

# 10.4. The background-position Property

The background-position property, as its name implies, allows you to control the placement of the background. The following table shows the possible values for the background-position property.

Property	Value
background-position	[<percentage>   <length> ]{1,2}   [ [top   center   bottom]    [left   center   right] ] Initial value: 0% 0%

At first glance, this property looks a little complicated; in truth, it isn't all that complex. The notation boils down to this: The property allows one or two values that express the position of the background. Square brackets are used to group the possible values. The following is the first subgrouping of values within the first grouping:

[<percentage> | <length> ]{1,2}

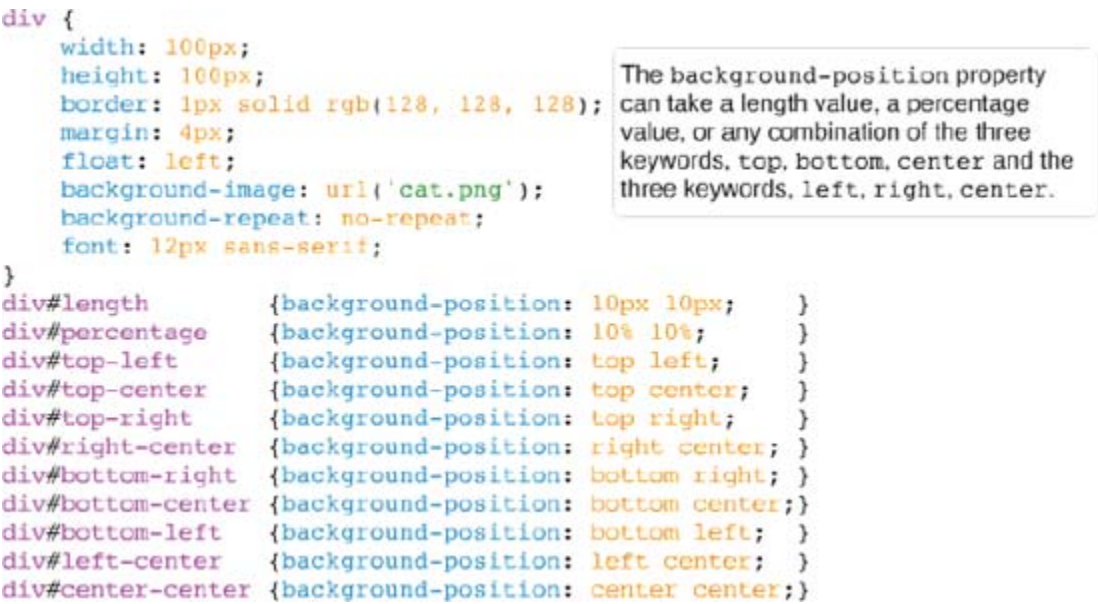
The first grouping indicates that the value may be a percentage or length value. Either one or two values may be provided. The second subgrouping is preceded by a vertical bar, which indicates another possibility for the value:

[ [top | center | bottom] || [left | center | right] ]

The second grouping indicates that either one or two keyword values may be provided. If two values are provided, it may be any keyword from the first grouping combined with any of the keywords from the second grouping. In addition, any of the keyword values can be mixed with either a <length> or <percentage> value.

[Figure 10-7](#) demonstrates some possible values for the background-position property.

