

The background of the slide features a slightly blurred image of a computer keyboard. Above the keyboard, the faint outlines of what appear to be code snippets or lines of text are visible, suggesting a theme related to programming or web development.

A Practical Guide

Web Development In 2019

Ajay Bhatia

About This

What Is In This Guide?

This is a step by step guide to all of the major technologies that are involved in becoming a front-end, back-end or full stack developer in 2019

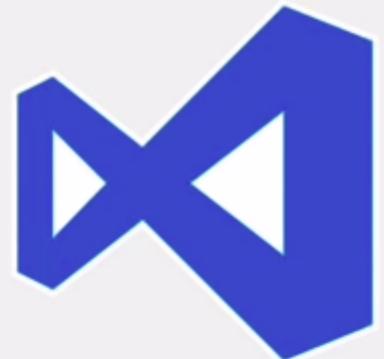
- You **do not** need to learn all of these technologies to be a web developer.
- This guide simply lays out options in a categorized way to help you find your path
- I will give my advice and suggestions based on experience and reference
- We will first look at the common needed technologies then look at the 2019 hot trends & predictions

Getting Started

Basic Software & Tools

Basic tools for web development

- Text Editor or IDE [VSCode](#), [Sublime Text](#), [Atom](#)
- Web Browser [Chrome](#), [Firefox](#)
- Design & Mockup (Optional) [XD](#), [Photoshop](#), [Sketch](#), [Figma](#)
- 3rd Party Terminal (Optional) [Git Bash](#), [WSL](#), [iTerm2](#), [Hyper](#)



The Building Blocks

HTML & CSS

Always the very first thing to learn

What To Learn

- Semantic HTML5 Elements
- Basic CSS (Positioning, box model, etc)
- Flexbox & CSS Grid
- CSS Variables (Custom Properties)
- Browser Dev Tools



Suggested YouTube Videos:

- HTML Crash Course For Beginners
- CSS Crash Course For Beginners
- CSS Grid Crash Course
- Flexbox In 20 Minutes

Suggested Udemy Course:

- Modern HTML & CSS From The Beginning

The Building Blocks

Responsive Layout

HTML



CSS



Responsive layouts are no longer a luxury, they are a necessity

- Set viewport
- Fluid widths
- Media queries
- rem over px
- Mobile first, stacked columns



Suggested YouTube Videos:

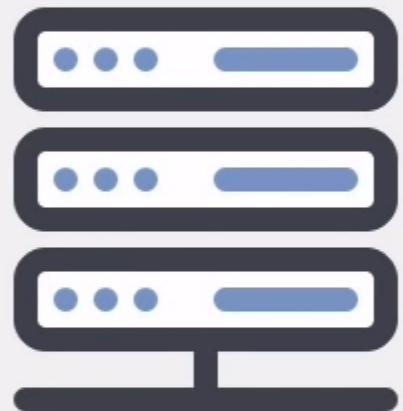
- Build an HTML5 Website With A Responsive Layout
- Responsive Landing Page
- Build a Responsive Mobile First Website
- Pluralsight Login Page Clone

Make It Live

Basic Deployment

Learn to deploy a static website

- Register a domain name [Namecheap](#), [Google Domains](#)
- Managed shared hosting or VPS [Inmotion](#), [Hostgator](#), [Bluehost](#)
- FTP, SFTP File Upload [Filezilla](#), [Cyberduck](#)
- Static Hosting (Need to know Git) [Netlify](#), [Github Pages](#)



Suggested YouTube Videos:

- Web Hosting & cPanel Guide
- Deploy Websites In Seconds With Netlify
- Github Deploy & Domain

Sass Pre-Processor

Not mandatory but recommended. It is easy enough to learn the basics

- Structured CSS
- Variables
- Nested CSS
- Mixins & Functions
- Inheritance



Suggested YouTube Videos:

- Responsive Portfolio Website Series
- Fancy Form UI From Scratch
- Sass Workflow With Gulp

