CHAPTER 29

Formatting Paragraphs

In This Chapter

The CSS box formatting model

Adding indentations

Controlling alignment

Choosing line spacing

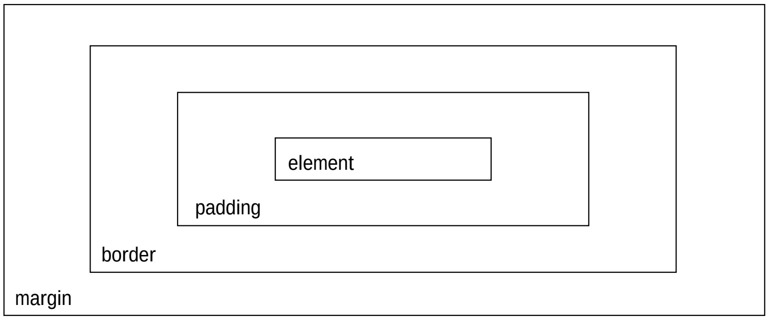
Controlling lists with styles

Adding borders

In this chapter, you learn about formatting paragraphs with the *box formatting model*. This CSS feature formats block-level elements of a document, including paragraphs. You learn about indenting text, controlling alignment, specifying line spacing, and controlling lists. You also learn how to add rules and borders.

Reviewing the CSS Box Formatting Model

You may recall the CSS box formatting model from Chapter 26: After a block-level object is rendered, the browser uses information in the style sheet to determine how to format around the box in which the object is rendered.



Padding summary

Padding — the white space between the element itself and the border (if there is one) — takes on the background color assigned to the element. These five properties are related to padding:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`
- `padding`

The first four take either a valid length or a percentage of the canvas. The fifth property, `padding`, is the shortcut for the first four. It can take between one and four values. It assumes that the values you provide are for the top, right, bottom, and left, in that order.

Margins summary

Margins take on the background color of the parent element — the container in which this block-level element is defined. For example, if the element being defined is a paragraph, the parent element might be the `BODY` element. Suppose two items are vertically aligned on the page, each with its own margin defined; for the top element, the margin would be the `margin-bottom` element, for the bottom element, it would be `margin-top`; the browser renders the two elements with a margin between them that equals the larger of these two margins, not the sum of the margins. For elements aligned horizontally, the browser renders each element with its own margin; it does not collapse the two margins into the greater of the two.

Five properties specify not only the margins, but also the white space between the border of the element (if it has one) and the margin of the next element. These properties are as follows:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`
- `margin`

The margin properties work exactly the same as the padding properties.

Units of length

To define padding or margins, first you need to know what units you can use to define them. Many properties in CSS take lengths in both *relative units* and *absolute units*. The relative units are as follows:

- **em** — the height of the element's font
- **ex** — the height of the letter x
- **px** — pixels, which are measurable on the canvas

The absolute units are as follows:

- **in** — inches
- **cm** — centimeters
- **mm** — millimeters
- **pt** — points; 72 points to an inch
- **pc** — picas; 12 points to a pica

Adding Indentations

Another way to format your paragraph is to indent the first line of text, or outdent the first line of text. *Outdenting,* also known as creating a hanging indent, is when the first line of text hangs out into the margin and subsequent lines of text begin further in. You can define text indenting using the `text-indent` property as follows:

```
text-indent: .5in;
text-indent: 2cm;
text-indent: -8em;
```

Controlling Alignment

You might also want to control what is normally called text justification. CSS calls it *alignment.* Text alignment can be set to `left`, which is the default; `center`;, `right`;, or `justify`, which is full alignment.

The property is called `text-align`. Here are some examples:

```
text-align: left;
text-align: center;
text-align: justify;
```

Choosing Line Height

By default, your browser renders your font size with one pixel above the tallest letter and one pixel below the lowest-reaching letter. So, if you specify you want the font-size to be 12pt, the browser assumes line spacing should be 14pt. You can, however, change that by using a greater `line-height` to spread out your text on the page.

The following example creates a paragraph with standard indentation and very tight line spacing:

```
P {
  font-size: medium;
  line-height: 12pt;
}
```

Controlling Lists with Styles

You can use most of the properties listed earlier in this chapter to format your lists as well. Each list item is formatted as its own block-level element, which means you can assign that element its own padding, borders, and margins. You can specify what bullets look like for unordered lists, the numbering system used for ordered lists, and (if you prefer) the image to be used as a bullet point. You can also specify whether the bullets are outdented (which is probably the way you're used to seeing them in lists), or flush with the left margin; if your text direction is right-to-left, your bullets can be flush with the right margin.

List style type

For ordered lists (`OL`), which typically have a number or a letter next to each list item, you can choose from among these types of list style:

- **decimal.** 1, 2, 3, 4, 5, and so on
- **lower-roman.** i, ii, iii, iv, v, and so on
- **upper-roman.** I, II, III, IV, V, and so on
- **lower-alpha.** a, b, c, d, e, and so on
- **upper-alpha.** A, B, C, D, E, and so on

For unordered lists, which typically have some shape of bullet next to each list item, you can choose between

- **disc.** standard solid circle

- **circle.** empty circle
- **square.** solid square

Some examples follow:

```
OL {
list-style-type: lower-alpha;   /* a, b, c, etc. */
}
UL {
       list-style-type: circle;
}
```

List style image

CSS enables you to specify an image instead of a disc, circle, or square for the bullet in unordered lists; you identify the image by its URL. The property is named `list-style-image`. If the browser can't find the URL of the image you have selected, it uses the style type you specified in the `list-style-type property`.

```
UL {
       list-style-type: square;
       list-style-image: url(../images/blue-box.gif);
}
```

List style position

The following example shows regular formatting of lists.

