

Positioning Graphics and Text

In this chapter, you learn more tricks to give you control over your page's layout using CSS and HTML together. Specifically, you learn how to have an image appear in the location of your choice. This chapter explains how to position your text where you want it to appear and how to wrap your text around floating graphics. Many examples help jog your own creativity.

Specifying Image Location

In the previous chapters, you learned enough CSS to specify where on the page an image should be located. In fact, you learned more than one way to do this.

Using frames

One way to ensure that your image ends up where you want it is to use frames. Consider placing an image in the upper right-hand corner of the screen. If you use frames, you can be sure the graphic will never move.

Your HTML might look like this:

```
<FRAMESET rows="200, *">
  <FRAMESET cols="*, 200">
    <FRAME name="banner" src="banner.html"
frameborder="0">
    <FRAME src="image.gif" scrolling="no"
frameborder="0">
  </FRAMESET>
<FRAME name="main" src="home.html">
</FRAMESET>
```



In This Chapter

Specifying the location of an image

Defining text positions

Floating text around an image

Floating an image next to text



In the preceding example, your image, called `image.gif`, should be exactly 200×200 pixels. By setting the `scrolling` attribute to `no`, you are assured that a scroll bar won't be to the right of your image. By setting the `frameborder` attribute to zero for both the columns in the first row of the page, you are assured a border won't be around the banner or around the image next to the banner.

Using CSS with absolute positioning

Another way to specify an image's position is to use the CSS `position` property with a value of `absolute`. You can use this method to position one or more graphics on the page. The problem with this technique is that it requires careful planning to make sure the elements positioned as `static` (or `relative`) don't run over your absolutely positioned element — in this case, an image. You can also use absolute positioning to position subtle graphics in a case where you want the text run right across the images.

Consider the following CSS:

```
IMG.no1 {
    position: absolute;
    top: 200;
    right: auto;
    bottom: auto;
    left: 350;
}
IMG.no2 {
    position: absolute;
    top: 800;
    right: auto;
    bottom: auto;
    left: 350;
}
```

Both images have absolute positioning from the top-left of the browser window. All images that use either of the two classes `no1`, and `no2`, will take their positioning from the class. The resulting page looks like Figure 34-1. Notice you only see one image. Because the absolute position of the second image (`no2`) is 800 pixels down the page and the viewable area is less than that, the second image is only visible when you scroll down the page.

Using CSS with relative positioning

Figure 34-2 uses relative positioning to place the image to the left of the text.

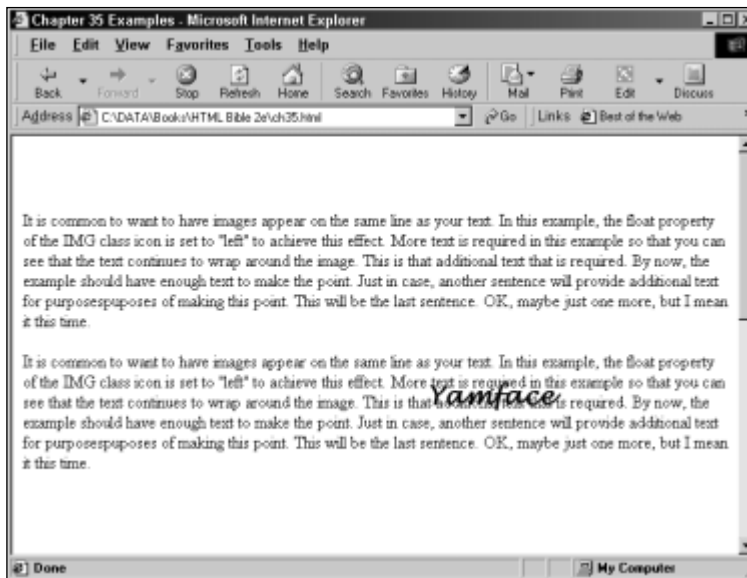


Figure 34-1: This absolutely positioned image has text overlaying the image.

