

## Hands-On Practice 6.1

You will explore the box model and normal flow in this Hands-On Practice as you work with the web pages shown in Figure 6.3 and Figure 6.4.

### Practice with Normal Flow

Launch a text editor and open `chapter6/starter1.html` in the student files. Save the file with the name `box1.html`. Edit the body of the web page and add the following code to configure two `div` elements:

```
<div class="div1">
This is the first box.
</div>
<div class="div2">
This is the second box.
</div>
```

Now let's add embedded CSS in the head section to configure the "boxes." Add a new style rule for a class named `div1` to configure a light-blue background, dashed border, width of 200 pixels, height of 200 pixels, and 5 pixels of padding. The code is

```
.div1 { width: 200px;
        height: 200px;
        background-color: #D1ECFF;
        border: 3px dashed #000000;
        padding: 5px; }
```

Create a style rule for a class named `div2` to configure a width and height of 100 pixels, white background color, ridged border, 10 pixel margin, and 5 pixels of padding. The code is

```
.div2 { width: 100px;
        height: 100px;
        background-color: #ffffff;
        border: 3px ridge #000000;
        margin: 10px;
        padding: 5px; }
```

Save the file. Launch a browser and test your page. It should look similar to the one shown in Figure 6.3. The student files contain a sample solution (see `chapter6/6.1/box1.html`).

### Practice with Normal Flow and Nested Elements

Launch a text editor and open your `box1.html` file. Save the file as `box2.html`.

Edit the code. Delete the content from the body section of the web page. Add the following code to configure two `div` elements—one nested inside the other.

```
<div class="div1">
This is the outer box.
  <div class="div2">
This is the inner box.
  </div>
</div>
```

Save the file. Launch a browser and test your page. It should look similar to the one shown in Figure 6.4. Notice how the browser renders the nested div elements: The second box is nested inside the first box because it is coded inside the first div element in the web page source code. This is an example of normal flow. The student files contain a sample solution (see chapter6/6.1/box2.html). The examples in this Hands-On Practice happened to use two div elements. However, the box model applies to block display HTML elements in general, not just to div elements. You will get more practice using the box model in this chapter.

## 6.3 CSS Float

Elements that seem to float on the right or left side of either the browser window or another element are often configured using the **float property**. The browser renders these elements using normal flow and then shifts them to either the right or left as far as possible within their container (usually either the browser viewport or a div element).

- Use `float: right;` to float the element on the right side of the container.
- Use `float: left;` to float the element on the left side of the container.
- Specify a width for a floated element unless the element already has an implicit width, such as an `img` element.
- Other elements and web page content will flow around the floated element.

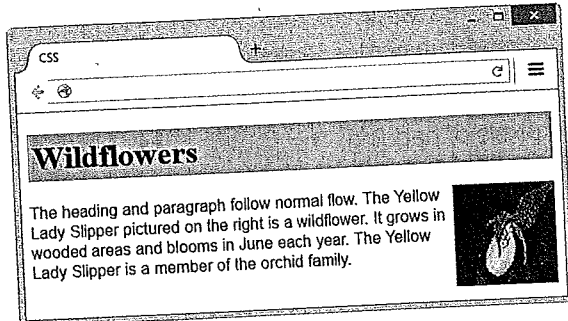


Figure 6.5 The image is configured to float

Figure 6.5 shows a web page with an image configured with `float: right;` to float on the right side of the browser viewport (see chapter6/float.html in the student files). When floating an image, the `margin` property is useful to configure empty space between the image and text on the page.

View Figure 6.5 and notice how the image stays on the right side of the browser viewport. An id called `yls` was created that applies the `float`, `margin`, and `border` properties. The attribute `id="yls"` was placed on the image tag. The CSS is

```
h1 { background-color: #A8C682;
      padding: 5px;
      color: #000000; }
p { font-family: Arial, sans-serif; }
#yls { float: right;
        margin: 0 0 5px 5px;
        border: 1px solid #000000; }
```

The HTML source code is

```
<h1>Wildflowers</h1>

<p>The heading and paragraph follow normal flow. The Yellow Lady
Slipper pictured on the right is a wildflower. It grows in wooded
areas and blooms in June each year. The Yellow Lady Slipper is a
member of the orchid family.</p>
```

## Hands-On Practice 6.2

In this Hands-On Practice, you'll use the CSS float property as you configure the web page shown in Figure 6.6.



Figure 6.6 The CSS float property left-aligns the image

Create a folder named `ch6float`. Copy the `starteryls.html` and `yls.jpg` files from the `chapter6` folder in the student files into your `ch6float` folder. Launch a text editor and open `starteryls.html`. Notice the order of the images and paragraphs. Note that there is no CSS configuration for floating the images. Display `starteryls.html` in a browser. The browser renders the page using normal flow and displays the HTML elements in the order they are coded.

Let's add CSS to float the image. Save the file as `index.html` in your `ch6float` folder. Open the file in a text editor and modify the code as follows:

1. Add a style rule for a class name `float` that configures float, margin, and border properties.

```
.float { float: left;
         margin-right: 10px;
         border: 3px ridge #000000; }
```

2. Assign the image element to the class named `float` (use `class="float"`).

Save the file. Launch a browser and test your page. It should look similar to the web page shown in Figure 6.6. The student files (`chapter6/6.2/index.html`) contain a sample solution.

