**Backgrounds and Links**

In this chapter, you will continue to learn how to use CSS to control presentation of XHTML web pages, starting with CSS properties that enable you to control presentation of links, backgrounds, list styles, table styles, and outlines around boxes. You will then learn a technique to add content to a page (even if it was not in the XHTML document) using the ::before and ::after pseudo elements. Finally, you will see how CSS can be used to position boxes on the page, which will allow you to create attractive layouts for your pages.

By the end of the chapter, you will know how to use CSS to control the following:

* Presentation of links
* Backgrounds of document
* Styles of bullet points and numbered lists
* Appearance of tables
* Outlines around boxes
* Elements that can gain focus or are active
* Addition of content to the XHTML document before or after an element
* The three positioning schemes that allow you to determine where on a page a box will appear — something that prepares you to use CSS to create layouts

*Occasionally in this chapter, you will come across a feature that is not yet implemented by the*

*main browsers, but it is worth learning these now as they are likely to be used as standard in*

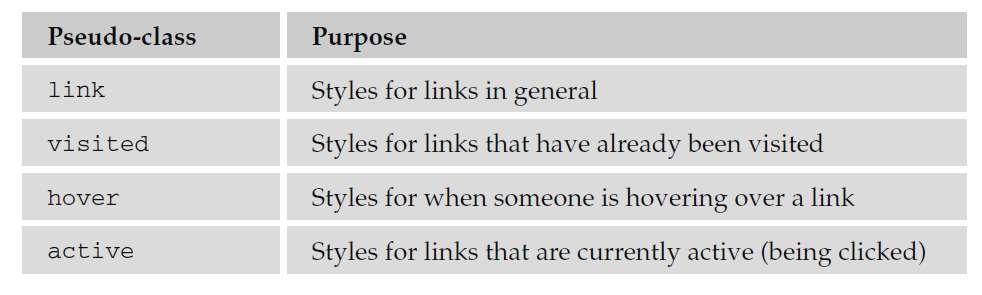
*the near future.* ***(Now it has been implemented)***

**Links:**

Most browsers show links in blue with an underline and change the color of links you have already visited, unless you tell them to do otherwise. The following are properties often used with links:

* color: Changes the colors of the links
* background - color: Highlights the link, as if it had been highlighted with a highlighter pen
* text - decoration: Commonly used to control whether the link is underlined or not, although it can also specify that text should have a strikethrough, blink, or be overlined

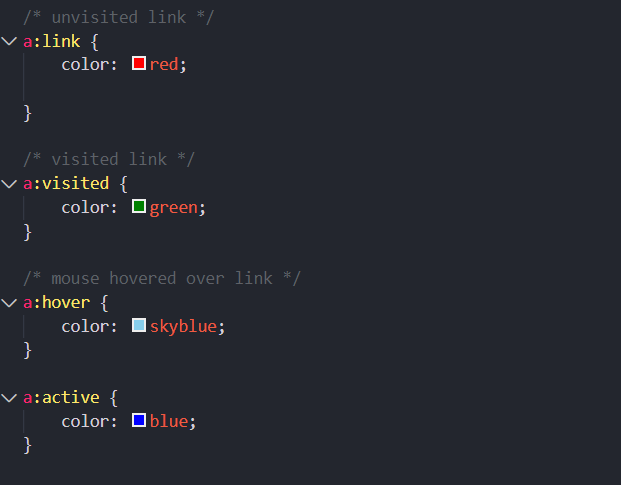
While you can just create rules that apply to the <a> element to set properties such as color and text-decoration, there are also four *pseudo-classes* that can give greater control over presentation of links.



Using these pseudo-classes allows you to change properties of links when the user hovers over them (making them a slightly different color, maybe adding a highlight and underlining them), and also the properties of links that have been visited (for example, making them a slightly different color — which helps users know where they have been).

