The Web Developer Bootcamp – Colt Steele, Udemy

Syllabus

Technologies Covered:

* HTML5
* CSS3
* JavaScript
* Bootstrap 4
  + CSS & JS Framework which helps with making websites nicely & easily
* jQuery
* NodeJS
* ExpressJS
  + NodeJS Framework
* MongoDB

Course Flow:

* 13+ Projects
  + Blog application
  + Browser games
  + Portfolio sites
  + HTML forms
  + Animated to-do list
  + User authentication
* Course Project -Yelp Camp
* 34 Units

Table of Contents

[Syllabus 1](#_Toc36712314)

[Unit 1: Introduction to Front End Development 3](#_Toc36712315)

[Scenario – How the Internet Works 3](#_Toc36712316)

[Unit 2: Introduction to HTML 5](#_Toc36712317)

[Unit 3: Intermediate HTML 7](#_Toc36712318)

[Unit 4: Intermediate HTML 8](#_Toc36712319)

Unit 1: Introduction to Front End Development

Objectives:

1. Setup Development Environment
   1. Google Chrome
   2. Sublime Text
2. Compare & Contrast Front-End & Back-End
3. Define roles of HTML, CSS & JavaScript

Developing for the Web:

* *Write* Code - Sublime Text 3
* *View* Code – Google Chrome/Mozilla Firefox

Internet Basics

Scenario – How the Internet Works

1. A visits example.com
2. A’s computer requests example.com’s server for a copy of its webpage
   * Request (HTTP **GET** Request) is sent in a packet, wrapped with information about and request & details like the IP address of example.com
   * Packet goes via Internet Hubs to be rerouted to locale of the server – imagine England Hub -> NYC Hub -> LA for LA hosted server (for traffic from UK to LA)
3. Server reads request, compresses web data into tiny packets to be rebuilt on users end – **POST** data back to A
4. Packets are rebuilt on A’s computer

www.website.com -> 23.235.47.175 (Unique IP resolved to Domain Name)

Finding the Right Address –

* When hitting enter to go to a website, query is sent to Internet Service Provider (ISP)
* Within ISP, the DNS (Domain Name Server) resolves Domain Name to IP

Going to that Address –

* Request is sent to IP address via HTTP (HyperText Transfer Protocol)
* Request routes fastest possible path to server with said IP
  + Utilizes server hopping – not always a direct journey due to worldwide Internet traffic, server location, etc.

website.com Responds –

* Server figures out what is being *requested*
* Server builds the right content (pulling from backend database, etc.)
* Server responds with combination of HTML, CSS, & JavaScript

Browser Translates –

* Browser translates combination of HTML, CSS, JavaScript into something readable/understandable for us

View Page Source

* + Underlying HTML, CSS, JavaScript can be inspected on any webpage
    - [Right-Click] -> “View Page Source”
    - Navigate to View -> Developer -> View Source

Front End vs Back End

* **Front End** is the stuff that you see & interact with – HTML, CSS, JavaScript
* **Back End** is everything else: Logic, DB, Choices [Server-Side Logic]

Pertinent Analogy – Restaurant

* Backend is everything that happens in the kitchen, while the front end is what is plated and sent to your table

HTML – HyperText Markup Language

* Defines Structure of a Webpage
  + “put an image here”
  + “put a heading here”
  + “put a form here”
* The “nouns” of a webpage

CSS – Cascading Style Sheets

* Defines the style of HTML
  + “make all text purple”
  + “give the first image a yellow border”
  + “make all bullet points green”
* The “adjectives” of a webpage – or the skin to the HTML skeleton

JavaScript

* Adds logic and interactivity to a page
  + “Do some math”
  + “Change color when the user clicks”
  + “Load new data from twitter”
  + “Get new NFL scores”
* The actions or “verbs” of a webpage

Unit 2: Introduction to HTML

Objectives:

1. Write properly structured HTML documents
2. Write common closing and self-closing tags
   1. Write tags with attributes
3. Recreate a simple website based on a provided photo

Resources:

* MDN - Mozilla
* W3Schools

History of HTML:

* Created in 1989/1990
* Allowed publishing and exchanging of scientific and technical documents
* Allowed electronic linking of documents via hyperlinks

The General Rule – <tagName> Content goes here </tagName>

* Some types of tags:
  + Links - href
  + Headings – h1, h2…
  + Paragraphs – p

Boilerplate of every HTML document – fleshed out @firstPage.html



* Title of Webpage goes between <title></title> tags
* Head
  + Metadata – Font files, stylesheets, scripts (what we don’t see)
* Body
  + Content
* Comments in HTML can be made with <!—comment goes here -->

Basic Tags

* Library can be found at MDN element reference
* Block elements (individual lines)
  + Heading Tags
    - <h1>…..<h6≥
  + Paragraph Tag
    - <p>
* In-line elements (within text)
  + Bolding -> now under CSS formally (<b> deprecated)
    - <strong>
  + Italicize -> also formally under CSS (<i> deprecated)
    - <em>

