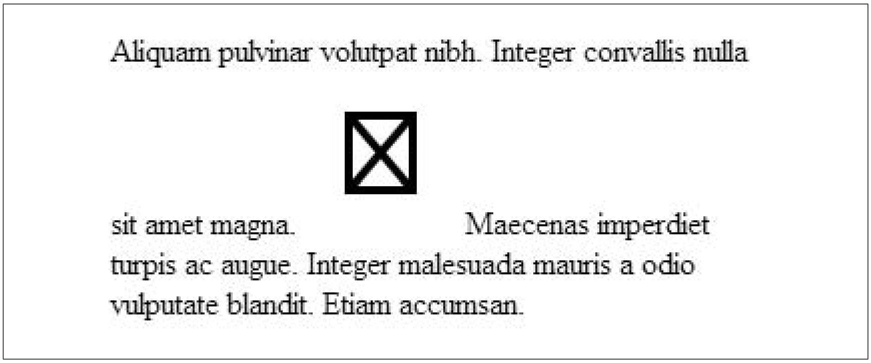
- Top and bottom margins applied to replaced inline elements (i.e., images and form inputs) *do* affect the height of the line. In other words, the margin is included in the inline box for replaced elements, and the line box is drawn larger to accommodate it.
- When an image has a margin, the bottom outer edge of the margin is placed on the baseline of the line (unless placement is altered with the `vertical-align` property on the `img`). The image in Figure 19-6 has 20 pixels of margin on all sides. The result is that the image is raised off the baseline by 20 pixels and the line height opens up to accommodate the image and its margin.



Aliquam pulvinar volutpat nibh. Integer convallis nulla
sit amet magna. Maecenas imperdiet
turpis ac augue. Integer malesuada mauris a odio
vulputate blandit. Etiam accumsan.

Figure 19-6. Margin settings on inline images

Borders

A border is simply a line drawn around the content area of an element and its (optional) padding. The three aspects of a border that can be specified are its style, width (thickness), and color. As for `margin`, each of these qualities may be specified for an individual side at a time or for several sides at once using shorthand properties.

There are only a few things to know about border style behavior:

- Borders are drawn on top of an element's background, so the background color or image will show through the gaps in the intermittent border styles.
- Borders applied to non-replaced inline elements (text elements) have no effect on the line height for that line. In other words, they are not included in the inline box for the element.
- Borders applied to replaced elements, however, do affect line height, just as margins do.

Border Style

The border style is the most important of the border qualities because, according to the CSS specification, if there is no border style specified, the border does not exist. In other words, you must always declare the style or other border settings will be ignored.

Figure 19-7 shows the nine border styles you have to choose from.

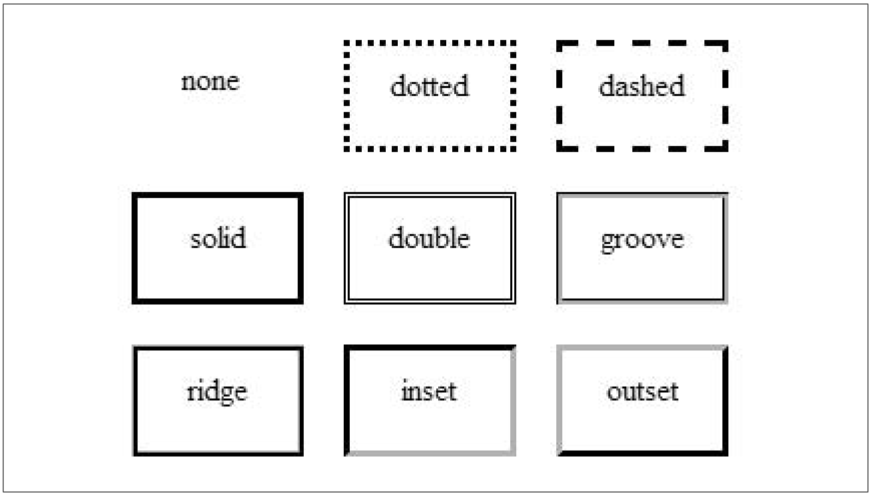


Figure 19-7. The nine available border styles

There is a bug in Internet Explorer 6 for Windows that causes borders specified as `dotted` to render as `dashed`.

Border styles can be applied one side at a time or by using the `border-style` shortcut property.

border-top-style, border-right-style, border-bottom-style, border-left-style

Values:	`none	dotted	dashed	solid	double	groove	ridge	inset	outset`
Initial values:	`none`
Applies to:	All elements
Inherited:	No

border-style

Values:	`[none	dotted	dashed	solid	double	groove	ridge	inset	outset]{1,4}	inherit`
Initial value:	Not defined
Applies to:	All elements
Inherited:	No

As you might expect, the border-top-style, border-right-style, border-bottom-style, and border-left-style properties allow you to specify a border style to one side of the element. If you do not specify a width for the border, the medium width value (the default) will be used. If there is no color specified, it uses the foreground color of the element (i.e., the text color). This example shows single-side border attributes in action (Figure 19-8).

```
div {border-top-style: solid;
     border-right-style: dashed;
     border-bottom-style: dotted;
     border-left-style: double; }
```

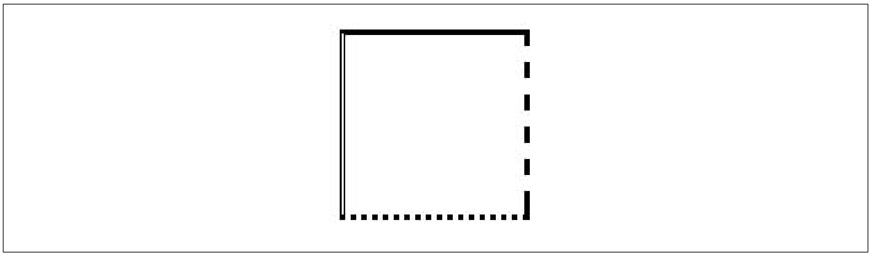


Figure 19-8. The border-style property

The border-style shortcut property works the same as the margin shortcut described earlier. Border style values for each side are provided in clockwise order: top, right, bottom, left. If fewer values are provided, some values are replicated. The right value will be used for a missing left value, the top value will be replicated for a missing bottom value; and if only one border style is provided, it will be applied to all four sides of the element.