Figure 10-7a. Figure 10-7a



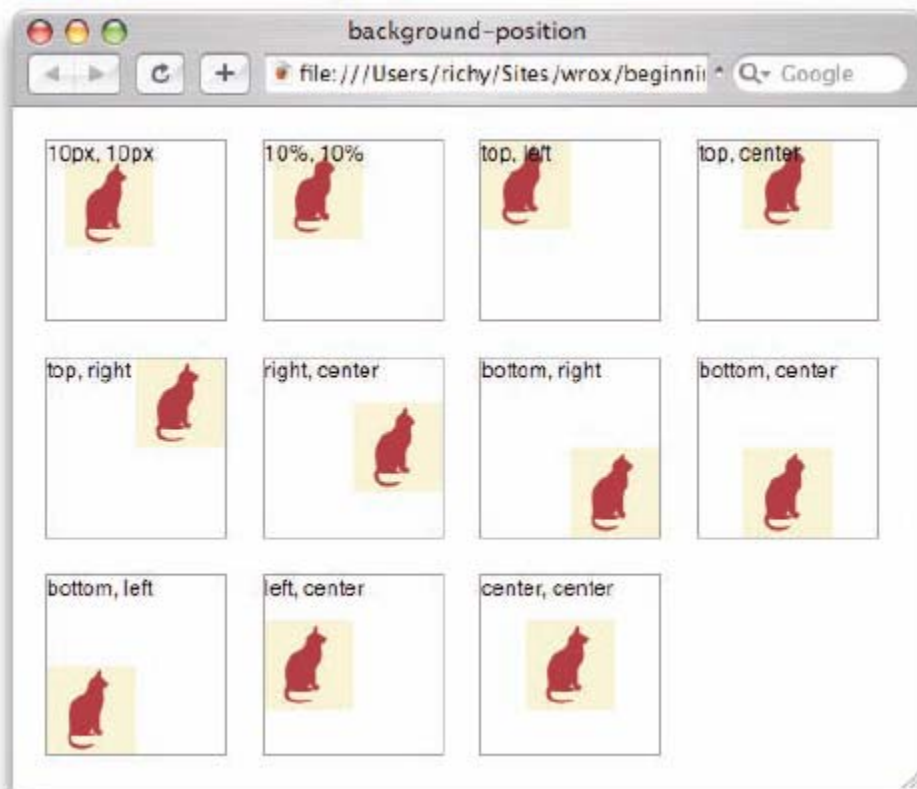
The CSS in [Figure 10-7a](#) is combined with the markup in [Figure 10-7b](#).

Figure 10-7b. Figure 10-7b

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en'>
  <head>
    <title>background-position</title>
    <link rel='stylesheet' type='text/css' href='096977320fg1007.css' />
  </head>
  <body>
    <div id='length'>10px, 10px</div>
    <div id='percentage'>10%, 10%</div>
    <div id='top-left'>top, left</div>
    <div id='top-center'>top, center</div>
    <div id='top-right'>top, right</div>
    <div id='right-center'>right, center</div>
    <div id='bottom-right'>bottom, right</div>
    <div id='bottom-center'>bottom, center</div>
    <div id='bottom-left'>bottom, left</div>
    <div id='left-center'>left, center</div>
    <div id='center-center'>center, center</div>
  </body>
</html>
```

The CSS in [Figure 10-7a](#) and the markup in [Figure 10-7b](#) result in the output you see in [Figure 10-7c](#).

Figure 10-7c. Figure 10-7c



In [Figure 10-7](#), you see what the background-position property with two values looks like. This figure shows what happens when both values are of the same ilk, that is to say: both length values, or both percentage values, or both keyword values.

### 10.4.1. Mixing Different Kinds of Position Values

What happens when you mix length with percentage, or percentage with a keyword? This question is answered by the example in [Figure 10-8](#).

Figure 10-8a. Figure 10-8a

```
div {
  width: 100px;
  height: 100px;
  border: 1px solid rgb(128, 128, 128);
  margin: 4px;
  float: left;
  background-image: url('tree.png');
  background-repeat: no-repeat;
  font: 12px sans-serif;
}
div#keyword-length {background-position: top 10px; }
div#length-keyword {background-position: 10px top; }
div#keyword-percentage {background-position: center 50%;}
div#percentage-keyword {background-position: 50% center;}
div#percentage-length {background-position: 50% 10px; }
div#length-percentage {background-position: 10px 50%; }
```

You can mix and match any type of measurement.

#### NOTE

Per the CSS 2.1 specification, when keywords are mixed with nonkeyword values, the first value must be left or right, if left or right is used, and the second value must be top or bottom, if top or bottom is used. Some technically invalid declarations appear in [Figure 10-8a](#) for proof-of-concept.

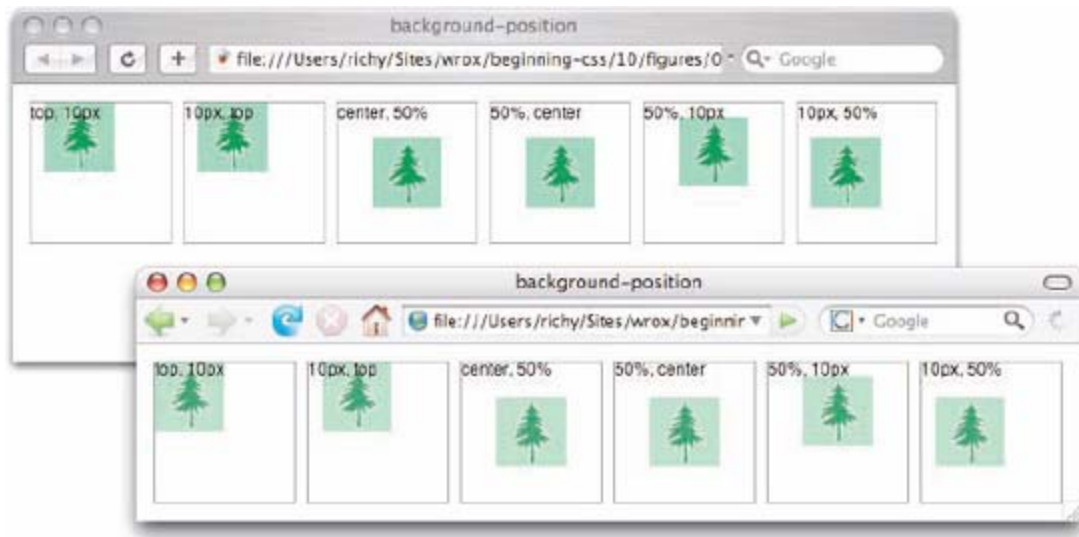
The CSS in [Figure 10-8a](#) is combined with the markup in [Figure 10-8b](#).

Figure 10-8b. Figure 10-8b

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>background-position</title>
    <link rel="stylesheet" type="text/css" href="096977%20fg1008.css" />
  </head>
  <body>
    <div id='keyword-length'>top, 10px</div>
    <div id='length-keyword'>10px, top</div>
    <div id='keyword-percentage'>center, 50%</div>
    <div id='percentage-keyword'>50%, center</div>
    <div id='percentage-length'>50%, 10px</div>
    <div id='length-percentage'>10px, 50%</div>
  </body>
</html>
```

