CHAPTER 12
LIST-O-MATIC

SITE MAP

- Famous Primates
- Thespians
  - King Kong
    - King Kong (2005)
    - King Kong (1976)
    - King Kong (1933)
  - Cheeta
    - Tarzan, The Ape
    - Tarzan Escapes (
  - Cornelius

As you might have guessed by the title, this chapter is all about lists. Why focus on lists? Simple. As designers have moved toward adopting web standards they've learned to embrace the humble list, turning lists from unstyled but semantically rich XHTML lists, styled with the browser's default style sheet, to well-styled lists that use CSS for design and presentation.

One area where XHTML and CSS lists come into their own is for the creation of web site navigation, and one of the primary focuses of this chapter will be on marking up and styling a list to create the Famous Primates web site's navigation.

Before we get started though, credit where credit's due. The title for this chapter is an homage to Max Design's excellent Listamatic:

`http://css.maxdesign.com.au/listamatic/`

Listamatic is an invaluable resource that's well worth a visit. Designed to showcase what's possible with CSS, Listamatic shows the power and flexibility of CSS when applied to a single, simple, well-structured list. By styling an identical XHTML list in a variety of ways, Listamatic demonstrates how CSS can be used to completely change the look and feel of lists, just by changing the CSS.

Once you've read this chapter, we strongly recommend you visit Listamatic to get a feel for what's possible when styling lists with CSS. You'll find a variety of inspirational approaches there that you might also like to experiment with.

In the last chapter we introduced a second column to our King Kong page's layout, creating a `sidebar div`. At that point our primary concern was to introduce basic layout principles. As a consequence the content of our `sidebar div` was a humble affair, a simple paragraph sitting under a lone h2; it served its purpose linking from our King Kong page to our Cheeta and Cornelius pages.

This was fine for the purposes of Chapter 11, but if you've been diligent and worked along with the homework, you'll be aware that our Famous Primates web site is a little more substantial than this . . .

We have a number of sections. Each section has its own "launch pad" page, providing the reader with some useful ape and monkey facts—our primates are thespians and pioneers after all. Each of these pages links to a number of additional primate-specific pages: homes for King Kong, Cheeta, Cornelius, Gordo, Miss Baker, and Albert. Finally, we have our Links and Contact Us pages.

In short, we have a great deal more content to link to. That content has an implicit structure and, as you'll see at the end of this chapter, we can use nested lists to give that content some additional semantic meaning, creating a well-styled site map for the Famous Primates web site.

In this chapter we'll replace the sidebar content we introduced in Chapter 11 with a list that we'll use to link to the different sections of the Famous Primates web site. This list will act as the navigation for our finished web site.

We'll use the background-image property, which we introduced in Chapter 10, to replace the browser's default bullets with our own custom star images, fitting for our ape thespians and monkey pioneers, creating a visually engaging home for our web site's navigation. Lastly, as we promised in Chapter 4, we'll take a look at styling ordered lists.

So, now that you know what we're covering, let's get started.

Styling lists

A long time ago in a galaxy far, far away . . . (in Chapter 4) we introduced you to the humble list. By now you're familiar with lists—both unordered and ordered—as we've included examples of both on the King Kong and Gordo web pages. This chapter is where we begin to style these lists using CSS.

As before, we'll use a step-by-step approach. We'll start by adding an unordered list to our sidebar, listing the main sections of the Famous Primates web site. In the first pass we'll style this list using CSS, showing you how to turn a humble but well-structured XHTML list into a well-designed and well-styled CSS list, add custom bullets to the list, and replace our browser's default bullets.

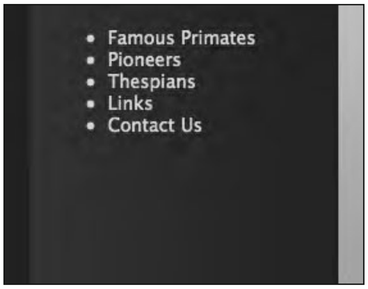
Once we've walked you through adding custom bullets, we'll add some links into the mix; these will form the backbone of our web site's navigation and will give us an opportunity to further style our list. Let's get started.

Styling a simple list

The first step in the process is to amend our sidebar content, replacing the content we added in Chapter 11 with an unordered list, listing the key sections of our Famous Primates web site. We replace the h2 and p elements in our sidebar with an unordered list as follows:

```
<div id="sidebar">
  <ul>
    <li>Famous Primates</li>
    <li>Pioneers</li>
    <li>Thespians</li>
    <li>Links</li>
    <li>Contact Us</li>
  </ul>
</div>
```

Without any additional CSS, this list renders in the browser as in Figure 12-1. This list will form the basis of our Famous Primates web site’s navigation.



- Famous Primates
- Pioneers
- Thespians
- Links
- Contact Us

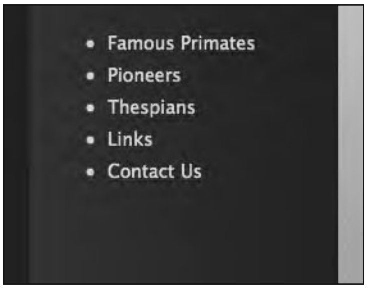
Figure 12-1.
Our unordered list as it appears in our `sidebar` with no additional CSS added

Although this list could benefit from some additional styling, we now have an unordered list in the `sidebar` that we can use as the basis for creating the navigation for our Famous Primates web site.

One thing you’ll notice is that, as it stands, our list items could do with some additional `line-height`. You might recall we added some `line-height` to the `p` elements of our King Kong page in Chapter 9. Although this took care of the `line-height` on our `p` elements, the browser’s default style sheet is applying a different `line-height` for our `ul` elements. We override this by adding the following rule to our style sheet, which targets all the unordered lists on our King Kong page:

```
ul
{
line-height: 1.8;
}
```

The result of adding this rule is shown in Figure 12-2.

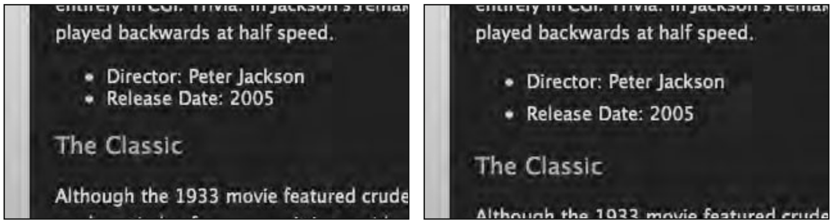


- Famous Primates
- Pioneers
- Thespians
- Links
- Contact Us

Figure 12-2.
Adding some `line-height` to all `ul` elements on the page inserts a little more vertical space between our unordered list items, giving them a little more breathing room.

A welcome by-product of this additional rule is that it takes care of *all* unordered lists on the page, including the two unordered lists beneath each of our King Kong films (listing the respective directors and release dates).