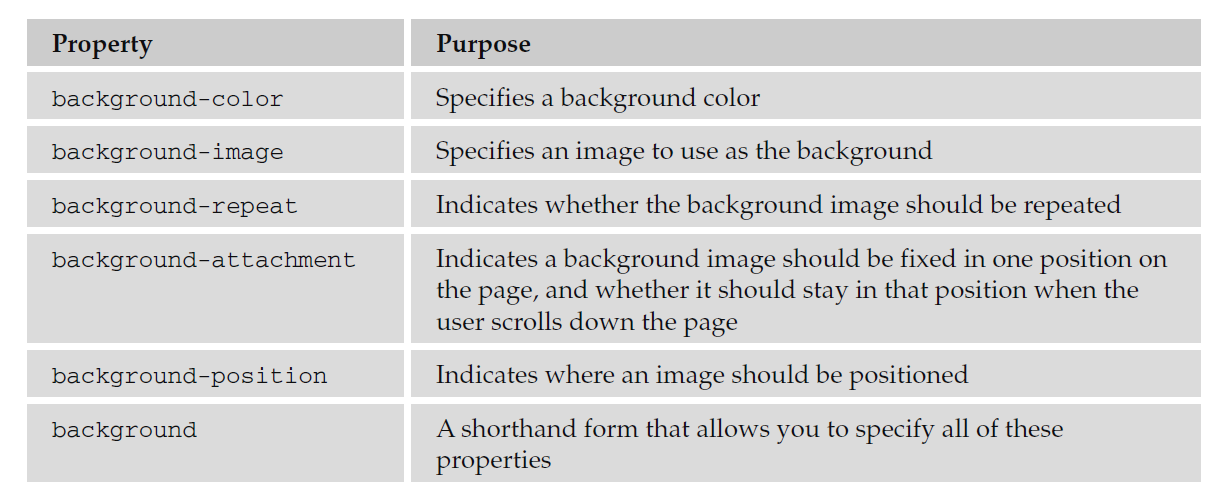
When used, these properties should be specified in the order listed in the table above. Here is an example that will change the styles of links as users interact with them:





**Backgrounds:**

As you saw in the last chapter, CSS treats each element as if it were its own box. You can control the background of these boxes using the following properties (when used on the <body> element they affect the entire browser window).



***The background-color Property:***

The background-color property allows you to specify a single solid color for the background of

any element.

When the background - color property is set for the < body > element, it affects the whole document, and when it is used on any other element it will use the specified color inside the border of the box created for that element.

The value of this property can be a color name, a hex code, or an RGB value:

body {background-color: #cccccc; color: #000000; }

b {background-color: #FF0000; color: #FFFFFF}

p {background-color: rgb(255,255,255); }

***The background-image Property:***

As its name suggests, the background - image property enables you to add an image to the background of any box in CSS. This can be very useful in many situations, from adding a subtle texture or shading to adding a distinctive design to the back of elements or entire pages.

The value for this property should start with the letters url , followed by the URL for the image in brackets and quotes like so:

Example:

body {

background-image: url(“images/background.gif”);

}

If both a background - image property and the background-color property is used, then the

background - image property takes precedence. It is good practice to supply a background – color property with a background image and give it a value similar to the main color in the background image because the page will use this color while the background image is loading or if it cannot display the image for any reason.

There are a few points to note about how background images work:

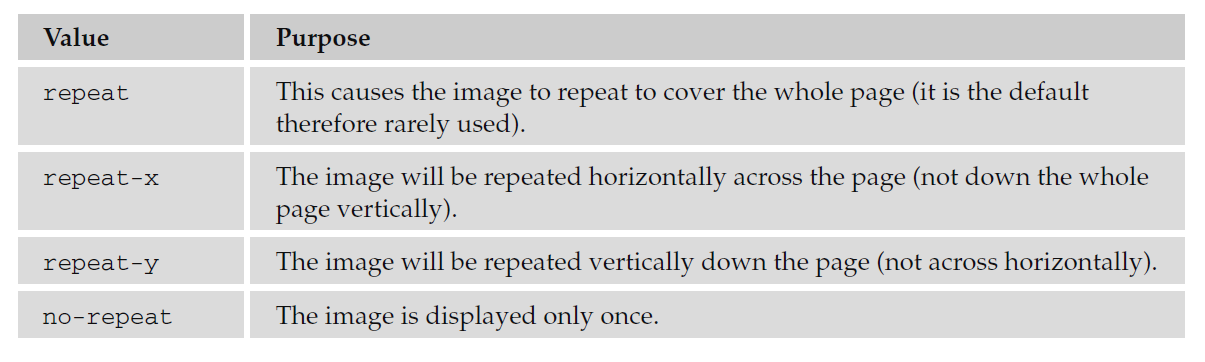
* There is no way to express the intended width and height of a background image, so you need to save it at the size you want it to appear.
* There is no equivalent to the alt attribute (alternate text for those not able to see the image for any reason); therefore, a background image should not be used to convey any important information that is not described on the page in text as well.
* Background images are often shown on the page after other items have been rendered, so it can look as if they take a long time to load.
* The background-image property works well with most block - level elements, although some older browsers can have problems showing background images in tables.

