

HTML / CSS

Introduction to Web Development

About Me

Instructor: Kevin Sholander

Web Developer and Business Owner

Instructor of Web Development

Email: kevin.sholander@learningfuze.com

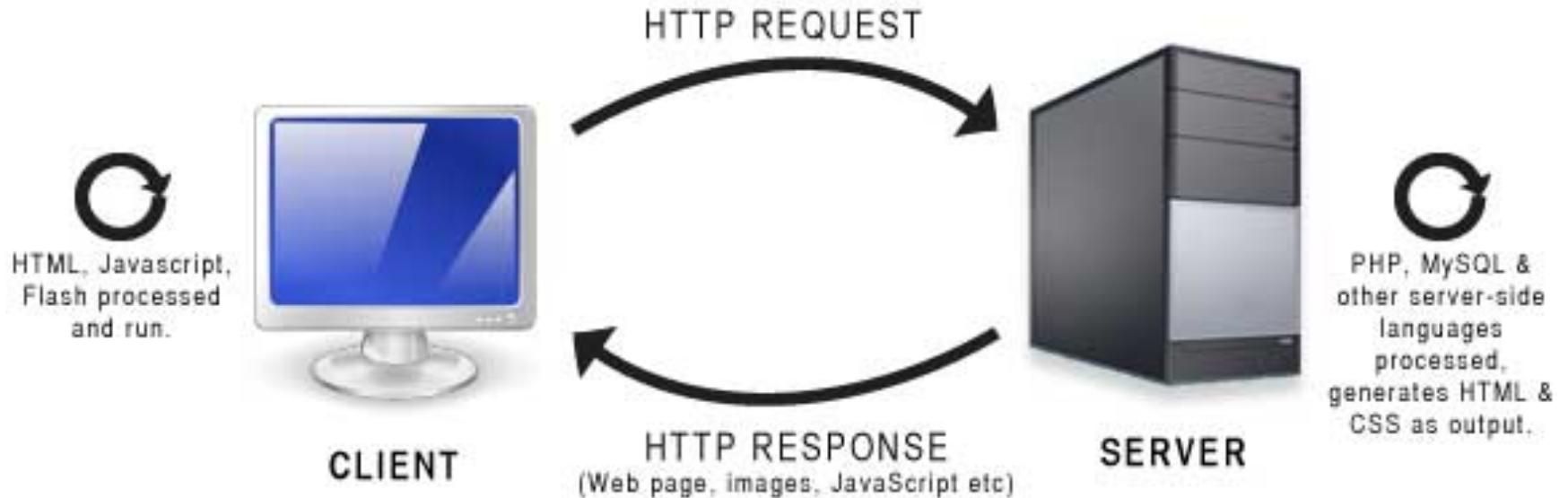
Agenda

1. **Intros & Setup**
2. **Web Overview & Setting Up Environment**
3. HTML Foundations
4. CSS Foundations
5. Project
6. Tools
7. Website Creation Checklist
8. Next Step Possibilities

Website Rendering Workflow

What happens when a user types a website address in a browser?

