CSS

Cascading Style Sheets

Originally prepared by Prof. Engelsma

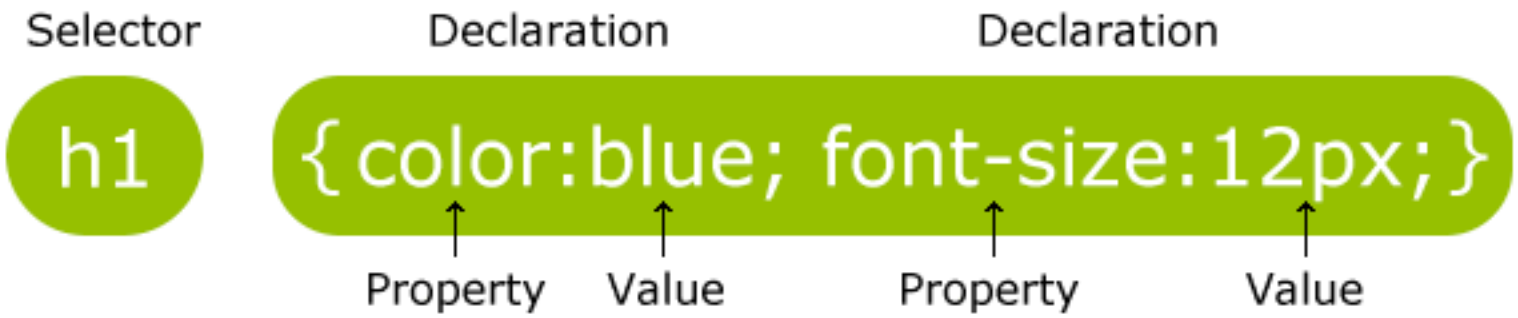
History

- HTML originally for structure only and did not have any format tags.
  - HTML 3.2 added a bunch of formatting tags, such as <font>. They had to be added to every page and became a big nightmare from a maintenance perspective.
- W3C created CSS to respond to the problem, and in HTML 4.0 all format tags were removed. Format info is now included in CSS.

Introduction

- CSS makes HTML pretty
- Three ways to attach style to HTML
  1. Use the `style` attribute: `<p style="color: blue">...`
  2. Use an *internal* style sheet defined in the HTML <HEAD> element:
     ```
     <style>
     p { color: blue; }
     </style>
     ```
     type="text/css" is no longer necessary.
  3. Use an *external* style sheet: `<link rel="stylesheet" type="text/css" href="someURL" />`

CSS Syntax



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

CSS Format

- A *stylesheet* is a collection of styles
- A style is a *selector* followed by a list of properties inside curly braces separated with semicolons
- Uses C-style comments
- Simple case: selector is the name of an HTML element

```
/* Style for the body */
body { font-size: 0.8em; color: green; }
```

What does “Cascading” mean?

- If multiple style specification approaches are used (inside HTML element, inline in <head> or external CSS file) all styles cascade into one.
- Cascading order:
  1. Browser default
  2. External style sheet
  3. Internal style sheet
  4. Inline style (in HTML element)

Increasing Priority!

CSS class vs. id

- In addition to using HTML elements as selectors, we can also use class and id.
- Id: use to specify a style for a unique element.
  - Uses the id attribute of the HTML element and defined with a #.
- Class: use to specify a style for a group of elements.
  - Uses the class attribute of the HTML element and is defined with a “.”

Class Selectors Example

- A *class* defines a style that can be applied to all elements with a matching `class` attribute
- Define a class selector:

```
.important { color: red; font-weight: bold; }
```

- Use it:

```
<p class="important">this is important</p>
```

Id Selector Example

- An *id* refers to exactly one element that has the matching `id` attribute
- Define an id selector:

```
#myheader { font-size: 24pt; font-style: italic; }
```

- Then use it:

```
<h1 id="myheader">Welcome!</h1>
```

Combining Styles

- The following style would only apply to important paragraphs:

```
p.important { ... }
```

- Apply common style to several selectors by separating them with commas

```
h2, h3, p { color: blue; }
```

