

CSC309 *Programming on the Web*

week 3: css, rwd

Amir H. Chinaei, Spring 2017

Office Hours: M 3:45-5:45 BA4222

ahchinaei@cs.toronto.edu

<http://www.cs.toronto.edu/~ahchinaei/>

survey 1

- ❖ in survey 1, you provide us with:
 - your UTorID,
 - your GitHub username, and
 - your familiarity with technologies related to this course
- ❖ before completing the survey
 - make sure you have a GitHub username,
 - if you don't, sign up here: <https://github.com/join>
 - and, get a student developer pack here: <https://education.github.com/pack>
- ❖ if you have the GitHub username and UTorID
 - complete the survey here: <https://goo.gl/forms/1sovPcFA1dLEo1k42>
(deadline: Jan 09)

review

❖ **design tip**

- separate **semantics** from **appearance**
- in developing html documents:
 - focus on structural **semantics**
 - not on appearance

❖ **advantages**

- faster development
- maintainability due to modularity
- use semantics elements in html5
- Use html validator

cascading style sheets

- **general syntax:**

selector {properties; }

- **example:**

h2, p { color: blue; }

h2 { font-size: 1.5em;
background-color: yellow }

- cascading term?

CSS

- ❖ it's the **w3c** standard
- ❖ to describe the **appearance** of information in a document (web page) defined by html elements
 - **appearance?** colors, backgrounds, fonts, layouts, borders, etc.
- ❖ history
 - **css** (1996), **css2.1** (2004), **css3** (2012)
- ❖ responsive **w**eb **d**esign
 - design features that keep appearance appealing on different devices

units

❖ **absolute length**

- good for **print**
- not recommended for screen
- common examples:
 - cm, mm, px, pt, in (=2.54cm =96px =72pt)

❖ **relative length**

- recommended for **screen** (rwd)
- common examples:
 - em, rem, %, vw, vh, vmin, vmax

designer styles

external style

embedded style

inline style

selectors

- ❖ wild card *
- ❖ **single**
- ❖ **group**
- ❖ **class**
 - .main { font-style: italic; color: red;}
- ❖ **id**
 - #123 { font-style: italic; color: red;}
- ❖ **attribute**
 - a[target] {background-color: yellow;}

selectors

❖ **pseudo classes**

- `a:hover {color: pink;}`

❖ **pseudo elements**

- `p::first-letter {
 color: #ff0000;
 font-size: xx-large;
}`

❖ **combinators**

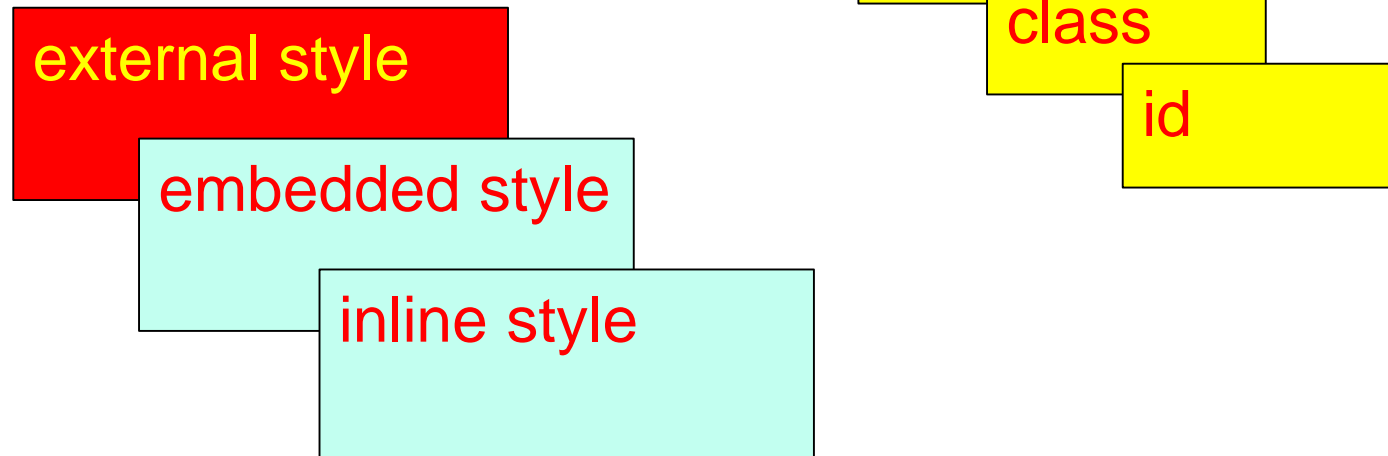
- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

