

# Forum:[Hack Forums](#)

Thread: [A Practical Guide: Web Development AZ 2019 trends](#)

**hackerkhan** - November 5, 2019, 7:39 am

## 1. Basic Software Tools:

[Basic tools for web development]

- ? Text Editor or IDE: VScode, Sublime text, Atom
- ? Browser: Chrome, Firefox
- ? Design & Mockup (Optional): XD, Ps, Sketch, Figma
- ? 3rd party Terminal (Optional): Git Bash, Wsb, iTerm2, Hyper-x

## 2. HTML & CSS:

[Always the very first thing to learn]

- ? Semantic HTML5 Elements
- ? Basic CSS (Positioning, box model, etc)
- ? Flexbox & CSS grid
- ? CSS variable (Custom Properties)
- ? Browser Dev tools

## 3. Responsive layout:

[Responsive Layouts are no longer a luxury; they are a necessity]

- ? Set Viewpoint
- ? Fluid Widths
- ? Media queries
- ? Rem over px
- ? Mobile First, stacked columns

## 4. Basic Deployment:

[Learn to deploy a static website]

- ? Register Domain name: Namecheap, Google Domains
- ? Managed Shared Hosting or VPS (Inmotion, Hostgator, Bluehost.
- ? FTP, SFTP File Uploads (FileZilla, Cyberduck )
- ? Static Hosting (Need to know Git) (Netlify, Github pages)

## 5. Sass pre -Processor:

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

- ? Structured CSS
  - ? Variables
  - ? Nested CSS
  - ? Mixins & functions
  - ? Inheritance
- {Learn From Youtube:
- ? Responsive Portfolio Website Series
  - ? Fancy Form UI From Scratch
  - ? Sass Workflow with Gulp}

## 6. JS: vanilla Java script:

[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)

#### 7. Basic Front-End Web Developer:

- ? Build static website (Small business , Informational)
- ? Build UI layout (Take a design and create HTML /CSS)
- ? Add Dynamic Functionality (modals /slideshows, etc)
- ? Deploy and Maintain Website

[It is possible to find an entry level job, but you are most likely going to need to know more. Freelancing is a possibility...]

#### 8. 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

#### 9. 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

#### 10. 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 (Installation Packages)
- ? Webpack or Parcel (Module bundling)
- ? Gulp or Grunt (Task runners)
- ? Editor Extension (ESLint, Prettier, Live Server, etc)

#### 11. Front end Framework:

[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

#### 12. 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

### 13. 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 back-end APIs & work with data
- [You should be able to get a front-end job or work for yourself and do pretty well]

### 14. 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

### 15. Server Side Framework (Choose One):

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

- ? Node.js ( Express, Koa, Adonis)
- ? Python (Django, Flask)
- ? PHP (Laravel, Symfony)
- ? C# (ASP.NET)

### 16. Database (Choose One or Two):

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

- ? Relational Database (MySQL, PostgreSQL, MS SQL)
- ? NoSQL (MongoDB, Couchbase)
- ? Cloud (Firebase, AWS, Azure DocumentDB)
- ? Lightweight (SQLite, NeDB, Redis)

### 17. 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)

### 18. 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]

- ? PHP Base (Wordpress, Drupal)

- ? JS Based (Ghost, Keystone)
- ? Python Based ( Mezzazine)
- ? .Net (Piranha, Orchard CMS)

#### 19. DevOps, Deployment & More:

[Learning language and frameworks is one thing, Setting up environments, testing & deployment is another]

- ? Deployment (Linux, SSH, Git, Server Software (Nginx, Apache)
- ? Platforms. (Digital Ocean, AWS, Heroku, Azure)
- ? Virtualization (Docker, Vagrant)
- Testing (Unit, Integration, Functional, System)

#### 20. Full Stack Badass:

- ? Setup Full Stack dev environment & Workflows
- ? Build Back-end APIs & micro-services
- ? Work with database
- ? Construct full-stack apps (Front-end framework & Server)
- ? Deploy to the cloud (SSH, Git , Servers, etc)

#### 21. 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#)

#### 22. 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