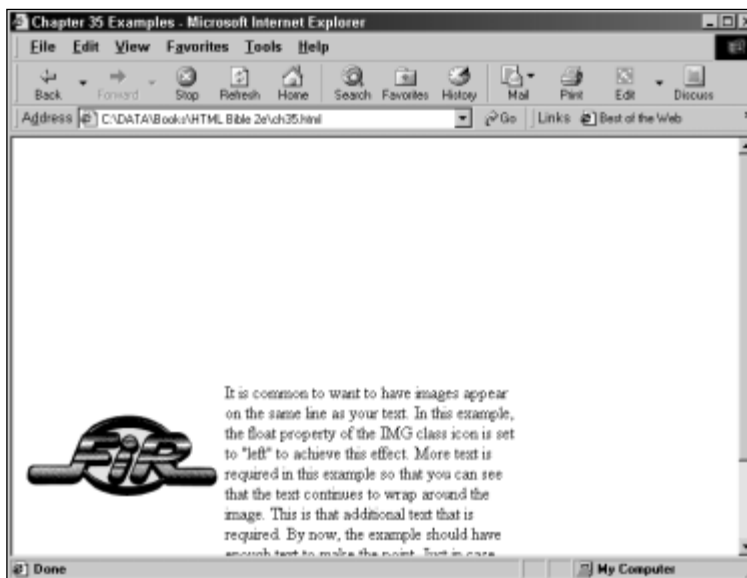


Figure 34-2: Relative positioning places the image to the left of the body of text.

Look at the following style sheet and notice the `BODY` element is formatted with a 200-pixel left-hand border. This leaves room to relatively position the image next to the text. What about all that white space above the paragraph? This is because when you use relative positioning for one element, all subsequent elements are positioned based on that element's normal position, not its new, relative position. What does this mean? It means the browser leaves space for the image *above* the paragraph of text, even though the image is positioned *next to* the paragraph of text. If you have multiple paragraphs, this could be ugly.

```
P.thisone {
    position: normal;
}
IMG.relative {
    position: relative;
    left: -190;
    top: 150;
}
BODY {
    margin-left: 200;
}
```

Using CSS to float the image

If the `float` property had been used instead, the result would have been Figure 34-3.



Figure 34-3: The float property wraps the text around the image.

Notice the text wraps around the image, rather than giving you a clean left margin. Either look has its place, depending on what you desire. The CSS for Figure 34-3 follows:

```
IMG.icon {
    float: left;
}
P {
    margin-left: 15px;
}
```

Defining Text Positions

How can you put text where you want it? You have several formatting choices. You can format your text by modifying the padding and margin of the text element itself, as in Figure 34-4.

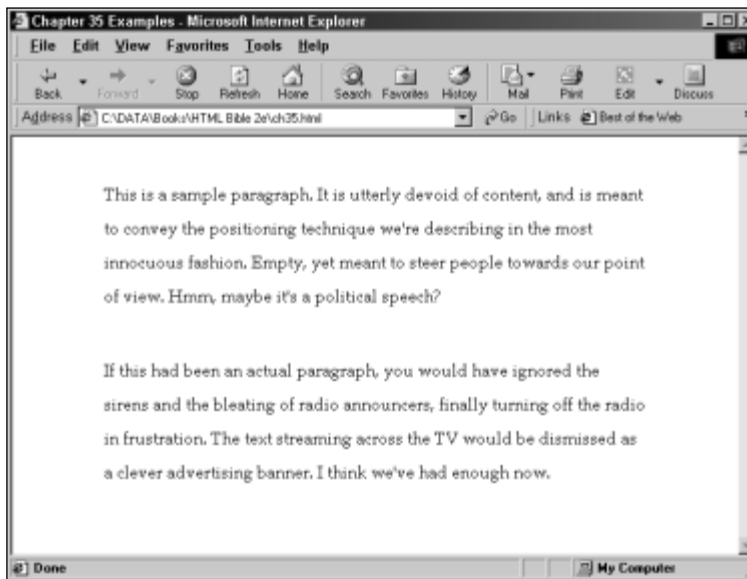


Figure 34-4: Text formatted with margins and padding

The CSS that follows was used to create the simple effects you see in Figure 34-4:

```
P {
    margin-top: 1cm;
    line-height: 200%;
    font-family: "Book Antiqua";
    padding: 0 2cm;
}
```

Changing the BODY element

Another option you have for formatting your page — so the text appears where you want it — is to change the formatting of the `BODY` element. Figure 34-5 uses `BODY` element formatting.

