1 Floating Elements
The CSS `float` property (reference)

```css
img.headericon {
  float: right; width: 130px;
}
```

Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>side to hover on; can be left, right, or none (default)</td>
</tr>
</tbody>
</table>

- removed from normal document flow; underlying text wraps around as necessary
Floating elements diagram
Common float bug: missing width

- Often floating block elements must have a width property value
- Let’s try “floating”
Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.
### The `clear` property (cont.)

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>disallows floating elements from overlapping this element; can be left, right, or none (default)</td>
</tr>
</tbody>
</table>
Clear diagram

div#sidebar { float: right; }
p { clear: right; }
Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.
Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.
### The *overflow* property (cont.)

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overflow</td>
<td>specifies what to do if an element's content is too large; can be auto, visible, hidden, or scroll</td>
</tr>
</tbody>
</table>
Multi-column layouts

```html
<div>
    <p>first paragraph</p>
    <p>second paragraph</p>
    <p>third paragraph</p>
    Some other text that is important
</div>
```

```css
p { float: right; width: 25%; margin: 0.5em; border: 2px solid black; }
div { border: 3px dotted green; overflow: hidden; }
```

output:

```
Some other text that is important

first paragraph  second paragraph  third paragraph
```
Sizing and Positioning
The position property (examples)

div#ad {
    position: fixed;
    right: 10%;
    top: 45%;
}

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>position</td>
<td>static</td>
<td>default position</td>
</tr>
<tr>
<td></td>
<td>relative</td>
<td>offset from its normal static position</td>
</tr>
<tr>
<td></td>
<td>absolute</td>
<td>a fixed position within its containing element</td>
</tr>
<tr>
<td></td>
<td>fixed</td>
<td>a fixed position within the browser window</td>
</tr>
<tr>
<td>top, bottom, left, right</td>
<td>positions of box's corners</td>
<td></td>
</tr>
</tbody>
</table>
Absolute positioning

```css
#sidebar {
  position: absolute;
  left: 400px;
  top: 50px;
}
```

- removed from normal flow
- positioned relative to the block element containing them
- actual position determined by `top`, `bottom`, `left`, `right`
- should often specify a `width` property as well
Relative positioning

```css
#area2 { position: relative; }
```

- absolute-positioned elements are normally positioned at an offset from the corner of the overall web page.
- to make the absolute element to position itself relative to some other element's corner, wrap the absolute element in an element whose position is relative.
Fixed positioning

- removed from normal flow
- positioned relative to the browser window even when the user scrolls; the window, element will remain in the same place

CSS

```css
#area2 { position: relative; }
```
Alignment vs. float vs. position

1. If possible, lay out an element by \textit{aligning} its content
   - \textbf{horizontal alignment: text-align}
     - set this on a block element; it aligns the content within it (not the block element itself)
   - \textbf{vertical alignment: vertical-align}
     - set this on an inline element, and it aligns it vertically within its containing element

2. If alignment won't work, try \textit{floating} the element

3. If floating won't work, try \textit{positioning} the element
Details about inline boxes

- Size properties (width, height, min-width, etc.) are ignored for inline boxes.
- margin-top and margin-bottom are ignored,
- but margin-left and margin-right are not ignored.
Details about inline boxes

- the containing block box's `text-align` property controls horizontal position of inline boxes within it
  - `text-align` does not align block boxes within the page
- each inline box's `vertical-align` property aligns it vertically within its block box
The vertical-align property

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical-align</td>
<td>specifies where an inline element should be aligned vertically, with respect to other content on the same line within its block element's box</td>
</tr>
</tbody>
</table>

- can be top, middle, bottom, baseline (default), sub, super, text-top, text-bottom, or a length value or %
- baseline means aligned with bottom hanging letters
Don't be sad! Turn that frown upside down! Smiling burns calories, you know. Anyway, look at this cute puppy; isn't he adorable! So cheer up, and have a nice day. The End.

HTML
vertical-align example
(cont.)

Don't be sad! Turn that frown upside down!

Smiling burns calories, you know.

Anyway, look at this cute puppy; isn't he adorable! So cheer up, and have a nice day. The End.
Common bug: space under image

```html
<p style="background-color: red; padding: 0px; margin: 0px">
<img src="images/smiley.png" alt="smile" />
</p>
```

- red space under the image, despite padding and margin of 0
- this is because the image is vertically aligned to the baseline of the paragraph (not the same as the bottom)
- setting vertical-align to bottom fixes the problem (so does setting line-height to 0px)
The **display** property

```css
h2 { display: inline; background-color: yellow; }
```

This is a heading | This is another heading

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>display</td>
<td>sets the type of CSS box model an element is displayed with</td>
</tr>
</tbody>
</table>

- values: none, inline, block, run-in, compact, ...

- use sparingly, because it can radically alter the page layout
The `display` property (cont.)

```css
.p.secret {
    visibility: hidden;
}
```

- hidden elements will still take up space onscreen, but will not be shown
  - to make it not take up any space, set display to none instead
- can be used to show/hide dynamic HTML content on the page in response to events
The display property

- lists and other block elements can be displayed inline
- flow left-to-right on same line
- width is determined by content