

WEB DEVELOPMENT TREND

A Road map for those considering a career in web development



Content Overload Warning

TYPE OF WEB DEVELOPER

Front end developer

Client facing websites / UI apps

HTML, CSS, JS, TypeScript, JS frameworks, DOM, HTTP, etc

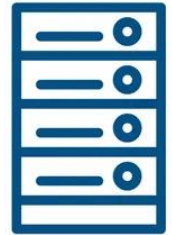


Backend developer

Server APIs, data, tasks

JS, Python, C#, PHP, etc.

Databases – relational, NoSQL, server configurations etc



Full Stack Developer

Combine the above two

Front End

- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation

Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

CAREER GOALS?

Professional dev at a company

Freelancer / consultant / own business

Building your own apps / SaaS

Content Creation

Hobby

BASIC TOOLS

Computer / OS

Mac, Windows, Linux

Text Editor

VS Code, VIM, Atom, Sublime Text

Web Browser

Chrome, Firefox, Safari, Edge

Terminal

Default, Hyper, Powershell, Git Bash

Design <optional>

Adobe XD, Photoshop, Sketch, Figma (mostly of exclusive UI design)

HTML AND CSS (FRONT END AND FULL STACK)

HTML 5 page structure and semantic tags

Basic CSS styling

Positioning

Alignments

Transitions and animations

Responsive design / media queries

CSS / UI FRAMEWORKS

Options available:

- Tailwind CSS

- Bootstrap (recommended)

- Materialize

- Bulma

Learn fundamentals of CSS first and then use these options to speed up the process of UI design