***The background-repeat Property:***

When you specify a background-image, and the box is bigger than the image, then the image is repeated to fill up the whole box, creating what is affectionately known as *wallpaper.*

If you do not want your image to repeat all over the background of the box, you should use the

background-repeat property, which has four helpful values, as you can see in the table that follows:



Example:

body {

background-image: url(“images/background\_small.gif”);

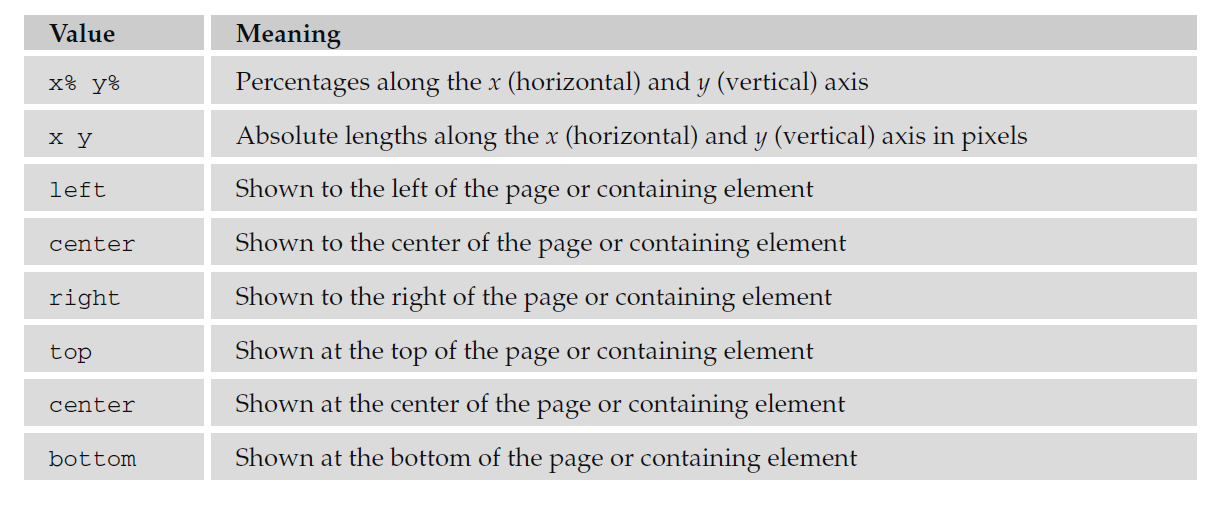
background-repeat: repeat-y;

background-color: #eaeaea;

}

***The background-position Property (for Fixing Position of Backgrounds):***

You may want to alter the position of this image, and you can do this using the background-position property, which takes the values shown in the table that follows:



Here is an example of fixing the position of the image:

body {

background-image: url(“images/background\_small.gif”);

background-position: 50% 20%;

background-repeat: no-repeat;

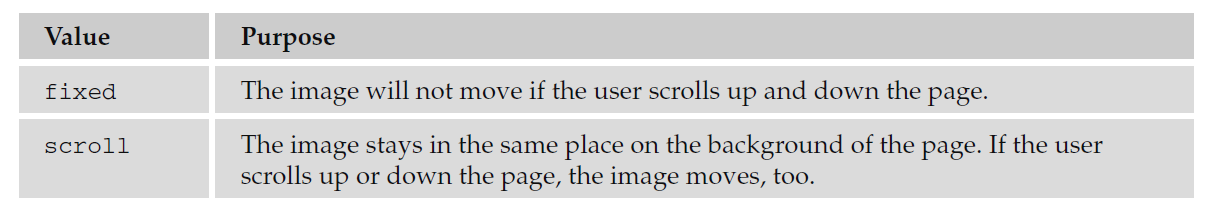
background-color: #eaeaea;

}

This image will be horizontally centered (because it should be 50 percent of the screen’s width from the left - hand side of the page) and a fifth of the way down from the top of the screen (because it is positioned 20 percent of the window height from the top of the screen). It is worth trying this example in the code download and changing the size of the browser window to see how the background image will remain in the center of the browser window horizontally and a fifth of the way down the window vertically when you change the size of the window.

***The background-attachment Property (for Watermarks):***

When you specify a background image you can use the background-attachment property to specify whether the image is fixed in its position, or whether it moves as the user scrolls up and down the page.



Here is an example where the image will stay in the middle of the page even when the user scrolls further down:

body {

background-image: url(“images/background\_small.gif”);

background-attachment: fixed;

background-position: center;

background-repeat: no-repeat;

background-color: #eaeaea;

}

***The background Property (the Shorthand):***

The background property allows you to specify several of the background properties at once. The values can be given in any order, and if you do not supply one of the values, the default value will be used.

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

For example, you can just write:

body {

background: #cc66ff; url(“images/background\_small.gif”) fixed no-repeat center;

}