Floating Elements

The CSS `float` property (reference)

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

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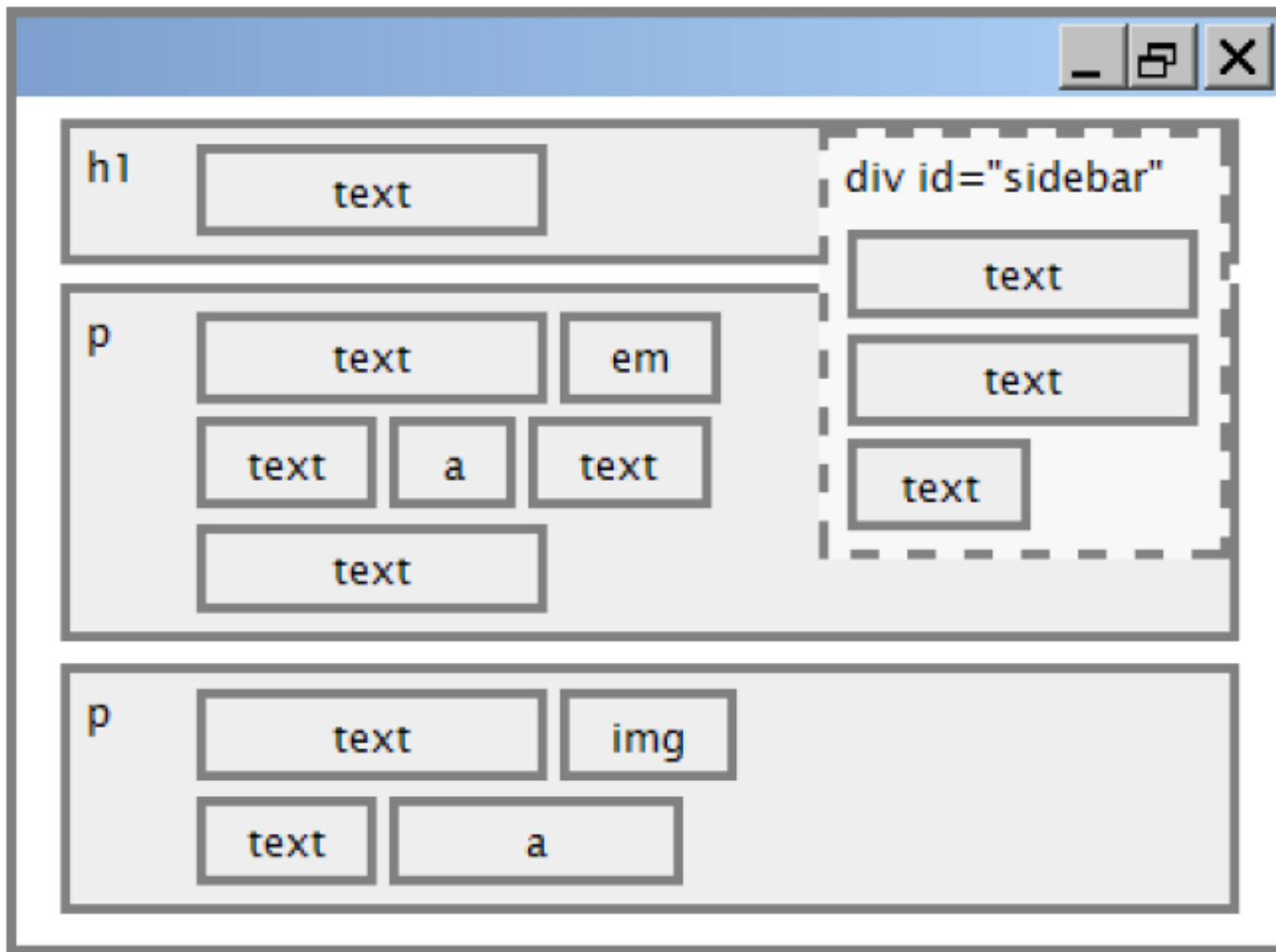parapsychologists-turned-ghost capturers.

output

property	description
float	side to hover on; can be left, right, or none (default)

- removed from normal document flow; underlying text wraps around as necessary

Floating elements diagram

h1

text

div id="sidebar"

text

text

p

text

em

text

a

text

text

text

p

text

img

text

a

Common `float` bug: missing width

I am not floating, no width

I am floating right, no width

I am not floating, 45% width

I am floating right, 45% width

- □ often floating block elements must have a width property value
- □ Let's try "floating"

CSC443: Web Programming

The `clear` property

```
p { background-color: fuchsia; }
h2 { clear: right; background-color: yellow; }
```
CSS