Get Dynamic

Vanilla JavaScript

Start learning JavaScript **without** any library or framework

- Data Types, functions, conditionals, loops, operators
- DOM Manipulation & Events
- JSON
- Fetch API
- ES6+ (Arrow functions, promises, async / await, destructuring)

JS

Suggested YouTube Videos:

- JS Fundamentals For Beginners
- JavaScript DOM Crash Course
- Vanilla JS Playlist (Projects)

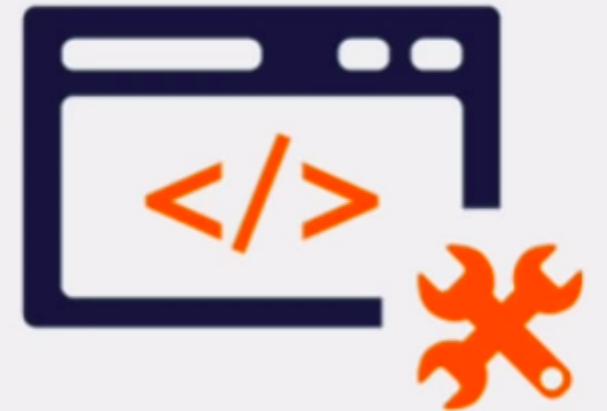
Suggested Udemy Course:

- Modern JS From The Beginning

You're skill set

Basic Front-End Web Developer

- Build static websites (Small business, informational)
- Build UI layouts (Take a design and create the html/css)
- Add dynamic functionality (modals, slideshows, etc)
- Deploy and maintain websites



It is possible to find an entry level job, but you are most likley going to need to know more.

Freelancing is a possibility...

Now what?

What To Learn Next

What to do next depends on what you want to do. You have a few choices

- HTML / CSS Framework
- Git & Tooling
- Front End JS Framework
- Server Side Language & Database



Easier UIs & Prototyping

HTML / CSS Framework **Choose One**

HTML / CSS frameworks are becoming a bit less relevant but I would still suggest learning one. They can make your life much easier and great for prototyping

- Bootstrap
- Materialize
- Bulma



- Suggested YouTube Videos:**
- Bootstrap Crash Course
 - Materialize Crash Course
 - Bulma Crash Course

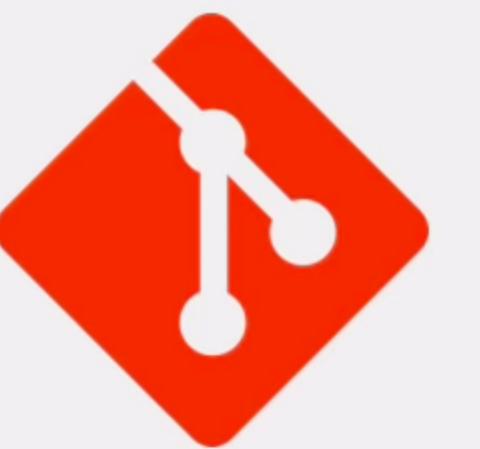
- Suggested Udemy Courses:**
- Bootstrap 4 From Scratch
 - Materialize CSS From Scratch

Your Workflow

Git & Tooling

Git is absolutely necessary for all web developers. Here are some other tooling suggestions

- Basic Command Line [Used for so much including these tools](#)
- Git [Version control](#)
- [NPM](#) or Yarn [Installing packages](#)
- Webpack or [Parcel](#) [module bundling](#)
- [Gulp](#) or Grunt [task runners](#)
- Editor Extensions [ESLint, Prettier, Live Server, etc](#)

