# Week 1 - HTML & CSS

* Intro to web development
* What is a browser?
* How does a browser show content?
* Basic Client - Server architecture
* What is web development?
* Course overview
* Introduction to HTML
* Setting up your environment
* Installing Sublime Text
* Installing Google Chrome
* Editors, Directory structure, Tools
* Creating a text document and displaying in a browser
* How to show file extensions in Windows
* Renaming your text file to show HTML
* **Exercise** Hello World: Your first web page
* Survey of HTML elements
  + Headings
  + Paragraphs
  + Images
  + Styles
  + Bold, Italic, Underline
  + Font: Family
  + Colors: Named and Hex
  + Size
  + Alignment
  + Links
  + Tables
  + Lists
  + Blocks: divs, spans
  + Forms
  + Text Field
  + Select
  + Radio Buttons
  + Submit Buttons
* **Exercise** About Me: Build a profile page about yourself with a form to contact
* Introduction to CSS
* Overview of the CSS syntax
* How to use CSS
  + Inline CSS
  + Embedding CSS into your HTML pages
  + Importing a CSS stylesheet
* CSS Selectors
  + Using selectors for Standard HTML elements
  + Using selectors with divs
  + Using selectors with spans
* Styling with CSS
  + Colors
  + Backgrounds
  + Borders
  + Margins & Padding
  + Size
  + Text Manipulations
* CSS Boxing For Layouts
* Responsive/Mobile design basics
* **Exercise** Style your about site. Create multiple pages with links. Add a responsive image gallery showing photos of you and your family.
* **HOMEWORK:**:
  + Review HTML/CSS on the W3 website and complete the exercises
  + https://www.w3schools.com/html/
  + https://www.w3schools.com/css/
  + Continue working on your about me website
  + Photos/descriptions of your "Favorite things"
  + At least 5 links to other website you like
  + Include anything else that reflects you

# Week 2 - Web Design

* Survey of CSS Frameworks
* Introduction to Bootstrap
* Ways of using bootstrap in your web applications (downloading vs CDN)
* Survey of bootstrap elements
* **Exercise**: Improve the aesthetic of your personal site by using bootstrap elements.
* TBD...

# Week 3 - Introduction to Coding With Javascript

* Intro to JS
* What is Javascript and what's it's role in web development
* Setting up your editor/tools to write javascript
* Basic syntax / Setting up your webpage to write Javascript
* Placement (head, body, external file)
* Displaying information (webpage, console, alert)
* Opening the Javascript Console
* Writing information to the Javascript Console
* **Exercise** Your first Javascript app: Hello World!
* Coding Basics
* Variables
  + What is a variable?
  + Survey of primitive datatypes in Javascript
  + string
  + number
  + boolean
  + null
  + undefined
  + Variable naming convention
  + **Exercise** Refactor your "Hello World" app to print "Hello [Name]" where [Name] is stored in a variable
* Operators
  + Assignment
  + Arithmetic
  + Increment
  + Comparison
  + typeof
  + **Exercise** Drum machine: Create a basic drum machine that plays different sounds when each drum has been clicked.
* Basic Flow Control
  + If/Else Statement
  + Switch Statement
* Functions
  + Definition
  + Calling Functions
  + Passing in data (parameters)
  + Returning data
* Introduction to the P5.js library and the canvas
  + Setting up your development environment to support P5
  + Drawing basic shapes
  + Interactions
  + Basic physics
  + **Exercise** Flappy Bird - Create a basic clone of flappy bird using basic shapes
* **HOMEWORK:**: Create 2 Javascript programs
  + Larger Number: Have the user enter in two numbers into an input field and output which number is larger
  + Complete the flappy bird game using your own graphics

# Week 4 - Introduction To Coding With Javascript Continued

* Intro to JS Continued
* Coding Basics Continued
  + Loops
  + For-Loops
  + While-Loops
  + Loop Control (break, continue)
  + **Excerise** Guessing Game: Ask the user to think of a number between 1 and 100. Then have the computer make guesses and ask the user if their number is higher or lower than the guess. The computer will then guess the difference / 2 or the difference \* 2 until they correctly guess the user's number.
* Intermediate Programming in Javascript
* Arrays
  + Declaration
  + Indexing
  + Length
  + Looping
* Introduction to Object Oriented Programming
* What are objects and how are they used?
* Declaring objects in Javascript
  + Properties
  + Methods
  + Constructors
  + This keyword
  + Prototype
  + Adding properties to objects
  + Adding methods to objects
  + Inheritance
* Manipulating the DOM with Javascript
  + Creating elements
  + Appending elements
  + Styling elements
  + Getting/Setting elements
  + Finding elements
  + **Exercise** Tic-Tac-Toe
* Debugging 101
  + Introduce Chrome Developer Tools
  + Elements Tab
    - DOM elements and associated properties
  + Network Tab
    - Monitor incoming/outgoing HTTP requests
    - Tracking AJAX requests
  + Sources Tab
    - Adding breakpoints
    - Stepping through script execution
  + Console
    - Running commands directly in the console
    - Using console.log(); for debugging
    - Handling errors (effective Googling, Stack Overflow use)
* **HOMEWORK:**
* Create 3 Javascript programs
  + Digital Clock: Create a program that draws a working digital clock in the browser. Bonus: Create an "alarm" that pops up an alert at a given time.
  + Average: Write a function that takes in 3 numbers and returns their average.
  + Hello World Translator: Write a function helloWorld that takes in a language and returns 'Hello World' in that language. Your program should work with at least 3 languages. *example:* helloWorld('spa') => Hola mundo