UI DESIGN

Things to keep in mind

Color and contrast

White Space

Scale

Visual hierarchy

Typography

JAVASCRIPT

Basics

Variables, arrays, functions, loops

DOM and Styling

Selecting and manipulating elements

Array Methods

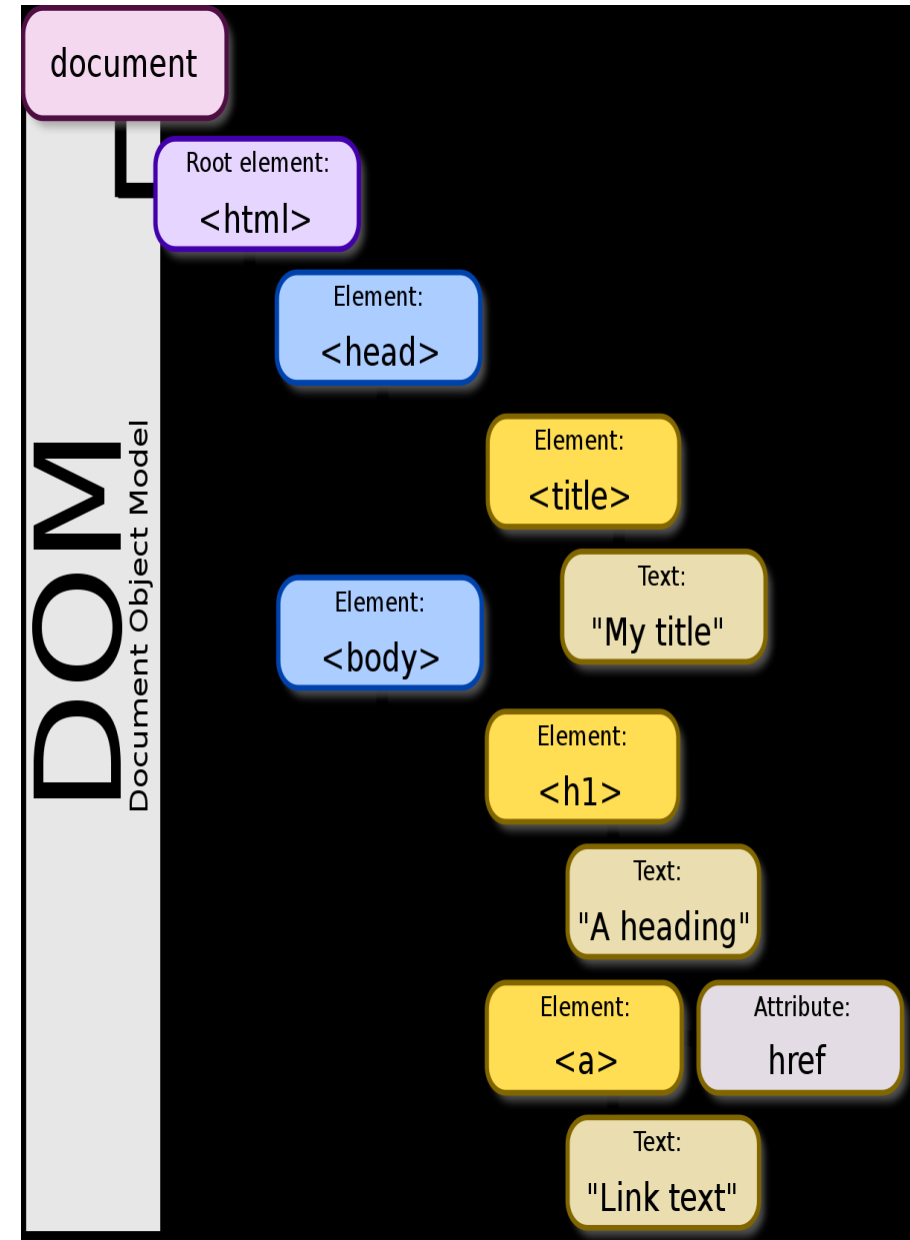
foreach, map, filter, reduce, etc

JSON

JavaScript Object Notation

HTTP Requests

Fetch API – GET, POST, PUT, DELETE



OTHER TOOLS OF IMPORTANCE

Version Control

Git, Subversion

Repo Manager

GitHub, Bitbucket, GitLab

Package Manager

NPM, Yarn

Module Bundler (only for Front end)

Parcel, Webpack, rollup

Browser Developer Tools (Console, network, storage, etc)

Editor extensions and helpers (linting, prettier etc)

BASIC FRONTEND DEPLOYMENT

Static hosting

Netlify, github pages, heroku

CPanel Hosting

InMotion, Hostgator, Bluehost

Methods of deploying

Git: continuous deployment by pushing to a repo

FTP / SFTP: File transfer protocols <not fast enough>

SSH: secure shell (using the terminal)

BASIC FRONTEND DEPLOYMENT

Domain Names

Namecheap, google domains, enom

Email Hosting

Namecheap, zoho mail, Cpanel

SSL certificates

Let's Encrypt, Cloudflare, Namecheap

FOUNDATIONAL FRONTEND DEVELOPER TASKS

Setup effective and efficient development environment

Code in HTML, CSS and JS

Use of a CSS framework

Develop responsive layouts

Develop websites with dynamic functionality and work with DOM

Connect to 3rd party APIs with fetch

Comprehend basic HTTP

Use Git with GitHub or some other Git repo

Deploy and manage a website or a small web app

BEYOND THE BASICS – FRONTEND DEVELOPMENT

Become proficient in JavaScript

Learn some front end frameworks

React, Vue, Angular

Learn a server side language / technology

Node JS, Python, PHP, C# etc

FRONTEND FRAMEWORKS

Become proficient in one of the following frontend frameworks

React: Currently most popular

State management using: Context, Redux, MobX

Vue

State management using: Vuex

Angular

State management using: Shared Service, NgRx

Svelte

State management using Context API

TYPESCRIPT

A superset of JS and can be paired with Frontend frameworks

Introduces strict type system to JS

Makes code robust and less prone to errors

OOP support – classes, interfaces, generics, modules

A good option for very large projects

TESTING YOUR PROJECT

Unit tests

Testing individual units like functions and classes

Integration tests

Modules being tested as a group

End to end tests

Test your workflow from start to end

Some example Testing frameworks

JS: Jest and Mocha

Python: PyTest and Robot

SERVER SIDE RENDERING

The option of running frameworks like React and Vue on the server for better SEO (search engine optimization), easy routing etc.

