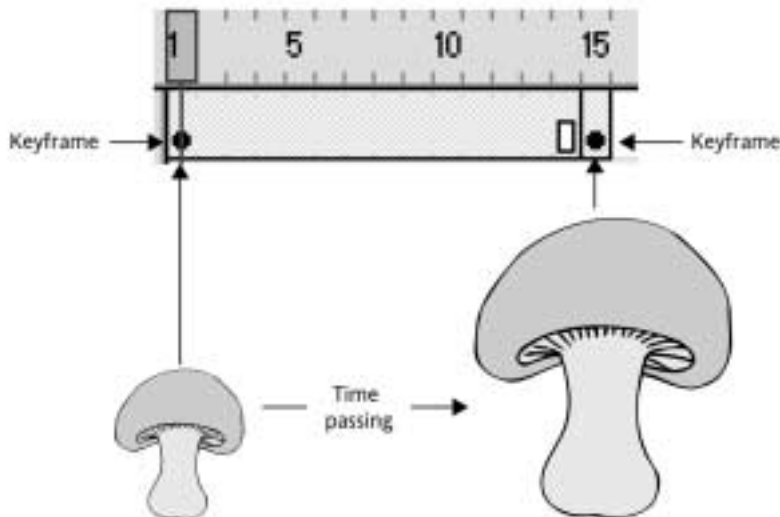
A keyframe indicates that something important happens at the point in the timeline where the keyframe is located: for example, we display some content, or we make something disappear, or we make a transition from one piece of content to another. Keyframes are markers in time, indicating start and end points for different pieces of action.

Essentially, a keyframe says to Flash, 'there's something significant here'. Your Flash movies will consist of a series of keyframes spaced out along the timeline, with each keyframe flagging up the start and end points of distinct pieces of content that are to be displayed in the user's browser.

Think back to the time-lapsed nature movie that we discussed earlier. Remember how we said that the start and end points of that movie would be the baby mushroom and the full-grown mushroom?



In Flash, those start and end points are defined by keyframes. We'd need a keyframe in frame 1 that contained the baby mushroom image, and a keyframe in a later frame (say frame 15) containing the image of the fully developed mushroom:



These two keyframes will tell Flash that we have images of our mushroom in two different states: 'starting to grow', and 'finished growing'. Once we have these two keyframes in place, we can get Flash to help us create and animate the intermediate growth states for the mushroom that should exist between the start and end images. We'll be looking at how to do that in more detail very shortly. For the moment, remember this:

If something significant changes on screen in your movie, there'll probably be a keyframe involved. Keyframes are the 'key' to making things happen.

Much more on frames in a moment. First, a bit of housekeeping...

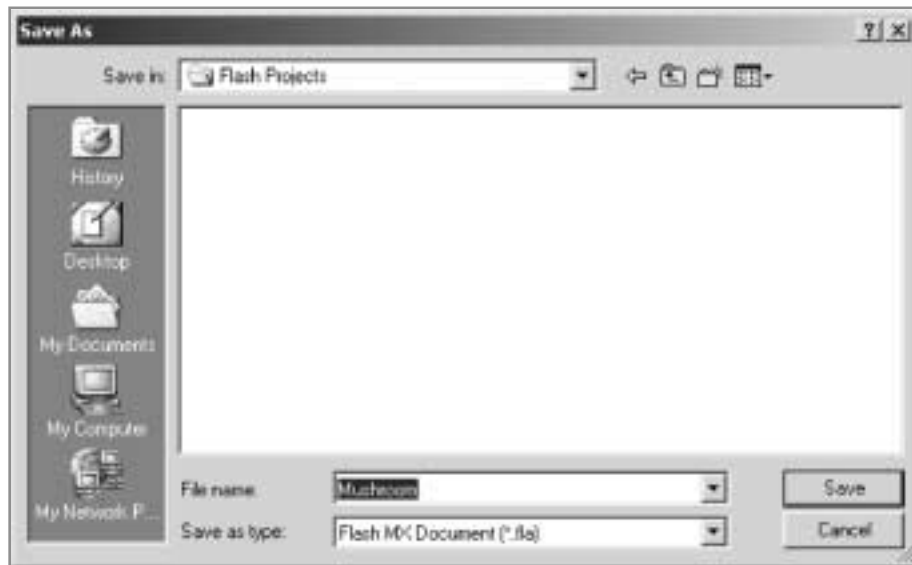
Saving your movie

It's good practice to save your work often – just in case your dog eats it or the computer crashes and burns. The way to save your embryonic Flash movie is the same way that you would in most other programs:

1. Click on **File > Save** in the main menu.

You'll be prompted with a dialog box asking where you want to save the movie, and what name you would like to give it.

2. Choose (or create) a suitably named folder to save your work into, and then type `Mushroom` into the **File name** box and click **Save**:



Flash will automatically save the movie that you're working on as an `FLA` file, which is the extension used for Flash authoring files.

Remember that there's a distinction between the file that you work on in the authoring environment (the FLA file) and the file that's loaded and played when a surfer visits your site (the SWF file): as an author, you create and modify your content inside the FLA, and the SWF is created when you publish your movie. The publishing process takes all of the drawings and other content that you created in your FLA file and compresses and encodes them into the SWF format. The SWF file is much smaller and more compact than its parent FLA file, and is thus much quicker and efficient to download for the end user.