The CSS and markup in [Figure 10-8a](#) and [Figure 10-8b](#) are combined to get the rendered output you see in [Figure 10-8c](#).

Figure 10-8c. Figure 10-8c



In [Figure 10-8](#), you see the combination of each different type of value for the background-position property. You'll note the difference in the rendering of the first box between Safari and Firefox. Firefox rejects the background position of the first box entirely because the top keyword appears first, rather than second as required by the CSS 2.1 specification. Safari tolerates the ordering being different.

#### 10.4.2. Tiling and Position

What happens when the background is tiled and a position is set? You see an example of positioning a tiled background with a length measurement in [Figure 10-9](#).

Figure 10-9a. Figure 10-9a

```

div {
  width: 100px;
  height: 100px;
  border: 1px solid rgb(128, 128, 128);
  margin: 4px;
  float: left;
  background-image: url('star.png');
}
div#repeat {
  background-position: 10px 10px;
}
div#repeat-single {
  background-position: 10px;
}
div#repeat-x-control {
  background-repeat: repeat-x;
}
div#repeat-x {
  background-position: 10px 10px;
  background-repeat: repeat-x;
}
div#repeat-x-single {
  background-position: 10px;
  background-repeat: repeat-x;
}
div#repeat-y-control {
  background-repeat: repeat-y;
}
div#repeat-y {
  background-position: 10px 10px;
  background-repeat: repeat-y;
}
div#repeat-y-single {
  background-position: 10px;
  background-repeat: repeat-y;
}
br {
  clear: both;
}

```

When both axes are tiled, the position specified adjusts when tiling of the image begins.

When just one axis is tiled, the position specified adjusts the offset of the tiled axis. The top position of the x-axis, for example, adjusts where the x-axis is drawn, but the left position adjusts where the tiling of the image begins.

The CSS in [Figure 10-9a](#) is combined with the markup in [Figure 10-9b](#).