- Outside list item 1

  has bullet outside the margin of subsequent lines
- Outside list item 2

  has bullet outside the margin of subsequent lines

This is how you specify it:

```
UL {
       list-style: outside;
}
```

The next example shows compact formatting of lists:

- Inside list item 1

  has flush subsequent lines

✦ Inside list item 2

has flush subsequent lines

You can specify compact formatting by using the CSS property called `list-style`. The code looks like this:

```
UL {
        list-style: inside;
}
```

Notice the property is associated with the `UL` element and not with the `LI` elements.

List style shorthand

As with many CSS property groups, you can use the `list-style` property to group your list formatting. The `list-style` property takes three values: the value of the `list-style-type` property, the value of the `list-style-position` property, and the value of the `list-style-image` property. The possible values of the three don't conflict; you can specify only one value — or any three values — and it will work. An example follows:

```
UL {
        list-style: square outside url(../images/blue-box.gif);
}
OL {
        list-style: decimal inside;
}
```

Adding Borders

Borders are a great way to break up your page and to draw attention to something important. CSS offers you superior flexibility when you specify which (if any) rules and borders you want to include with your box-formatted elements. You can choose `border-width` properties, a `border-color` property, a `border-style` property, and a variety of shorthand notations to define borders.

Border width

You can use five properties to define the width of the border. They are `border-top-width`, `border-right-width`, `border-bottom-width`, `border-left-width`, and `border-width`. CSS is nothing if not predictable. You can specify any of the first four or you can use the fifth one to specify up to all four of the widths in shorthand, just as with the margin and padding properties.

If you specify one value for the `border-width` property, all four sides will be that width. If you specify two values for the `border-width` property, the top and bottom are assigned the first value, and the sides are assigned the second value. If you specify three values (and this is completely unintuitive), the top is assigned the first value, the sides are assigned the second value, and the bottom is assigned the third value.

In addition to specifying a valid length (using one of the measurements listed earlier), you can also use one of three keywords to instruct the browser to render your border width: `thin`, `medium`, and `thick`. The default is `medium`. An example of such a border rendering follows:

```
border-width: thin;           /* all 4 sides will be thin */
border-width: medium thick;   /* top, bottom medium, sides
thick */
border-width: 4px thin 2px;   /* top 4px, sides thin, bottom
2px */
border-width: 3mm 4mm 2mm 1mm; /* top, right, bottom, left, in
that order */
```

Border color

The `border-color` property takes up to four values; it works exactly like the `border-width` property. You can specify border color for all four sides at once or for each side individually. The property is called `border-color`. Here is what it looks like in action:

```
border-color: #FF0000;           /* makes all four borders
red */
border-color: red blue red blue;  /* makes top and bottom
borders red, sides blue */
border-color: yellow black;       /* makes top and bottom
yellow, sides black */
```

Border style

The `border-style` property tells the browser what kind of border to use. As with the `border-width` property, you can specify one to four values. Netscape 4.7 supports more styles then Internet Expolorer 5.0. Your choices are as follows:

- **none.** No border; overrides `border-width` specification.
- **dotted.** Border is a dotted line.
- **dashed.** Border is a dashed line.
- **solid.** Border is a solid line (the default).

- **double.** Border is a double line. The `border-width` value specifies the total width of both lines and the white space between them.
- **groove.** Border is a 3D groove of the color assigned in `border-color` property.
- **ridge.** Border is a 3D ridge of the color assigned in `border-color` property.
- **inset.** Border is a 3D inset of the color assigned in `border-color` property.
- **outset.** Border is a 3D outset of the color assigned in `border-color` property.

```
border-style: solid none;     /* makes top and bottom solid,
none on sides */
border-style: ridge;         /* ridge on all four sides */
border-style: dotted dashed solid   /* top dotted, sides
dashed, bottom solid */
border-style: solid dotted double dashed   /* top, right,
bottom, left in that order */
```

Shorthand techniques

CSS provides several techniques for creating shorthand definitions related to borders. Four properties — `border-top`, `border-right`, `border-bottom`, and `border-left` — can be used to specify values for width, style, and color, in that order. A brief example follows:

```
border-top: thick dashed blue;
border-bottom: thick dashed red;
```

A `border` property can be used to set values for width, style, and color — but only if you want to assign the same values to all four sides, as in the following example:

```
border: thin solid black;     /* all four sides will be thin,
solid, and black */
border: 6px double yellow;    /* all four sides will be 6px,
double, and yellow */
```

From Here

Cross-Reference

Jump to Chapter 33 and learn about using CSS for absolute positioning.

Proceed to Chapter 30 and learn about CSS and tables.

Summary

In this chapter, you reviewed the CSS box-formatting model. You learned more about specifying margins and padding for paragraphs. You learned the units of measurement to use when specifying many CSS property values. You also learned about including paragraph indentation, paragraph alignment, and the property that specifies line spacing. You learned about formatting lists, either with the bullet outdented or flush with the text. Finally, you learned about specifying borders for your block-level elements.

✦ ✦ ✦