Website Rendering Workflow



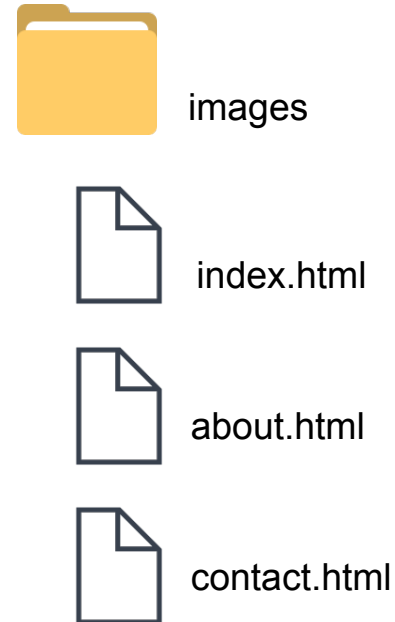
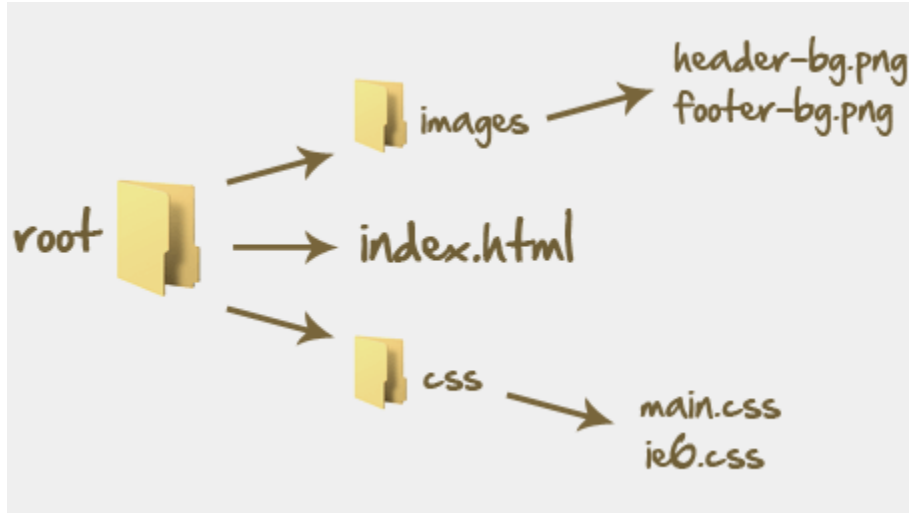
Website Analysis

Let's examine a couple popular websites

Websites Overview

- A website is made up of web pages linked together using hyperlinks.
- Websites are built by writing code using the standard web languages:
 - Markup language (HTML)
 - Style Sheet language (CSS)
 - Scripting language (JavaScript)
- Web standards can be found at [w3schools.com](https://www.w3schools.com)

Website Structure



Setting Up Your Dev. Environment

- Download and install Sublime:
 - <http://www.sublimetext.com>
- Set up a folder to store your work
 - On your desktop create a folder called “WebDev”
 - In “WebDev”
 - Create a folder called “HTMLpractice”
 - Create a folder called “CSSpractice”
 - Create a folder called “project1”

Agenda - 5 Min. Break

1. Intros & Setup
2. Web Overview & Setting Up Environment
3. **HTML Foundations**
4. CSS Foundations
5. Project
6. Tools & Website Creation Practices
7. Next Step Possibilities

HTML Basics

- What is HTML?
 - Stands for “**HyperText Markup Language**”
 - A document is "marked up" by inserting tags into the text
 - Tags indicate the purpose of each of the text pieces
 - NOT used to control how the document will appear in the browser
- How is it created?
 - It consists of a simple text file saved as “filename.html” or “.htm”
 - The filename cannot have spaces, i.e. “file name.html”
 - The file is built up of elements. An element has a tag, optional attributes and content. The content can be other elements.

HTML Basics - Structure

- Basic Page Structure:

`<!DOCTYPE html>` (not an actual tag - tells browser the html version)

`<html>`

`<head>`

information about the page

`</head>`

`<body>`

actual visible content

`</body>`

`</html>`

HTML - Tags

- Rules:
 - Tags are case insensitive (but use lowercase)
 - Most tags require an opening `<p>` and closing `</p>` tag
 - Close all tags in correct order (nesting):

`<p>Let's emphasize this word</p>`

- Exception: Some elements are "self-closing"
 - These are elements that can't have any content
 - examples:

`<input />,
, <hr />, `

HTML - Element “Attributes”

- What are attributes:
 - Attributes are added inside the **opening tag** to provide additional information about that element
 - Attributes are specified with a name and a value
 - Some tags require particular attributes. Other attributes can be used with any element
 - Examples:

```

```

```
<p id="main-paragraph">This is my main paragraph</p>
```

http://www.w3schools.com/tags/tag_p.asp

HTML - Ex. 1

In your “HTMLpractice” folder:

- Create a new file. Name it "basic.html"
- Edit the file with Sublime
- Declare the document type
- Create the html, head, body tags
- Inside the head add a <title> tag and inside add the text:
"Basic HTML"
- Save the file and open in your browser (double click the file to open)

HTML - Ex. 2

In your “HTMLpractice” folder:

- Copy the file "basic.html"
- Name the copy "elements.html"
- Edit "elements.html" in Sublime
- Change the title to "Basic Elements"
- Save the file

Basic HTML Tags

headers - `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

paragraph - `<p>`

lists - ``, ``

tables - `<table>`

forms - `<form>`

anchors(links) - `<a>`

images - ``

Basic Elements - Ex. 1

In your “elements.html” file:

- Add six elements, one for each header (h1-h6)
- Use the text “header one”, “header two”, and so on as the content of each element
- Save the file and run in your browser

http://www.w3schools.com/tags/tag_hn.asp

Basic Elements - Ex. 2

In your “basic-elements.html” file:

- Keep the first header, replace the rest with <h2>
- Under each header add a paragraph with any text inside it
- Save the file and run in your browser

http://www.w3schools.com/tags/tag_p.asp

Basic Elements - Lists

- Ordered Lists
 - numbered
 - attribute styles: numbers, letters, Roman Numerals, (upper & lowercase)
- Unordered Lists
 - unordered
 - attribute styles: disc, circle, square, none

Basic Elements - &

List items in ordered/unordered lists use the tag.

Example:

```
<ol>
```

```
  <li>Item number one</li>
```

```
  <li>Item number two</li>
```

```
</ol>
```

http://www.w3schools.com/html/html_lists.asp

Basic Elements - Ex. 3

- Create a new file called “ordered-lists.html”
- Add the basic HTML structure
- In the body of the document add two ordered lists
- Add three items to each list
- In the first item of each list, insert another list with two items
- Save the file and open in the browser. Notice how the nested lists are formatted

<http://jsfiddle.net/bpqwmnxg/>

Basic Elements - Ex. 4

- Create a new file called “unordered-lists.html”
- Add the basic HTML structure
- In the body of the document add two unordered lists
- Add three items to each list
- In the first item of each list, insert another list with two items
- Save the file and open in the browser. Notice how the nested lists are formatted

<http://jsfiddle.net/bpqwmnxg/>

Basic Elements - Links

- Links are created with the anchor tag: <a>
 - The href attribute is used to specify the url of the target page

The Anchor Text

http://www.w3schools.com/tags/tag_a.asp

Fun with Links

- Link to send an email using client's default email client:

`Mail Me!`

- Open up a link in a new browser tab:

`Google`

- Create a bookmark in your document:

``

- Jump to a bookmark in your document:

`Go to bookmark 1`

Basic Elements - Ex. 5

In your “elements.html” file:

- Insert a tag into your paragraph
- Add links to your current files
- i.e.:

```
<p><a href="ordered-lists.html">Ordered Lists</a></p>
```

http://www.w3schools.com/tags/tag_a.asp

Basic Elements - Images

- Images are declared with the `` tag
 - “src=” attribute defines the path to the image
 - “alt=” attribute defines alternate text
 - `` is a self closing tag - it has no content other than attributes

```

```

http://www.w3schools.com/html/html_images.asp

Basic Elements - Ex. 6

- **Create a file “images.html”**
 - a. add an image that uses a “relative path”
 - b. add a second image that uses an “absolute path”
 - i. Go to Google images and find 2 images
 - ii. Save one image into your “htmlpractice” folder
 - iii. Copy the url of the other image and add it to your “src=” attribute
 - c. Add a link to this page from your “elements.html” page under one of the paragraphs

Basic Elements - Tables

- Tables are used to display data and are created with the `<table>` tag
 - tables all have table rows: `<tr>`
 - table rows have table data: `<td>`
 - tables rows can also have table headers: `<th>`

http://www.w3schools.com/html/html_tables.asp

```
<table>
```

```
  <tr>
```

```
    <th>Name</th>
```

```
    <th>Grade</th>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Jane</td>
```

```
    <td>B+</td>
```

```
  </tr>
```

```
</table>
```

Example Table

Name	Grade
Jane	B+

Lunch!

