CSS

COEN 161

Cascading Style Sheets

Cascading Style Sheets

- In the beginning…
  - No CSS
  - Style determined by markup
    - <h1> - Big header
    - <b> - Bold text
  - Markup gets interpreted by the browser
  - Developer can’t customize how the markup gets rendered in the browser

Cascading Style Sheets

- The Solution
  - HTML 3.2 added the tag

This is some text!

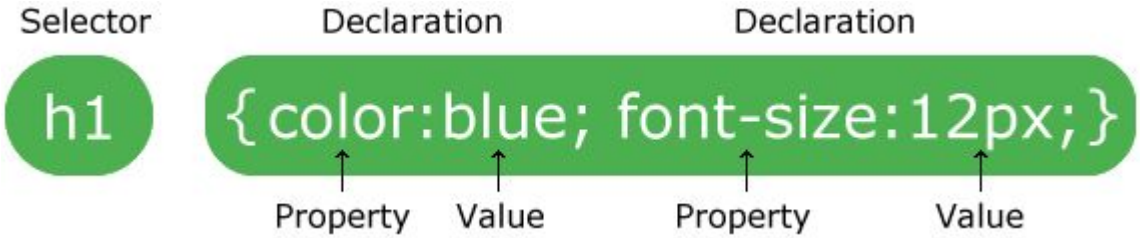
- However…
  - This was not the best solution
  - Adding fonts to large websites was a long process
  - Each page had its own fonts
  - Updating fonts meant changing each font tag’s attributes

Cascading Style Sheets

- The *Real* Solution...CSS!
  - Formatting information is separated from content
    - HTML defines the content and the structure of the page
    - CSS defines how to display the content
    - Attributes tie HTML + CSS
  - Customization!
    - Introduced a rich set of style attributes such as font and background-color
  - Reusability
    - CSS can be defined once and used across multiple pages, it can even be used across multiple sites
  - Inheritance
    - Some style attributes can be passed down from parent elements to their children

Syntax

- CSS consists of rule-sets
- Each rule-set consists of a ***selector*** and a ***declaration block***
- Each declaration contains a ***style attribute*** and a ***value***



Adding CSS

- 3 Ways to add styling
- Link an external .css file in the head of an HTML file

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

Adding CSS

- Writing css in the <style> tag in the head of the HTML file

```
<head>
      ...
      <style>
            body {
                  background-color: linen;
            }

            h1 {
                  color: maroon;
                  margin-left: 40px;
            }
      </style>
      ...
</head>
```

Adding CSS

- ***Inline*** css style using the ***style*** attribute in an HTML element

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

- Inline styles do not have a selector, just style declarations

Selectors

- Selectors are patterns that tell CSS which elements get styling applied

```
h1 {
      color: maroon;
      margin-left: 40px;
}
```

- Selectors usually have the HTML element which is getting styled
  - In the example above, all <h1> elements get the two styles applied

Types of Selectors

Selector	Example	Example description
element	p	Selects all <p> elements
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
[attribute]	input[type="text"]	Selects all <input> elements with the type attribute set to "text"
*	*	Selects all elements
element,element	div,p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div>p	Selects all <p> elements where the parent is a element
element+element	div+p	Selects all <p> elements that are placed immediately after elements
element.class	p.intro	Select all <p> elements with class=“intro”

Using Selectors: id

HTML

```
<body>
    <p id="para1">The first paragraph</p>
    <p>The second paragraph</p>
</body>
```

CSS

```
#para1 {
    color:red; /* make the first paragraph red */
}
```

Using Selectors: class

HTML

```
<body>

    <p>The first paragraph</p>

    <p class="important">The second paragraph</p>

    <p class="important">The third paragraph</p>

</body>
```

CSS

```
.important {

    /* make important
       things red */
    color:red;

}
```

Grouping Selectors

HTML

```
<body>

    <h1>A header</h1>

    <p id="para1">The first paragraph</p>

    <p class="important">The second paragraph</p>

</body>
```

CSS

```
h1 {
    color:red;
}

#para1 {
    color:red;
}

.important {
    color:red;
}
```

Grouping Selectors

HTML

```
<body>

    <h1>A header</h1>

    <p id="para1">The first paragraph</p>

    <p class="important">The second paragraph</p>

</body>
```

CSS

```
h1, #para1,  .important{
        color:red;
}
```

CSS Comments

- Start with /* and end with */

```
/* Comments can be
   Outside a rule-set */
p {
        /* they can be in a rule-set */
        color:red; /* they can be after a rule */
        /* ...or
           after */
}

/* .important { color:red; } */
```

Pseudo-classes

- Added to a selector to select an element that has a special state

Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:hover*	a:hover	Selects links on mouse over
:active**	a:active	Selects the active link
:focus	input:focus	Selects the input element which has focus
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:nth-child()	p:nth-child(2)	Selects every <p> element that is the second child of its parent

* :hover must come ***after*** :link and :visited
** :active must come ***after*** :hover

Using Pseudo-classes: nth-child()