By simply adding this one rule, we’ve added some `line-height` to *all* the unordered lists on the King Kong page. You can see the result of this rule on one of these unordered lists in Figure 12-3.


played backwards at half speed.

- Director: Peter Jackson
- Release Date: 2005

The Classic

Although the 1933 movie featured crude

played backwards at half speed.

- Director: Peter Jackson
- Release Date: 2005

The Classic

Figure 12-3. On the left our ul elements as they stood at the end of Chapter 11, a little tight. On the right, as they now stand, a little more line-height added makes a big difference.

The benefit of applying the line-height to the ul element on our King Kong page is that it uses inheritance to take care of *all* of the unordered lists on the page. This allows us to take care of all of the unordered lists in one pass and later apply more specific rules to those lists we'd like to style differently.

In the last pass we specified a line-height for *all* the ul elements on our King Kong page. As we move forward, however, we'd like to target the changes we're making to the unordered list in our sidebar only. In order to do this, we need to identify this specific list, singling it out for the changes we're about to make.

In Chapter 10 we introduced you to id and class attributes and covered how we could use these to target specific elements of our web pages. We'll now add an id to our sidebar ul; this will allow us to write additional rules that target *just this list*.

We add an id to our markup as follows:

```
<ul id="nav">
  <li>Famous Primates</li>
  <li>Pioneers</li>
  <li>Thespians</li>
  <li>Links</li>
  <li>Contact Us</li>
</ul>
```

Now that we've identified this specific list, we can write rules targeted *at this list only*, enabling us to specify properties for this list alone.

12

Before we move on to adding our own custom bullets to our sidebar list, it's worth noting that we can also use CSS to style the bullets of our unordered lists using any of a variety of generic browser styles, including none, disc, circle, and square. We do this by simply adding a rule to our style sheet targeting the list-style property as follows:

```
#nav
{
list-style: circle;
}
```

Adding this rule replaces our default, unstyled list items' bullets (styled with the disc value) with the circle value, as you can see in Figure 12-4.

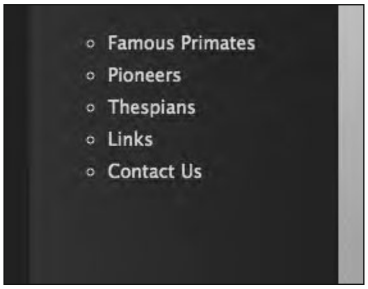


Figure 12-4.
By adding a rule targeting the ul with the id of nav, we can change the default list-style to circle instead of the browser's default disc.

This is a good start; however, we can go further than this by defining a background-image in CSS to specify custom bullets that relate to our site's theme a little more closely. In order to do this, we first need to switch off the browser's default bullets.

We do this by amending our previous declaration as follows:

```
#nav
{
list-style: none;
}
```

The result of changing this rule is shown in Figure 12-5.

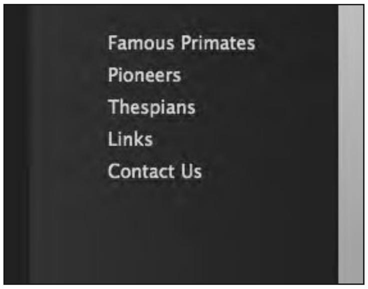


Figure 12-5.
By changing our declaration to list-style: none; we can switch off the bullets of the browser's default style sheet. This is the first stage in replacing these generic bullets with our own.

Although we've switched off the default bullets of the browser's default style sheet, the default indentation for list items remains. This is due to the fact that, by default, unordered lists have a certain amount of padding or margin (depending on which browser you're using) to accommodate the bullets. Before we apply our own bullet using a background-image, we'll switch off the default padding and margin by adding the following declaration:

```
#nav
{
list-style: none;
padding: 0;
margin: 0;
}
```

The results of removing the browser's default padding and margin can be seen in Figure 12-6.

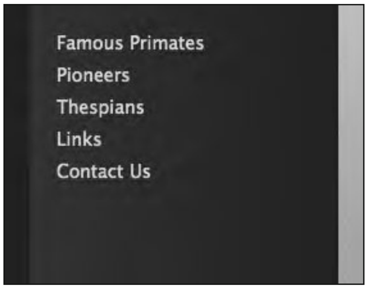


Figure 12-6.
Removing the browser's default padding and margin turns off our list items' indentation.

Now that we've removed the default bullets and removed the indentation, it's time to add our own custom bullets using a background-image specified in our style sheet. We could do this using the list-style-image property in CSS; however, this produces inconsistent results with our bullets' vertical positioning. A better approach is to use a background-image on each li element.

We add the following rule, which targets only the list items in the ul with the id of nav; note how we're using an id to differentiate the ul in our sidebar from the other ul elements on the page:

```
#nav li
{
background-image: url(../images/star.png);
background-repeat: no-repeat;
background-position: 5px center;
padding: 7px 0 7px 20px;
}
```

Adding this rule results in the changes shown in Figure 12-7.

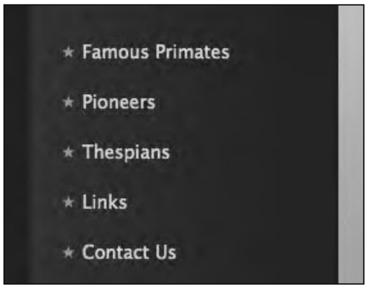


Figure 12-7.
By using the background-image property to add a new, custom bullet made from a star image, we can create a list that's more in keeping with our Famous Primates web site.

What we've done with the last rule first of all is to specify a background-image for all of our sidebar list items, targeted at the ul with the id of nav. We've set the background-image to be a PNG we've created called star.png, which is located in our images folder. We've then set the background-repeat property to no-repeat, instructing the browser not to tile the background-image. We covered both of these properties in Chapter 10 when we introduced background images; however, in this example we've introduced a new background-related property, background-position, which we'll introduce properly now.

The background-position property allows us to position background images in relation to the elements we apply them to. When no background-position is specified, the browser will default to background-position: 0 0;, which sets the background-image to the top left edge of the element (the first value specifies the position from the left, and the second value specifies the position from the top).

In this case we've used a background-position of 5px center to position our star.png custom bullet 5 pixels from the left of each li element and to center the image vertically in relation to each list item.

Lastly, we've set some padding on each list item to allow our 9 × 9 pixel star.png image to show through completely, giving it 7 pixels of space at the top and bottom and 20 pixels of space on the left-hand side. Without this padding on the left, our list item text would sit on top of the background-image. The amount of padding you add to the left-hand side of the li will depend on the size of your custom bullets.

