



BEGINNING HTML AND CSS

CLASS 2

CSS

Your Content
+ HTML: Structure
+ CSS: Presentation
= Your Website

With CSS you specify colors, position, fonts, borders, spacing and many many other effects.

WHAT WE'LL LEARN TODAY

- A little CSS history
- Terminology and syntax
- Selectors
- Defining Colors and Fonts
- How to attach CSS to your web page

HISTORY OF CSS

CSS = Cascading Style Sheets

- The 90s
 - HTML pages read from top to bottom, black font, no color, and all default browser styles.
 - Fine for science papers, but designers said "We Want More!"
 - 1993: The first graphical browser is born — "Mosaic"
 - 1994: World Wide Web Consortium is inaugurated (W3C) and the World Wide Web is born.

HISTORY OF CSS

- Late 90s
 - 1996: Specifications for CSS1 are released (a year before HTML 4.0).
 - CSS1 is buggy and poorly adopted.
 - 1998: W3C releases CSS2.
 - CSS2 is buggy and poorly adopted.
 - Meanwhile, table-based layouts and browser wars are rampant!

HISTORY OF CSS

- The 00s
 - 1999 - 2000: Work is begun on CSS2.1 to fix bugs in CSS 2.
 - 2004: The working draft becomes a candidate for adoption by the W3C. It reverts back to working draft in 2005.
 - 2007: The working draft again becomes a candidate for adoption by the W3C
 - 2010: It reverts back to a working draft.
 - 2011: June 7th, it is finally published as the W3C recommendation.

HISTORY OF CSS

- CSS3
 - CSS3, begun in 2000, is still mostly in working-draft stage.
 - Modular release (rather than one single adoption).
 - 2013: Most modules still in working-draft stage
 - ...some released and adopted by modern browsers.

CSS: WHAT DOES IT LOOK LIKE?

```
}-
body {-
  > font: 100% "Gotham", sans-serif;-
  > width: 100%;-
  > padding: 0;-
  > margin: 0;-
}-
header{-
  > clear:both;-
  > width: 100%;-
}-
h3{-
  > font-size: 110%;-
  > font-weight: normal;-
  > letter-spacing: 1px;-
  > margin-bottom: 12px;-
  > margin-top: 10px;-
  > text-transform: uppercase;-
}-
a{-
  > color: #01A9B4;-
  > text-decoration: none;-
  > background-color: transparent;-
}-
a:hover-
{-
  > color: #222;-
}-
#page, #site_bar_content, .header, .footer {-
  max-width: 1200px;-
  min-width: 480px;-
  margin: 0px auto;-
}-
.header{-
  > padding: 10px 15px;-
}-
```

LET'S CODE SOME CSS!

- <head>
- <title>
- My Very First Web Page!
- </title>
- <style>
- h1 {
- color: blue;
- background-color: yellow;
- }
- </style>
- </head>

NOW SAVE YOUR PAGE

Open it up in a browser

Does your heading look different?

CSS TERMINOLOGY:

CSS is composed of style "rules"

Here is a style rule:



STYNAX IS IMPORTANT!

- `h1 {`
 - `color: blue;`
 - `background-color: yellow;`
 - `}`
- There are no limits to the number of declarations in a style rule.
- Common convention is to use lower case throughout.
- Don't forget the semicolon at the end of the declarations!
- Don't forget the closing curly bracket!

WHAT CAN WE USE AS SELECTORS?

- HTML tags
- Classes and ids
- Pseudo classes
- Any combination of the above!

HTML TAGS:

- p {
- property: value;
- }

This would select every paragraph element.

- img {
- property: value;
- }

This would select every image element.

...but what if you need more control?

CLASSES AND IDS

- "Class" and "ID" are HTML *attributes*.
- Attributes "describe" elements and are followed by *values*.
- In your HTML, it looks like this:
 - `<p id="intro">`
 - ``

CLASSES

- `<p class="warning">`
- `.warning {`
- `property: value;`
- `}`
- A class name is preceded by a period in your style rule.

IDS

- `<p id="intro">`
- `#intro {`
- `property: value;`
- `}`
- An id name is preceeded by a pound sign in your style rule.

IDS VS. CLASSES

ID: An id can only be used once *on a page*. Refers to a singular page element (like a footer).

Think ~ A student ID number

Class: Lots of elements can have the same class. I.E. There can be many spans with a class of "warning".

Think ~ A student as a member of a class

NAMING YOUR CLASS OR ID:

- Can use letters, numbers, underscore or dash (but don't start with a number or a dash followed by number).
- No spaces — use a hyphen or underscore
- CSS is case-insensitive, but the convention is to use all lowercase letters.
- In your HTML, class and id names are in quotes (just like all other attribute values).

LET'S CODE IT!

- Add these rules inside your style tags
 - `#intro {`
 - `color: blue;`
 - `}`
 - `.warning {`
 - `color: red;`
 - `}`

Add an id of "intro" to your first paragraph

Add two spans with a class of "warning"

PSEUDO CLASSES

Describes a "current condition" of an HTML element,
rather than an "attribute"

Link pseudo classes are the most common

LINK PSEUDO CLASSES

- `a:link` *~unvisited link*
 - `a:visited` *~visited link*
 - `a:hover` *~mouse over link*
 - `a:active` *~selected link*
-
- If present, `a:hover` *must* come after `a:link` and `a:visited`.
 - If present, `a:active` *must* come after `a:hover`.

LET'S SPICE UP OUR LINKS!