- e:nth-child(n) selects the element, e, that is the ***nth*** child of its parent, regardless of type
- **n** can be a number, keyword, or formula

```
p:nth-child(2){background:red;}

p:nth-child(odd){background:red;} /* even is the other keyword */

p:nth-child(3n+1){background:red;} /* every p that is the
                                      1st, 4th, 7th,…,  child of its parent */
```

Pseudo-elements

- Used to select a specific part of an element

Selector	Example	Example description
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion in a p that has been selected by a user
::before	p::before	Insert content before every <p> element
::after	p::after	Insert content after every <p> element

Note: pseudo-elements use two colons instead of one

CSS Selector Practice

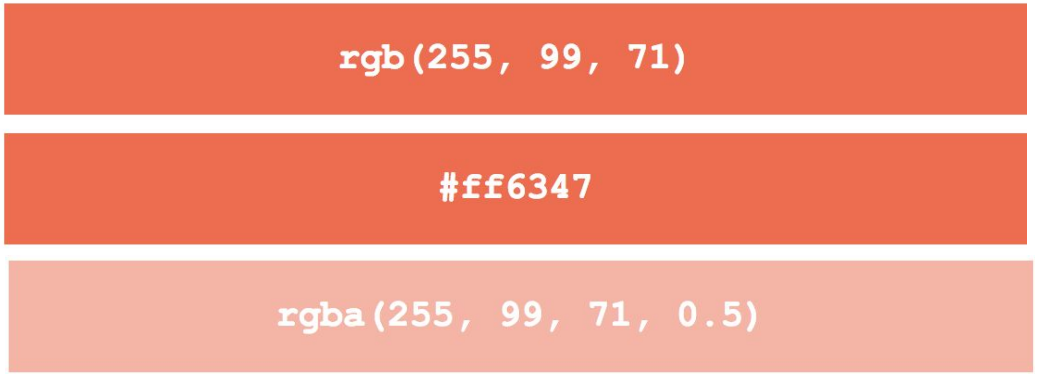
CSS Diner

CSS Properties

- Some basic CSS properties include
  - color - sets the foreground color
  - background - sets the background color
  - font - lets you set which font to use, what size, what style

CSS color

- HTML supports 140 standard color names
- You can specify your own colors using RGB, HEX, RGBA, HSLA



- Tip: when R,G,B are the same, it becomes gray

CSS background

- background-color - uses the same properties as color
- background-image - set a background image *use the url(“”) function*
- background-repeat - repeats the image horizontally and/or vertically
- background-attachment - set if the image scrolls or stays fixed on the page
- background-position - sets where the background image starts
- background - a short for all the properties above

background: #ffffff url("img_tree.png") no-repeat right top;

CSS font

- font-family
  - Generic: serif, sans-serif
  - Specific: "Times New Roman", "Arial"
  - font-family uses a *fallback* system

    font-family: "Times New Roman", Times, serif;

- font-style
  - Has three values: normal, italic, oblique
- font-weight
  - normal, bold, bolder, lighter, *number* (100 to 900)

CSS font-size

- font-size can be specified in terms of
  - pixels i.e. 16px ← default font size for most browsers
  - points i.e 24pt - like what a text processor uses
  - em - a multiple of the current font size i.e. 2em by default would be 32px
  - Percentage font size such as 90%, 120%
  - Absolute font sizes such as x-small, small, medium, large, x-large
  - Relative font size such as smaller, larger

CSS Text

property	description
text-align	alignment of text within its element, e.g. left, right, center, or justify
text-decoration	decorations such as underlining, e.g. overline, line-through, blink, or none
line-height, word-spacing, letter-spacing	gaps between the various portions of the text
text-indent	indents the first letter of each paragraph

Other CSS Properties

- opacity - sets the opacity of *any* element
  - Takes a decimal value from 0.0 (fully transparent) to 1.0 (fully opaque)
- list styles

Property	**Description**
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow
list-style-type	Specifies the type of list-item marker

Inheritance

- Some elements pass down their style properties to their children
  - font-family and color are inherited, the whole document uses the same font and color
  - margin, padding, border don’t inherit from their parent, we want elements to set their own spacing
- You can control the inheritance of all CSS properties
  - inherit - take the property value from the parent
  - initial - use *the browser’s* default value for this property
  - unset - set the property back to its default value
    - This could be like inherit if the property inherits by default, otherwise it is like initial

Cascading

- When styling, we might have multiple rules that apply to one element
- We need to decide which selector wins out
- The order in which we decide is:
  1. Importance
  2. Specificity
  3. Source Order

Importance

- CSS has a special syntax to determine the importance of a rule

!important

- Declarations that are declared !important *always win!*
- But…….
- The only way to beat !important is with *another !important*
- **Use it carefully**
  - Use it for styles that you know *always* need to be the same

```
.blue { background-color: #3B5998 !important; }
```

Importance

```
<p class="better">This is a paragraph.</p>
<p class="better" id="winning">Winning Paragraph!</p>
```


```
#winning {
  background-color: red;
  border: 1px solid black;
}

.better {
  background-color: gray;
  border: none !important;
}

p {
  background-color: blue;
  color: white;
  padding: 5px;
}
```

