CSS: Inheritance, Conflict, Positioning and More

CISC 282

January 23, 2014

Inheritance

- Nested elements inherit styles from parent(s)
  - Styles carry downwards

```html
<p>
  text-align: right;
</p>

<ul>
  color: red;
</ul>

<em>
  text-decoration: underline;
</em>

<p>
  Paragraph text
  <ul>
    <li>List text</li>
    <li>
      <em>Em text</em>
    </li>
  </ul>
</p>
```
Conflict

- An element may be subject to many styles
  - Directly applied
  - Inherited

- Some styles may conflict
  - Same properties
  - Different values

```css
p {
    color: red;
}

em {
    color: blue;
}

<p>
    Text and <em>more text</em>
</p>
```

Types of Conflict

- Between parent elements
  - Closest in hierarchy wins

- Between application to same element
  - Most specific wins
    - ID, class, tag

- Between definitions for same selector
  - Last one defined wins
    - In same file
    - In different files

- A mix of the above
  - Generally, closest & most specific
Shorthands

- Properties that set several other properties

```css
p.reallyEmphasize {
  font: italic bold 14pt sans-serif;
}
```

```css
.lightlyOutlined {
  border: thin groove gray
}
```

Internet Explorer 6

- Shipped with Windows XP
  - Was widely used
  - Poorly interprets some CSS properties
    - Particularly with respect to layout

- Box model standard
  - `width` and `height` applied to content

- IE6 interpretation of box model
  - `width` and `height` applied to content **and** padding
  - **no support for** mid-width, max-height, ...

- There are also issues with floats
  - These persist in IE7
Centering

- Use `text-align` for inline content
  - Text, images, ...
- Can use `margin` for blocks
  - Must assign a width
    - This approach doesn't work otherwise
  - Set `margin-left: auto;`
  - Set `margin-right: auto;`

Additional Layout Tips

- Some unit values may be negative (e.g., -50px)
  - Can create overlapping or off-page effects
    - Apply to margin(s) or position(s)
- reset.css
  - Sets basic values for tags
    - `line-height: 1.0, margin: 0, etc`
  - Goal is to eliminate cross-browser inconsistencies
  - Must be the first stylesheet loaded
Positioning

- Removes element from standard flow
  - Precisely specifies position ... 
  - ... with respect to an element or the page

- **position**
  - Sets the orientation of an element
  - **Value**: static, relative, absolute, fixed
    - static is the default

- **top, bottom, left, right**
  - Specifies the offset of an element ...
  - ... with respect to another element or the page
  - Somewhat like setting a direction-oriented margin
  - **Value**: Unit value or %

---

Positioning

- Relative positioning
  - With respect to default (static) position

- Absolute positioning
  - With respect to containing block element ...
  - ... if block also uses absolute or relative
    - If not, defaults to the page

- Fixed positioning
  - With respect to the page
  - **Value**: Unit value or %