The final stage in the process is to apply a little typographic style to the text of our list items, a small but important touch. We add the following declarations to our #nav li rule:

```
#nav li
{
...
text-transform: uppercase;
letter-spacing: 0.1em;
}
```

Figure 12-8 shows the results of adding these two declarations.

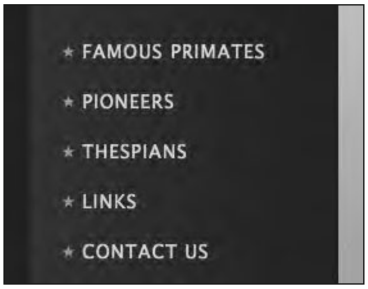


Figure 12-8.
To differentiate our unordered list in the sidebar from the other unordered lists of the page, we use the text-transform property to transform our sidebar list into uppercase and we add a little letter-spacing.

This transforms the list items in the sidebar into uppercase and adds a little letter-spacing, both properties covered in Chapter 9 when we covered styling text.

That's it. Although we've not added any links yet, the preceding walkthrough covers everything you need to know to transform your well-structured XHTML lists into well-styled lists using CSS. By experimenting with the bullets you apply using the background-image property, you can achieve a variety of striking effects while still creating well-structured lists that will display perfectly when CSS is switched off (as it might be in a nonbrowser context).

All we need to do now is to add some links to the unordered list in our `sidebar`, and we're on our way to creating some much-needed navigation for our Famous Primates web site.

Styling a navigation list

In the last section we used CSS's `background-image` property to create custom bullets for the list items in our `sidebar`, switching off the browser's default bullets in favor of an image that we defined using CSS. The one thing the last list was missing, however, was links.

In this section we'll add links to the unordered list in our `sidebar`. Before we do that, however, let's regroup and refresh where we are, as we'll be building on our existing XHTML markup and CSS as we move forward. Our `sidebar` list is currently styled with the following rules:

```
ul
{
line-height: 1.8;
}

#nav
{
list-style: none;
padding: 0;
margin: 0;
}

#nav li
{
background-image: url(../images/star.png);
background-repeat: no-repeat;
background-position: 5px center;
padding: 7px 0 7px 20px;
text-transform: uppercase;
letter-spacing: 0.1em;
}
```

We showed you the result of these rules in Figure 12-8; we're repeating the screenshot here in Figure 12-9 as this is the starting point from which we'll build as we move forward, adding links into the mix.

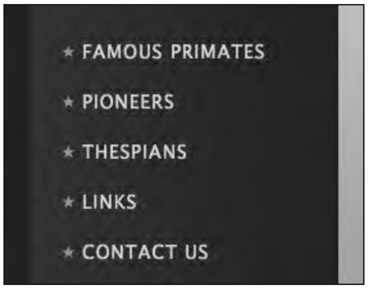


Figure 12-9.
Our unordered list as it stood at the end of the last section. This will form the foundation on which we build moving forward.

Before we add links to our unordered list, we'll put in place a little advance groundwork, adding some borders above and below our different list items; this will help to further differentiate the Famous Primates web site's different sections. We add the following declarations to the rules shown previously:

```
#nav
{
list-style: none;
padding: 0;
margin: 0;
border-top: 1px solid #39322D;
}

#nav li
{
background-image: url(../images/star.png);
background-repeat: no-repeat;
background-position: 5px center;
padding: 7px 0 7px 20px;
text-transform: uppercase;
letter-spacing: 0.1em;
border-bottom: 1px solid #39322D;
}
```

Adding these two `border` declarations results in the changes shown in Figure 12-10.

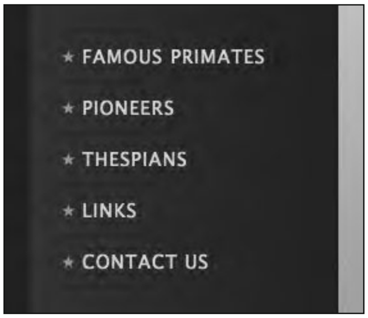


Figure 12-10.
By adding a border to the top of the unordered list in our `sidebar` and to the bottom of each of our list items, we can help to draw out the structure of the navigation list.

The two previous rules have added a border (1px solid #39322D;) to the top of our unordered list in the sidebar, targeted at this list's unique id of nav; it's also added a border with the same specifications to the bottom of every li element in this particular list. This helps to distinguish the different navigation sections of the Famous Primates web site.

We'll now complete the process of building the navigation list by adding links to our sidebar list. We add the following relative links to our sidebar's list items:

```
<ul id="nav">
  <li><a href="../index.html">Famous Primates</a></li>
  <li><a href="../pioneers/index.html">Pioneers</a></li>
  <li><a href="index.html">Thespians</a></li>
  <li><a href="../links.html">Links</a></li>
  <li><a href="../contact.html">Contact Us</a></li>
</ul>
```

Figure 12-11 shows the results of adding these links to the unordered list in our sidebar.

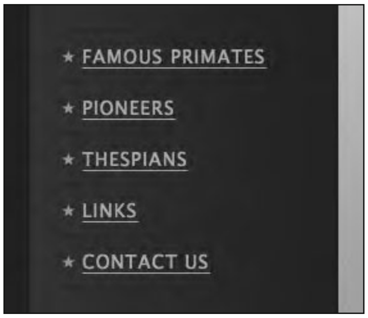


Figure 12-11.
Adding the links to the list in our sidebar gives us a navigation list for the Famous Primates web site.

Adding the links to our unordered list adds a border-bottom to the list of links (this is inherited from the rule we wrote in Chapter 9 targeting our a:link pseudo-class).

The next stage is a subtle one, but we'll reap the benefits of it in a moment. We change the selector targeting our sidebar list items (#nav li) to a selector targeting anchors located in our sidebar list items (#nav li a).

By using a descendant selector, we're targeting only a elements that are li elements in the ul with an id of nav. (You might want to read that again, but it makes perfect sense—essentially we're targeting just these links with laserlike precision.) We covered the use of descendant selectors to target specific elements based on the context in Chapter 10; essentially we're only targeting links that are list items in our sidebar list.

We do this by changing the selector for our existing rule as follows:

```
#nav li a
{
background-image: url(../images/star.png);
background-repeat: no-repeat;
background-position: 5px center;
padding: 7px 0 7px 20px;
text-transform: uppercase;
letter-spacing: 0.1em;
border-bottom: 1px solid #39322D;
}
```

Changing the selector results in what you see in Figure 12-12.

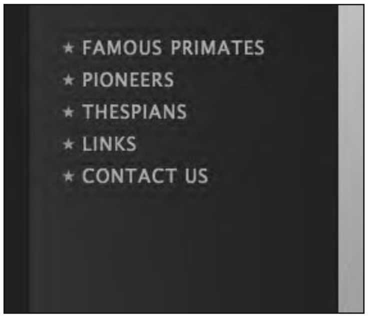


Figure 12-12.
In Figure 12-11 we used a selector of #nav li; we're now using a more precise selector of #nav li a.