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

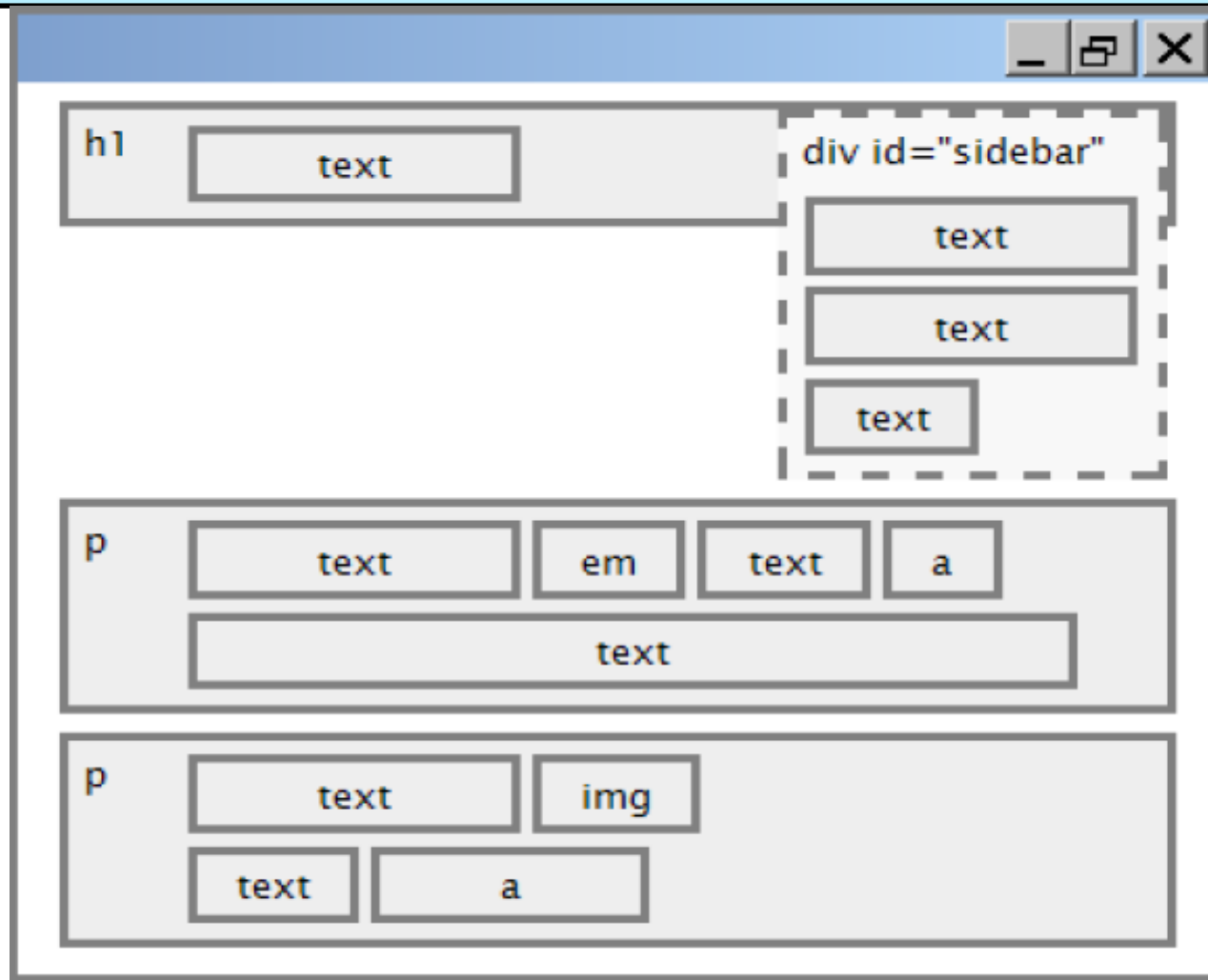**Super Mario Fan Site!**

output

The `clear` property (cont.)

property	description
clear	disallows floating elements from overlapping this element; can be left, right, or none (default)

Clear diagram

```css
div#sidebar { float: right; }
p { clear: right; }
```
CSS

Common error: container too short

```
<p><img src="images/mario.png" alt="super mario" />
Mario is a fictional character in his video game
series.....</p>
```
HTML

```
p { border: 2px dashed black; }
img { float: right; }
```
CSS

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.