- `a:link {`
- `color: blue;`
- `}`
- `a:visited {`
- `color: yellow;`
- `}`
- `a:hover {`
- `color: green;`
- `}`
- `a:active {`
- `color: white;`
- `}`

COMPOUND SELECTORS

Combining selectors to get really specific!

- `p em {`
- `property: value;`
- `}`

Selects all em elements that are within a paragraph

- `#intro a {`
- `property: value;`
- `}`

Selects all link elements in elements with an id of "intro".

LET'S ADD A COMPOUND SELECTOR RULE!

- `<style>`
 - `#intro a {`
 - `font-style: italic;`
 - `}`
- `</style>`

STYLING WITH COLOR AND FONTS

COLOR

The *color* property sets the color of the font.

The *background-color* property sets the color of the background.

Color value can be defined in one of three ways:

- By a recognized color name
- By a *hexadecimal* value
- By an *RGB* value

RECOGNIZED COLOR NAMES

The 17 standard colors are:

aqua, black, blue, fuchsia, gray,
grey, green, lime, maroon, navy,
olive, purple, red, silver, teal,
white, and yellow.

There are 141 named colors:

http://www.w3schools.com/cssref/css_colornames.asp

HEXADECIMAL VALUES

Example — color: #A53C8D

Three pairs of numbers or letters, making a string of six
Includes 0 - 9 and A - F

- The first pair equals red
- The second pair equals green
- The third pair equals blue

Hex values are always preceded by a pound sign

RGB VALUES

Example — `color: rgb(165, 60, 141)`

Three comma-separated numbers from 0 to 255

- The first number equals red
- The second number equals green
- The third number equals blue

CSS3 introduces a 4th value, "a", setting opacity

Example — `color: rgba(165, 60, 141, 0.5)`

FONT

5 DIFFERENT PROPERTIES TO STYLE FONT!

1. font-style

- example: `font-style: italic;`
- values: *"normal"*, *"italic"*, or *"oblique"*

2. font-variant

- example: `font-variant: small-caps;`
- values: *"normal"*, *"small-caps"*, or *"inherit"*

3. font-weight

- example: **font-weight: bold;**
- values: *"normal", "bold", "bolder", "lighter", "100", "200", "300", ... "900"*

4. font-size

- example: **font-size: 12px;**
- values:
 - absolute: pixels (ie 12px)
 - relative: percents (ie 100%) and ems (ie 1.5em)

5. font-family

- example:

`font-family: Corbel, 'Helvetica Neue', Helvetica, Arial, sans-serif;`

- Computers don't all have the same fonts installed...so provide alternatives
- Specific to general, in a comma-separated list.
- Fonts with two-word names are in quotes

BONUS FONT PROPERTIES!

6. text-transform

- example: `text-transform: uppercase;`
- values: *"capitalize"*, *"uppercase"*, *"lowercase"*, or *"none"*

7. line-height

- example: `line-height: 1.5;`
- values: *numbers*, *percents*, *pixels*, or *"ems"*

SHORTHAND FONT DECLARATION

example:

```
font: italic small-caps bold 34px/150% "Times New Roman", Times,  
serif;
```

font-style → font-variant → font-weight → font-size /
line height → font-family

SHORTHAND FONT DECLARATION

To use shorthand, you must declare at least font-size and font-family

Family has to come last

If you're using them, font-style, font-variant and font-weight have to come first.

LET'S CODE IT!

- Add the shorthand font rule to your heading
- `h1 {`
- `font: italic small-caps bold 34px/150% "Times New Roman", Times, serif;`
- `}`

ATTACHING CSS TO YOUR WEB PAGE

There are three ways

- Inline
- Embedded
- Linked

INLINE

```
<p style="color: red;">Some text.</p>
```

Uses "style", which is an HTML attribute

Difficult to use in large projects

Not preferred

EMBEDDED

This is how we have been doing it in our exercises.

- "Embedded" inside the `<head>` element.
- Uses `<style>` tag.

WHAT IS A BIG DRAWBACK TO THIS METHOD?

LINKED

- <head>
- <title>
- My Very First Web Page!
- </title>
- <link rel="stylesheet" type="text/css" href="style.css">
- </head>

ADVANTAGES OF LINKED (EXTERNAL) STYLE SHEETS:

- Shared resource for several pages.
- Reduced file size & bandwidth
- Easy to maintain in larger projects.

LET'S CODE IT (PART 1)

1. Open a new tab in your text editor
2. Copy all the rules between the style tags on your index.html
3. Paste them on your new page
4. Save your file inside the "styles" folder you created earlier. Name it "styles.css".

LET'S CODE IT (PART 2)

5. Delete the style tags and everything within them on your index.html page
6. Where the styles used to be, code the following:

```
<link rel="stylesheet" type="text/css"  
href="styles/style.css">
```

7. Save your index.html page and open it in a browser

Are your styles still there?

CASCADING

Styles "cascade" down until changed

```
p{
  color:blue;
  font-family: 'Helvetica';
}
.red{
  color:red;
}
#special{
  font-family: Arial;
}
```

```
<p>Paragraph</p>
<p class = "green">Paragraph</p>
<p class = "red">Paragraph</p>
<p class = "red" id = "special">Paragraph</p>
```

CSS PROPERTIES

Many CSS properties have self-explanatory names:

- background-color
- font-family
- font-size
- color
- width
- height

Comprehensive list of all CSS properties

QUESTIONS?