Now that we've got our example movie safely saved away, we can continue our exploration of frames.

We've already drawn a couple of ovals in frame 1: these represent our baby mushroom – the starting point for our movie. If we were to play the movie now, we'd see a static image of our baby mushroom in the browser (you can try this by pressing CTRL+F12. This will open up your default browser and render your movie – you'll need to close the browser or click back into your Flash MX window once you've finished viewing your masterpiece).

The reason that we see this static image is that we've only created a single frame with any content in – the movie has nowhere else to go except this single keyframe. When Flash plays back the movie and looks at the timeline, it only finds this single keyframe. Flash's default publishing settings tell it to **loop the playback** of our movie, so what we're seeing in the browser is the same single frame, being shown over and over again as Flash repeatedly loops through the timeline. Flash will continue to do this until we close the browser or move onto another web page.

Next, we're going to see how to add an *end* point for our 'mushroom growth' movie, and how to create the sequence of images that imitate the passage of time and show the mushroom growing. This will mean that we'll have a proper movie to play rather than a single-frame loop.

Making the movie really move

We've already seen that keyframes mark the beginning and end points of pieces of action in our movie. In this sense, Flash can be compared to traditional animated cartoons. In traditional 'toons, the animator would plan out the action sequence that they wanted to create – for example, a car driving from one side of the screen to the other. They would create the background for the action – maybe a desert setting with some distant mountains – and this background would typically remain static.

Next, the animator would create their start and end images of the car: the car on the left of the screen, and the car on the right of the screen. Clearly, you'd also need to have a series of intermediate images that showed the car in progress across the desert. Each image would be drawn onto a separate sheet of transparent acetate. The plan would be to use a movie camera to photograph the start, intermediate, and end images against the static background to create a number of different frames of film. Running the frames in sequence would show the car in motion against the static background.

This was a time-consuming business, and cartoon production companies didn't want to tie up their star animators in slaving over frame after frame of minutely-changing action. The solution was to get the lead

animators to create the key images – start and end points, and important transitional images – and then use specialized ‘in-betweeners’ to sit down and draw up the images that came in between the keyframes that the expert animator had created. These in-between frames were critical to ensuring that the cartoon action was smooth and convincing.

In Flash, we do things similarly. We can create keyframes that define significant stages in the action that we want to show, and then get Flash to generate the in-between frames that link the keyframes together. This saves us a lot of time and effort in creating the transitional frames, and is an important factor in making Flash the successful animation package that it has become. (Note that you *can* mimic the traditional animation method in Flash if you wish, by hand-drawing each frame individually. This can be a powerful way to express yourself and create great animation, but it’s really too big a subject to tackle. Maybe in another book...)

Let’s get back to our mushroom movie. Currently, we have that single starting keyframe containing the baby mushroom. Next, we want to create a keyframe holding the fully-grown version, and then get Flash to create all the in-between frames that show the mushroom growing over time – let’s work through that in an exercise now.

Working with frames

As we’ve said already, a keyframe marks a significant change in your movie. In order to create the image showing our mushroom when it’s fully grown, we need to add a keyframe to the timeline. This will tell Flash that we have some important new content that it needs to be aware of.

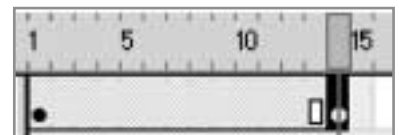
To insert a new keyframe to the timeline, you simply click in the timeline at the position where you want to add your keyframe. Once you’ve selected the frame position, you can then choose the **Insert > Keyframe** menu option, or use the keyboard shortcut F6, and a new keyframe will appear.

It might occur to you to ask, ‘how far along the timeline should I put my second keyframe?’ The answer to this depends on how long you want your piece of action to last. The thing to remember here is the frame rate setting that we saw in the Property inspector. Remember we said that the 12 frames per second default rate was about right for most basic movies? This frame rate setting means that **for every 12 frames our movie takes up on the timeline, there will be one second of action when the movie plays back**. So, the math is fairly simple: estimate how long you want the action to last in seconds, and multiply the number of seconds by the frame rate. You can always change the position of the keyframes later on if you need to tweak the timing of different pieces of action.

Let’s make that mushroom grow.

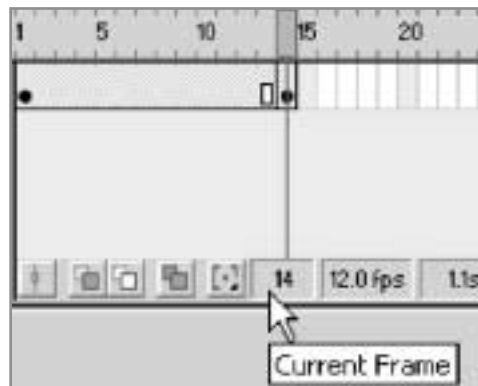
1. Click on frame 14 of your mushroom movie and press F6 to insert a keyframe.

