

*CS 641, Haik Sahakian*

---

# Mobile Web Development

---

An Introduction to HTML,  
CSS, JavaScript, and HTTP

# An Introduction to HTML, CSS, JavaScript, and HTTP

*Getting Quickly Up to Speed On*

# HTML and CSS

---

# Introductory Videos

---

If you need to get up to speed on HTML & CSS quickly I think the best way is to watch some short videos, then view more detailed overviews, then check reference sites as needed.

The CSS video will make more sense if you watch the HTML videos first. These videos by Jake Wright are each 12 minutes in length, and are excellent.

HTML: <https://www.youtube.com/watch?v=bWPMSSsVdPk>

HTML part 2: <https://www.youtube.com/watch?v=KJ13lX20FqU>

CSS: <https://www.youtube.com/watch?v=0afZj1G0BIE>

---

# Detailed Tutorials

---

Code Academy is a well known and excellent site for learning HTML. However, W3Schools may be much faster, as you can easily skip parts you have already figured out. It's older and not interactive, but for programmers I think it will usually be better as I think you'll learn the basics quickly just by looking at its examples.

W3Schools: [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

Code Academy: <http://www.codecademy.com/en/tracks/web>

---

# HTML and CSS Reference

---

Mozilla's site below is nicely organized and full of useful information. MSDN and W3Schools are excellent sites too.

HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element#S>

CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>



*Getting Quickly Up to Speed On*

# JavaScript

---

# Code Monster

---

If you're new to programming, Crunchzilla's Code Monster site is an excellent place to start. Its visuals are aimed at school-age kids, but don't mind them — it's the best way to learn programming I've seen.

Another advantage to Code Monster is that it happens to teach programming by using JavaScript and the browser's Canvas object to draw graphics. So you get a good introduction to web graphics programming too.

<http://www.crunchzilla.com/code-monster>



# Text and Image Editors

---

# Text Editor

---

- ❖ If you're looking for a good text editor for all web development including Ionic, I recommend WebStorm. It's free for students.
- ❖ You can download it at <https://www.jetbrains.com/webstorm/>.

---

# Image Editor

---

- ❖ Choose and install an image editing program to use in the course. PhotoShop is excellent but expensive. Gimp is harder to use, but is free. Adobe's PhotoShop Elements is about \$30 second-hand. There are also many good OS-specific image editors available around \$30.
- ❖ Check you can:
  1. Save transparent images in PNG format
  2. Cut out part of one image and paste it into another
  3. Change the color of an object in an image to a new given color

HTML

---

# Basics

---

- ❖ Elements and Attributes
- ❖ Paragraphs and Line Breaks
- ❖ Images and Video
- ❖ Links

---

# Slightly More Complicated

---

- ❖ Block vs Inline
- ❖ Content vs Formatting



---

# Document Structure

---

- ❖ The DOCTYPE
- ❖ HEAD and BODY
- ❖ What lives where

---

# Layout

---

- ❖ DIVs
- ❖ Tables
- ❖ Text
- ❖ Layout Frameworks
- ❖ Best practices

---

# Useful Tags

---

- ❖ Header
- ❖ Label
- ❖ H1 - H6
- ❖ Meta
- ❖ Pre
- ❖ Script
- ❖ Form
- ❖ Button
- ❖ Input
- ❖ Textarea
- ❖ Select
- ❖ Fieldset

---

# Growth of HTML

---

- ❖ HTML 5 is the current spec: <http://www.w3.org/html/wg/drafts/html/master/>
- ❖ Complete list (2001) at <http://www.willcam.com/cmat/html/crossref.html>
- ❖ New HTML 5 tags: [http://www.w3schools.com/html/html5\\_new\\_elements.asp](http://www.w3schools.com/html/html5_new_elements.asp)
- ❖ Complete current list: <http://www.w3schools.com/tags/>

HTTP

---

# HTTP Request Structure

---

- ❖ User Agent identification
- ❖ Query strings
- ❖ GET and POST requests
- ❖ Being stateless
- ❖ Best practices



---

# HTTP Response Structure

---

- ❖ HTTP Headers
- ❖ Cookies
- ❖ Server Response Codes

---

# Caching

---

- ❖ Snappy performance
- ❖ Browsers caches in memory and on disk
- ❖ Server caches
- ❖ Distributed caches
- ❖ Cache-control and Expires headers
- ❖ Filename and query string considerations
- ❖ Best practices

CSS

---

# Basics

---

- ❖ Fonts
- ❖ Borders, Margins, and Padding
- ❖ Backgrounds

---

# Basics

---

```
color: #9fa834;  
background-color: #080808;  
background: url("plastic.png") red repeat-y fixed;  
font-family: "Source Sans Pro", Arial, Helvetica, sans-serif;  
border: 1px solid red;  
margin: 10px;  
margin-top: 20px;  
padding: 30px;
```

---

# Defining Colors

---

- ❖ By name, eg. “red”
- ❖ By hex value, eg. “#a37d80”
- ❖ RGB hex values are made up of 3 hex numbers, each from 00 to FF, representing Red, Green, and Blue. For example, #FF0000 is a bright red and #303050 is a blue-ish grey.



