



Course Outline: Basics This is a rough draft of the course outline I'm thinking about developing. It still needs a lot of work but it's a good start.

1. Introduction * What is the web? * A little history * Components of the web * What the course will not cover * A Technical Overview * Course Project: Blog * What * Why * How
2. HTML * What is HTML? * Markup Language * Differences with CSS and Javascript * Elements by groupings * Structure * Styles * Multimedia * Linking and Embedding * Course Project * Build the Markup
3. CSS (Structure based on [Web.dev's Learn CSS](<https://web.dev/learn/css/>)) * Box Model * Specificity and the cascade * Stacking Context and Z-index * Selectors * Inheritance * Color * Measuring units in CSS * Absolute units * Relative units * Viewport units * Layouts * Flexbox * Grid * Multi column * Spacing * Pseudo Elements and Pseudo Classes * Focus * Advanced Features * Borders and Shadows * Gradients * Animations * Filters * Blend modes * Logical Properties and writing modes * Course Project * Style the blog
4. Javascript * Javascript history * Unusual Inheritance model * About Javascript versions * Annual release * Browser support * [caniuse.com](<https://caniuse.com/>) * Values ([Eloquent Javascript](https://eloquentjavascript.net/01_values.html)) * Numbers * Special numbers * Strings * Booleans * Logical Operators * Empty Values * Null * Undefined * Program structure * Functions * Expressions and statements * Variables and Constants * Difference between var, let and const * Return values * There is always a return value * Control flow and conditional execution * If / Then / Else * While / Do * For * Switch * Functions ([Eloquent Javascript](https://eloquentjavascript.net/03_functions.html)) * Defining * Function Expressions * Function declarations * IIFE * Arrow functions (ES6) * Shortcomings * Classes * Bindings and Scopes * Modules ([Eloquent Javascript](https://eloquentjavascript.net/04_modules.html)) * Special type of script * Import / Export * Different types of modules * CommonJS * ECMAScript Modules (ESM) * Javascript on the browser * The DOM * Looking for elements on the page * `getElementBy{id, class, tagName}` * `querySelector` and `querySelectorAll` * Using scripts on a page * Inline * Linking * Events * Event listeners * Event handlers
5. Asynchronous Programming (pull the content from ``mdn-async-js.md``) * `setTimeout` and `setInterval` * Callbacks * Promises * Async / Await
6. Javascript on the server * Node.js * History: What came before and why now * How it works * Advantages and Disadvantages
7. Building a server with Express.js (1) * Why not a framework? * Endpoints and actions * APIs * REST APIs * Database Integration * Firebase * MongoDB * PostgreSQL * Running the server * Course Project * Build a server for the blog * Pick a database to integrate the server with
8. Building a server with Express.js (2) * APIs * Validation * [Express Validator](<https://express-validator.github.io/docs/>) * Authentication * [Passport.js](<http://www.passportjs.org/>) * Course Project: Validate your data before submission * Course Project add authentication for adding and editing content for the blog
9. Converting the blog into a PWA * The PWA concept * The manifest file * The service worker * The offline page * Course Project: Add a Manifest file * Course Project: Add a service worker
10. Publishing your work
11. Accessibility