You’ll see immediately that the timeline has changed:



Frame 14 now has a black circle in it and a black border (note that the circle is white and the frame black in the image above because the frame is selected), and all the frames between frames 1 and 14 have turned gray. Additionally, there's a small white rectangle in frame 13, and the playhead has jumped to frame 14. Let's take a look at what this all means:

- The black border around frame 14 indicates that this is a keyframe. We're going to use this keyframe to hold some content that's different from the frames that precede it.
- The white rectangle in frame 13 tells us that this is the last frame before a new keyframe, and that all of the frames to the left of this rectangle contain the same content as the **previous** keyframe – in this case, frame 1. The frames are grayed to show you that they contain the same content as frame 1 – every frame in the black-bordered box running from frame 1 through frame 13 contains the same image of the baby mushroom that we created in frame 1.
- The black circle in the keyframe at frame 14 indicates that this frame contains some content. But how come? We didn't add any content to this frame yet, did we? What's happened is that Flash has carried over the content from the previous keyframe (frame 1). This is the default behavior when you add a new keyframe. This feature can be very useful, as you'll see in a moment.
- There's one other thing to note here: when you move the playhead backwards and forwards through your movie, the current frame number is also indicated in the area just underneath the timeline:



So far, then, we've got an opening keyframe (frame 1), followed by some intermediate frames (frames 2 – 13), and a new keyframe (frame 14) that will soon contain the image of our fully-grown mushroom.

All of the frames from 2 to 13 now 'belong' to the keyframe in frame 1, and reflect what is in it. If you changed the image in frame 1, the following 'slave' frames (2 – 13) would change to reflect the new picture. Frame 14 would remain as it is, however, as we've told Flash that we want this frame to be self-contained so that it can contain new content.

2. Double-click anywhere on the timeline between the two keyframes. You'll see that frames 1 through 13 turn black:



This shows you that these frames all hold the same content – they’re all dependent on the content in the keyframe at frame 1. You can’t alter the content of frames 2 through 13 by editing them directly – you can only change their contents *indirectly* by amending what’s in frame 1. **You can only directly edit content that’s in a keyframe.**

3. Deselect the black-highlighted frames by clicking away from them. Next, click anywhere between the keyframes. This time only the frame that you clicked on is selected:



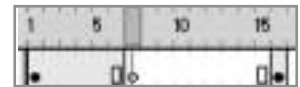
4. Use the **Insert > Frame** menu option or press the F5 key twice to insert two new frames into the timeline at the position where you just clicked with the mouse. This will add two frames to the timeline, and these new frames will inherit the contents from the preceding keyframe (frame 1).

Using the playhead to move through your movie, you can see that it is now 16 frames long, and each of the frames contains the same picture of a mushroom. We promise that your mushroom will grow soon, but first there are a few more tricks we can perform with frames.

5. Click between the keyframes again to select a single frame.

This time, rather than inserting ‘slave’ frames, we’re going to insert a **blank keyframe**. A blank keyframe is just what its name suggests, a keyframe that has no content in it. It is, however, independent of the content of the keyframe that precedes it.

6. Use the **Insert > Blank Keyframe** menu option or press the F7 key to convert one of the normal frames in your timeline into a blank keyframe:



A blank keyframe is represented in a similar way to a keyframe, except that it shows an unfilled circle in the timeline – this is because it doesn’t yet have any content. The remaining white frames are all dependent on the blank keyframe that we just inserted.

7. If you now scroll through your movie using the playhead, you’ll see that all of the dependent frames after the blank keyframe are also blank, reflecting the fact that the keyframe before them was empty. Blank keyframes are useful for stopping animations or dividing different pieces of content that exist on the same layer (stay tuned, there’s more on layers later).

You’ll also see that the keyframe at frame 16 still has the baby mushroom image in it. That’s because it inherited the content of frame 1 when we created it. However, there is no link between the inherited image and the current (possibly amended) content of frame 1 – frame 16’s keyframe is completely independent.

We don’t actually want half of the frames in our movie to be blank, so somehow we’ll have to get rid of that blank keyframe and its dependent frames. There are two ways to do this: you can either delete the blank keyframe or convert it into a normal frame. We’re going to do the latter – the advantage of this is that the movie’s length remains the same.

8. Click on the blank keyframe and use the **Insert > Clear Keyframe** menu option or press **SHIFT+F6**. This menu option name is a little confusing – what we’re actually doing here is converting the keyframe to a normal frame – we’re not actually *inserting* anything as such. The result of doing this

is that the white frames become shaded again, and if you run the movie through now, you'll see that they've all been refilled with your baby mushroom picture.

So you now have 16 frames in your movie, all filled with exactly the same mushroom image. We only wanted the movie to last for 15 frames because it's a nice round number, so you'll have to remove one of those frames now...

9. Click to select a single frame anywhere between the two keyframes and use the **Insert > Remove Frames** menu option or press Shift+F5 to delete the highlighted frame. You should now be left with two keyframes on your timeline (in frames 1 and 15), and a set of identical normal frames between the keyframes, containing the same image of your mushroom that you created in frame 1.

The 'normal' frames that separate out the keyframes may appear to be plain and boring at the moment, but don't dismiss them: they're the Flash equivalent of pawns on a chessboard or foot soldiers in an army – not as glamorous as the other elements, but just as important.

*In Flash there are three different methods for adding, converting, and deleting frames, each with its own advantages and disadvantages. The first, and most 'formal' method is to use the **Insert** menu. This menu, found on the main menu bar, contains all of the manipulation actions associated with frames. The second, and probably most commonly used method is to use the keyboard shortcuts that mimic the **Insert** menu options. The major problem with this is that you have to learn them first, although this is something that most people pick up quite quickly through use. The third method is to right-click on the relevant frame in the timeline to highlight it, and then to select the appropriate command from the context-sensitive menu that appears. Throughout the tutorial sections in this book, we'll use a mixture of these methods. There is no single 'best' method – whichever one you find easiest is the one that you should use.*

As promised, let's make our mushroom live and...well, breathe.

