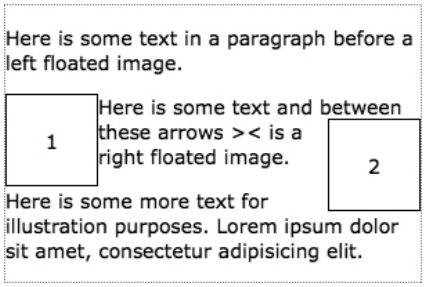
`float`

An element with a `float` value of `left` or `right` is taken out of the normal flow of a document and shifted to one side of the containing box. The content that follows it in the document wraps around the floated element's new position.

- `none`: An element is not floated and behaves as normal.
- `left`: An element is taken out of the normal flow and is shifted to the left of where it was to otherwise appear, with content flowing around it on the right side of the element.
- `right`: An element is taken out of the normal flow and is shifted to the left of where it was to otherwise appear, with content flowing around it on the right side of the element.

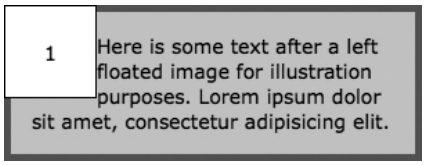
Think of a small photo that is moved to the right and has text flowing around it; you now have the basic idea of how an element (the photo) behaves when floated (**Figure 6.5** on the next page).

Figure 6.5. Two small elements floated inside text content.



Margins and padding set on a floated element behave as they would on any box in the standard box model and can be used to create a gutter around that element. However, floating has the effect of interrupting the boxes of the wrapping elements leaving their padding, margins, or borders drawn in their normal rectangle, as shown in **Figure 6.6**, rather than moved or redrawn for each individual row of text. Note the way the text touches the right edge of the floated image in this example.

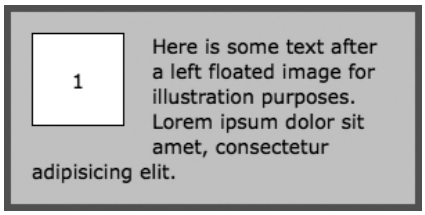
Figure 6.6. Text wrapped around a floated element.




```
img {
   float: left;
}
p {
   border: 5px solid #666;
   padding: 15px;
   background: #ccc;
}
[...]
<img src="images/1.png" width="67" height="67" alt="1" />
<p>Here is some text after a left floated image for illustration purposes. Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p
```

Adding a 20-pixel margin around the floated image would create a gutter between the opposite side of the floated element and the content wrapping around it (**Figure 6.7**). After doing this, it is also obvious how the rectangular box drawn by the paragraph including its border and background color are not "wrapping." Only the content inside of those blocks is wrapping.

Figure 6.7. A margin placed on the floated element to create space between it and the content wrapped around it.




```
img {
   float: left;
   margin: 20px;
}
```

If multiple elements are floated and would appear along the same edge, then they are arranged horizontally with the earliest element in the source appearing closest to the containing block's edge. If the combined width of the floated elements becomes too wide for the containing block's content area, then the later floated elements will wrap. This behaves a bit less like the normal flow of text from line to line and more like how marbles of different sizes will settle as they are placed in a cup. Elements of various sizes are left floated in **Figure 6.8** and right floated in **Figure 6.9** to illustrate the wrapping and ordering behaviors.

Figure 6.8. Left-floated blocks of various sizes.

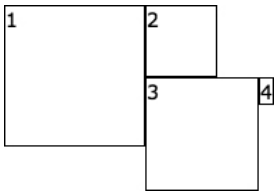
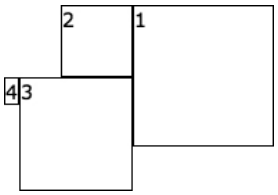


Figure 6.9. Right-floated blocks of various sizes.



Tip

Floated elements cannot appear before (vertically) elements that come before them in the document, which is why box 4 does not appear next to box 2 even if there were room for it.

With modern layout techniques, floated elements are used for much more than small bits of content in elements and text wrapping. They can be the basis for placing elements on opposite sides of the same line, controlling a series of tabs, and even controlling multiple-column layout grids.

Note

If an element is positioned with absolute positioning, that positioning will take precedence over the behavior of `float` .