Lists in HTML

* Ordered Lists <ol>
  + Numbered list elements <li>
* Unordered Lists <ul>
  + Bulleted list elements <li>

Divs & Spans

* <div> element – generic container for flow content
  + Block level element – takes its own line
* <span> element – generic container for content
  + In-line element

In-line CSS (brief)

* <tag style=”atrr:val;”>

HTML Attributes

* MDN attribute reference
* Adding Additional Information to Tags
  + <tag name = “value”></tag>
  + <img src=”sample.png”>
* Not every attribute works on every element

Links

* Anchor tag <a>
  + Links to external sites – via HTTP protocol HTTP://
    - <a href=”http://url”> Link Text </a>
  + Links to other files
    - <a href=”/filepath”> Link Text </a>

Unit 3: Intermediate HTML

Objectives:

1. Write Valid HTML Tables
   1. Exercise – Pokémon Table Exercise
2. Write validated HTML Forms using <form>, <input>, <select>, and <label> tags
   1. Exercise – Recreate Complex Form

HTML Table

* <tr> -- table rows
* <td> -- table data cell
* <th> -- table headings
* <thead> list headings here>
* <tbody> list body here>

Forms – Getting User Input

* <form> tag
  + action – URL to send form data to
  + method – type of HTTP request (GET/POST)
  + container for inputs
* <input> tag
  + creates interactive controls
  + type – attribute determines the type of input
    - text
    - date
    - color
    - file
    - checkbox
* <label> tag
  + Associate parts of the form to the labels (for accessibility)
* Form Validation
  + The “required” attribute validates that inputs are not empty
    - Presence validation
    - Will not work in every browser!!
  + Type validations – text vs email
* Drop Down – Select tags
* Check Boxes – Radio Buttons
* TextArea

Unit 4: Intro to CSS

Objectives:

1. Define the “General Rule” of CSS – writing valid CSS files
2. Correctly include/incorporate CSS into your HTML files
3. Select elements by tag name, classes, and IDs
4. Style elements with basic properties like color & background
5. Use Chrome CSS Inspector to debug HTML & CSS

Resources/Inspiration –

* CSSZENGARDEN.com

CSS (Cascading Style Sheets) Basics

* The “adjectives” of the webpage
* Separate documents, included in our HTML
* The General Rule:

selector {

property: value;

anotherProperty: value;

}

* Writing Style:
  + Inline
    - <h3 style=”color: pink;”> blah blah blah </h3>
  + Style Tag

<style type=”text/css”>

li {

color: red;

}

</style>

* When conflicting style is present, last used will be present

Connecting CSS files

* link stylesheet with <link> tag

Colors in CSS

* Resource for named colors in CSS – colours.neilorangepeel.com
* Color systems
  + Hexadecimal – String of 6 hexadecimal numbers (0-9, A-F)
    - (R) \_ \_ (G) \_ \_ (B) \_ \_
  + RGB
    - 3 channels – Red, Green, & Blue – each ranging from 0 – 255
    - color: rgb(0, 255, 0); /\* This is Green \*/
* Use color pickers (online)
* RGBA – Just like RGB, but with an alpha(transparency) channel
  + Ranges from 0.0 – 1.0

Background & Border

* User ‘color’ to set text color & ‘background’ for background color
* background –
  + background-repeat: no-repeat;
  + background-size: cover;
* border –
  + border-color
  + border-width
  + border-style
  + can be combined into one attr –
    - border: width style color;

Selectors

* Element Selectors – select all instances of a given element
* ID Selector – Selects an element with a given ID – Only one per page
* Class Selector – Selects all elements with a given class
* Star Selectors – Selects all elements on page
* Descendant Selectors – Selects nested elements
  + li a { }
    - all anchor tags within li tags
* Adjacent Selectors – Selects elements of adjacent level
* Attribute Selectors – Selects attributes of tags
* Nth of type Selectors – Selects every nth element
  + element:nth-of-type(#) { }

Specificity & Inheritance

* Children inherit properties/style from their parents
* Specificity Priority
  + Inline Styles <- IDs <- Classes, Attributes <- Type Selectors

Unit 5: Intermediate CSS

Objectives:

1. Manipulate common font and text properties using CSS
2. Include external fonts using Google Fonts
3. Define and manipulate the four components of the Box Model

Exercises:

1. Tic Tac Toe Board (no JS/no logic)
2. Photo Portfolio/Gallery
3. Blog

Texts & Fonts

* Resources – cssfontstack.com
* Google Fonts (available online)
* font-family – change font
* font-size – change font size
  + em – references parent
    - font-size: 2.0em;
* font-weight – thickness of font
* line-height – paragraph spacing
* text-align – alignment of text on page
* text-decoration – over/underline, strikethrough

Custom Fonts

* fonts.google.com
* choose font, add to cart – embed link and add as stylesheet
* remember to check page load!!

Introduction to the Box Model

* In a document, each element is represented as a rectangular box
* Components
  + Margin Edge – between border and outside
  + Border Edge
  + Padding Edge – area between content and border
    - can be localized to left right above below, or all around by default
  + Content Edge