Making the mushroom grow

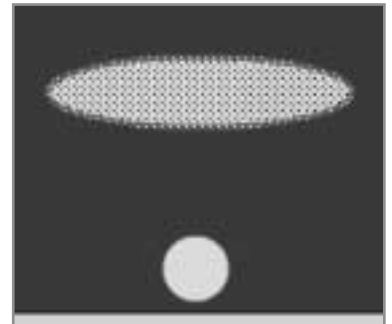
Frame 15 is going to be the final frame of our movie, which means that we need to populate it with the content that represents the final growth stage of our mushroom. Let's add the fully-grown mushroom image to the keyframe in frame 15. What we want to do is have our mushroom grow from the baby form in frame 1 to an adult version in frame 15.

1. Start by clicking on frame 15 in the timeline. Notice that when you click on the frame the whole mushroom (the two ovals we drew earlier) is already selected. Remember, these ovals were inherited from the keyframe in frame 1, so they're identical in shape and position to the mushroom

in frame 1. This in turn means that we don't have to worry about positioning the image of the mushroom in frame 15 – it's in exactly the same place as it in frame 1 – this will make our animation easier to perform.

*A related, and very useful feature in Flash is the ability to copy (or cut) and paste content into exactly the same location on the stage. This is particularly useful when you want to paste images or other components into other keyframes or onto other layers and still have them occupy the same coordinates on the stage as the original image. To achieve this, you copy (or cut) the original component, and then use the **Edit > Paste in Place** menu option. This way you're sure to place your object exactly where you want it. We guarantee that you'll find this feature immensely useful in your Flash career.*

2. Click on the Arrow tool in the top left corner of the Tools panel, and then click on the background of the stage to deselect the mushroom.
3. Now double-click in the middle of the cap of your mushroom: this will select both the stroke and the fill. If you had only clicked once in the middle of the cap, you would only have selected the fill.
4. Next, click on the mushroom cap and drag it up to where you want the cap of your full-grown mushroom to be. (Alternatively, holding **SHIFT** and using the arrow keys makes your selection move in units of 10 pixels):

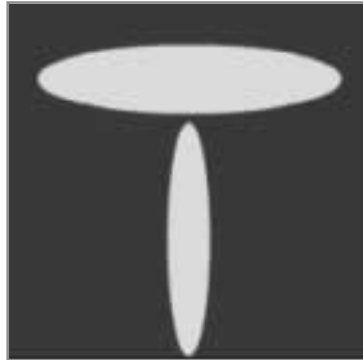


*If you hold down the **SHIFT** key while you drag, Flash will help you drag the mushroom cap upwards in a straight line (provided that you've turned on the **Snap to Objects** option).*

5. Double-click on the stalk (the lower oval) to select it, and then press **BACKSPACE**. This will delete the baby version of the stalk from frame 15 (although the old, smaller stalk is still intact in frames 1 through 14).

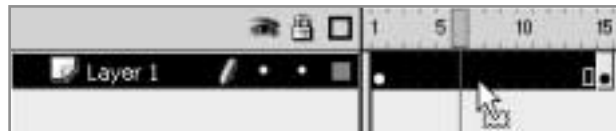
Our next task is to create the fully-grown version of the mushroom stalk.

6. Click on the Oval tool in the Tools panel again, and draw a long thin oval from the bottom of the cap of your mushroom down to the bottom of the stage, again ensuring that there's a small gap left between the edges of the two ovals. This is your adult mushroom stalk:



*If you need to move your new stalk when you've drawn it, select the Arrow tool, double-click on the stalk, and drag it to the desired position. If things go wrong, you can always use the 'undo' option by selecting **Edit > Undo** from the menu bar, or pressing (CTRL+Z).*

7. OK. Press the **ENTER** key to preview your movie. Hmm...not particularly convincing, is it? We're getting the same picture for 14 frames, and then a sudden jump to a fully-grown fungus. What we really want is a smooth transition from 'button' mushroom to 'big mushroom', and Flash can do this for us. You're about to see how Flash can perform as an underpaid, unappreciated 'in-betweener'.
8. Double-click somewhere in between the two keyframes on the timeline to select the first keyframe and all of the normal frames that depend on it:



What we're going to do now is change the behavior of these frames. We're going to tell Flash that we want to create an animation that smoothly transforms the small mushroom in frame 1 into the fully-grown version that we've just created in frame 15. Our ability to do this is entirely dependent

on the existence of our two keyframes: the keyframes define the two different states of the mushroom, and we're asking Flash to create all of the in-between frames that'll represent the growth of the mushroom. Let's do that.

9. If you don't have your Property inspector open, right-click on the frames and select **Properties** from the context-sensitive menu. Using the Property inspector, we're going to give these frames a label, and create our growth animation:



*In Flash, animation is called **tweening**. This might seem an odd name, until you realize that it's short for 'in-betweening', or the process of creating the transitional frames that go in-between the keyframes.*

10. Click inside the furthest left box (under the word **Frame**) in the Property inspector and give your frames a label that identifies them – we've called ours `Mushroom Growth`. This attaches the label to frame 1.