Next.js (React)***

Nuxt.js (Vue)

Angular Universal (Angular)

STATIC SITE GENERATORS (SSG)

Generates website pages at build-time as opposed to real-time. Making them very fast and secure

Gatsby (React based)**

Gridsome (Vue based)

11ty (JS alternative to Jekyll)

Jekyll (Ruby based)

Hugo (Go based)

HEADLESS CMS

Backend only content management systems that is typically used with SSGs

Strapi ***

Sanity.io

Contentful

Prismic

Wordpress

JAMSTACK – JS, APIS & MARKUP

Web architecture with high performance, security, scalability at a low cost and good developer experience

Static sites and assets

Markdown

Serverless

Headless CMS for content

Hosting with services like Netlify

THE FRONTEND DEVELOPER RESUME

Can build apps and interfaces with a frontend framework

Work with component and global state

Connect to backend JSON data and integrate it into the apps used

Write and test clean and efficient code

Optional:

- Use of TypeScript

- Server side rendering

- Static site generators / Jamstack

SERVER SIDE LANGUAGE / BACKEND

Focus is on data, modeling and HTTP requests/responses.

Pick a language:

Node.js (JavaScript based)

Deno (JavaScript based)

Python

C#

GoLang

Ruby

PHP

Java

Kotlin

SERVER SIDE FRAMEWORKS

Pick a framework:

Node: Express, Koa, Nest, loopback

Python: Django, Flask

PHP: Laravel, Symfony, Slim

C#: ASP.NET

Java: Spring MVC

Ruby: Ruby on rails, Sinatra

Kotlin: Javalin, KTor

MANAGE YOUR DATA -

Pick a Database

PostgreSQL

MongoDB

MySQL

MS SQL Server

Firebase

Elasticsearch

THE RISING STAR - GRAPHQL

Query language for your API

Send a query similar to JSON to your API and get precisely what you need

Setup a GraphQL server and query using a client like Apollo

Easily integration with React and other frameworks

REAL-TIME TECHNOLOGIES

Socket.io allows for real-time bidirectional communication

In today's apps, it is sort of expected that you have some real time features

Instant messaging and chat

Realtime analytics

Document collaborations

Binary streaming etc.

DEPLOYMENT, SERVERS AND DEVOPS

Hosting Platforms

Heroku, Digital Ocean, AWS, Azure

Web Servers

NGINX, Apache

Containers

Docker / Kubernetes, Vagrant

Image/ Video

Cloudinary, S3

Continuous Integration (CI) / Continuous Delivery/Deployment (CD)

Jenkins, Travis CI, Circle CI

THE FULL STACK DEVELOPER RESUME

Can develop both frontend UIs and Servers

Knowledgeable of a server side language and technology

Knows of working with and developing databases

Comprehends HTTP and can develop RESTful APIs

Can deploy <successfully> full stack projects

Very comfortable with the terminal

GOING BEYOND – MOBILE DEVELOPMENT

You can jump into mobile app dev using web app dev related technologies

Flutter / Dart

React Native

Ionic

Xamarin

Kotlin

Swift

PROGRESSIVE WEB APPS (PWA)

Web apps with native feel with respect to layout, functionality, etc irrespective of the device platform

For all device screen sizes

Offline content

HTTPS

Native experience

DESKTOP APPS WITH WEB TECHNOLOGIES

Web technologies that can be used to create desktop apps

Electron

NW.js

Python and Tkinter

OTHER SUBJECTS OF IMPORTANCE

Algorithm Design and Analysis

Improve logical and critical thinking by many folds

Knowledge of data structures