Figure 10-9b. Figure 10-9b



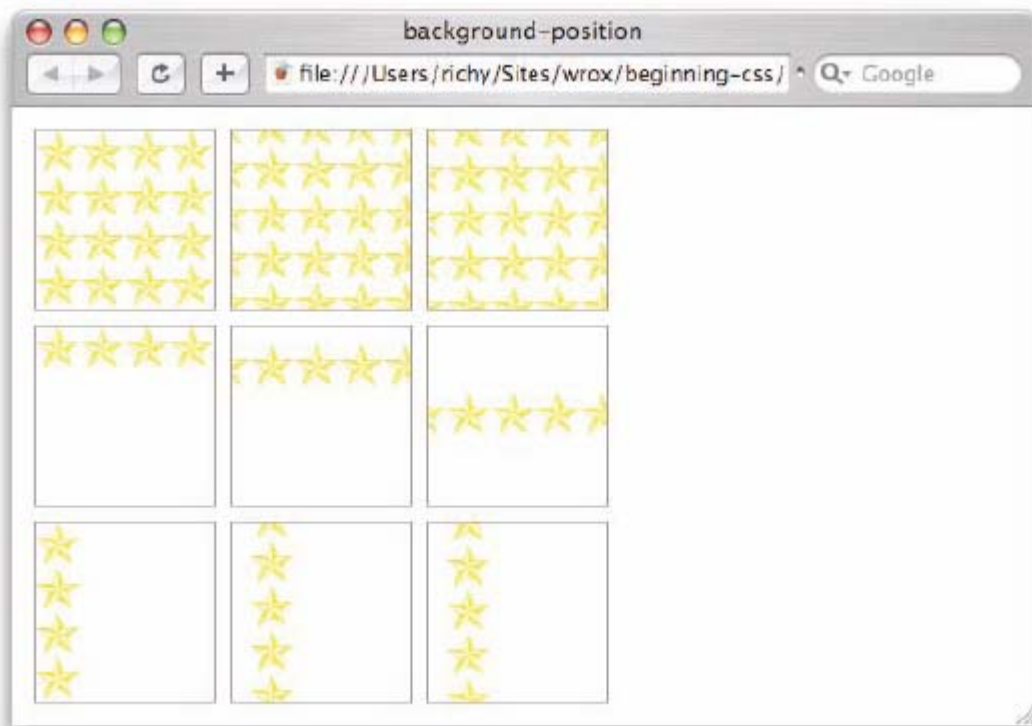
```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en'>
  <head>
    <title>background-position</title>
    <link rel='stylesheet' type='text/css' href='096977%20fgl009.css' />
  </head>
  <body>
    <div id='repeat-control'></div>
    <div id='repeat'></div>
    <div id='repeat-single'></div>
    <br />
    <div id='repeat-x-control'></div>
    <div id='repeat-x'></div>
    <div id='repeat-x-single'></div>
    <br />
    <div id='repeat-y-control'></div>
    <div id='repeat-y'></div>
    <div id='repeat-y-single'></div>
  </body>
</html>

```

The CSS in [Figure 10-9a](#) and the markup in [Figure 10-9b](#) produce the output you see in [Figure 10-9c](#).

**Figure 10-9c. Figure 10-9c**



In [Figure 10-9](#), you see how specifying a background position affects the tiling of a background image. When both axes are tiled, the position that you specify determines where the image tiling begins.

Specifying `background-position: 10px 10px;` causes the tiling to begin with the first ten pixels of the image clipped. When the same declaration is applied to an element with `background-repeat: repeat-x;`, you can see that the tiling of the image also begins with the first ten pixels of the image clipped for the value of the left position. The value of the top position causes the axis of tiled images to be offset ten pixels from the top

border.

Just for the sake of completeness, what happens when keywords are used instead of lengths to position a tiled image? The answer is found in [Figure 10-10](#).

Figure 10-10a. Figure 10-10a

```
div {  
  width: 100px;  
  height: 100px;  
  border: 1px solid rgb(223, 223, 200);  
  margin: 4px;  
  float: left;  
  background-image: url('star.png');  
}  
div#repeat {  
  background-position: center center;  
}  
div#repeat-single {  
  background-position: center;  
}  
div#repeat-x-control {  
  background-repeat: repeat-x;  
}  
div#repeat-x {  
  background-position: center center;  
  background-repeat: repeat-x;  
}  
div#repeat-x-single {  
  background-position: center;  
  background-repeat: repeat-x;  
}  
div#repeat-y-control {  
  background-repeat: repeat-y;  
}  
div#repeat-y {  
  background-position: center center;  
  background-repeat: repeat-y;  
}  
div#repeat-y-single {  
  background-position: center;  
  background-repeat: repeat-y;  
}  
br {  
  clear: both;  
}
```

When both axes are tiled, the position specified adjusts when tiling of the image begins.

When just one axis is tiled, the position specified adjusts the offset of the tiled axis. The top position of the x-axis, for example, adjusts where the x-axis is drawn, but the left position adjusts where the tiling of the image begins.

The CSS in [Figure 10-10a](#) is combined with the markup in [Figure 10-10b](#).