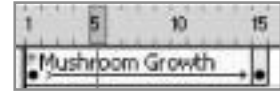
Using labels makes it easier to identify specific bits of action inside a large and complex movie. If, as we've suggested, you keep objects on separate layers, and name those layers, then you should not need to use large amounts of labels. They'll come in useful later on when we start to deal with actions, but this will be discussed in greater detail when we come to it.

11. Now click on the drop-down menu next to **Tween**, and select **Shape** from it:



This will automatically create a **shape tween**: Flash will decide that we want the stalk in frame 1 to morph into the stalk in frame 15, and that we want the cap in frame 1 to morph into the cap in frame 15, and it'll automatically generate the in-between images in frames 2 through 14 that will produce this effect.

12. Click on the playhead and notice that the tweened frames on the timeline have now been colored green by Flash, indicating a shape tween, and that there's an arrow pointing from frame 1 to frame 15 – this indicates the length of the tween. Finally, Flash has added our label to the tweened frames, and appended a little red flag to draw our attention to it:



(For clarity, we've removed the label from subsequent screenshots.)

13. Slowly drag the playhead back to Frame 1, noticing that Flash has automatically filled in all of the animation frames between 1 and 15. Now press ENTER to preview the movie, and the mushroom will steadily grow to its full-size.
14. Save your happy-grow-lucky mushroom.

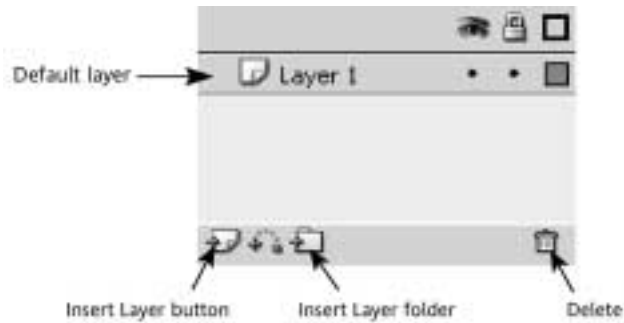
We've spent some time looking at frames in Flash, and started to see how they can help us achieve the effects that we're after in our movies. What we'll do next is introduce you to another vital element of Flash authoring files – **layers**.

Layers

Whereas the timeline and its frames help us organize and manipulate content over time, layers help us add depth to the movie, and allow us to separate out pieces of content and action that would otherwise get tangled up. If you had to place your entire movie content on a single layer, it would be horrendously difficult to achieve anything complicated. By separating out the action onto different layers, you can create much more convincing and complex movies, and make full use of the flexibility and power that Flash's timeline gives you. Multiple layers mean that your movies can have a host of different elements on the stage, all acting completely independently of each other.

A traditional animator would have a different set of sheets of acetate for each part of their cartoon. For example, the background forest would be on one set of sheets, Red Riding-Hood would be happily skipping on another set, and the Big Bad Wolf would be stalking her from yet another set. By keeping the parts separate, the animator had much greater control over the individual aspects of their cartoon. It meant that if something needed to be altered, the animator could change just one set – say, adding an evil twinkle to the Wolf's eye – without having to redraw the forest or Miss Riding-Hood as well. Another benefit of having separate animations on different sets of acetate is that they can be re-used later in the cartoon, or indeed in a completely different cartoon, so while Jack and Jill are running up the hill in cartoon two, cartoon one's Little Red Riding-Hood can be happily skipping along below them without having to be redrawn. The content on the different layers is independent and portable.

In Flash, layers are shown to the left of the timeline. Each new Flash movie comes with a single layer by default:



This layer – **Layer 1** – is the default layer that we’ve been working with in our mushroom movie so far.

In Flash, layers are the equivalent of those separate sheets of acetate containing different visual components. Layers make movies easier to alter, and allow for much greater richness of content. Let’s take a look at what layers allow us to do in our sample movie.

Working with layers

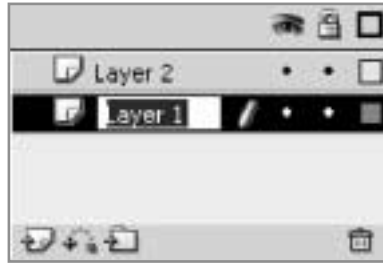
It’s good practice to keep each element of your movie on a separate layer for ease of editing and for neatness. We’ll see how this works in this exercise.

1. The active layer in Flash is the highlighted layer with the pencil icon next to its name: this pencil icon indicates that this is the layer that you’ve currently selected to work on.
2. Click on the Insert Layer button. Flash will create a new layer above **Layer 1** and call it **Layer 2**.
3. Notice that Flash has automatically made **Layer 2** 15 frames long to match the length of **Layer 1**. If you look at any of these frames though, they’ll still all be empty:



Flash always calls a new layer **Layer n** where *n* is the next number up from the last layer that you created. This means that even if you subsequently deleted a layer, Flash would still increment the next layer’s number as if the deleted layer still existed. For example, if you deleted **Layer 2** and then added another layer, it would still be called **Layer 3** even though there was no **Layer 2**. Luckily, the good people at Macromedia understand just how confused you are at the moment, so instead of trying to work out which layer is which, they’ve given us the ability to uniquely name each layer – which means that we can forget about the whole numbers shebang, and work with meaningful, descriptive layer names instead. Giving names to your movie’s layers is another good habit to get into, and it’ll save you a lot of heartache.