CSC443: Web Programming

output

The `overflow` property

```css
p { border: 2px dashed black;
overflow: hidden; }                                          CSS
```

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.

output

CSC443: Web Programming

The `overflow` property (cont.)

property	description
overflow	specifies what to do if an element's content is too large; can be auto, visible, hidden, or scroll

Multi-column layouts

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

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

Some other text that is important

| third paragraph | second paragraph | first paragraph |

output

CSC443: Web Programming

Sizing and Positioning

12

The position property (examples)

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

property	value	description
position	static	default position
	relative	offset from its normal static position
	absolute	a fixed position within its containing element
	fixed	a fixed position within the browser window
top, bottom, left, right		positions of box's corners

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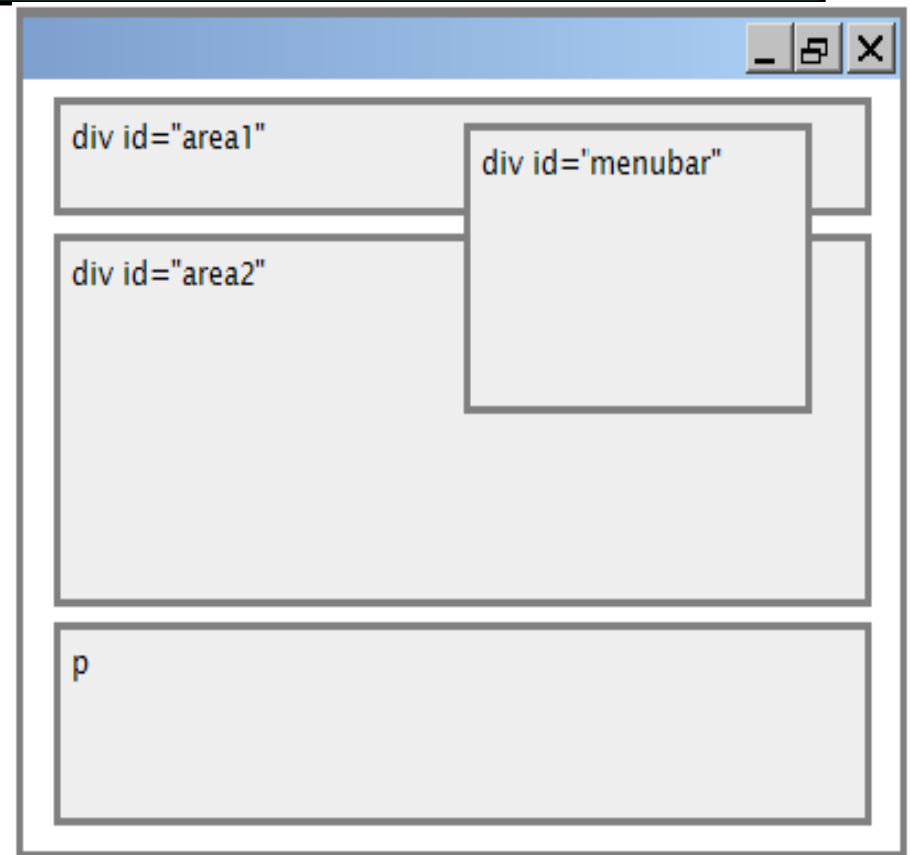
```css
#sidebar {
position: absolute;
left: 400px;
top: 50px;
}                                                    CSS
```