The anchor elements that we're now targeting are inline-level and so occupy less vertical space than the block-level list items; however, we'll resolve this in a moment by setting our anchors to `display: block;`.

Although this change to our selector has resulted in the vertical space between the links being reduced, we'll resolve this now. Making this change allows us to more accurately target the links within our `sidebar` list, which will prove useful in a moment. We add the following to our rule:

```
#nav li a
{
background-image: url(../images/star.png);
background-repeat: no-repeat;
background-position: 5px center;
padding: 7px 0 7px 20px;
text-transform: uppercase;
letter-spacing: 0.1em;
border-bottom: 1px #39322D solid;
display: block;
}
```

Adding `display: block;`, coupled with a more precise selector targeting the anchors in the `sidebar` list, instructs the browser to display our `sidebar` links as block-level elements as shown in Figure 12-13.

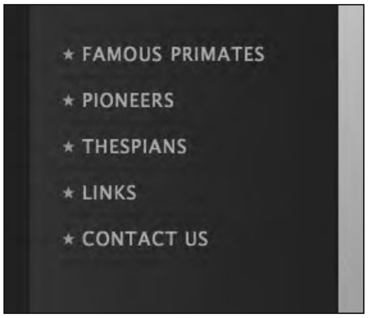


Figure 12-13.
Adding display: block; to the a elements in our sidebar list forces them to display as block-level elements, taking up the vertical space previously occupied by the block-level list items.

By setting our a elements in the sidebar to display: block;, they will display as block-level elements as opposed to inline-level elements; this forces them to occupy the entire vertical space that each block-level list item previously occupied, increasing the rollover area of our links. The result of this is shown in Figure 12-14.

Hovering over these block-level a elements now gives us a larger a:hover state, resulting in a more generous rollover area in our sidebar links.

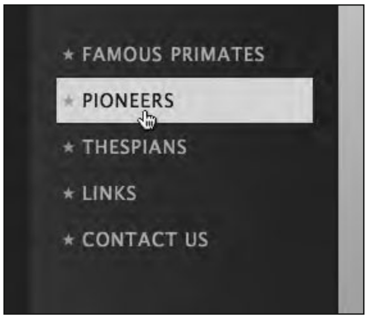


Figure 12-14.
Setting display: block; gives us a more generous rollover state.

With the changes we've made to this rule, when we hover over the links, the background-color is the same as that set on the links in our content div (#E0DFDA). We could leave this as is, but we'll add one other twist by creating an additional rule for the a:hover pseudo-class of our sidebar links.

By setting the background-color and color (the text color) on the a:hover pseudo-class of our sidebar links, we can create a more compelling hover effect. We add a new rule as follows:

```
#nav li a:hover
{
background-color: #39322D;
color: #88B4DB;
}
```

By simply changing the background-color on the a:hover state on our sidebar links, we can create the effect of our link's background-color fading into the background-color of the sidebar, with the text of the links on the a:hover state set to the pale blue we've been using for our links. The result of this change is shown in Figure 12-15.

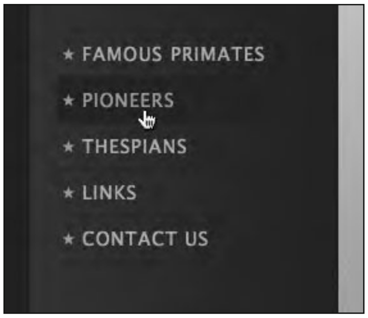


Figure 12-15.
The addition of this one extra rule gives our sidebar links an elegant fading a:hover effect.

With just a few additional rules we've created a well-styled, but equally well-structured, navigation list for our Famous Primates web site in the sidebar.

Creating horizontal lists

In the first half of this chapter, we enabled you to add custom bullets to your lists using CSS, allowing you to add to the design of your web pages. There are a number of other presentational aspects, however, that we can also control with CSS, not least taking lists—which are often imagined to be vertical—out of the normal flow of the document and, as a consequence, creating horizontal lists.

We can achieve a horizontal list in one of two ways: using either float: left—which you now know removes the list items from the normal document flow—or, alternatively, setting the list items, which are block-level elements, to display: inline, instructing the browser to display them as inline-level elements.

In this section, we'll explain how to use display: inline to change a simple unordered list of links into a "breadcrumb trail" of links, enabling the user to see at a glance where they are on the Famous Primates web site.

The first step is to create the list itself. We'll do this by adding some additional markup to our King Kong page, where each of our list items represents a level within the Famous Primates web site, allowing you to easily traverse back up through the levels of the Famous Primates web site using the links in the breadcrumb trail.

We add the following list to our King Kong web page (note the magic escalator we introduced in Chapter 6):

```
<ul id="crumbs">
  <li><a href="../index.html">Famous Primates</a></li>
  <li><a href="index.html">Apes in the Movies</a></li>
  <li>King Kong</li>
</ul>
```

We've given the ul an id of crumbs, which allows us to target this specific list with its own set of rules. Before we start adding the CSS, the list looks as shown in Figure 12-16.

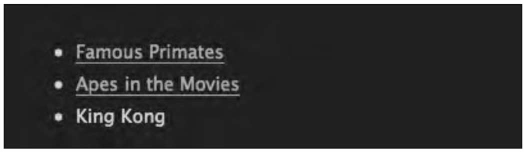

- Famous Primates
- Apes in the Movies
- King Kong

Figure 12-16. Our unordered list with links to the Famous Primates home page and the Apes in the Movies section page. As we are on the King Kong page, there's no need to link to it.

This gives us a solid foundation on which to build and some markup to apply our CSS to. Let's get started with the process of turning this into our horizontal breadcrumb trail. The first stage in the process is to remove the default bullets, margins, and padding using the following rule:

```
#crumbs
{
list-style: none;
margin: 0;
padding: 0;
}
```

The result of this change—which should come as no surprise—is shown in Figure 12-17.

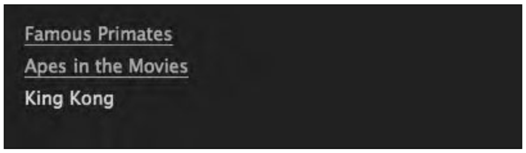

Famous Primates
Apes in the Movies
King Kong

Figure 12-17. Removing the default bullets, margins, and padding is the first step in the process of creating our horizontal list.

The real magic resides in the following rule. Short, but to the point, it instructs the browser to display the list items in our list as inline-level elements rather than block-level elements. We add a rule styling all list items in the ul with the id of crumbs; this styles just these particular list items.

```
#crumbs li
{
display: inline;
}
```

As a result of adding this short rule, the default block-level line breaks above and below these specific li elements are removed, and the list is now horizontal instead of vertical, as shown in Figure 12-18.

