Margins

Margins are an amount of space that may be added around the outside of the element's border. There are properties for specifying a margin amount for one side at a time or by using the shorthand `margin` property.

margin-top, margin-right, margin-bottom, margin-left

Values:	`<length> | <percentage> | auto | inherit`
Initial value:	`0`
Applies to:	All elements (except elements with table display types other than `table` and `inline-table`)
Inherited:	No

margin

Values:	`[<length> | <percentage> | auto]{1,4} | inherit`
Initial value:	See individual properties
Applies to:	All elements (except elements with table display types other than `table` and `inline-table`)
Inherited:	No

With the `margin-top`, `margin-right`, `margin-bottom`, and `margin-left` properties, you can specify a margin for one side of an element. Margin size may be specified in any of the accepted units of length. Negative values are permitted. Figure 19-4 shows examples of adding margins to individual sides of an element. Note that the dotted lines are a device to point out the outer edge of the margin and would not display in the browser.

```
h1 { margin-top: 3px; }
h1 { margin-right: 20px; }
h1 { margin-bottom: 3px; }
h1 { margin-left: 20px; }
```

Percentage values are also permitted, but be aware that percentages are calculated based on the *width* of the parent element. If the parent element gets narrower (perhaps as the result of the browser window resizing) the margins on all sides of the child elements will be recalculated.

Margins may also be set using the keyword `auto`, which allows the user agent to fill in the amount of margin necessary to fit or fill the containing block.

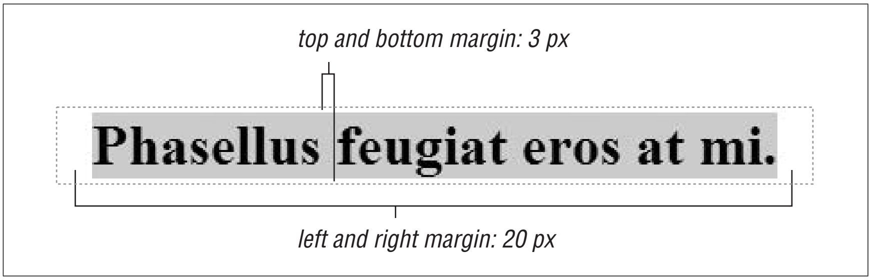
top and bottom margin: 3 px
Phasellus feugiat eros at mi.
left and right margin: 20 px

Figure 19-4. Individual margin settings

> The proper way to horizontally center an element in CSS is to set the left and right margins to `auto`. This technique (as well as the workaround required for Internet Explorer in anything but Standards mode) is addressed in Chapter 24.

The Shorthand margin Property

As an alternative to setting margins one side at a time, there is the shorthand `margin` property. The accepted values are the same as those previously listed. What changes slightly is the syntax as the `margin` property provides a lot of flexibility for specifying values.

In the values listed for `margin` above, the `{1,4}` notation means that you can provide one, two, three, or four values for a single `margin` property. Here's how it works.

When you provide four values, the values are applied around the edges of the element in clockwise order, like this (some people use the mnemonic device "TRouBLe" for the order Top, Right, Bottom, Left):

```
{ margin: top right bottom left }
```

The four margin properties listed in Figure 19-4 could be condensed using the margin property as so:

```
{ margin: 3px 20px 2px 20px; }
```

When one or more of the four values is missing, certain provided values are replicated for the missing values.

If three values are provided, it is assumed the value for the left margin is missing, so the value for right is used for left (`{margin: top right/left bottom}`). This rule, therefore, is equivalent to the previous example:

```
{ margin: 3px 20px 3px; }
```

If two values are provided, the right value is replicated for the missing left value, and the top value is replicated for the missing bottom value (`{margin: top/bottom right/left}`). Again, the same effect achieved by the previous two examples could be accomplished with this rule:

```
{ margin: 3px 20px; }
```

Finally, if only one value is provided, it is replicated for all four values. This declaration applies 20 pixels of space on all sides of an element:

```
{ margin: 20px; }
```

Margin Behavior

It is helpful to be aware of these general margin behaviors.

- Margins are always transparent, allowing the background color or image of the parent element to show through.
- Elements may have negative margins, which may cause elements to break out of containing blocks of their parent elements or overlap other elements on the page.
- The vertical (top and bottom) margins of adjacent block elements in the normal document flow will *collapse*. That means that the space held between adjacent block elements will be the larger of the two margin values, rather than the sum of their margin values. The collapsing margins in the following examples are demonstrated in Figure 19-5.

```
h2#top {margin: 10px 20px 10px 20px;}

h2#bottom {margin: 20px 20px 20px 20px; }

    <h2 id="top" >Lorem ipsum dolor sit amet,</h2>
    <h2 id="bottom" >consectetuer adipiscing elit.</h2>
```

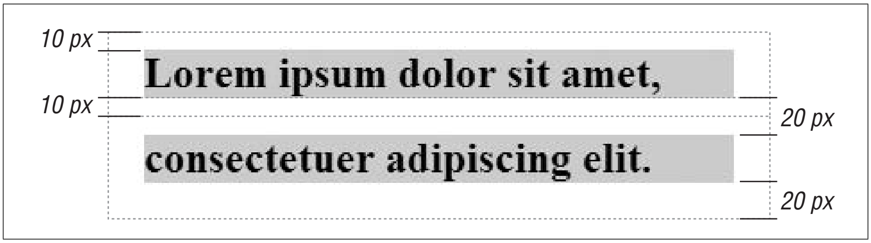

Figure 19-5. Collapsing margins

- The vertical margins do not collapse for floated elements, absolutely positioned elements, and inline block elements.
- In CSS 2.1, horizontal (left and right) margins never collapse.
- Top and bottom margins applied to non-replaced inline elements (text elements such as `em` or `strong`) have no effect on the height of the line. In other words, top and bottom margins are not calculated as part of the element's inline box or the height of the line box for that line.
- Left and right margins applied to non-replaced inline (text) elements *do* cause the specified amount of space to be held before and after (to the left of the first character and right of the last character) the inline element, even if it is broken over two lines.