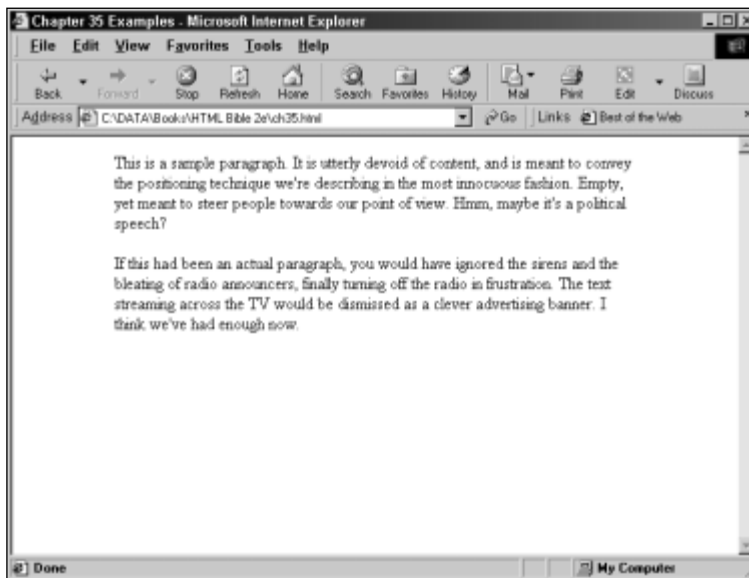


Figure 34-5: Text formatting with the `BODY` element

Figure 34-5 was created with the following CSS:

```
BODY {  
    margin-left: 1in;  
}
```

If you wanted to add page decorations to the left margin, you could put it in the one-inch, left-hand margin. Or, you could put an absolutely positioned graphic, with a negative horizontal position, in that space.

Positioning text with relative positioning

You might need to pull text out of the body of the page for some reason. Figure 34-6 uses relative positioning to pull a paragraph of text out of the body of the page and draw attention to it.

The following CSS was used to create Figure 34-6:

```

BODY {
    margin-right: 1.5in;
}
P.pull {

    position: relative;
    padding: 1em 1em 1em 3em;
    left: 150px;
    top: auto;

    border-left: red .25cm solid;
    border-top: red 1px solid;
    border-bottom: red 1px solid;
}

```

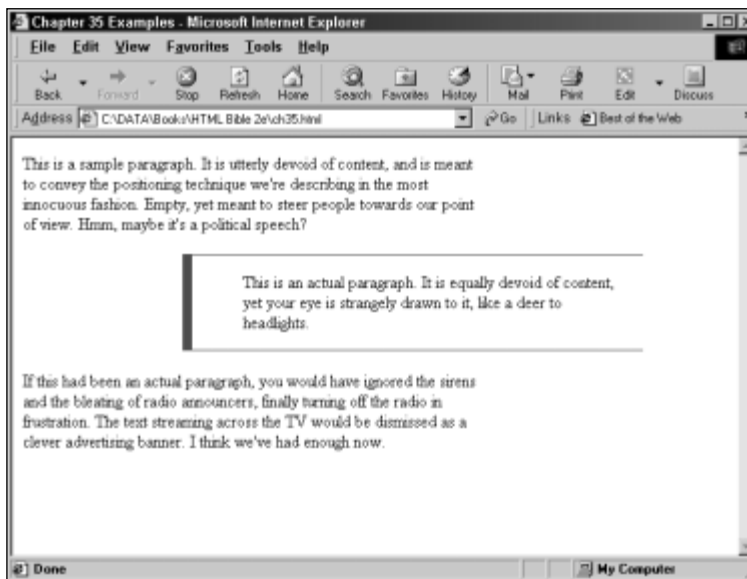


Figure 34-6: Text formatted with relative positioning

Floating an Image Next to Text

In Figure 34-7, you can see the same boring text and the same boring images are rearranged, so the text floats around the image, first to the right of the image and then to the left.