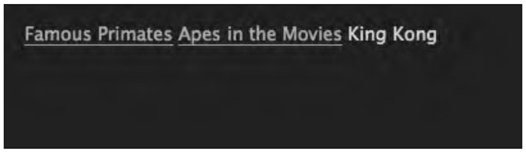

Famous Primates Apes in the Movies King Kong

Figure 12-18. Setting our list items to display: inline; creates a horizontal list.

We'll now begin to fine-tune our design a little, starting by removing the border-bottom we set on our links in Chapter 9. We do this by adding a descendant selector, targeting only anchors within list items in the unordered list with the id of crumbs:

```
#crumbs li a
{
border: 0;
}
```

Figure 12-19 shows the result of adding this rule.

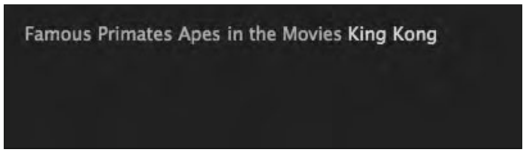

Famous Primates Apes in the Movies King Kong

Figure 12-19. We remove the border-bottom we set on links in Chapter 9 with a rule targeting just this list.

Our breadcrumb trail is taking shape, but would benefit from some horizontal space between the list items, which, at this point, are sitting tight against each other, separated only by a blank space. We'll once again leverage the power of margins and padding, combined with a carefully chosen background-image to insert some arrows into our horizontal list to give the breadcrumb trail some style.

We add the following declarations to our #crumbs li a rule:

```
#crumbs li a
{
border: 0;
padding-right: 20px;
margin-right: 5px;
background-image: url(../images/arrow.png);
background-repeat: no-repeat;
background-position: right center;
}
```

Adding these additional declarations results in the changes you see in Figure 12-20.

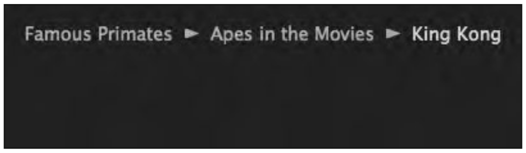


Figure 12-20. Using an arrow as a background-image and adding some margins and padding draws out the structure of the breadcrumb trail.

Let's have a look at what the declarations we've added are doing. First we've added 20px of padding to the right-hand side of our a elements; this makes room for the background-image, which we position to the right of the links and center vertically using background-position: right center;. We also set the background-image to display just once, using background-repeat: no-repeat;. We use margin-right; 5px; to create 5 pixels of space to the right of the arrow image; this inserts space between the arrow and the next list item.

By using a descendant selector to target *just the anchors in this list*, our arrows are applied only to the right of the links. Note that the words King Kong don't get the arrow, as they're not acting as an anchor element. Perfect. Figure 12-21 shows the list in action.

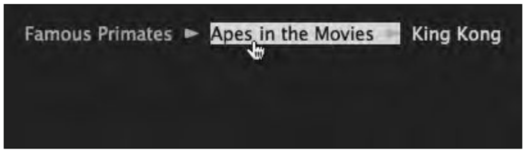


Figure 12-21. Our breadcrumb trail allows users to retrace their steps back up the levels of the Famous Primates web site, improving usability.

That's it—targeting these three simple rules using an id allows us to differentiate this list from the others on our King Kong page, creating a useful navigation breadcrumb trail that improves our web site's usability.

Lists are not only semantically the right choice when marking up lists of items, but are also extremely versatile when it comes to styling them with CSS. The ability to change a list's direction from vertical to horizontal makes the list a versatile and useful tool in any self-respecting Web Standardista's tool box.

Styling nested lists

Although we've focused on the King Kong and Gordo web pages in our walkthroughs and homework, you should by now have an awareness of the entire Famous Primates web site's structure. In this section we'll look at using nested lists to create a site map for the Famous Primates web site.

Nested lists are the perfect choice for a site map, indicating a site's logical structure and different sections' and pages' relationships. Admittedly our site—with an extensive eleven pages—is a little small to merit a full site map; however, the principles we discuss here can be scaled up and applied to larger sites with more pages, making this is a topic worth covering to prepare you for your ongoing journey as a Web Standardista.

Styling a site map with a nested list

In the following example we've added some additional, fictitious content to our Famous Primates web site to give us a little more to material to work with. Our King Kong and Cheeta entries now feature further nested lists, where we've added some imaginary subsections covering various films starring these two renowned ape thespians.

For the purposes of simplicity, we've restricted ourselves to just a handful of the films these two ape thespians starred in; however, you can see a more extensive version of this site map at the book's companion web site:

```
www.webstandardistas.com/12/site_map.html
```

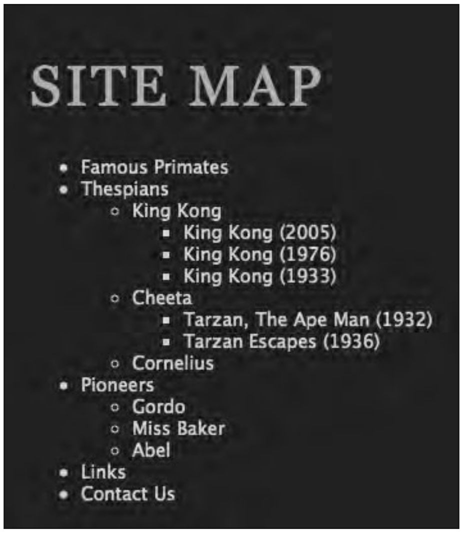
We're focusing on styling nested lists, particularly looking at using descendant selectors to target lists nested within other lists. With all of this nesting, things can get quite complicated, so to keep things simple, we're not adding links to our list items. With everything we've covered so far, however, you should be more than capable of combining our different examples to create nested navigation lists of some complexity.

To give us some material to work with, we create the following nested list (for a refresher on nested lists you might like to refer back to Chapter 4):

```
<ul>
  <li>Famous Primates</li>
  <li>Thespians
    <ul>
      <li>King Kong
        <ul>
          <li>King Kong (2005)</li>
          <li>King Kong (1976)</li>
          <li>King Kong (1933)</li>
        </ul>
      </li>
      <li>Cheeta
        <ul>
          <li>Tarzan, The Ape Man (1932)</li>
          <li>Tarzan Escapes (1936)</li>
        </ul>
      </li>
        <li>Cornelius</li>
    </ul>
  </li>
  <li>Pioneers
```

```
        <ul>
          <li>Gordo</li>
          <li>Miss Baker</li>
          <li>Abel</li>
        </ul>
      </li>
      <li>Links</li>
      <li>Contact Us</li>
    </ul>
```

We now have a reasonably long nested list that we can add some style to using CSS. Figure 12-22 shows how the preceding markup renders in a browser using our existing style sheet.