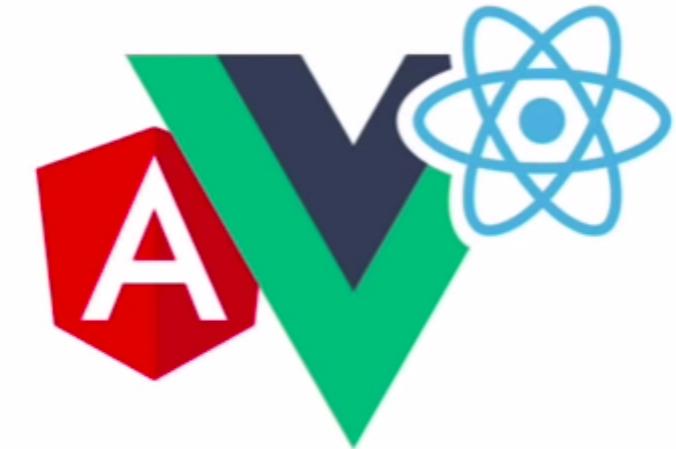


- Suggested YouTube Videos:
- Linux Command Line Basics
 - Git Crash Course
 - NPM Crash Course
 - Yarn Crash Course
 - Webpack Crash Course
 - Gulp Crash Course
 - VSCode Setup Video

JavaScript

Front-End Framework **Choose One**

It is becoming a necessity to learn a JS front-end framework



- Very popular in the industry
- More interactive & interesting UIs
- Components & modular front end code
- Good for teams

React - Most popular in the industry

Vue - Easy to use and really gaining traction

Angular - Fading A Bit - Used in enterprise

Suggested YouTube Videos:

- React Crash Course
- Vue.js in 60 Minutes
- Angular in 60 Minutes
- Project Series

Suggested Udemy Courses:

- React Front To Back
- Angular Front To Back

JavaScript

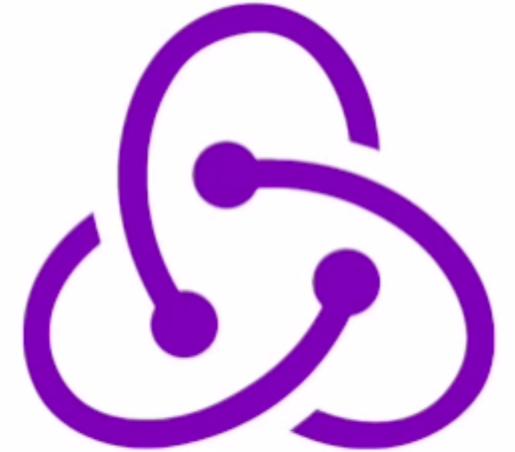
State Management

For larger apps with a framework, you may need to learn methods to manage app-level state

- Redux, Context API
- Apollo (GraphQL Client)
- VueX
- NgRx

Concepts

Immutable State
Stores
Reducers
Mutations
Getters
Actions
Observables



Suggested YouTube Videos:

- Redux Crash Course
- React & Context API Project

Suggested Udemy Courses:

- React Front To Back
- MERN Stack Front To Back

You're skill set

Full Fledged Front-End Web Developer

- Build incredible front-end applications
- Smooth & steady front-end workflow
- Work well with teams & familiar with Git
- Connect to backend APIs & work with data



You should be able to get a front-end job or work for yourself and do pretty well



Backend Programming

Server Side Language **Choose One**



To be a full stack or software engineer, you will need to learn a server-side language/technology



- [Node.js](#)
- Python
- PHP
- C#
- Go

Things To Learn

- Fundamental Syntax
- Structure & Workflow
- Package Management
- HTTP / Routing

Suggested YouTube Videos:

- Node.js For Beginners
- Python Fundamentals
- PHP Front To Back
- C# Project in 60 Seconds
- Go Crash Course

Suggested Udemy Courses:

- Node.js Dev To Deployment
- MVC PHP OOP
- Python Django Dev To Deployment

express

Backend Programming

Server Side Framework Choose One



Do not reinvent the wheel. Learn a framework to build better and faster

django

- Node.js
- Python
- PHP
- C#

[Express](#), [Koa](#), [Adonis](#)
[Django](#), [Flask](#)
[Laravel](#), [Symfony](#)
[ASP.NET](#)

Suggested YouTube Videos:

- Express Crash Course
- Koa Crash Course
- Adonis Series
- Django Crash Course
- Flask Series
- Laravel Series
- Symfony Series
- C# .NET App in 60 Minutes

Suggested Udemy Courses:

- Node.js Dev To Deployment
- Python Django Dev To Deployment

Database **Choose One Or Two**



Most applications will use some kind of database. There are different types, here are some options



mongoDB

- Relational Database
- NoSQL
- Cloud
- Lightweight

[MySQL](#), [PostgreSQL](#), [MS SQL](#)

[MongoDB](#), Couchbase

[Firebase](#), AWS, Azure DocumentDB

[SQLite](#), [NeDB](#), [Redis](#)



Suggested YouTube Videos:

- SQL Crash Course
- MongoDB In 30 Minutes
- Redis Crash Course

Server Rendered Pages



Frameworks like React, Vue & Angular can also be rendered on the server which can actually make things relatively easier

- Next.js React
- Nuxt.js Vue
- Angular Universal Angular

Suggested YouTube Videos:
• Next.js Crash Course

Fast Projects



CMS Choose One



Content management systems allow for quick development and give your clients the ability to update their content. May not be a bad idea to pick one up. **Great for freelancers**

= ghost

- PHP Based
- JS Based
- Python Based
- .NET

[Wordpress](#), [Drupal](#)
[Ghost](#), [Keystone](#)
[Mezzanine](#)
[Piranha](#), [Orchard CMS](#)

Suggested YouTube Videos:

- Express Crash Course
- Koa Crash Course
- Adonis Series
- Django Crash Course
- Flask Series
- Laravel Series
- Symfony Series
- C# .NET App in 60 Minutes

Suggested Udemy Courses:

- Node.js Dev To Deployment
- Python Django Dev To Deployment

DevOps, Deployment & More

Learning languages and frameworks is one thing, setting up environments, testing & deployment is another

- Deployment.
- Platforms.
- Virtualization.
- Testing.

[Linux, SSH, Git, Server Software\(Nginx, Apache\)](#)
[Digital Ocean, AWS, Heroku, Azure](#)
[Docker, Vagrant](#)
[Unit, Integration, Functional, System](#)



- Suggested YouTube Videos:
- Getting Started With Docker
 - Vagrant Crash Course
 - Jest Testing
 - Deploy Node Apps

You're skill set

Full Stack Badass

- Setup full stack dev environments & workflows
- Build back-end APIs & micro-services
- Work with databases
- Construct full-stack apps (Front-end framework & server)
- Deploy to the cloud (SSH, Git, Servers, etc)



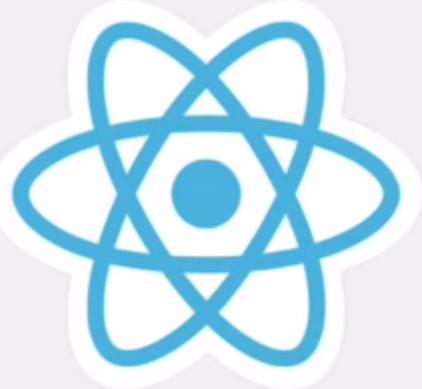
You are a frigging rockstar!

Beyond The Browser

Mobile Development **Choose One**

There are some frameworks that allow us to create native apps with web technologies

- React Native Build Native apps with React
- NativeScript Angular, TypeScript, JavaScript
- Ionic Hybrid apps with HTML/CSS/JS
- Flutter Mobile SDK for Android & iOS (Uses Dart)
- Xamarin Mobile apps with C#



Suggested YouTube Videos:

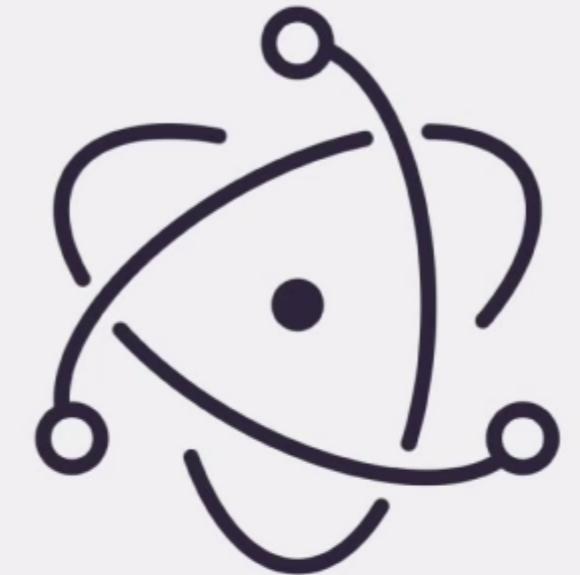
- React Native Crash Course
- Ionic Series