# Week 5 - Introduction to Server-side Development Using Node.js

* What is "server side" development?
* Overview of how a web server works and delivers content
* Survey of the different server-side development technologies and why we chose to teach node.js
* Introduction to the command line
* Putty for Windows, iTerm for Mac
* Basic Commands and Navigation
* Creating and deleting folders
* Introduction to Node.js
* Setting up the tools / environment
* Overview of the node file structure
* **Excercise** Hello World - Your first server
* Introduction to Node Package Manager
* What are package managers and why do we need them?
* Installing Node packages using npm
* **Exercise** Your first file server. Using the Node package node-static, write a server that serves up your about website.
* Introduction to the MVC Design pattern
* What is MVC?
  + Model
  + View
  + Controller
* How does the MVC design pattern apply to Node.js applications?
* How do you structure a node project to support MVC?
* Introduction to Express: Routing
* Where does Express fit into the MVC equation?
* Creating a basic express application from the express template.
* Basics of routing
  + HTTP Verbs
  + GET
  + POST
  + PUT
  + DELETE
* Serving dynamic content from routes
* **Exercise** - Create a multi-page website using at least 4 routes. POST form content to one route and display it.

# Week 6 - Creating Multi-page Websites Using Node.js & Intro To Version Control

* Creating View Templates
  + What are view templates and why should we use them?
  + Survey of the different view templates (EJS, Jade, Pug)
  + Refactoring your multi-page site to use view templates
  + **Exercise** - Convert your multi-page site to using view templates
* Models
  + How to create model objects in Node.js
  + How to import and use those models in your application
  + **Exercise** - Favorite things website - Create an MVC website displaying a list of your favorite things using a Favorite model
* Introduction to version control with Git
* What is Git/GitHub and why should we use them?
* Downloading and installing Git
* Setting up GitHub account
* Basic Git workflow/useful commands
  + git init
  + git clone
  + git add
  + git add \*
  + git commit -m "Commit message"
  + git push
  + git branch
  + git checkout
  + git pull
  + git merge
* **Homework**
  + Publish your about me website to github pages https://pages.github.com
  + Sign up for https://www.freemysqlhosting.net and create your first database

# Week 7 - Databases and Data Persistence

* What is a database?
* Why do we need databases
* An introduction to MySQL
* Getting the tools
* Installing Sequel Pro
* Connecting to a database
* Database tables
* Database Columns
  + Datatypes
* Database rows
* Basic Sequel Statements
  + INSERT
  + SELECT
  + UPDATE
  + DELETE
* Setting up Node.js to talk to a database
* Survey of libraries
* Setting up Sequelize
* Connecting to your mysql database
* Creating your first model
  + Creating
  + Searching
  + Updating
  + Deleting
* **Exercise/HOMEWORK**
* Create a basic blog engine using Express + Sequelize
  + Create/Edit/Delete posts
  + POST /posts
  + List posts
  + GET /posts
  + List a single post
  + GET /posts/{post\_id}

# Week 8 - Creating And Interacting With JSON APIs: The language of the Internet

* What is REST?
* POST vs PUT (or PATCH, if you guys prefer PATCH)
* DELETE
* This is only assuming we didn't go over these http methods in great detail back in week 4
* How do modern web applications talk to each other
* What is JSON?
* How does JSON work with Javascript?
* **Excercise:** Building an API on top of your blog
* Show how to send back JSON data
* Allow Posts to be viewed and created via your api
* Intro to JQuery & Asynchronous Javascript
* Basic JQuery syntax
* Using the $ operator and gaining access to DOM elements
* Manipulating DOM elements
* Interacting with web APIs
* Converting form data to JSON
* **HOMEWORK**
* Update your blog admin to create posts using your API without refreshing the page using $.post

# Week 9 - Deployment & Project Pitches

* Differences between staging/local and production
* Setting up a basic web server
* Setting up a Digital Ocean instance http://digitalocean.com
* A survey of the different server types
* Creating a new server instance
* Creating an ssh key-value pair
* ssh-ing into your server
* Installing node.js on your server
* Deploying your blog API application
* Connecting your server to github
* Tuning your application to run in production
* Brainstorming ideas
* Project pitches
* Get approximately 3 minutes to pitch idea
* Project voting
* Everyone gets 3 votes
* Spread out votes across ideas or stack them all on one (including their own)
* Top 3 - 4 ideas are what they work on (give or take, just make sure there's 3 to 4 people per team)
* Identifying a Minimum Viable Product (MVP)
* Pare it down to 3 sprints
* Creating a requirements document
* Creating a features list
* Data modeling
* Designing necessary routes
* **HOMEWORK**
* Come up with 3 project ides for community project.
  + Be prepared to pitch your best one
  + Have a 3 slide presentation ready to show the class
  + Make sure that the project is relevant to your community and can be achieved in 3 weeks

# Week 10-12 - Intro to Scrum Project Management & Community Project

* Intro to SCRUM Project Management
* Populating the backlog with project features
* Sprint Planning
* Daily Standups
* Sprint Retrospective
* **Exercise** Sign up for Trello http://trello.com and populate the backlog with your team's requirements
* Sprint 1 (Week 10)
* Begin work on wireframes of the community project
* Start building server and setting up toolchain
* Complete data modeling and start on API work
* Sprint 2 (Week 11)
* Build Data base
* Complete API work
* Work on front-end of community project
* Complete
* Sprint 3 (Week 12)
* Complete application implementation
* Bug Fixes
* Prepare presentations
* **Exercise/Homework**
* Creating a presentation / demo to show off the application during "demo day"
* Presenting your team's project
* Presenting your personal project