SITE MAP

- Famous Primates
- Thespians
  - King Kong
    - King Kong (2005)
    - King Kong (1976)
    - King Kong (1933)
  - Cheeta
    - Tarzan, The Ape Man (1932)
    - Tarzan Escapes (1936)
  - Cornelius
- Pioneers
  - Gordo
  - Miss Baker
  - Abel
- Links
- Contact Us

Figure 12-22. With our existing style sheet, our nested list is displayed with the browser's default bullets.

Even without any additional CSS our Famous Primates web site's structure is clearly indicated, with the hierarchy of pages obvious. As it stands, however, this is looking a little bland. We can use CSS to draw out the hierarchy further, highlighting the top-level pages in the site—Famous Primates, Thespians, Pioneers, Links, and Contact Us—differentiating them from the subsections and their related pages.

The first thing we need to do is give our unordered list an id so that we can target our CSS at this specific list without affecting any other lists on the page. We apply an id of sitemap to the containing ul as follows:

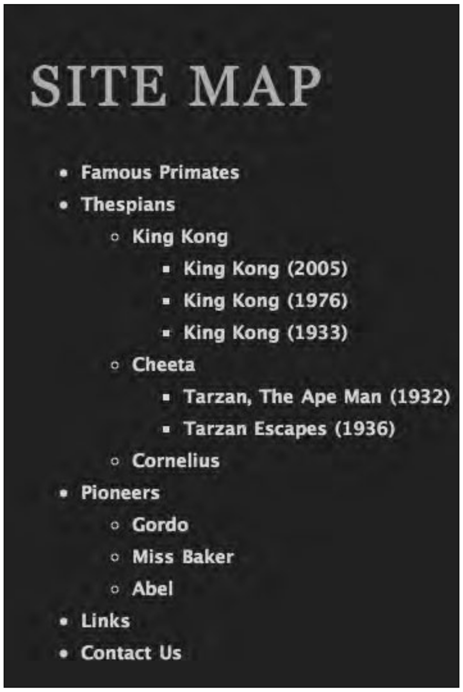
```
    <ul id="sitemap">
      <li>Famous Primates</li>
      ...
    </ul>
```

We can now write a rule targeting the `sitemap` `id`, applying some general style to the list. We'll then use shades of color to differentiate the different levels of our list, using a lighter shade for the top-level list and progressively darker shades for the nested lists.

We add the following rule to take care of some basic styling:

```
#sitemap
{
font-weight: bold;
color: #E0DFDA;
line-height: 1.8;
}
```

This displays in the browser as shown in Figure 12-23.



SITE MAP

- Famous Primates
- Thespians
  - King Kong
    - King Kong (2005)
    - King Kong (1976)
    - King Kong (1933)
  - Cheeta
    - Tarzan, The Ape Man (1932)
    - Tarzan Escapes (1936)
  - Cornelius
- Pioneers
  - Gordo
  - Miss Baker
  - Abel
- Links
- Contact Us

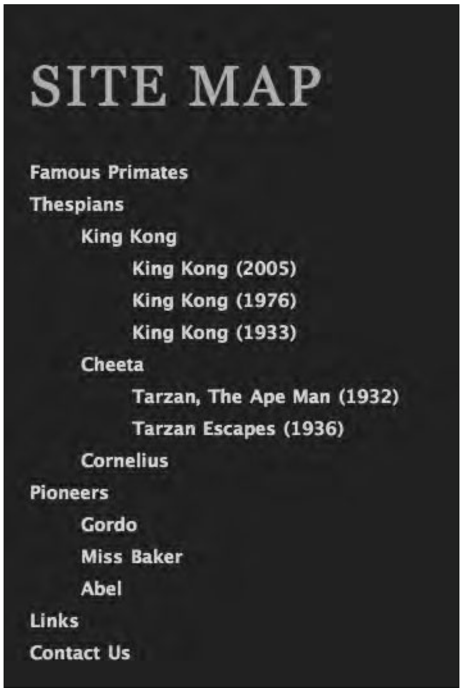
Figure 12-23. Adding some `line-height` to our nested list gives the list a little more breathing room. We've also set the list items to display in a bold weight using the `font-weight` property.

Setting the `font-weight` property to `bold` and adding a little `line-height` is the first stage in the process of styling our list. We're also setting a color for the list items to `#E0DFDA` (the light creamy text color we've been using). We'll overwrite this in a moment for darker shades on the nested list items.

We now amend our #sitemap rule as follows and add a second rule to remove all of the list items' bullets:

```
#sitemap
{
font-weight: bold;
color: #E0DFDA;
line-height: 1.8;
padding: 0;
margin: 0;
}

#sitemap li
{
list-style: none;
}
```

This displays in the browser as shown in Figure 12-24.



SITE MAP

Famous Primates
Thespians
    King Kong
        King Kong (2005)
        King Kong (1976)
        King Kong (1933)
    Cheeta
        Tarzan, The Ape Man (1932)
        Tarzan Escapes (1936)
    Cornelius
Pioneers
    Gordo
    Miss Baker
    Abel
Links
Contact Us

Figure 12-24. With the `list-style: none;` declaration, we switch off the browser's default bullets; we also remove the indent on our first level list.

One point to note here is that resetting the `margin` and `padding` on the top-level list (with the `id` of `sitemap`) only resets the margin on *this* list. Remember, there are other lists nested inside this list with their own default margin and padding. We're leaving the margin and padding on the nested lists untouched to reveal the hierarchy of the web site.

The second key point here is that our list-style: none; declaration is taking care of *all* list items in the nested list. Essentially, the selector #sitemap li instructs the browser to "look for any list items in the list with the id of sitemap and remove the bullets."

Now that we've removed the browser's default bullets, we'll add our own on the second- and third-level lists, adding stars in keeping with the design we've created. We'll target these at the nested list items.

We add the following additional rules:

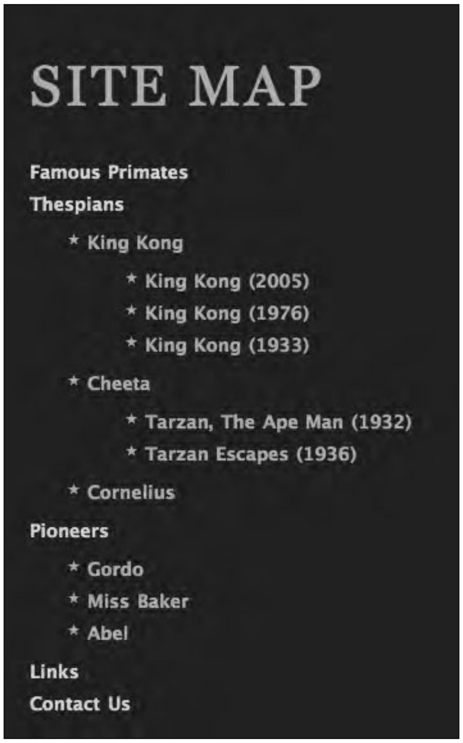
```
#sitemap ul
{
color: #C0BDB8;
margin: 5px 15px;
padding: 0 15px;
}

#sitemap ul li
{
background-image: url(images/star_first.png);
background-repeat: no-repeat;
background-position: left 6px;
padding-left: 15px;
}
```

This displays in the browser as shown in Figure 12-25.

