

Web Dev 101 - Frontend Fundamentals (5-12wk)

- Install and Configure a Code Editor (1d)**
 - Install code editor, install helpful packages & plugins
 - Learn basics of using a code editor and how to get around
 - Build a first basic webpage
- HTML (1-3d)**
 - How HTML tags and attributes work
 - The main 20 tags
 - HTML folder structure and how to navigate between pages
- Basic CSS (2-6wk)**
 - <style> tags, <link> tags, style attributes
 - Creating CSS rules, how CSS selectors work, changing basic properties
 - Get really good at drawing boxes (layouts) with flexbox and grid
 - Start learning the "tricks" to doing things you don't know how to do
 - e.g. show an image as a circle, fade colors when you hover
 - Using a CSS reset
 - Learn to build entire websites from just a design/mockup
- JavaScript Fundamentals (2-6wk)**
 - Programming basics: variables, functions, flow control (conditionals, loops)
 - Querying, selecting, working with DOM (HTML) elements
 - Event listeners and doing things in JS as users interact with your webpage
 - XHR requests to load data into your app
 - How to embed other scripts into your pages
- Understand How a Webpage Works (1d)**
 - What is DNS?
 - What is a webserver/webhost?
 - What happens when I type somewebsite.com into a browser and get a webpage back?
 - What process does the browser go through to turn html into a visible page?
- How to publish websites at this point?**
 - Hostgator: SIMPLE & cheap way to get a website up
 - Video: [How to publish a website](#)
 - Link: learncode.academy/hostgator - 100% simple, cheap too.
 - Promo Codes: DROP25PERC, DROP10BUCKS, LETSMAKEADEAL

access/print this doc at <https://learncode.academy/webdev>

Web Dev 102 - Intermediate Frontend (6-16wk)

- Basic Terminal Usage (1-3d)...that's right, you're going to stop fearing the terminal now**
 - Navigating your file folders using the terminal
 - How to run commands in the terminal
 - How to change and configure your shell
- Installing Node.js packages (1-3d)**
 - Install Node.js
 - What are packages (utilities, libraries, frameworks), and how to install global packages
 - How to install NPM packages for your project
- Version Control with Git (1-2wk)**
 - SSH: what is it, and how to make a keypair
 - Git: what is version control, why it's essential, how Git does it, Git basics
 - GitHub: creating, cloning, forking remote repos
- Intermediate CSS (2-6wk)**
 - Media queries & responsive layouts
 - Responsive images
 - Write SASS instead of CSS & compiling SASS code into CSS code
 - Break your CSS into modules for better organization
 - Using CSS frameworks like Tailwind or Bootstrap
- Intermediate JavaScript (2-6wk)**
 - Asynchronous programming: callbacks, promises, async/await
 - Intermediate coding concepts: classes, scope, context, first-class functions, closures
 - Persisting state: cookies vs local storage vs session storage
 - Break your JS code into modules for better organization
 - Transpile & minify your JS with Webpack
 - Using NPM packages in your JS code
- Code Quality Basics (1-2wk)**
 - Lint your code with ESLint
 - Format your code with Prettier
 - Type-check your code with TypeScript
 - Write JS tests with Jest
- How to publish websites at this point?**
 - GitHub pages: it's FREE!
 - Netlify: even better (only public GitHub repos are free)
 - Vercel

access/print this doc at <https://learncode.academy/webdev>

Web Dev 103 - Advanced Frontend (6-12wk)

Static Site Generators (1-2wk)

- Build a statically-generated HTML site with Eleventy
- Deploy an Eleventy site to Netlify (free)

The Basics of a single page app (1-2wk)

- Using modular JS to build a basic single page web app
- Memory leaks in JS and how to avoid them
- Hash-based and HTML5 routing

React Basics (1-2wk)

- Make a React page with create-react-app
- Basic state management with useState
- Learn how/when to use useEffect
- Async actions in React & fetching data from an API
- Understand what is actually happening with React (unidirectional data flow, virtual DOM, etc)

CSS in React (1-2wk)

- Inline CSS with POJOs (plain old JavaScript objects) & CSS imports
- CSS Modules
- styled-components
- Tailwind.css (Recommended!)

Routing & Frameworks (1-2wk)

- React Router
- Next.js (Recommended!)

Advanced State Management Libraries (1-2wk)

- Redux (not recommended, but most popular)
- MobX (Recommended!)
- MobX State Tree (if you need more structure for MobX e.g. a massive app)

How to publish websites at this point?

- GitHub pages: it's FREE!
- Netlify: even better (only public GitHub repos are free)
- Vercel (Recommended!)