- □ 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`
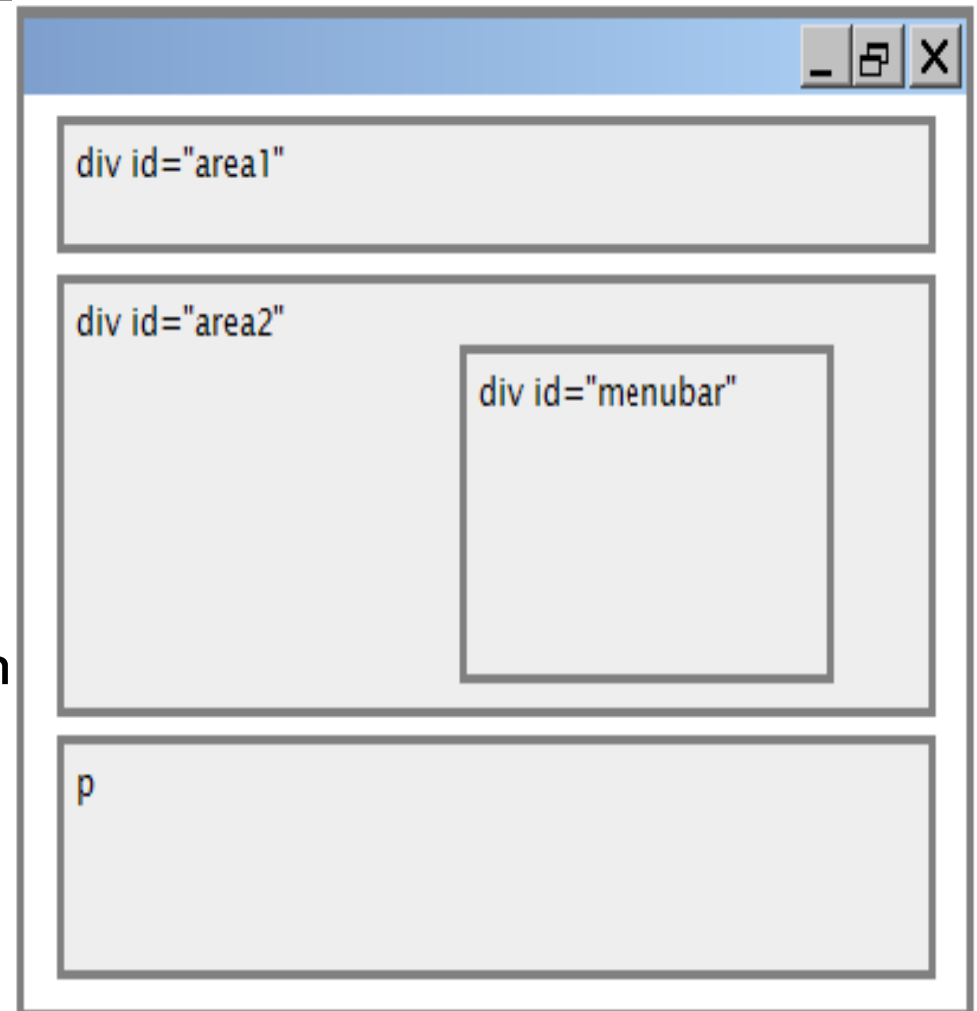 property as well

CSC443: Web Programming

Relative positioning

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

□ 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
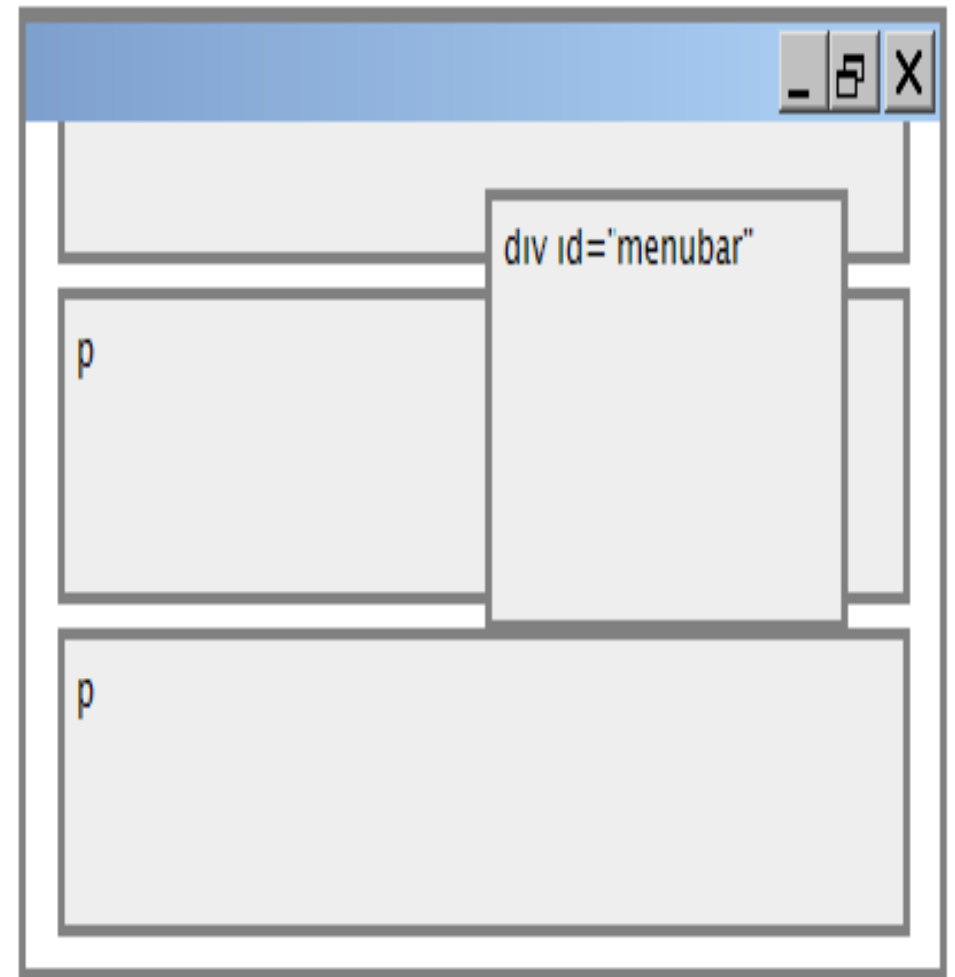
div id="area1"