4. Double-click on the name **Layer 1** in the timeline. When you double-click it, it will become editable, allowing you to change the name of the layer:



5. Type `mushroom` and press `ENTER`.
6. We want **Layer 2** to contain a picture of the moon in front of our night-sky background, so double-click where it says **Layer 2**, type `moon` and press `ENTER`. We now have our two layers meaningfully named and we can instantly infer what's on each of them.
7. Click on the Oval tool. Our moon will be a full one, so we'll use this tool to create the celestial body.
8. Click on the **Default Colors** button under the two color boxes at the bottom of the Tools panel:



This button will reset the colors to a black stroke with a white fill.

9. Still on the **moon** layer, use the playhead to go to the final frame of the movie, and then draw a circle (remember, you can hold down the `SHIFT` key to help you draw a perfect circle) over the top of the right-hand side of your fully-grown mushroom cap:



If you look at the timeline, you'll notice that Flash has automatically created a keyframe at frame 1 of the **moon** layer, and that it has also populated all of the frames on this layer with the image of the moon. This is a handy effect, as we want the moon to be visible in the sky throughout our movie. If we'd wanted the moon to appear in the movie from frame 15 onwards (and not before), we'd have needed to create a keyframe at frame 15 in the **moon** layer.

10. Press **ENTER** to play your movie. Flash has zipped through the timeline and displayed the content of both layers for us.

But something looks slightly wrong, doesn't it? The mushroom seems to be further away than the moon, and bigger, too. If we're going to get the perspective right, then you'll need to put the moon *behind* the mushroom. We need to get the layers' **stacking order** right.

The higher up a layer appears in the layer list on the left of the timeline, the closer the contents of that layer appear to you on the screen. So to get your mushroom in front of the moon, you just have to move its layer above the moon's layer in the layer list. Let's do that now.

11. Click on the name of the **mushroom** layer in the layer list and drag it above the **moon** layer. You'll see a shaded bar appear while you're dragging the layer:



This indicates where the layer will go when you release the mouse.

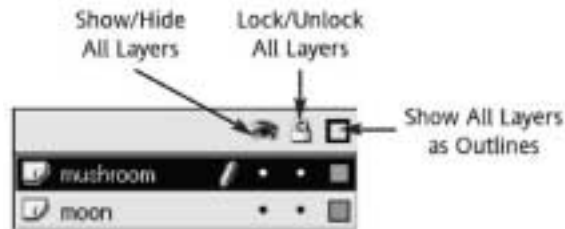
12. Now when you play your movie, the mushroom should rise in front of the moon. Perspective has been returned, and the tale of the very big mushroom and the very small moon has been consigned once more to legend.

Deleting a layer is as easy as adding a layer – it only takes the click of a button. To delete a layer, you'd just click on the layer to highlight it, and then click on the **Delete Layer** button, which is indicated by the trash can icon underneath the layer listing. You should note that Flash will not let you delete *all* the layers in a movie: there must always be at least one layer left. If you're trying to delete everything from a movie and start again, it's easier to close the current movie and start a new one.

As we've seen, layers are a useful way to manage and control our content. Flash MX has the added ability of using layer folders to bundle together similar layers for further ease – more on these in Chapter 4.

Layer modes

Layer modes define how you view and alter specific layers in the authoring environment. There are three layer modes in Flash and, by default, they're all turned 'off'. The three modes are controlled by clicking on the icons in the three columns after the layer name, and the status of the modes for a particular layer is indicated by the two dots and the square next to that layer's name:



The first column controls showing and hiding a layer, the second controls locking and unlocking a layer, and the third column is used for viewing the contents of that layer as outlines. Let's take a look at how these different modes interact in the authoring environment, and see what benefits they give us.

Working with layer modes

With all three of the layer mode selector icons in the 'off' position, the authoring environment will behave exactly as it has done so far: if you were to start drawing a circle with the Oval tool right now, it would appear as normal on the currently selected (active) layer. Let's see what happens when we start switching the layer modes 'on'.

1. In the **mushroom** layer, click on the dot underneath the picture of the eye.

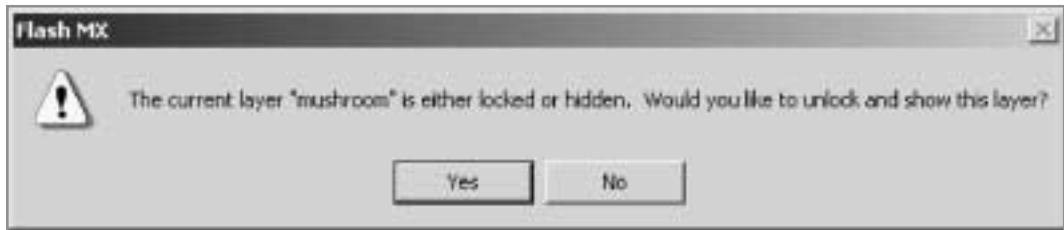
Three things will happen:

- As we've hidden the layer, its contents will disappear from the stage in the authoring environment.
- A red cross will replace the dot in the eye column, reminding us that this layer is currently hidden.
- The pencil icon will have a red line struck through it, indicating that the contents of the hidden layer cannot be edited. If you try to draw on the stage now, you'll find that the cursor has changed to a pencil with a warning circle next to it:



The warning circle tells us that the currently selected tool cannot be used at the moment. The logic here is that if you were able to draw on the hidden layer, you'd be able to unwittingly draw all over the content that you'd created so carefully already. Flash is protecting you from yourself.

