Cascading Style Sheets, CSS

BWA Slides #4

Cascading Style Sheets, CSS

Language for describing the presentation semantics of markup documents

- To produce a (nice) 2D view of DOM trees (XML, HTML, XHTML, ...)
- Possible to set styles (fonts, colors, backgrounds,...), easy ...
- Possible to do layout (positioning), complex ...

Cascade

- There can be many CSSs involved: author, user, user agent (browser), may overlap !!
- The cascade resolves the interaction
- Command "!important" can change precedence

Why CSS?

Prior to CSS, nearly all of the presentational attributes of HTML documents were contained within the HTML markup
- Duplicate code
- Unmaintainable

CSS allows authors to move much of that information to a separate style sheet resulting in considerably simpler HTML markup

CSS has had severe problems (getting better)
- Browser incompatibility
- Bad specifications
- Bugs

CSS Specification

W3C CSS specifications (enormous collection of specifications …)

- "… breaks the specification into more manageable chunks and allows more immediate, incremental improvement to CSS."
- Not all stable
- We say we use CSS3, a very nice snapshot

CSS Language Basics

CSS program in *.css file

Basically
- Program contains a list of statements. We say **rules** (simplified)
- A rule has a **selector**, which element(s) to apply rule to
- Rule has block of **properties** with values (fonts, colors, width, z-index, …)
- White space may surround statements
- Case insensitive
- Illegal parts ignored (no result in Browser, check spelling)!
- Comments: /* */ (not nested)
- Inheritance (not like OO)

CSS Rule and Selectors

Rule ::= Selector followed by Declaration block

```
/* A rule */
h1 {                    /* Selector,block start */
    font-size: 34px;   /* Property name and value */
    font-weight: bold;
 }                      /* Block end */
```

Selector index

CSS Properties

Property and property values index

CSS Styles

Add rules for font, colors, borders, backgrounds etc. to the rules

- NetBeans have CSS style builder (had at least?)

NOTE: Graphical design and typography normally not a topic for programmers (don't pay too much attention)

CSS Box Model

All elements in HTML surrounded by a 2D box, the box model

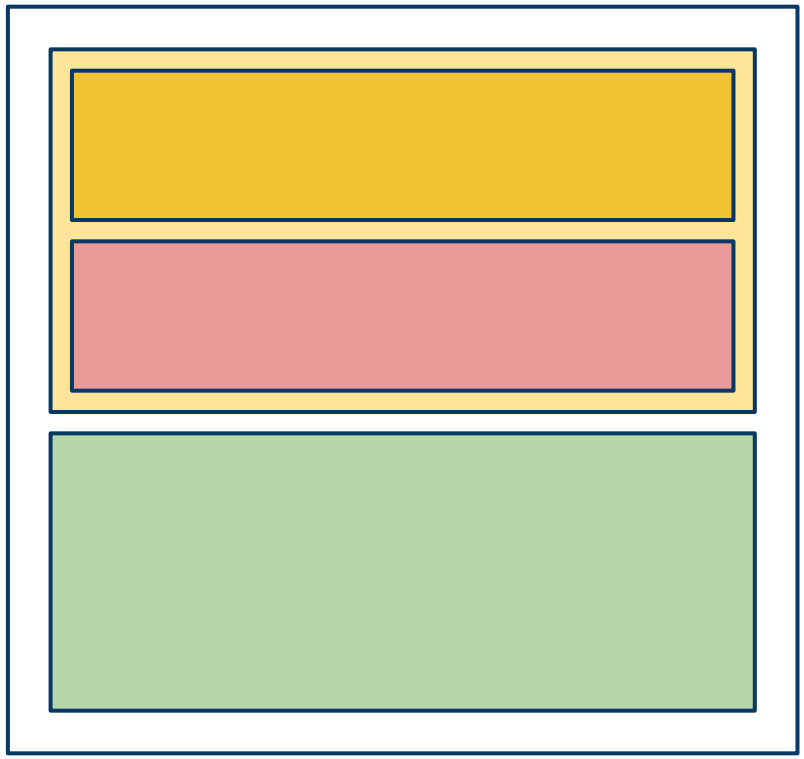
Basically

- Some elements have a **blockbox** i.e ('s)
  - Element that takes up the full width available
  - Has a line break before and after
- Others have an **inline box**; not full width, no breaks

Normal Flow Layout

If no CSS, browser uses normal flow and boxes for layout

```
<div id="white">
    <div id="yellow">
        <div id="orange">
        </div>
        <div id="red">
        </div>
    </div>
    <div id="green">
    </div>
</div>
```