Beyond The Browser

Desktop Apps With Electron

Electron is used to build powerful cross-platform desktop applications using JavaScript

- Uses Chromium & Node.js
- Compatible with Windows, Mac & Linux
- Crash reporting, debugging & profiling



Suggested YouTube Videos:

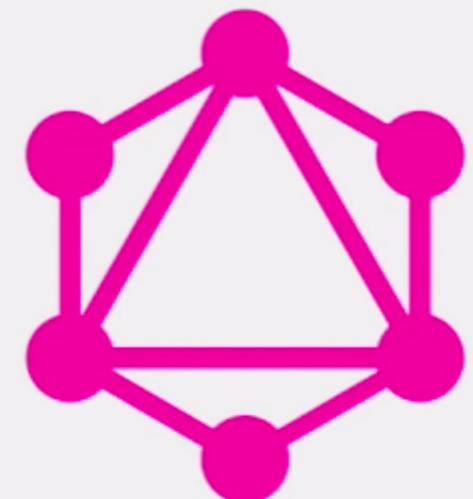
- Intro To Electron
- Build An Electron App

2019 Trends

GraphQL & Apollo

GraphQL is a revolutionary new way to think about APIs. Query language that is much less rigid than standard **REST**

- Ask for only what you want
- Front & back end can collaborate more smoothly
- Writing queries are very easy and similar to JSON
- Apollo is a client to make requests to a GraphQL server
- Used with the Gatsby static site generator



Suggested YouTube Videos:

- Build a GraphQL Server
- App With React, Apollo & GraphQL

2019 Trends

TypeScript

TypeScript is a superset of JS with additional features including static typing

- Types for variables, functions, etc
- Classes
- Other ES6-like features
- Used in Angular but can be implemented in React & Vue



Suggested YouTube Videos:

- TypeScript Crash Course

2019 Trends

Serverless Architecture

Eliminate the need for creating and managing your own server

- Use 3rd party services to execute “**Serverless Functions**” (FaaS)
- Examples are **AWS**, **Netlify** & **Firebase**
- Popular with **Gatsby** static sites
- Serverless framework **Toolkit for building serverless apps**



Suggested YouTube Videos:

- Lambda Functions With Netlify

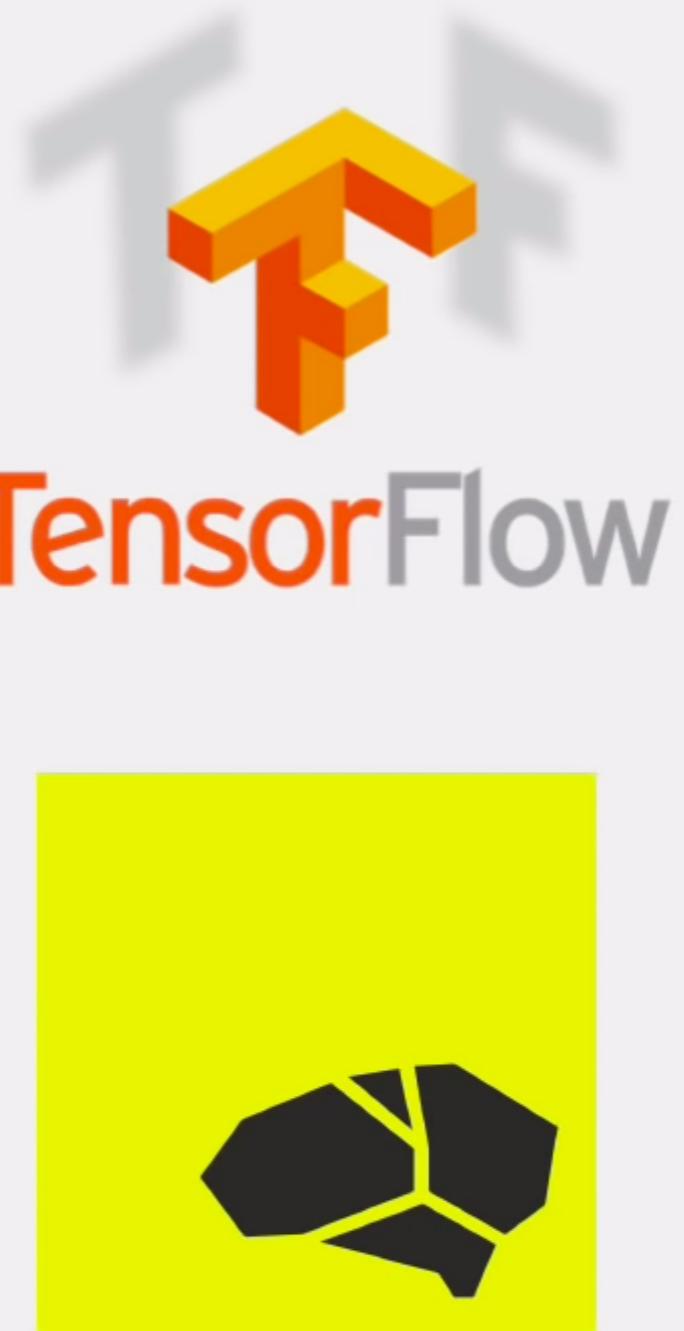
2019 Trends

AI & Machine Learning

AI & Machine Learning have been huge in almost every area of programming & technology including web development

- Machine learning can allow web apps to adapt over time
- AI has a long way to go but I suspect we will see more of it in web dev
- Used heavily in **Python** but we also have JS libraries like

Tensorflow.js and **Brain.js**



Suggested YouTube Videos:

- Machine Learning In JavaScript With Brain.js

2019 Trends

Blockchain Technology

Companies are using **Blockchain** for digital transactions in order to make them more efficient and secure

- Solidity Language for implementing contracts
- Mist Used for storing Ethereum, sending transactions & contracts
- Coinbase API Blockchain devs can easily build apps and integrate Bitcoin



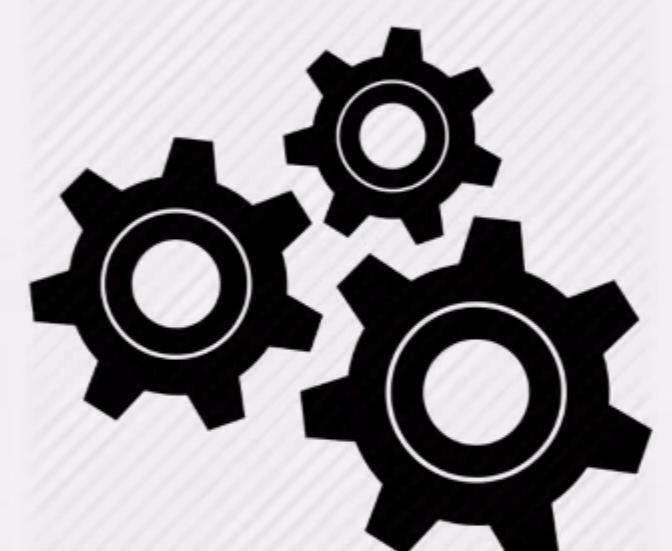
2019 Trends

PWA

Progressive Web Apps are regular web apps but give the user a native app experience in terms of layout and functionality

- **Responsive** to fit any form factor
- **Service workers** for offline availability
- App-like interactions
- HTTPS
- Reliable, Fast & Engaging

PWA



Suggested YouTube Videos:
• Intro To Service Workers

2019 Trends

Web Assembly

Assembly-like binary format for code that can be executed by web browsers. Can be generated from higher level languages like **C/C++ & Rust**.

- Faster than JavaScript
- Secure - Enforces same origin & security policies in the browser
- Open & debuggable



The End

Thank You

Follow Me On Social Media @profajaybhatia