div id="area2"

div id="menubar"

p

Fixed positioning

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

- □ removed from normal flow
- □ positioned relative to the browser

  window even when the user scrolls the window, element will remain in the same place

div id='menubar'

p

p

CSC443: Web Programming

Alignment vs. float vs. position

1. If possible, lay out an element by *aligning* its content

   - horizontal alignment: text-align
     - set this on a block element; it aligns the content within it (not the block element itself)
   - 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 *floating* the element

3. If floating won't work, try *positioning* the element

   - absolute/fixed positioning are a last resort and should not be overused

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**
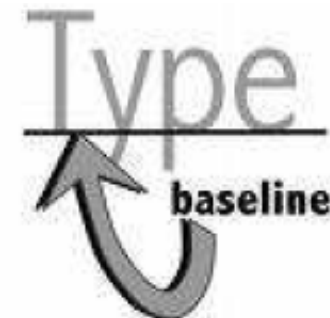
Details about inline boxes

- 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

property	description
vertical-align	specifies where an inline element should be aligned vertically, with respect to other content on the same line within its block element's box

- can be top, middle, bottom, baseline (default), sub, super, text-top, text-bottom, or a length value or %
    - baseline means aligned with bottom of non-hanging letters

vertical-align **example**

```html
<p style="background-color: yellow;">
    <span style="vertical-align: top; border: 1px solid
red;">
    Don't be sad! Turn that frown
    <img src="images/sad.jpg" alt="sad" /> upside down!
    <img style="vertical-align: bottom"
src="images/smiley.jpg" alt="smile" />
    Smiling burns calories, you know.
    <img style="vertical-align: middle"
src="images/puppy.jpg" alt="puppy" /> Anyway, look at this
cute puppy; isn't he adorable! So cheer up, and have a
nice day. The End.
    </span>
</p>
```
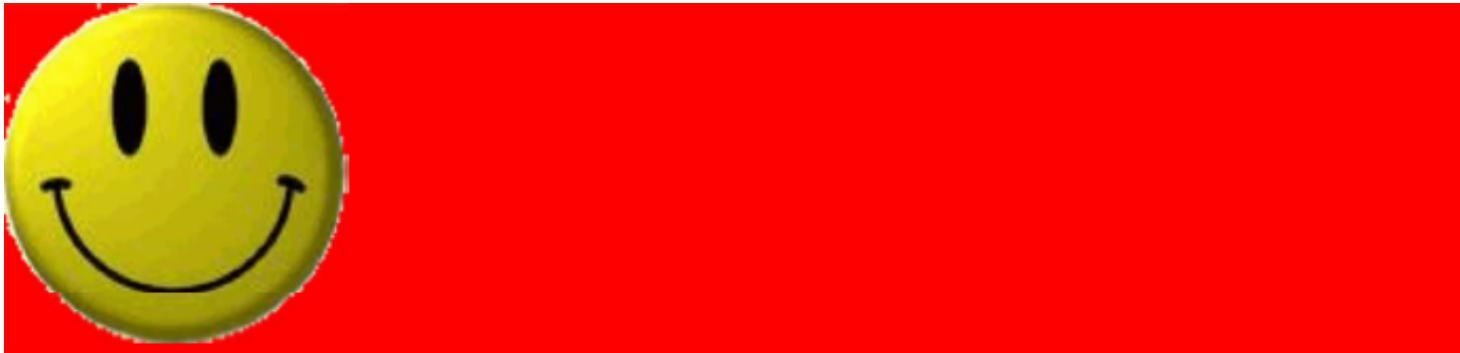*HTML*

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.

output

Common bug: space under image

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


□ 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

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

This is a heading **This is another heading**

output

property	description
display	sets the type of CSS box model an element is displayed with

- **values:** `none, inline, block, run-in, compact,` …

- use sparingly, because it can radically alter the page layout

CSC443: Web Programming

The `display` property (cont.)

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

output

- 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

CSC443: Web Programming

The display property

```html
<ul id="topmenu">
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
</ul>
```
HTML

```css
#topmenu li {
display: inline;
border: 2px solid gray;
margin-right: 1em;
}
```
CSS

Item 1   Item 2   Item 3

output

- lists and other block elements can be displayed inline

- flow left-to-right on same line

- width is determined by content