Next:

- Block Vs. Inline Level Elements**
- CSS Foundations**
- Website Project**

Agenda

1. Intros & Setup
2. Web Overview & Setting Up Environment
3. HTML Foundations
4. **CSS Foundations**
5. Project
6. Tools & Website Creation Practices
7. Next Step Possibilities

Block vs. Inline Elements

- **Block Level:**
 - Take up entire width of the containing element
 - Default margins & padding
 - Can contain both block & inline elements
 - `<h1>`, `<p>`, ``, ``, `<div>`
 - In CSS:
 - `display: block;`
- **Inline Level:**
 - Occupies only space of content size
 - Coexists with other elements on same line
 - Can only contain other inline elements
 - `<a>`, ``, ``, ``, ``

Cascading Style Sheets (CSS)

Cascading Style Style Sheets are used for:

- 1. Styling (color, font, underlining, etc)**
- 2. Layout (placement, size)**
- 3. <http://www.csszengarden.com/>**

CSS1 (2000), CSS2 (2009), CSS3(2013+)

CSS Foundations - Adding CSS

There are 3 ways in which you can apply or add CSS rules to your web page:

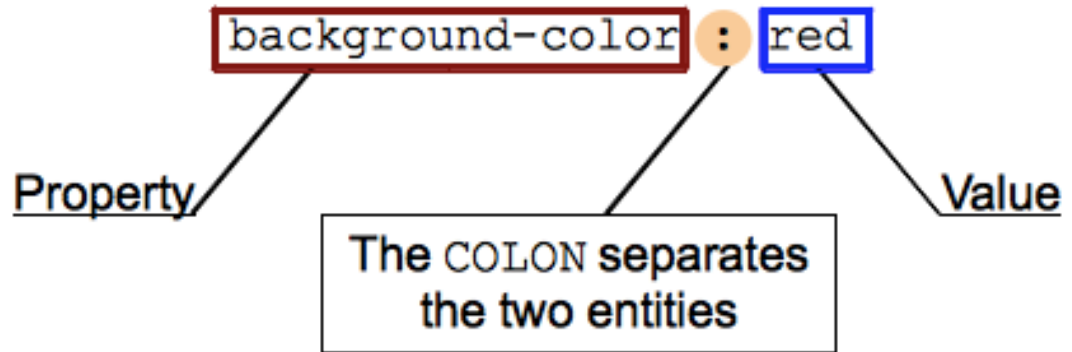
- 1. Inline - elements may have a style attribute**
- 2. Internal - html files may have style tags**
- 3. External - a file that contains just css. Can be linked into multiple pages as needed**

CSS Foundations - Basic Syntax

CSS is made up of rules that contain a selector and a list of declarations.

```
Selector(s) {  
  property: value;  
}
```

A CSS declaration :



CSS Foundations - Ex. 1

In your “csspractice” folder:

- create a new file “css-syntax.html”
- add the basic html skeleton
- add the following:
 - a. title
 - b. a main header <h1> with 2 secondary headers <h2>
 - c. a paragraph after the main header
 - d. an ordered list after the first <h2>
 - e. an unordered list after the second <h2>
- add inline styling to the first paragraph <p style="color:blue;">
- add an internal style targeting the ordered list (color:purple;)

CSS Foundations - External CSS

- Adding an external stylesheet is the most common practice
- External style sheets may be reused across multiple pages or an entire site
- Rules are added to a ".css" file
- The file is included into HTML using a link tag:

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

CSS Foundations - Ex. 2

In your “csspractice” folder:

- create a new file “main.css”
- add a link to this stylesheet in your current file

CSS Foundations - Cascade/Inheritance

- The “cascade” in CSS comes from rules being applied in a cascading manner
- Rules are applied according to Parent/Child relationships
- Certain properties applied to parent elements are “inherited” by their child elements

CSS Foundations - Selectors

A CSS selector is a pattern to match elements in an HTML document. The associated style rules are applied to the element that match that selector

Universal: ***** {..}

Element Type: **p** {..}

Group: **body, h1, p** {..}

Descendant: **ul li a** {..}

Pseudo hover selector: **:hover** {..}

class selector: **.class** {..}

id selector: **#id** {..}

CSS Foundations - Styling

Styling Content Blocks:

Colors, backgrounds, fonts, list styles, text-decoration, display properties

Styling values:

Colors: color names, hexadecimal values, rgb

Backgrounds: colors, images

CSS Foundations - Box Model

The Box Model:

Padding, Borders, Margins

Dimensions:

width, height

Dimension Values:

px, %, em, etc.