Specificity

- A measure of how specific a selector is - how many elements it *could* match.
  - element selectors have very low specificity
  - class selectors have higher specificity, but not as high as id
  - The only way to win against id selectors is with !important
- To calculate specificity

style	id	class, pseudo-class, attribute	elements, pseudo-elements
thousands	hundreds	tens	ones
0	0	0	0

Specificity

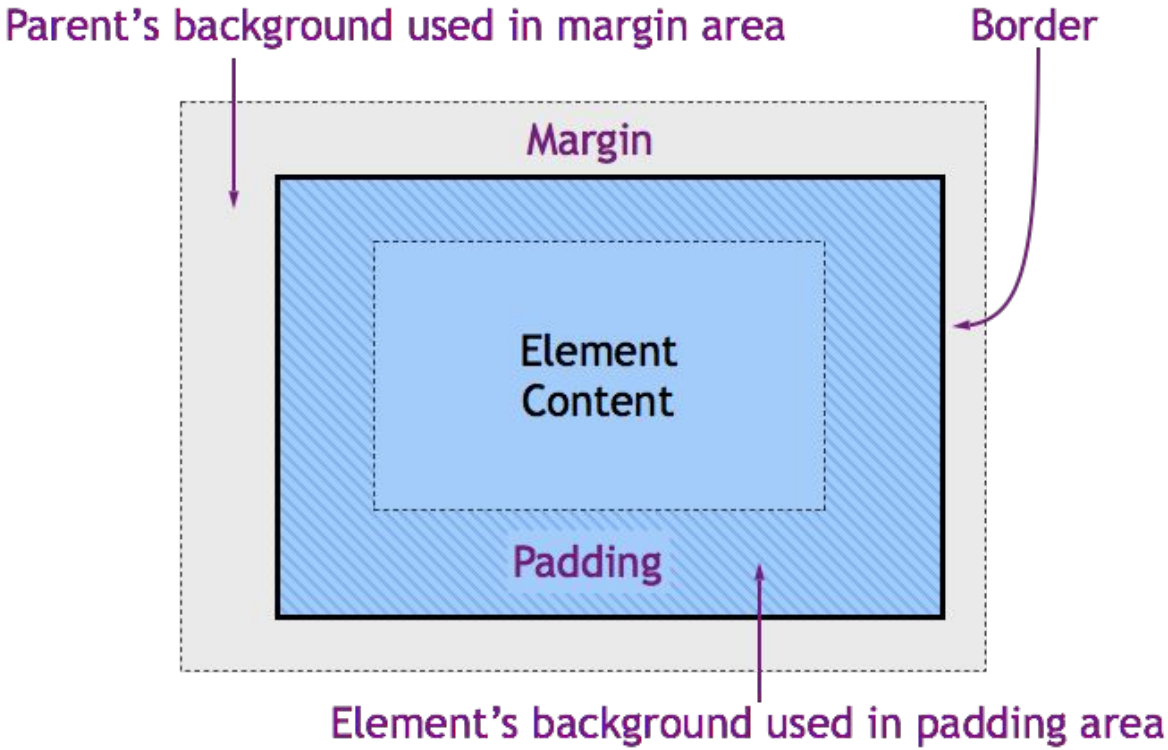
Selector	Thousands	Hundreds	Tens	Ones	Total specificity
h1	0	0	0	1	0001
#important	0	1	0	0	0100
h1 + p::first-letter	0	0	0	3	0003
li > a[href*="en-US"] > .inline-warning	0	0	2	2	0022
#important div > div > a:hover, inside an element's style attribute	1	1	1	3	1113

Source Order

1. Inline styles
2. <style> tag in the <head> of the file
3. External CSS

Note: Importance always wins, so even if you have inline styles, if you have an !important style in any of the other two source it will overwrite your inline style

CSS Box Model



CSS Box Model

- For layout purposes, every element is composed of:
  - The actual element's content
  - A border around the element
  - A padding between the content and the border (inside)
  - A margin between the border and other content (outside)

width = content width + L/R padding + L/R border + L/R margin

height = content height + T/B padding + T/B border + T/B margin

Padding

- Size in terms of px, pt, %, etc.

Property	Description
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

Border

- thickness: specified in px, pt, em, or thin, medium, thick
- style (none, hidden, dotted, dashed, double, groove,inset, outset, ridge, solid)
- color (specified as seen previously for text and background colors)

property	description
border-color, border-width, border-style	specific properties of border on all 4 sides
border-bottom, border-left, border-right, border-top	all properties of border on a particular side
border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-top-color, border-top-style, border-top-width	properties of border on a particular side

Margin

- Same units as padding

Property	Description
margin	A shorthand property for setting the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

Resources

https://flukeout.github.io/

https://css-tricks.com/specifics-on-css-specificity/

https://www.w3schools.com/css/css_boxmodel.asp

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Cascade_and_inheritance