What's important to note here is that we're using our knowledge of descendant selectors (targeted at specific elements) coupled with the use of inheritance to take care of the styling.

The first of the two rules we've added targets any unordered lists nested in the ul with the id of sitemap (that's all the nested lists at the second and third level) and sets their color to a slightly darker shade of brown (#C0BDB8) to differentiate them from the list items at the top level. We also add some additional margin and padding to give our site map a little more breathing room.

The second of the rules uses inheritance to apply a background-image to our nested list items. We've created a star image that's the same color as the color we set in the last rule (#C0BDB8). In the next, and final, stage we'll create a more specific rule that takes care of the third-level list items, differentiating them further.


SITE MAP

Famous Primates
Thespians
* King Kong
  * King Kong (2005)
  * King Kong (1976)
  * King Kong (1933)
* Cheeta
  * Tarzan, The Ape Man (1932)
  * Tarzan Escapes (1936)
* Cornelius

Pioneers
* Gordo
* Miss Baker
* Abel

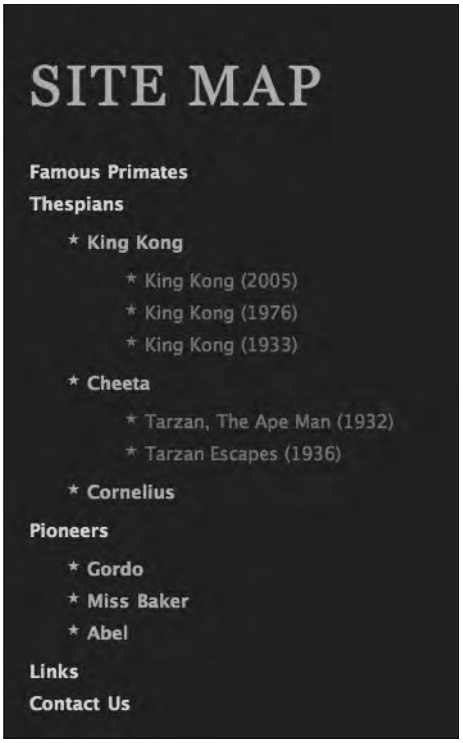
Links
Contact Us

Figure 12-25. Adding custom bullets to our list items using a variant of the star image we used for the sidebar navigation list ties this list into our design.

We do this by adding the following rule, which targets our third-level lists with laserlike precision:

```
#sitemap ul ul li
{
color: #8D8984;
background-image: url(images/star_second.png);
font-weight: normal;
}
```

This displays in the browser as shown in Figure 12-26.


SITE MAP

Famous Primates

Thespians

* King Kong
  * King Kong (2005)
  * King Kong (1976)
  * King Kong (1933)
* Cheeta
  * Tarzan, The Ape Man (1932)
  * Tarzan Escapes (1936)
* Cornelius

Pioneers

* Gordo
* Miss Baker
* Abel

Links

Contact Us

Figure 12-26. Changing the text `color` and setting the `font-weight` of our third-level list to `normal` differentiates it from the lists above it in the hierarchy.

In the final stage of the process we've differentiated our third-level lists from the lists above it in the hierarchy. Changing the `color` of the text and the associated color of our custom bullet in addition to using the `font-weight: normal;` declaration has helped to further underline the hierarchy of our site map.

As with the previous stages in the walkthrough, we're progressively targeting the elements we want to style: in this case all instances of `li` nested within a `ul`, nested within another `ul`, nested within the `ul` with the `id` of `sidebar`. That might seem a little complicated, and in this case it is, but it's completely logical!

Let's take a look at our finished CSS with comments added to indicate which level of the list the rules are targeting:

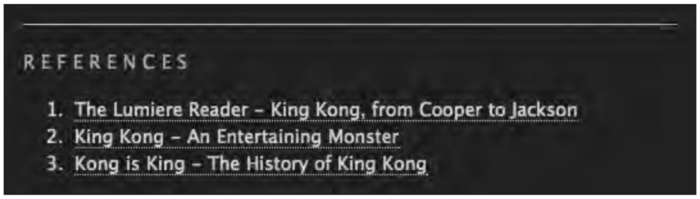
```
/* First-Level List */

#sitemap
{
font-weight: bold;
color: #E0DFDA;
line-height: 1.8;
padding: 0;
margin: 0;
}

#sitemap li
{
list-style: none;
}

/* Second-Level List */

#sitemap ul
{
color: #C0BDB8;
margin: 5px 15px;
padding: 0 15px;
}

#sitemap ul li
{
background-image: url(images/star_first.png);
background-repeat: no-repeat;
background-position: left 6px;
padding-left: 15px;
}

/* Third-Level List */

#sitemap ul ul li
{
color: #8D8984;
background-image: url(images/star_second.png);
font-weight: normal;
}
```

Styling nested lists can take a bit of getting used to and is certainly not a beginner-level topic. Dealing with inheritance in nested lists can be a little like a baptism of fire, but as ever, practice makes perfect. (Remember Mr. Miyagi? "Wax on . . . wax off.")

Styling an ordered list

In Chapter 4 we promised we'd introduce you to some of the alternative values we can use when styling ordered lists. Good news, this is the section where we cover this in action.

To give this some context, let's take a look at the references on our King Kong page where we've used an ordered list. At present this list is styled using the browser's default style sheet, which gives each list item a numeral (using a default value of `decimal` on the `list-style` property). This displays as shown in Figure 12-27.


REFERENCES

1. The Lumiere Reader – King Kong, from Cooper to Jackson
2. King Kong – An Entertaining Monster
3. Kong is King – The History of King Kong

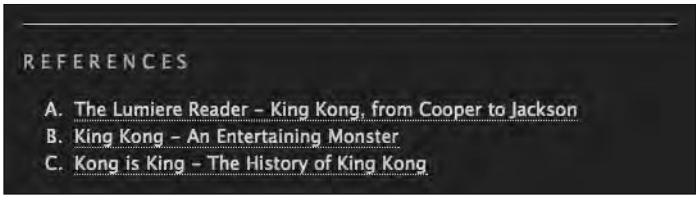
Figure 12-27. Our references as they are currently styled, with the browser's default style sheet specifying a `list-style` value of `decimal`

However, as we indicated in Chapter 4, we're not restricted to numbers for our list items; we can also use, among others, `upper-alpha` and `lower-alpha` (to change the numbers for letters), or `upper-roman` and `lower-roman` (to change the numbers for Roman numerals).