CSS Foundations - Positioning/Layout

Positioning:

Account for Padding, Borders, Margins (Box Model)

Account for Block or Inline Level Elements

floats

Static, Fixed, Absolute, Relative

Agenda

1. Intros & Setup
2. Web Overview & Setting Up Environment
3. HTML Foundations
4. CSS Foundations
- 5. Project**
- 6. Tools & Website Creation Practices**
7. Next Step Possibilities

Project - Personal Website

Goal: Create a website that emulates the [airbnb.com](https://www.airbnb.com) layout.

We will build the home page together

Afterwards, try and add at least 2 more pages or build your own!

Step 1 - Structure

Let's Start with the Home Page Structure:

1. In your “project1” folder:
2. Create a basic HTML skeleton
3. Add the header, middle section, main content, and footer
4. Add the dimensions (height only):
 - a. header: 500px
 - b. middle section: 100px
 - c. main content area: 500px
 - d. footer: 150px

Step 2 - Add Border/Reset

We will next add a border to all page elements using the universal (*) selector to visualize the layout

This is a common technique for visualizing and debugging layouts

1. Add margin, padding resets
2. Examine layout & approach

Step 3 - Start Filling the Header

Next we will start tackling each section 1 by one

1. Google “landscape image” and find an image larger than 1600x800
2. I found this image:

http://digital-photography-school.com/wp-content/uploads/flickr/5661878892_15fba42846_o.jpg

3. Add background image to header
4. use background-position: x y (horizontal, vertical) if needed

Step 4 - Create navigation (nav)

HTML5 provides the new `<nav>` element to contain your main navigation

1. Use an unordered list
2. Create 4 list items: Home, About, Help, Contact Us
3. horizontally align with `display:inline-block`
4. float to the right with `float:right`
5. change color, remove list style, create button, add `:hover`
6. Button color: `#70AAE2`, on `:hover`: `#50AAE2`
7. Change cursor to pointer

Step 5 - Add Logo

Google “logo” and find an image that is suitable

1. Try finding a logo that is wider
2. Save the logo into your “images” file and link to it
3. change the width dimensions
4. add proper margins
5. I found:

<https://www.thefreshconnection.biz/wp-content/uploads/2014/12/DSM-Logo1.png>

Step 6 - Add Banner Title

inside a `<div>` container add an `<h1>` and `<p>` content

1. Add a class attribute to the container
2. align text, style header/paragraph
3. add proper margins
4. Font: Arial, Color: white
5. use text transform property to uppercase

Step 7 - Add Middle Section

The middle section will contain a form input and button

1. Add `<form>` and inside add
2. An input tag with attribute `type="text"`
3. A submit button with content "Sign Up"
4. Align section properly, add transparency with "opacity" property
5. Button color: #E60000

Step 8 - Change BG Color

Our main content section has an “off-white” color

1. We will add it as a background color to the entire body
2. color: #edefec

Step 9 - Tackle Main Content Section

Our main content area will have a header and 3 centered images

1. create a div to contain our `<h2>` and `<p>` title
2. Add a class to it and center the content
3. Adjust margins, position, font size/family, and color: `#565a5c`
4. Google 3 images width x height will be 350x300
5. add proper margins

Step 10 - Tackle the Footer

Our footer contains a placeholder title and 4 social media icons

1. I went to iconfinder.com (found 4 icons) and linked to them
2. add proper styling, center, and adjust margins

Move CSS to an External Sheet

Now that we have our base styling we can move it out of our file to make it smaller and easy to maintain

1. Cut the entire style and remove the style tags
2. Create a new file and save it as main.css in the same root
3. Paste your style, save
4. Link your html file to your new stylesheet

AND Voila!! Your First Website!

Possible Next Steps

1. Build out other pages
2. Add anchor tag links to your navigation and link to your new pages
3. Link your main stylesheet to every page, and try to keep within your layout but if you wish, try changing it but stay consistent
4. Turn the social media icons into real links

Reference: <http://www.w3schools.com/>

Try their Color Picker: http://www.w3schools.com/tags/ref_colorpicker.asp

<http://chrispederick.com/work/web-developer/>