- Apply style to nested elements by separating them with spaces

```
.important { color: red; }
.important p { font-size: 12pt; }
.important p b { font-style: italic; }
```

es

How can we specify the background color of each block without having to add a color class to each span?

```
<div class='row red'>
  <span class='block'>2</span>
  <span class='block'>3</span>
  …
</div>
<div class='row yellow'>
  <span class='block'>2</span>
  <span class='block'>3</span>
  …
</div>
```

```
.row {
    width: 800px;
    border-radius: 5px;
}

.red {
  background-color: red;
}

.yellow {
  background-color: gold;
}
```

es

How can we specify the background color of each block without having to add a color class to each span?

```
<div class='row red'>
  <span class='block'>2</span>
  <span class='block'>3</span>
  …
</div>
<div class='row yellow'>
  <span class='block'>2</span>
  <span class='block'>3</span>
  …
</div>
```

```
.row {
    width: 800px;
    border-radius: 5px;
}

…
.red .block {
   background-color: pink;
}

.yellow .block {
   background-color: white;
}
```

Selector Specificity Weights

- Each CSS selector has a specificity weight which along with its placement in the cascade identifies how styles will be rendered:
  - **Type Selectors:**
    - **(low) weight: 0-0-1**
  - **Class Selectors:**
    - **(medium) weight: 0-1-0**
  - **ID Selectors:**
    - **(high) weight: 1-0-0**
- Specificity weights are used to resolve styling conflicts.

Example

HTML

```
<p id="food">...</p>
```

CSS

```
#food {
  background: green;
}
p {
  background: orange;
}
```

ID selector (food) has higher weight than type selector, so text in paragraph will be green, not orange.

Example

HTML

```html
<div class="hotdog">
  <p>...</p>
  <p>...</p>
  <p class="mustard">...</p>
</div>
```

CSS

```css
.hotdog p {
  background: brown;
}
.hotdog p.mustard {
  background: yellow;
}
```

.hotdog p:
0-1-1
.hotdog p.mustard:
0-2-1
(first two paragraphs brown, the third yellow)

Colors

- Many ways to specify a color
  1. name, e.g., `blue`
  2. rgb(red, green, blue), e.g., `rgb(0%,0%,100%)`
  3. rgba(red,green,blue,alpha)
     e.g., `rgba(120,200,45,.25)`
  4. hexidecimal (two digits for red, green, and blue), e.g., `#0000ff`

Common Properties

- background-color
- background-image
- background-position
- background-repeat
- color
- font-family
- font-size
- font-style
- font-weight
- letter-spacing
- line-height
- text-align
- text-decoration
- text-indent
- text-transform
- word-spacing
- border

Borders

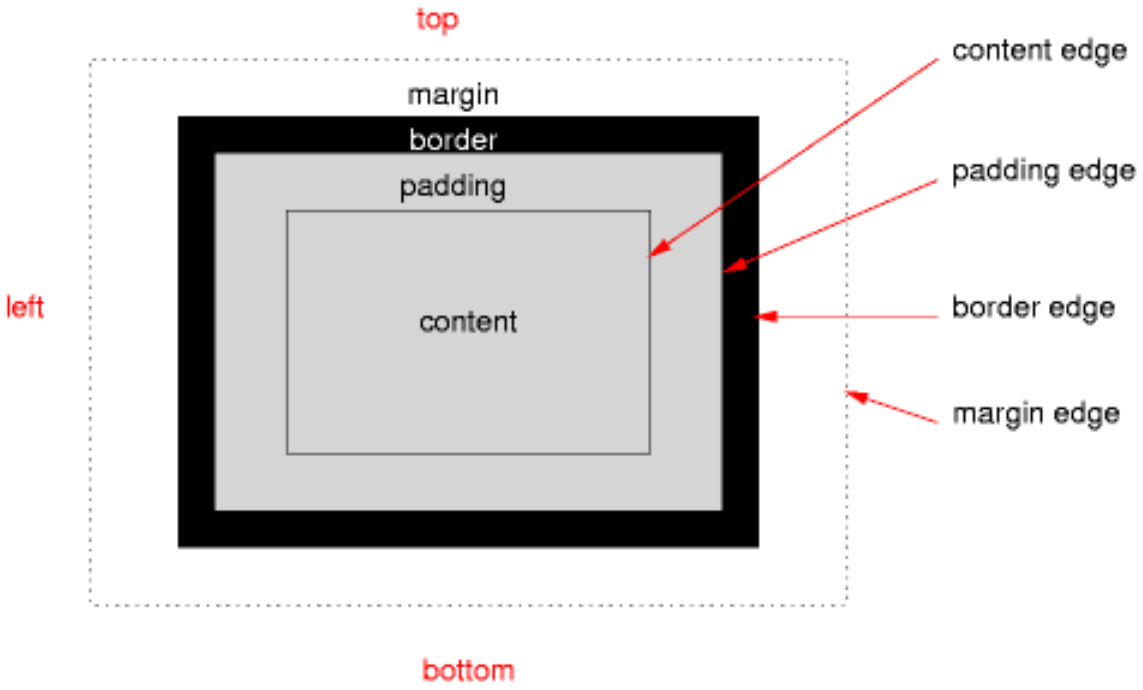
- border-style
- border-width
- border-color
- border

```
border: 1px blue solid;
```