Figure 10-10b. Figure 10-10b

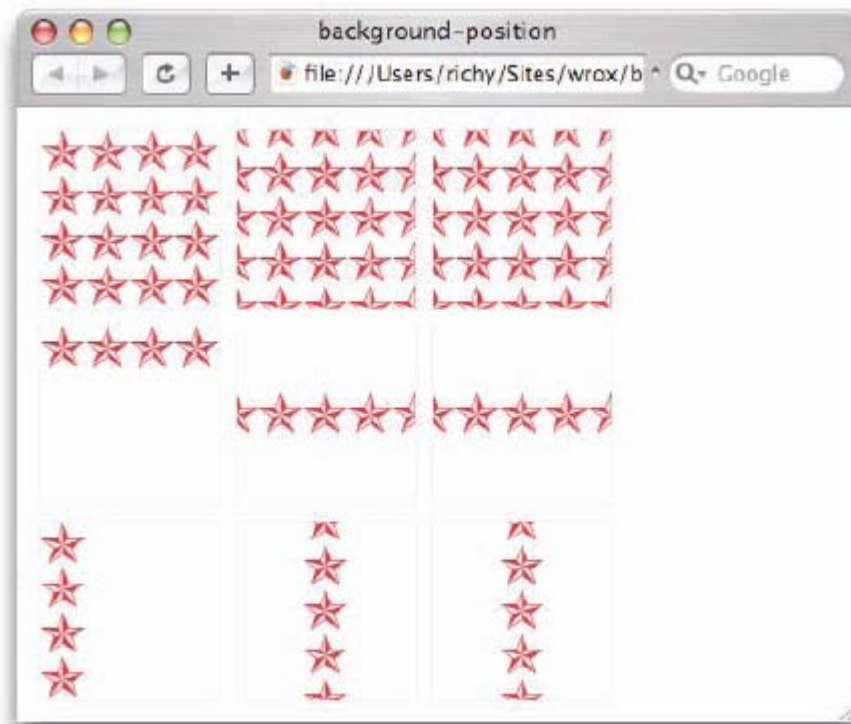
```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en'>
  <head>
    <title>background-position</title>
    <link rel='stylesheet' type='text/css' href='096977%20fg1010.css' />
  </head>
  <body>
    <div id='repeat-control'></div>
    <div id='repeat'></div>
    <div id='repeat-single'></div>
    <br />
    <div id='repeat-x-control'></div>
    <div id='repeat-x'></div>
    <div id='repeat-x-single'></div>
    <br />
    <div id='repeat-y-control'></div>
    <div id='repeat-y'></div>
    <div id='repeat-y-single'></div>
  </body>
</html>

```

The result of the CSS in [Figure 10-10a](#) and the markup in [Figure 10-10b](#) is seen in [Figure 10-10c](#).

Figure 10-10c. Figure 10-10c



In [Figure 10-10](#) you used the center keyword instead of a length measurement. When the tiling is along the x-axis, one center keyword centers the tiled images along the y-axis, and the other center keyword causes the tiling of each image to begin with the center of the image, rather than the left border of the image. This result is the same in every browser.

---

## Try It Out: Controlling the Background's Position



---

*Example 10-4.* The following steps recap how you can use the background-position property in a web page.