The same effect shown in Figure 19-8 can be replicated using this border-style declaration:

```
div {border-style: solid dashed dotted double; }
```

Border Width (Thickness)

The thickness of the rule is controlled with one of the border width properties. As we've seen for margin and border-style, you can control the width of each individual side or use the border-width shorthand property to specify several sides at once. The shorthand values are provided in clockwise (top, right, bottom, left) order and replicate as described for the margin shorthand property earlier in this chapter.

border-top-width, border-right-width, border-bottom-width, border-left-width

Values: thin | medium | thick | <length> | inherit

Initial values: medium

Applies to: All elements

Inherited: No

border-width

Values: `[thin | medium | thick | <length> ]{1,4} | inherit`

Initial value: Not defined

Applies to: All elements

Inherited: No

The properties may use the keyword values `thin`, `medium`, and `thick`, in order of increasing width. The actual pixel value for each keyword is left up to the user agent, but must be consistent throughout the document. Border width can be specified in units of length as well (pixels are common). Negative length values are not permitted for borders.

Figure 19-9 shows an example of keyword and pixel-measurement border widths.

```
div {border-style: solid;
    border-top-width: thin;
    border-right-width: medium;
    border-bottom-width: thick;
    border-left-width: 12px; }
```

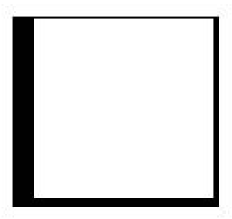


Figure 19-9. The border-width property

Border Color

Use one of the side-specific color properties or the `border-color` shorthand to specify a color for the border. Values for `border-color` are provided in clockwise (top, right, bottom, left) order and replicate as described for the `margin` shorthand property earlier in this chapter.

border-top-color, border-right-color, border-bottom-color, border-left-color

Values: `<color> | transparent | inherit`

Initial values: The value of the `color` property for the element

Applies to: All elements

Inherited: No

border-color

Values: `[<color> | transparent]{1,4} | inherit`

Initial value: Not defined

Applies to: All elements

Inherited: No

Colors values may be specified in any of the methods outlined in Chapter 16 and Appendix D. If no border color is declared, the default is the foreground color for the element (i.e., the text color for text elements).

The `border-color` shorthand property is demonstrated in this example and in Figure 19-10.

```
div {border: 6px solid;
    border-color: #333 #666 #999 #CCC; }
```

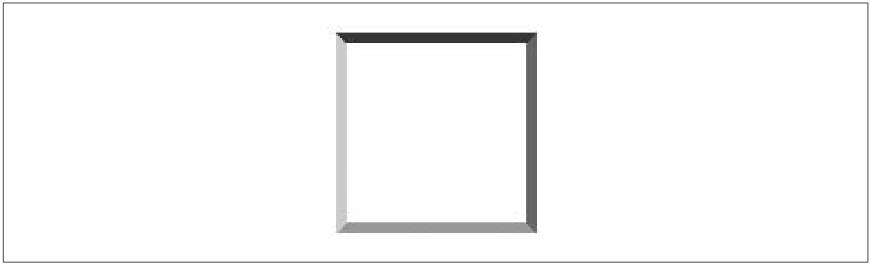


Figure 19-10. The border-color property

CSS 2 added the `transparent` value that allows the background of the parent element to show through the border, yet holds the width of the border as specified. This may be useful when creating rollover effects with CSS (this technique is explained in Chapter 24), because the space where the border will appear is maintained when the mouse is not over the element.

> Unfortunately, the `transparent` value is not supported in Internet Explorer for Windows through Version 6. Support in IE 7 (in beta as of this writing) is possible, but not documented.

Combining Style, Width, and Color

There is no shortage of shortcuts for specifying border appearance. Once again, we have rules that apply combinations of style, width, and color to one side at a time and the `border` property that applies the values to all sides of the element.

border-top, border-right, border-bottom, border-left

Values: `[<border-style> || <border-width> || <border-color> ] | inherit`

Initial value: Not defined

Applies to: All elements

Inherited: No

border

Values: `[<border-style> || <border-width> || <border-color> ] | inherit`

Initial value: Refer to individual properties

Applies to: All elements

Inherited: No

The side-specific and the shorthand border properties may include a `border-style` value, a `border-width` value, and a `border-color` value. They do not need to be in any particular order. You do not need to declare all three border qualities, but keep in mind that if the `border-style` is not declared, there will be no border.

The `border` shorthand is somewhat different from the other shorthand properties discussed so far in that it can be used to apply border properties to all four sides of the element only. It does not provide a way to target certain borders and there is no system of value replication.

The rules listed here are all valid examples of the border shortcut properties.

```
h1 {border-left: .5em solid blue; }
h1 {border-left: solid blue .5em; }
h1 {border-left: solid .5em; }

p.example {border: 2px dotted #666633; }
p.example {border: dotted 2px; }
```