---

# Units

---

- ❖ Pixels
- ❖ Points, ems and rems
- ❖ Percentages
- ❖ Setting a style to zero
- ❖ Setting a style to its default
- ❖ Best practices

---

# Attaching Styles

---

- ❖ Style attributes
- ❖ Class attributes
- ❖ ID attributes
- ❖ Inline styles
- ❖ External CSS files
- ❖ Best practices

---

# Box Model

---

## ❖ The Box Model

- `box-sizing: content-box; // default`
- `box-sizing: border-box; // IE`

## ❖ See <https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>

---

# Cool CSS

---

- ❖ Shadows
- ❖ Gradients
- ❖ Cursors

---

# Cool CSS

---

```
box-shadow: 3px 3px 5px 6px #ccc;
```

```
background: linear-gradient(to bottom, black, blue 90%,  
orange);
```

```
cursor: crosshair;
```

---

# Cool CSS

---

- ❖ CSS Shadow Generator: <http://www.cssmatic.com/box-shadow>



---

# Very Cool CSS

---

- ❖ Transformations
- ❖ Animations
- ❖ Transitions

---

# Very Cool CSS

---

```
transform: rotate3d(10, 10, 10, 180deg);
```

```
@keyframes cool {  
    0% {background-color: orange;}  
    100% {background-color: yellow;}  
}
```

```
.importantLink {  
    animation: 2s cool;  
    color: black;  
    transition: color 1s ease-in;  
}
```

```
.importantLink:hover {  
    color: red;  
}
```

---

# Very Cool CSS

---

- ❖ Animations and transitions are similar.
- ❖ Animations can have many in-between states, called keyframes. Transitions cannot.
- ❖ Animations start immediately; transitions are triggered.
- ❖ Animations can be looped easily; transitions generally run once.

---

# CSS Animations

---

- ❖ Tutorial: <http://webdesign.tutsplus.com/tutorials/a-beginners-introduction-to-css-animation--cms-21068>
- ❖ Example: [http://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation1](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1)
- ❖ Video tutorial by Zevan Rosser on 2D and 3D CSS3 transforms: <https://www.youtube.com/watch?v=7zw-Yab7BrY>
- ❖ Animations vs Transitions: [https://www.kirupa.com/html5/css3\\_animations\\_vs\\_transitions.htm](https://www.kirupa.com/html5/css3_animations_vs_transitions.htm)

---

# Pseudo Classes

---

- ❖ CSS applies styles with selectors
- ❖ Style examples: color, text-shadow, corner-radius
- ❖ Selector examples: div.pane, #SSN, body
- ❖ Pseudo classes extend regular selectors to style either certain states of them, or insert content around them
- ❖ Pseudo class examples: a:hover, .copyright::after, nth-child()
- ❖ All 27 pseudo classes: [http://www.w3schools.com/css/css\\_pseudo\\_classes.asp](http://www.w3schools.com/css/css_pseudo_classes.asp)



---

# Before and After Pseudo Classes

---

- ❖ Inserts *content*, not markup
- ❖ IE 8 supports the old single colon versions, IE 9 supports double colons
- ❖ <http://css-tricks.com/almanac/selectors/a/after-and-before/>
- ❖ <https://msdn.microsoft.com/en-us/library/ie/cc304076%28v=vs.85%29.aspx>

# Summary



---

# HTML

---

- ❖ Document (HTML, HEAD, BODY, IFRAME)
- ❖ Inline tags (B, IMG, SPAN)
- ❖ Block tags (P, DIV, TABLE, UL, H1)
- ❖ Form tags (FORM, BUTTON, INPUT)
- ❖ Semantic tags (HEADER, FOOTER, NAV)
- ❖ Adding CSS and JavaScript (STYLE, SCRIPT)

---

# CSS

---

- ❖ Formatting (color, background-color, border, padding, margin, width, height)
- ❖ Positioning (position, top, left)
- ❖ CSS pseudo-classes (hover, nth-child)
- ❖ Changing (2d and 3d transforms, animation, transitions)

---

# JavaScript

---

- ❖ Basics (alert, var, function)
- ❖ Event handlers (onClick, onLoad)
- ❖ Page modification (getElementById, img.src, style.top, style.left)
- ❖ The DOM

---

# Concepts

---

- ❖ Tags and Attributes
- ❖ Newlines don't matter, spaces are condensed
- ❖ Block vs Inline
- ❖ Loading in HEAD vs BODY
- ❖ Limited Network Connections
- ❖ External CSS and JavaScript files

---

# Browser Debugging

---

- ❖ Chrome is the most advanced browser, closest to mobile browsers, and has the best debugging tools.
- ❖ The network tab in Chrome's developer tools lets you inspect network connections. The elements tab inspects HTML and CSS. Its console tab shows JavaScript messages and errors.

---

# Example

---

I put the code from class up at <http://webpage.pace.edu/hsahakian/examples/intro-html/intro.html>.