Education is what survives when what has been learned has been forgotten.

B. F. Skinner
Floating Elements
Announcements

- BetaWeb is up
  - You should have username/initial password email
  - You are a member of your Scrum Team’s Unix group
  - Git class was this morning

- Assignment 1 is Due February 6, midnight
  - The Scrum Master emails a pointer to the web site to martin@cs.rochester.edu
  - Code should is stored in Git repository on Betaweb
  - Recommendation: submit assignment in Lab Class tomorrow
Due Dates

- I have given a week to submit Assignment 1 because this is traditional.

- In the future, I will give much less time
  - You should always have working code.
  - Turn in whatever works.
  - Turn ins are check points—I want to see how far the teams have gotten; they are not necessarily final submissions.
<table>
<thead>
<tr>
<th>Team</th>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backslash</td>
<td>ANDREW</td>
<td>NOCKA</td>
</tr>
<tr>
<td>C.O.D.E.</td>
<td>THADDEUS</td>
<td>SPENCER</td>
</tr>
<tr>
<td>Cellar</td>
<td>RAGHAV</td>
<td>MEHTA</td>
</tr>
<tr>
<td>ContraWeb</td>
<td>KEDARNATH</td>
<td>SHASHIDHAR</td>
</tr>
<tr>
<td>Hacklemore</td>
<td>EMILY</td>
<td>ANSLEY</td>
</tr>
<tr>
<td>Lanister</td>
<td>EDWARD</td>
<td>BARTHELEMY</td>
</tr>
<tr>
<td>Llama</td>
<td>CHARLES</td>
<td>HENICK</td>
</tr>
<tr>
<td>Scrum City</td>
<td>CHARLES</td>
<td>KELMAN</td>
</tr>
<tr>
<td>Sk3m Team</td>
<td>MARCEL</td>
<td>PAWLOWSKI</td>
</tr>
<tr>
<td>SqlThePrql</td>
<td>THOMAS</td>
<td>TAVOLARA</td>
</tr>
<tr>
<td>Synapps</td>
<td>JENNIFER</td>
<td>UVINA</td>
</tr>
<tr>
<td>Tautology</td>
<td>HUA</td>
<td>ZHOU</td>
</tr>
<tr>
<td>Team RNG</td>
<td>MOHAMAD</td>
<td>SYED AHMED</td>
</tr>
</tbody>
</table>
Monday will be Demo day

Each team will show what they have done so far to the TA and/or me.

So far, we have a backlog of:
- Assignment 0
- Assignment 1
- Assignment 2

I have moved Assignment 3 to Thursday to give us time to make sure that Git is working
Git

- We have set up a Git server in the CS department
  - Betaweb.ugcs.rochester.edu
  - Each person in the class should have an account on this server.
  - Each team is a Unix group, so you can set up you Git permissions so that only those in the group can see it.
  - Each team should have a member who has attended a tutorial this morning on setting up Git for the team.
  - Getting this right is the goal for the Lab on Tuesday.
Standup

- What did you do?
- What will you do today?
- Barriers?
Quiz

- Should you use Class or ID for the following?
  1. Styling a single paragraph.
  2. Important items in a list should be red.

- HTML Code
  3. Write the line of HTML code to attach mystyle.css to your pages.

- Should you use “div” or “span”
  4. Red title of movie at start of each paragraph
  5. Section of the page that contains instructions in bold
Quiz (Answer)

□ Should you use Class or ID for the following?

1. Styling a single paragraph. (id)
2. Important items in a list should be red. (class)

□ HTML Code

3. Write the line of HTML code to attach mystyle.css to your pages. (<link rel="stylesheet" href="mystyle.css" type="text/css"/>)

□ Should you use “div” or “span”

4. Red title of movie at start of each paragraph (<span>)

CSC 210 Section of the page that contains instructions in bold (div)
Floating Elements
The CSS `float` property

```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”
The `clear` property

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

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!
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; }
```

Some other text that is important

- third paragraph
- second paragraph
- first paragraph

output
Sizing and Positioning
The position property (examples)

```css
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

CSC 210
Absolute-positioned elements are normally positioned at an offset from the corner of the overall web page. To make the absolute element 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
Alignment vs. float vs. position

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

<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 of non-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

- 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

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

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

- Lists and other block elements can be displayed inline.
- Flow left-to-right on same line.
- Width is determined by content.