CSS Positioning

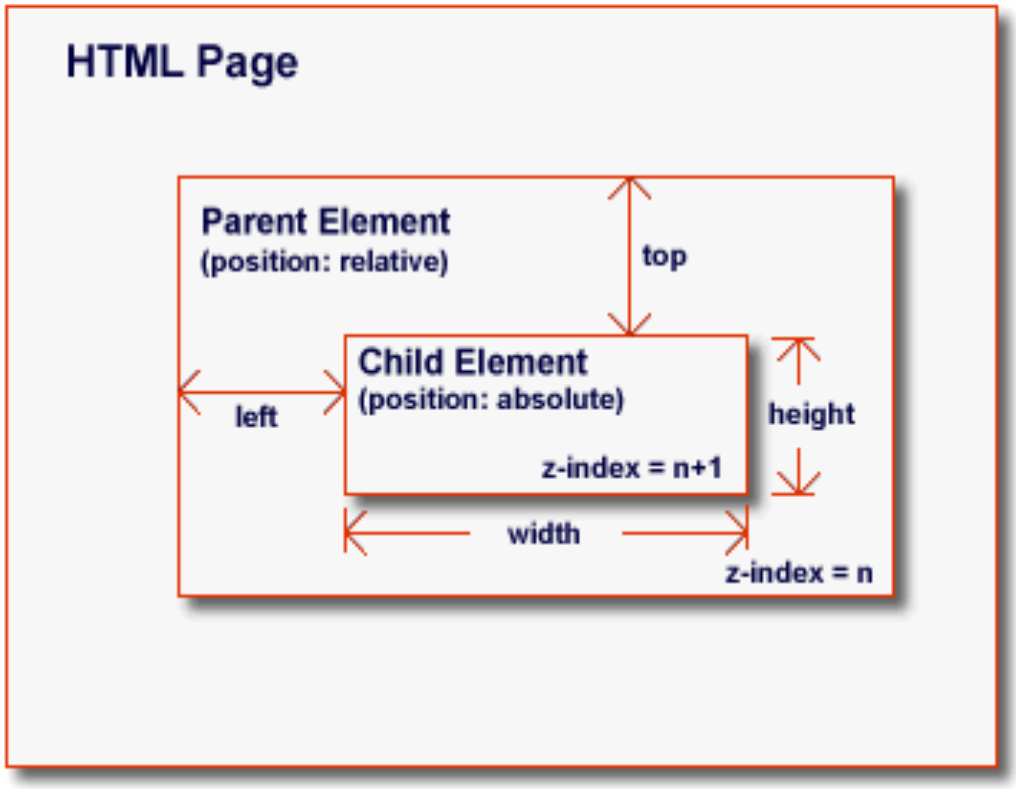
How boxes are laid out relative other (containing) boxes

- Relative, offset from normal flow
- Absolute positioning. Removed from normal flow, positioned relative first non-static positioned parent (if none, browser window)
- Fixed, fixed relative browser window
- Complex: Boxes (position, size) affects each other

Floats (floating boxes)

- Shifted to the left or right on the current line (if no room shifted downwards)
- Content flows down sides of float (prohibit with “clear”) Must have width set

CSS Positioning cont.



CSS Positioning

Adding CSS to HTML

Browser will download (and cache) and apply style sheet

HTML/XHTML

```
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

XML

```
<?xml-stylesheet type="text/css" href="mystyle.css"?>
```

CSS Advanced

Previously need of JavaScript replaced by advanced CSS3 features
- 2D/3D transformation

Modularization
- Possible to break up style sheets and import

```
<!-- In HTML -->
<link rel="stylesheet" type="text/css" href="/css/styles.css" />

// In styles.css (first in file)
@import url('/css/typography.css');
@import url('/css/layout.css');
@import url('/css/color.css');
```

Responsive Web Design

Hard to know on with devise the page will be displayed (size, etc). Using media queries
we can tailor different CSS to match different devices

A media query consists of a media type and zero or more expressions that check for the conditions of particular media features

Example Media Queries

```
<!-- Use in HTML, if device passes width test then
use shetland.css else skip -->
<link rel="stylesheet" type="text/css"
  media="screen and (max-device-width: 480px)"
  href="shetland.css" />

// Use in CSS
@media screen and (max-device-width: 480px) {
  .column {
    float: none;
  }
}

// Use in import in CSS
@import url("shetland.css") screen and (max-device-width: 480px);
```

Bootstrap

To avoid complexity of CSS better go higher level

Bootstrap is a HTML, CSS, and JavaScript framework
- By Twitter
- Reusable components
- Have to download some CSS and JavaScript files and add to your application
- Add Bootstrap defined values to the class attribute
- Add bootstrap.css to HTML