Padding and Margins

- `margin` - sets all four margins (top, right, bottom, left)
- `padding` - sets all four padding lengths
- Can also use individual properties like `margin-left`

Borders, Margins, and Padding



From W3C

Width and Height of an Element

```
div.ex
{
        width:320px;
        padding:10px;
        border:5px solid gray;
        margin:2px;
}
```

- Any div element of class ex will be 354px wide!
  - 320px + 20px (left and right padding) + 10px (left and right margin) + 4px (left + right margin).
- `Width` only respected by `block` and `inline-block`

Display

- Block
  - Each element starts a new paragraph
- Inline
  - Width and height properties don’t apply
  - Margin and padding not used when determining place of next line.
- Inline-block
  - Like a block, but elements don’t begin a new line.

Display vs. visibility

- `visibility:hidden` keeps the space and shows nothing:
  - Element still affects layout
- `display:none` behaves as if element isn't in the DOM

Position

- `absolute` - uses the top, bottom, left, and right properties to place the element in an absolute position
- `relative` - offsets the element by a given amount. Again, using the properties top, bottom, left, right. (relative to normal)
- `static` - default positioning style, normal flow
- `fixed` - fixes the element in a position relative to the browser window, not the web page
  - sticky

```
#footer { position: fixed; bottom: 0; left: 0; }
```

CSS Units

- px = pixels
  - One pixel on the screen … kind of
  - Retina displays let you specify the “virtual” resolution
  - “Zooming” in the browser also creates a “virtual” resolution.
- em = units are relative to the current font size (1em)
- pt = points
- % = percentage of its container/parent element
- `calc(100% - 50px)`

Float

- `float` - "left" or "right"
- Moves the element to either the left or the right end of the line and flows content around it
- Use the `clear` ("left", "right", or "both") property to stop wrapping content from the given side(s)

```
#sidebar { float: right; }
```

SASS

- Syntactically Awesome Style Sheets
- Supports two syntaxes SASS and SCSS
  - Same engine, different inputs
- Preprocessor for CSS
  - (i.e., ‘compiles’ down to CSS)
  - Provides helpful features
    - variables
    - nesting
    - importing / mixin / extend
  - The above features can be done with CSS, but it is just more verbose.

SASS vs. SCSS

- SASS syntax
  - Older.
  - More terse
  - Indent/whitespace based
  - Similar to .haml
- SCSS syntax
  - Newer
  - Similar to CSS
  - Every CSS file is valid SCSS

Installing / Running SASS

- There are many ways to install/run SASS
  - https://sass-lang.com/install
- One common technique
  - `npm install sass`
  - `./node_modules/.bin/sass the_file.scss > the_file.css`

Flexbox

- https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Reading Assignment

- Complete a CSS Tutorial (as needed):
  - https://www.w3schools.com/css/default.asp
  - https://learn.shayhowe.com