The CSS used to create this effect follows:

```
IMG.left {
    float: left;
    margin-left: 1.5cm;
    margin-right: 10pt;
}
IMG.right {
    float: right;
    margin-right: 1.5cm;
    margin-left: 10pt;
}
```

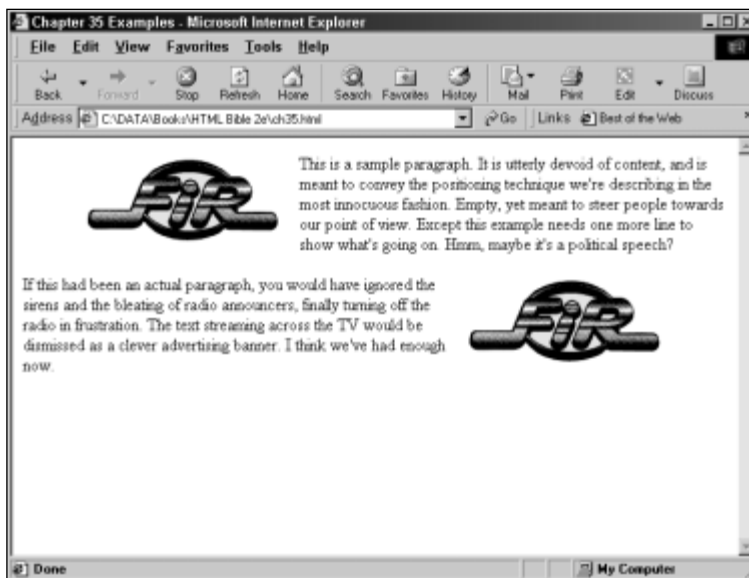


Figure 34-7: Text floating around images

The float property is not perfectly implemented in either Internet Explorer 5 or Netscape 4.7. Generally, Internet Explorer supports more of the CSS properties. Some combinations of properties give strange and unpredictable results, so you need to test your work carefully.

Floating both the text and the image

Figure 34-8 is the result of applying the float property to both the P elements and the IMG elements. In Internet Explorer, this combination has no visible effect

and the result looks exactly the same as Figure 34-7, where the float property is only applied to the IMG element. As you can see, Netscape does not handle this combination well.



Figure 34-8: Netscape has problems with the float property being applied to both the P element and the IMG element.

Floating only the text

Neither browser wants to have the float property assigned to the text. Both browsers want to have the float property assigned to the IMG element. The result of applying the float element to the text element (in this case, the P element) varies by browser, but is equally unappealing in both browsers, as can be seen in Figures 34-9 and 34-10.

The order of the HTML matters

If you use the same CSS that was effective in Figure 34-7, but change the HTML so the IMG element appear *after* both P elements, then you get different and unsatisfactory results, as you can see in Figures 34-11 and 34-12.

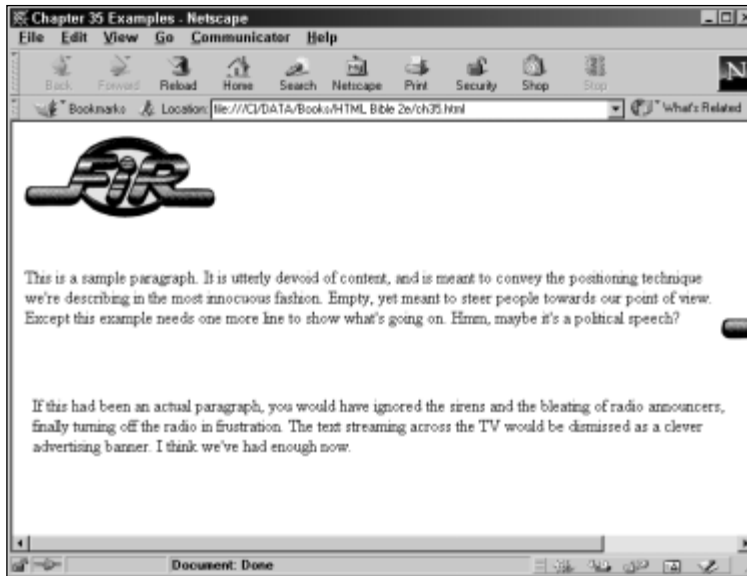


Figure 34-9: Netscape can't handle floating when the float property is applied to the text, rather than to the IMG element.

