```
clear
```
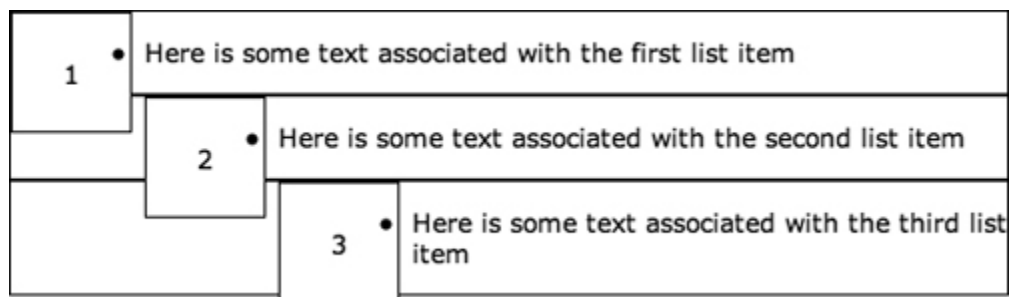
Floating is often used instead of positioning or large margins because the content-wrapping effects are desired, but the wrapping of a tall floated element can often extend into later content in an undesirable way. The `clear` property is used to insert a break when encountering a new section of the layout or another unique record in a list or some other reason for wanting to stop the wrapping effect.

- `none`: The element does not clear any floats.

- `left`: The element will be shifted down to sufficiently clear the bottom edge of any previous left-floated elements.

- `right`: The element will be shifted down to sufficiently clear the bottom edge of any previous right-floated elements.

- `both`: The element will be shifted down to clear and begin after all floated elements.

In **Figure 6.10** you can see what happens when the floated element from one list item bleeds into the following one; things really start to go haywire.

Figure 6.10. The effect of floated elements extending past its parent element into the following content.


```
li {
    border: 1px solid #000;
}
img {
    float: left;
    margin-right: 0.5em;
}
[...]
<ul>
    <li>
        <img src="images/1.png" width="67" height="67" alt="1" />
        <p>Here is some text associated with the first list item</p>
    </li>
    <li>
        <img src="images/2.png" width="67" height="67" alt="2" />
        <p>Here is some text associated with the second list
    item</p>
```

```
    </li>
    <li>
        <img src="images/3.png" width="67" height="67" alt="3" />
        <p>Here is some text associated with the third list item</p>
    </li>
  </ul>
```

You can set the clear property to left or both on the <li> element, as shown in **Figure 6.11**, to make sure that for each new item nothing from the previous item interferes.

Figure 6.11. clear:left used to stop the previous floats before each new item.


```
li {
    border: 1px solid #000;
    clear: left;
}
```