cascading principles

1. inheritance

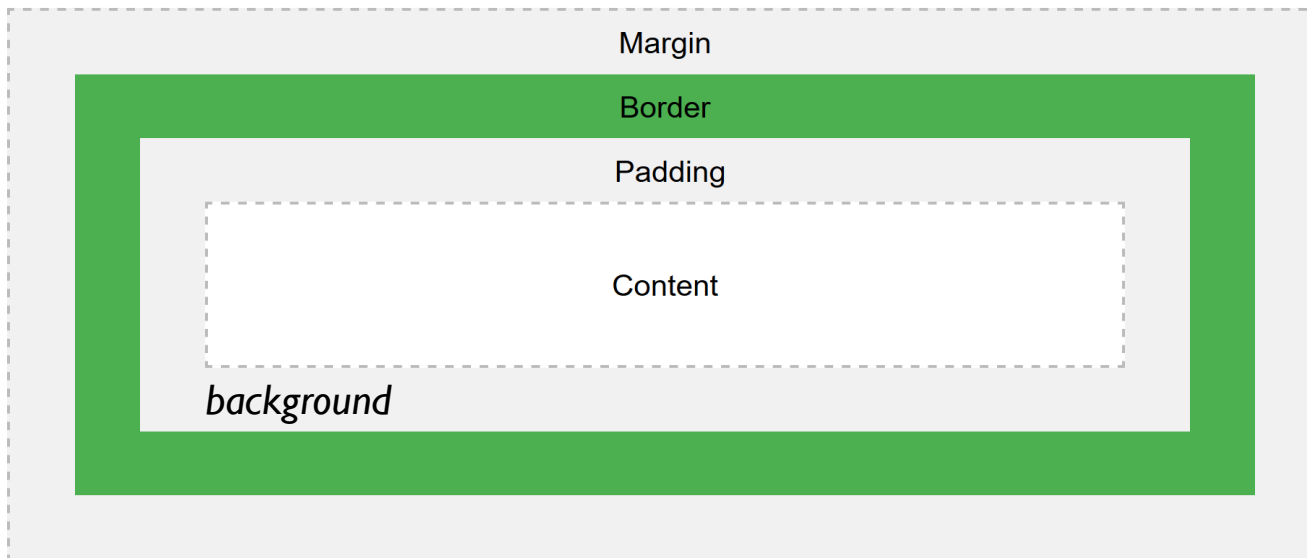
- font, color, list, and text are inherited
- border, layout, margin, padding are not, unless specified

2. specificity



the box model

- ❖ all html elements can be considered as boxes



box border

❖ border

- border-style

- solid, dotted, dashed, double, groove, ridge, inset, outset

- border-width

- border-color

- border-top-color: red
- border-right-color: blue
- ...

❖ border-radius

❖ border-image

box background

- ❖ background
- ❖ background-attachment: *scroll* | *fixed*
- ❖ background-color
- ❖ background-image
- ❖ background-position
- ❖ background-repeat: *repeat* | *repeat-x* | *repeat-y* | *no-repeat*
- ❖ background-size:

design tips

❖ for rwd:

- use relative length for *width* and *height*
- use *max-width* and *min-width*
- use *border-box* for *box-sizing*
- Use *float*

design tips

❖ **browser dev tools**

- firefox: right-click → inspect element

❖ **css validator**

- w3c jigsaw css validator
<https://jigsaw.w3.org/css-validator/>

styling the text

❖ **font styling**

- font-style, font-weight, font-variant
- font-stretch, font-size

❖ **paragraph styling**

- font-family
 - p {font-family: Calibri, Arial, Sans-Serif;}
- generic font families
 - **Serif, Sans-Serif, Monospace, Cursive, Fantasy**

@font-face

- @font-face { font-family: myFirstFont;
src: url(sansation_light.woff);}

layout

❖ **normal**

▪ **block elements**

- ...

▪ **inline elements**

- ...

▪ **note:** the display property can change this

- `ul {display: inline;}`

layout

❖ **position property**

- absolute | relative | fixed | static
 - figure {...; position: **relative**; top:10px; left:20px;}
 - figcaption {...; position: **absolute**; top:115px; left:25px;}

❖ **float property**

- left, right, both, none
 - figure {...; width: 150px; float: **right**;}
- ...

layout

❖ **fixed layout**

- absolute size/length (px)

❖ **fluid layout**

- relative size/length (%)

❖ **hybrid**

- some elements fixed, some fluid

rwd

❖ responsive web design

1. setting viewport

<meta

name="viewport" content="width=device-width, initial-scale=1.0">

2. sizes relative to viewport (fluid elements)

img {max-width: 100%;}

3. grid design

what print designers do

4. media queries

@media

@media

❖ **define breakpoints**

- phone (small devices) first
- tablet (medium devices) next
- desktop (large devices) last

note: this is called **progressive enhancement**

❖ **example:**

{...}

@media only screen and (min-width: 600px) {...}

@media only screen and (min-width: 768px) {...}

@media

❖ examples

BP1: small devices:

```
[class*="col-"] { width: 100%;}
```

BP2: medium devices

```
@media only screen and (min-width: 600px) {  
    /* for tablets*/  
    col-m-1 {width: 8.33%;}  
    col-m-2 {width: 16.66%;}  
    col-m-3 {width: 25%;}  
    ...  
    col-m-12 {width: 100%;}  
}
```

@media

❖ examples

BP3: large devices

```
@media only screen and (min-width: 768px) {  
    /* for desktop*/  
    col-1 {width: 8.33%;}  
    col-2 {width: 16.66%;}  
    col-3 {width: 25%;}  
    ...  
    col-12 {width: 100%;}  
}
```

more design tips

- ❖ use a CSS framework
 - such as a grid frameworks (e.g. bootstrap)
- ❖ then customize the styles for your needs

- ❖ use a CSS template
 - such as *art, food, fashion*, etc.
- ❖ then customize the styles for your needs