### The Floated Element and Normal Flow

Take a moment to examine your file in a browser (see Figure 6.6) and consider how the browser rendered the page. The `div` element is configured with a light background color to demonstrate how floated elements are rendered outside of normal flow. Observe that the image and the first paragraph are rendered outside of normal flow. The `h2` element follows the `div`. If all the elements were rendered using normal flow, the area with the light background color would contain both child elements of the `div`: the image and the first paragraph. In addition, the `h2` element would be placed on its own line under the `div` element.

However, once the image is placed vertically on the page, it is floated outside of normal flow—that's why the light background color only appears behind the first paragraph and why the h2 element's text begins immediately after the first paragraph and appears next to the floated image. In the following sections, you'll explore properties that can "clear" this float and improve the display.

## 6.4 CSS: Clearing a Float

### The clear Property

The **clear** property is often used to terminate, or clear, a float. You can set the value of the clear property to `left`, `right`, or `both`, depending on the type of float you need to clear.

Review Figure 6.6 and the code sample (see `chapter6/6.2/index.html` in the student files). Notice that although the `div` element contains both an image and the first paragraph, the light background color of the `div` only displays behind the screen area occupied by the first paragraph—it stops a bit earlier than expected. Clearing the float will help take care of this display issue.

### Clear a Float with a Line Break

A common technique to clear a float within a container element is to add a line break element configured with the `clear` property. See `chapter6/clear1.html` in the student files for an example. Observe that a CSS class is configured to clear the left float:

```
.clearleft { clear: left; }
```

Also, a line break tag assigned to the `clearleft` class is coded before the closing `div` tag. The code for the `div` element is

```
<div>  
  
<p>The Yellow Lady Slipper grows in wooded areas and blooms in June  
each year. The flower is a member of the orchid family.</p>  
<br class="clearleft">  
</div>
```

Figure 6.7 displays a screen shot of this page. Review Figure 6.6 and note two changes: the light background color of the `div` element extends farther down the page and the `h2` element's text begins on its own line under the image.

### Another Technique to Clear a Float

If you are not concerned about the light background color display, another option is to omit the line break tag and, instead, apply the `clearleft` class to the `h2` element, which is the first block display element after the `div`. This does not change the display of the light background color, but it does force the `h2` element's text to begin on its own line, as shown in Figure 6.8 (see `chapter6/clear2.html` in the student files).



Figure 6.7 The clear property is applied to a line break tag



Figure 6.8 The clear property is applied to the h2 element

## The overflow Property

The **overflow property** is often used to clear a float, although its intended purpose is to configure how content should display if it is too large for the area allocated. See Table 6.2 for a list of commonly used values for the overflow property.

Table 6.2 The overflow property

| Value   | Purpose   |
|---------|---|
| visible | Default value; the content is displayed, and if it's too large, the content will overflow and extend outside the area allocated to it |
| hidden  | The content is clipped to fit the area allocated to the element in the browser viewport   |
| auto    | The content fills the area allocated to it and, if needed, scrollbars are displayed to allow access to the remaining content          |
| scroll  | The content is rendered in the area allocated to it and scrollbars are displayed  |

## Clear a Float

Review Figure 6.6 and the code sample (chapter6/6.2/index.html in the student files). Observe the div element, which contains the floated image and first paragraph on the page. Notice that although the div element contains both an image and the first paragraph, the light background color of the div element does not extend as far as expected; it is only visible in the area occupied by the first paragraph. You can use the `overflow` property assigned to the container element to resolve this display issue and clear the float. In this case, we'll apply the `overflow` and `width` properties to the div selector. The CSS to configure the div in this manner is

```
div { background-color: #F3F1BF;
      overflow: auto;
      width: 100%; }
```

This will clear the float. The web page will display as shown in Figure 6.9 (see chapter6/overflow.html in the student files).



Figure 6.9 The overflow property is applied to the div selector



Figure 6.10 The browser displays scrollbars

Notice that using the `overflow` property (see Figure 6.9) and applying the `clear` property to a line break tag (see Figure 6.7) result in a similar web page display. You may be wondering about which CSS property (`clear` or `overflow`) is the best one to use when you need to clear a float.

Although the `clear` property is widely used, in this example, it is more efficient to apply the `overflow` property to the container element (for example, a div element). This will clear the float, avoid adding an extra line break tag, and ensure that the container element expands to enclose the entire floated element. You'll get more practice with the `float`, `clear`, and `overflow` properties as you continue working through this textbook. Floating elements is a key technique for designing multicolumn page layouts with CSS.

### Configure Scrollbars

The web page in Figure 6.10 demonstrates the use of `overflow: auto;` to automatically display scrollbars if the content exceeds the space allocated to it. In this case, the div that contains the paragraph and the floated image was configured with a width of 300px and a height of 100px.

See the example web page (`chapter6/scroll.html` in the student files). The CSS for the div is shown below:

```
div { background-color: #F3F1BF;
      overflow: scroll;
      width: 300px;
      height: 100px; }
```



### Checkpoint 6.1

1. List the components of the box model from innermost to outermost.
2. Describe the purpose of the CSS float property.
3. Which two CSS properties can be used to clear a float?