Let's take a look at these in action. If we'd like to change our list markers from sequential numbers to sequential letters (A, B, C . . .), we simply add the following rule to our style sheet:

```
ol
{
list-style: upper-alpha;
}
```

This results in the changes shown in Figure 12-28.


REFERENCES

A. The Lumiere Reader – King Kong, from Cooper to Jackson
B. King Kong – An Entertaining Monster
C. Kong is King – The History of King Kong

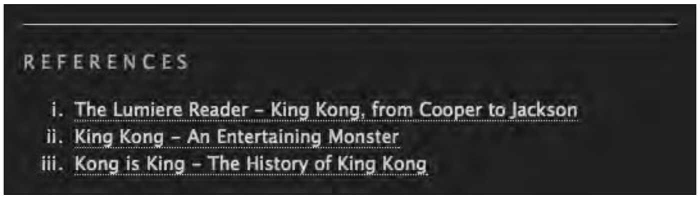
Figure 12-28. Our references are now ordered with sequential letters in uppercase.

Our `list-style` declaration is similar to the one we used previously to turn off the bullets on our unordered lists earlier in this chapter, in this case replacing our numbering with an ordered list of letters.

If we'd prefer a more Roman flavor, we can simply change the value of the `list-style` declaration to, for example, `lower-roman` as follows:

```
ol
{
list-style: lower-roman;
}
```

Figure 12-29 show the results of this change.


REFERENCES

i. The Lumiere Reader – King Kong, from Cooper to Jackson
ii. King Kong – An Entertaining Monster
iii. Kong is King – The History of King Kong

Figure 12-29. Our references are now numbered with lowercase Roman numerals.

Although this barely scratches the surface of styling ordered lists, it should get you started on your ordered list journey. There are a number of even more exotic ways to display ordered lists, including Greek, Armenian, and katakana list markers.

Noted CSS and web standards advocate Eric Meyer has a comprehensive test page at his web site that lists (pun intended) all available markers:

```
http://meyerweb.com/eric/css/tests/css2/sec12-06-02a.htm
```

Summary

So what have we covered? In this chapter we've looked at a number of the ways you can use CSS to style lists. We've looked at adding custom bullets to your lists by using the `background-image` property, and we've created both vertical and horizontal lists of links, creating block-level rollovers for the navigation we added to the `sidebar` of our Famous Primates web site.

We also looked at styling nested lists to enable you to create well-structured and well-styled site maps. Lastly, we briefly looked at some ways we can style ordered lists, introducing a variety of alternatives to plain old `decimal`.

An understanding of this chapter will enable you to create navigation lists (and other lists) for the web sites you build that are not only well-structured using semantic markup, but also well-styled with CSS. Combining the techniques we've covered will give you a flexible array of list styles to apply to your projects. As usual, experimenting with your own lists will help you develop your abilities further.

In the next chapter we'll introduce you to the benefits of using external style sheets, where a web standards approach really comes into its own. We'll also introduce you to a number of useful troubleshooting tools and techniques to serve you as you continue on your journey toward Web Standardistas' nirvana.

Homework: Adding the Famous Primates web site's navigation

Over the last five chapters we've progressively added to your knowledge of CSS to enable you to create well-styled web pages that are built on a solid foundation of well-structured markup. This is the backbone of the Web Standardistas' approach and one we'll conclude in the following chapter when we create an external style sheet for your Famous Primates web site, tying all of the web site's separate pages together.

In this chapter we showed you how to use CSS to create well-styled vertical and horizontal lists; we also showed you how to add custom bullets to these lists to improve their look and feel. We replaced the placeholder content we added to our `sidebar div` in Chapter 11 with a list of links to form the Famous Primates web site's navigation.

This chapter's homework is to create the navigation list for the `sidebar` of your Gordo page, once again following along with our King Kong page as an example. Once you've added your navigation list to your Gordo page, we'd like you to implement the navigation across the remainder of your Famous Primates web pages.

Lastly, we'd like you to explore the different ways you can style the ordered list of links in your Gordo page's references section by creating a rule to change the default `decimal` style with something a little more interesting.

1. Add the ul to the sidebar

In the last chapter we added some placeholder content to the `sidebar div` when we created our two-column CSS layout. In this chapter we'd like you to replace this content with an unordered list. This list will form the basis of your Famous Primates web site's navigation.

Following along with our King Kong example, give your `sidebar ul` an `id` of `nav`. This will allow you to create rules targeting this specific list on the page while leaving the other lists on your Gordo page untouched.

2. Add the custom bullets to your sidebar list

As with the previous chapters we've created everything you need to complete the homework. You can download the assets here:

```
www.webstandardistas.com/12/assets.zip
```

Once you've downloaded the preceding files, transfer the images to your `images` folder. You'll be using the star images we've supplied for you to replace the bullets of the browser's default style sheet with images more in keeping with Gordo, your space-traveling friend.

Following along with our King Kong example earlier, remove the bullets of the browser's default style sheet; reset your margins and padding; and specify the stars provided as `background-images` on each of the list items in your `sidebar ul`.

3. Add the links to the nav ul in the sidebar

Taking care to ensure your relative links are correct, add links to each of the list items in the navigation list in the sidebar div.

Following along with the second part of our King Kong navigation list walkthrough, use a descendant selector to target only the a elements that are li elements in the ul with the id of nav; that is, you'll be creating a rule for the following:

```
#nav li a
{
...
}
```

This is a moving into slightly more complicated territory, but if you've followed along, you should be ready. Before you embark on this aspect of the homework, you might want to re-read the second part of our navigation list walkthrough along with the introduction to descendant selectors in Chapter 10, to see how you're using descendant selectors to target these specific a elements with laserlike accuracy.

4. Style the ordered list in the references section

The final part of this chapter's homework is to style the ordered list in your references section. We set the list-style on our ol to upper-alpha and lower-roman—why not try something different? You might like to refer to Eric Meyer's comprehensive list of all available list-style markers here:

```
http://meyerweb.com/eric/css/tests/css2/sec12-06-02a.htm
```

As usual, to help you with the different stages of this chapter's homework, we've created our own, similarly styled, page about King Kong featuring our newly added sidebar navigation list. We've also styled the references section, changing the default list-style to decimal-leading-zero. You can refer to this, using your browser's View Source menu command to see how we've updated our CSS, here:

```
www.webstandardistas.com/12/king_kong.html
```

Once you've added the navigation list to the sidebar of your Gordo web page and styled the references, put the kettle on and enjoy a cup of *Keemun Orchid* as you prepare yourself for the next chapter.