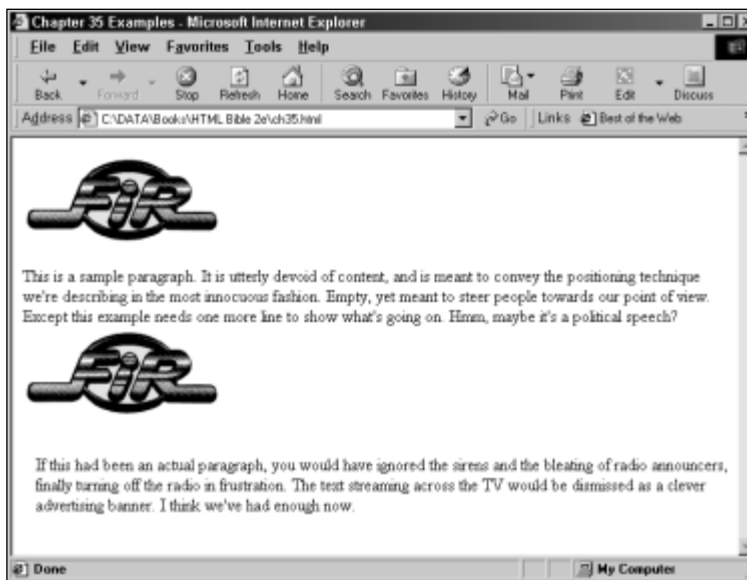


Figure 34-10: Internet Explorer doesn't even try to float anything when the float property is applied to the text, rather than to the image.



Figure 34-11: Netscape chokes if the IMG element is listed in your HTML after the P element.

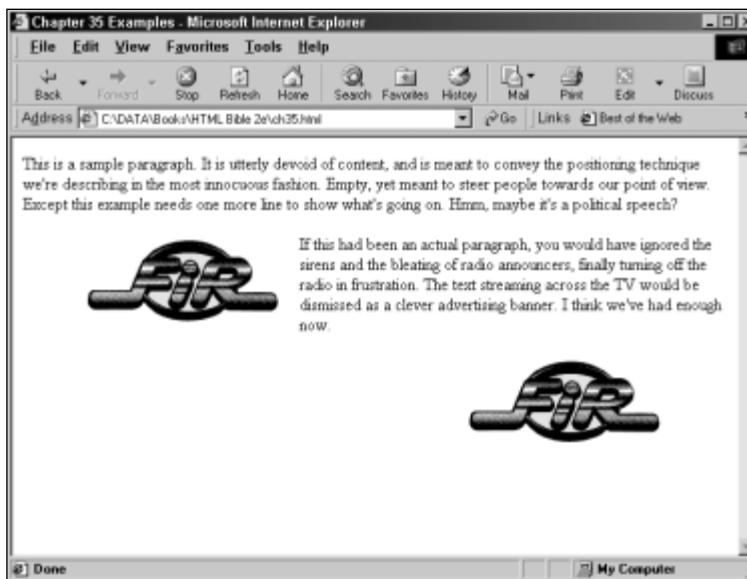


Figure 34-12: Internet Explorer doesn't seem to understand that the IMG element should float next to the preceding P element, even when the IMG element will be floating to the right of the P element.

From Here



Jump to Chapter 37 and learn how to create still graphics for the Web.

Proceed to Chapter 35 and learn more cool CSS positioning tips and tricks.

Summary

In this chapter, you learned how to position text and graphics using a variety of techniques, including frames, the position property, the float property, changing margins and padding of text blocks, and changing margins of the `BODY` element. You also learned some of the implementation shortcomings of the two leading browsers: Internet Explorer 5 and Netscape 4.7.

