

Relative and Absolute Positioning

You've seen how normal flow causes the browser to render the elements in the order that they appear in the HTML source code. When using CSS for page layout there are situations when you may want more control over the position of an element. This section introduces relative and absolute positioning.

Relative Positioning

Use **relative positioning** to change the location of an element slightly, relative to where it would otherwise appear in normal flow. Configure relative positioning with the `position: relative;` property along with one or more of the following offset properties: `left`, `right`, `top`, `bottom`. Table 8.1 lists CSS position and offset properties.

Property	Value	Purpose
position	static	Default value; the element is rendered in normal flow
	relative	Configures the location of an element relative to where it would otherwise render in normal flow
	absolute	Configures the location of an element outside of normal flow precisely in its container element
	fixed	Configures the location of an element within the browser viewport; the element does not move when the page is scrolled
left	Numeric value or percentage	The position of the element offset from the left side of the container element
right	Numeric value or percentage	The position of the element offset from the right side of the container element
top	Numeric value or percentage	The position of the element offset from the top of the container element
bottom	Numeric value or percentage	The position of the element offset from the bottom of the container element

TABLE 8.1 CSS Properties for Relative and Absolute Positioning

Figure 8.3 shows a web page (see the student files `chapter8/relative.html`) that uses relative positioning along with the `left` property to configure the placement of an element in relation to the normal flow. In this case, the container element is the body of the web page.

The result is that the content of the element is rendered as being offset or shifted by 30 pixels from the left where it would normally be placed at the browser's left margin. Notice also how the padding and background-color properties configure the heading element.

The CSS is

```
#myContent { position: relative;
             left: 30px;
             font-family: Arial, sans-serif; }
h1 { background-color: #cccccc;
      padding: 5px;
      color: #000000; }
```

The HTML source code follows:

```
<h1>Relative Positioning</h1>
<div id="myContent">
<p>This paragraph uses CSS relative
positioning to be placed 30 pixels
in from the left side.</p>
</div>
```



FIGURE 8.3 The paragraph is configured using relative positioning.

Absolute Positioning

Use **absolute positioning** to specify the location of an element outside of normal flow precisely in its container element. Configure absolute positioning with the `position: absolute;` property along with one or more of the following offset properties: `left`, `right`, `top`, `bottom`. Table 8.1 lists CSS position and offset properties.

Figure 8.4 depicts a web page that configures a `div` element using absolute positioning to display the content 200 pixels in from the left margin and 100 pixels down from the top of its container element, which is the body of the document. Padding and background-color are used to configure the heading element. An example is in the student files, chapter8/absolute.html).

The CSS is

```
#content { position: absolute;
           left: 200px;
           top: 100px;
           font-family: Arial, sans-serif;
           width: 300px; }
h1 { background-color: #cccccc;
      padding: 5px;
      color: #000000; }
```

The HTML source code is

```
<h1>Absolute Positioning</h1>
<div id="content">
<p>This paragraph is contained in a div configured to be 300 pixels
wide and uses CSS absolute positioning to be placed 200 pixels in from
the left and 100 pixels down from the top of the browser window.</p>
</div>
```

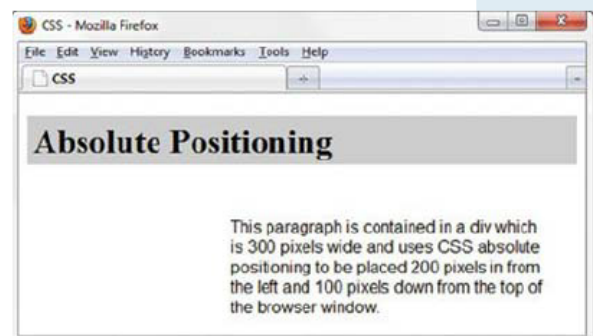


FIGURE 8.4 The `div` element is configured using absolute positioning.