If you were to actually click on the stage with the Oval tool, you'd get an error prompt:



Hiding layers is very helpful when you need to concentrate on some content on a particular layer and you don't want to have the contents of other layers obstructing your view. For example, in the final frame of your mushroom movie, the mushroom covers part of the moon. If you wanted to draw a face on your moon it would be hard to see what you were doing because the mushroom would be in the way. By hiding the **mushroom** layer, you can see the whole of the moon, select the **moon** layer and draw away to your heart's content. You could then click on the cross in the eye column and the **mushroom** layer would be visible again.

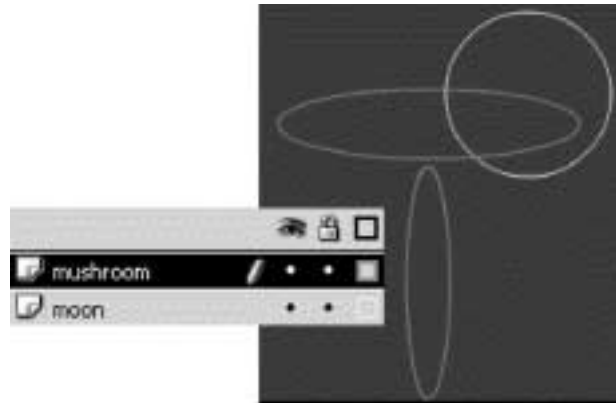
2. Click on the cross in the eye column to return everything to normal. Now click on the dot in the 'lock' column. A small padlock will replace the dot, and the pencil will again have a line through it. This time, although the layer is locked and you cannot draw or select objects on it, the mushroom will still be visible.

Locking layers allows you to work with objects above or below them without accidentally selecting anything in the locked layers. This is useful when you're drawing or modifying on one layer and you want to see the contents of the other layers to keep things in context. Locking the surrounding layers means you can draw and edit confidently, secure in the knowledge that you won't mess anything up on the other layers.

The final mode allows you to display all objects on a layer as **outlines only**, rather than as filled shapes.

3. Unlock the **mushroom** layer by clicking on the small padlock across from the layer name. Now click on the colored box to the right – the mushroom will be reduced to just an outline. The outline takes its color from the color of the box that you just clicked on. Each layer will have its own dedicated outline color (automatically allocated by Flash), so you can easily make out which objects belong to which layer.

4. Click on the outline button on the **moon** layer, and (on your monitor, at least!) you'll see that the moon's outline is a different color to the mushroom's:



Outline mode is useful for helping you get a grip on exactly what's in your movie across all layers. Things start to get complicated when you have lots of different things on lots of layers and you can't quite see what is going on. Outline mode lets you step back from the jumble and get a clear view. Outline mode is also helpful in previewing your movie, as Flash finds animating outlines easier to render than fully filled shapes, meaning that the movie preview displays quicker.

You can easily change the layer mode for *all* of the layers by using the icons at the top of each of the columns. So, to turn outline mode *off* for both of our movie's layers, just click on the black box next to the eye and the lock: This will instantly return your mushroom and moon to their full-color glory.

If you're working with a lot of layers and want to lock every layer except the layer you're currently working on, it is easiest to click on the **Lock/Unlock All Layers** button to lock every layer, and then click on the small padlock on the layer that you want to use to just unlock that layer. The same principle applies to the show/hide feature invoked via the eye icon.

Using layer folders (see Chapter 4), it's possible to lock or hide the content of a folder with one click. Locking or hiding the folder layer also affects all of the content within the layer in the same way. This makes it easy to show/hide or lock/unlock similarly grouped elements while editing other content on the stage.

Layer modes only influence how you see the layers as you are constructing them: they have no effect on the final movie that you create, so layers that are hidden in the authoring environment will be visible in the final movie. Similarly, layers that are in outline mode will be seen in full color when they're rendered in the browser.

Now let's move on to the last Flash concept that we want to introduce into this chapter – **scenes**.

Scenes

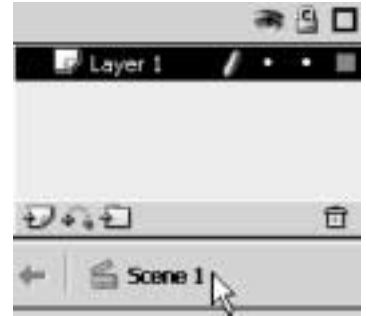
Scenes are used to organize your movie into sections that you can view as independent pieces of the whole movie. The ability to have a multi-scene movie allows you to break up your content into logical chunks and helps you organize things efficiently.

Flash movies can be large or small, simple or complex. Small, simple movies can usually be contained in a single scene with no problem, but when you're talking about a large movie with many components, or a movie that has long animations or multiple navigation elements, then multiple scenes can be the way to go.