1. Enter the following HTML document into your text editor:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en'>
  <head>
    <title>background-position</title>
    <link rel='stylesheet' type='text/css' href='Example_10-4.css' />
  </head>
  <body>
    <p>
      The background-position property allows you to specify a
      position using one of three primary methods, by length,
      by percentage, or by keyword.
    </p>
    <div id='length'>
    </div>
    <div id='percentage'>
    </div>
    <div id='keyword'>
    </div>
    <p>
      You can mix and match different types of positions.
    </p>
    <div id='length-percentage'>
    </div>
    <div id='percentage-keyword'>
    </div>
    <div id='length-keyword'>
    </div>
    <p>
      When positioning a tiled image, the position can adjust where
      tiling of the image begins with respect to the image itself, or
      the position of the axis of tiled images.
    </p>
    <div id='tiled'>
    </div>
    <div id='x-tiled'>
    </div>
    <div id='y-tiled'>
    </div>
  </body>
</html>
```

2. Save the HTML document as `Example_10-4.html` .
3. Enter the following CSS in your text editor:

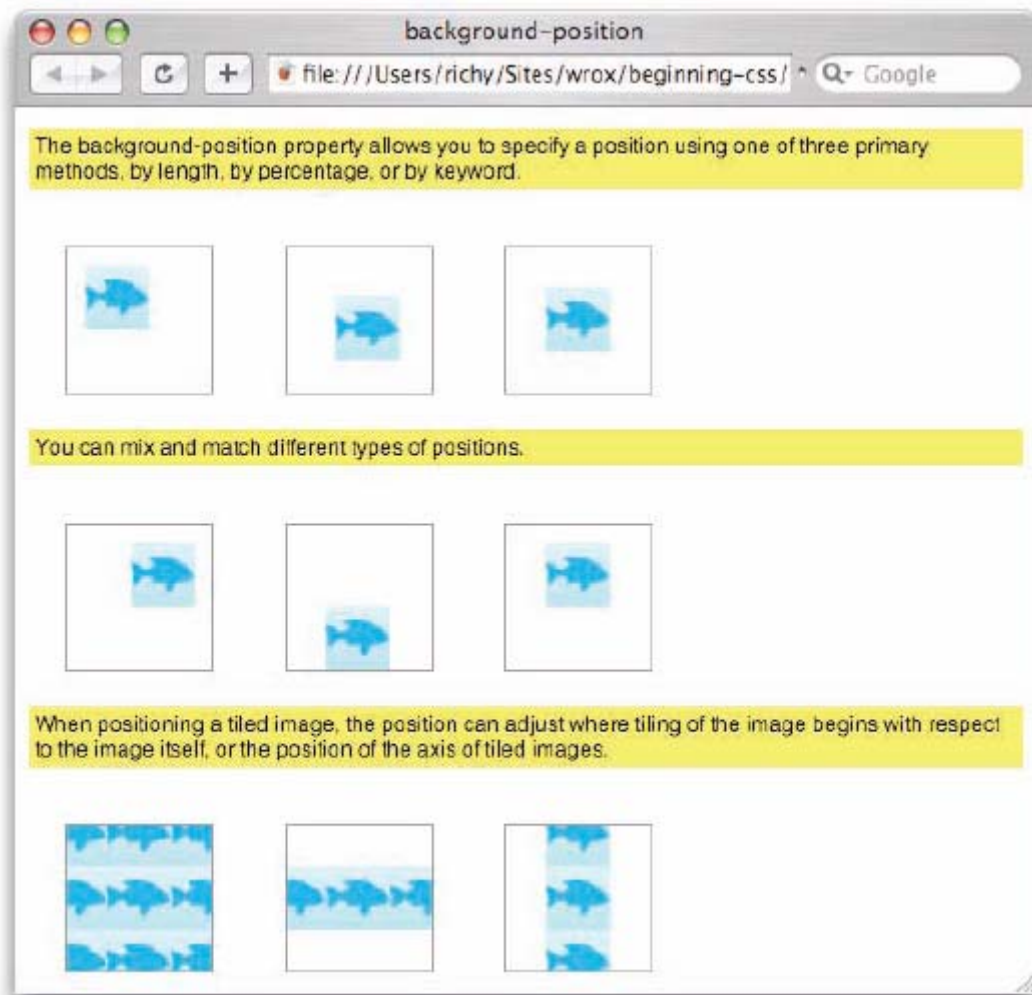
```
body {
```

---

```
    font: 12px sans-serif;
}-
p {
    background: yellow;
    padding: 3px;
    clear: left;
}
div {
    height: 81px;
    width: 81px;
    margin: 20px;
    background-image: url('fish.png');
    background-repeat: no-repeat;
    float: left;
    border: 1px solid rgb(128, 128, 128);
}
div#length {
    background-position: 10px 10px;
}
div#percentage {
    background-position: 60% 60%;
}
div#keyword {
    background-position: center center;
}
div#length-percentage {
    background-position: 80% 10px;
}
div#percentage-keyword {
    background-position: center 100%;
}
div#length-keyword {
    background-position: center 10px;
}
div#tiled {
    background-repeat: repeat;
    background-position: center center;
}
div#x-tiled {
    background-repeat: repeat-x;
    background-position: center center;
}
div#y-tiled {
    background-repeat: repeat-y;
    background-position: center center;
}
```

4. Save the CSS document as `Example_10-4.css`. The source code in [Example 10-4](#) renders something like what you see in [Figure 10-11](#).

Figure 10-11. Figure 10-11



### How It Works

In [Example 10-4](#), you recapped the different ways that a background image can be positioned with the background-position property. You can choose one of three different methods of positioning, keyword, length, or percentage, and any one of those will get the job done. You can also mix different methods, such as percentage with a keyword, or length with a keyword, and the browser can handle that. You can also use the background-position property to adjust the position of a tiled background image, be it the axis of tiled images, or where tiling of the image begins.

In the next section, I describe how to control the background-position when the page is scrolled with the background-attachment property.