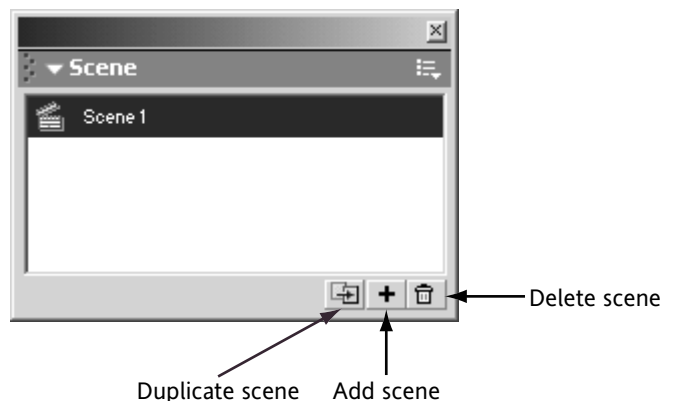
You can think of scenes as an extension of your timeline, giving you the facility to break up the action and continue from one scene to another. The benefit to you as an author is that your authoring files are more manageable, and the benefit to the user can be a more navigable movie. Many developers simply use separate scenes for their preloader (an element of a Flash movie that downloads movie components in advance of their being used), the introduction, and the main movie.

Some developers also use separate scenes in the same way as traditional designers use HTML pages on their sites – *one* scene is equivalent to *one* HTML page. This can be a useful model when you're designing a whole site in Flash – you can create a front page for your site, and then jump to different scenes (pages) depending on which buttons the user presses. We'll discuss this design approach in depth in a later chapter.

Each new Flash movie starts out with one default scene, named **Scene 1**. You can tell what scene you're currently in by looking at the scene name directly below your layers:



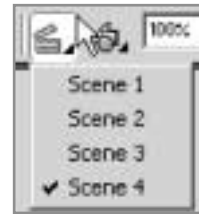
You can add and delete scenes as you wish, using the **Insert > Scene** and **Insert > Remove Scene** menu options. You can see the names of all of your scenes by opening up the **Scene** panel. You do this via the **Window > Scene** menu option or by pressing **SHIFT+F2**:



Here, you can use the buttons to manipulate scenes, and you can also drag scenes to change the order of playback. Usefully, you can also double-click on any scene in the Scene panel and give it a meaningful name. Once again, this is useful for bringing clarity to your movie authoring files.

By default, Flash will always play the scenes in the sequence in which the scenes are listed in the Scene panel, so make sure that you keep the right scenes in the right order here. However, you can also use Flash ActionScript to jump from scene to scene and play them in different sequences.

Another way to swap between scenes in the authoring environment is to click on the **Scene** button directly below the frames on the right, and click on the scene you want to edit in the drop-down list:



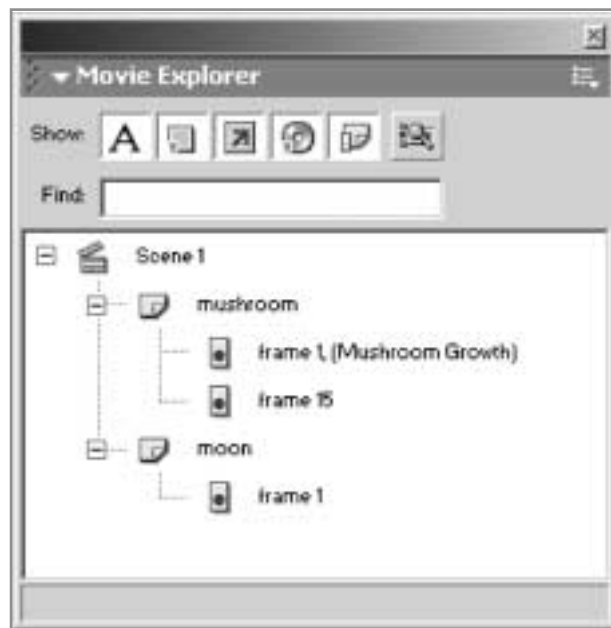
There's yet another way of navigating between your movie's different scenes in the authoring environment: the **Movie Explorer**.

The Movie Explorer

The Movie Explorer gives you the ability to browse your way through your whole movie at different levels of detail. You can open up the Movie Explorer by choosing the **Window > Movie Explorer** menu option or by pressing ALT+F3 (OPTION + F3 on a Mac).

The Movie Explorer window looks like this:

As you can see from this view from our mushroom movie, the Movie Explorer can reveal the contents of your movie to you in depth, in terms of its component scenes, layers, and keyframes (provided that you've chosen the relevant options from the **Show** buttons along the top). You can click on any node in this displayed hierarchy and be taken to that point in the movie. This is a really powerful way of helping you navigate through your movies – especially as your movies grow in size and scope. We'll be seeing more of the Movie Explorer as we progress through the book.



For now, though, let's just recap where we've been before we move on to the next chapter.

Summary

In this chapter, we've introduced you to the Flash authoring environment and demonstrated some of the essential elements of Flash movie creation.

We saw that:

- We create movie content on the **stage**, which has a surrounding **work area**.
- We can use the Property inspector to change the global characteristics of our movie, such as size and background color.
- The content we add to the stage is displayed in the viewer's browser when the movie plays back.
- Our movie is a series of points in time: these are played back in sequence as the **playhead** moves along the **timeline**.
- We use **keyframes** to hold new or changed content, and to indicate to Flash that something significant is happening. Keyframes are separated by 'normal' **frames** that influence how long the action between keyframes lasts.
- **Layers** add depth and manageability to your movies.
- **Scenes** allow us to separate our movies into distinct chunks.
- The **Movie Explorer** lets us navigate through our entire movie and browse through its content.

In the next chapter, we're going to look in more depth at the built-in Flash tools that let us create movie content. As we've already seen, these tools live in the Flash **Tools** panel.

