SPA vs MPA

Lesson Time: 30 Minutes

There are three approaches for building web sites.

1. Static HTML (Traditional)
   - The oldest way of creating pages. A website can have many html pages and all of the content is saved in the HTML.
   - Updates are done manually by editing the HTML. No dynamic HTML/JS is used to display content. JS might be used for visual effects only.
   - This approach is not suitable for creating web apps, and it's only good for small websites where the content does not change often, and there is no customized content.

2. Single Page Application - Dynamics HTML (SPA)
   - The newest approach to create web apps.
   - The web app is made up of only one single HTML file, and a large amount of JS scripts.
   - TypeScript and AngularJS or React are used to make all of the pages update and change. AJAX is used to get data from the web server without page loads. These tools are often used due to the complexity of SPA.
   - The biggest advantage of this approach is less page loads
   - The browser is responsible for most of the code, not the server.
   - The pages have dynamic content created by the web browser using JS, TypeScript, AngularJS, and/or React.
   - Examples are twitter, facebook, instagram.

3. Multi-page application using MVC - Dynamics HTML (MPA)
   - The most common way of creating web apps.
   - Tried and true method and very popular
   - The web site is made up of many pages on the web server.
   - The web server is responsible for most of the code, not the browser.
   - The pages have dynamic content created by the web server.
   - Doesn't need additional tools such as Angular or TypeScript.
   - Examples include CNN & most news organizations, Amazon, Stack Overflow, and Ebay

There is no right or wrong approach here. SPA and MPA both have strengths and weaknesses.
Our full stack program focuses on teaching the MPA / MVC approach.

Key Terms	Single Page Application Multi Page Application
Lesson Files	
Additional Resources	
Further Learning	https://www.scalablepath.com/blog/single-page-applications/

Next Steps for front-end development

Lesson Time: 30 Minutes

We are coming to a close on the JS portion of the class, and this feels like a good point to stop and take a look back at what you've learned and outline what comes next.

First, you've learned the fundamentals of front end web development using HTML, CSS, and JS. You've seen how all 3 technologies work together to build pages. You've learned foundational programming skills in JS that can be applied in many other languages.

We are stopping JS here and not going into Object Oriented Programming. We'll cover OOP in the C# course, and all of the concepts of OOP learned in C# will apply to JS.

TypeScript is typically the next step past JS. TypeScript is a popular alternative to vanilla JS and is currently popular in large web apps.

After TypeScript, AngularJS or React two front-end frameworks that are popular for building large, advanced websites using Single Page Application (SPA) design. SPA web apps only have one .html page. The entire app is created using JS or TypeScript, and a framework such as AngularJS.

Typescript, AngularJS, and React are beyond the scope of this course and program, and they are optional tools. You can create any website with just HTMl, CSS, JS, but mastering these optional tools can save time and land other jobs, and teach you how to create SPA web apps.

For in the final lessons of this course, we are going to round out your knowledge with JQuery and Bootstrap, your first framework.

The remaining courses in this program will all be focuses on backend development on the web server, SQL server, and programming with the MPA / MVC design approach.

We strongly encourage you to get practicing your HTML, CSS, and JS skills. The only way to get better at web development is to **DO MORE WEB DEVELOPMENT.**

Below we are linking to a large amount of resources you can use as reference as practice on your own.

Mozilla Developer Network
https://developer.mozilla.org/en-US/
One of the largest and best sites for front end web development, they have massive amounts of tutorials on HTML, CSS, & JS, along with code examples and practice exercises.

W3Schools
https://www.w3schools.com/
An Alternative to MDN, W3Schools offers full tutorials on many web technologies. Their reference section is excellent for quickly learning syntax.

Javascript.info
http://javascript.info/
Perhaps the best guide on introduction to programming using Js.

Google Grasshopper
https://grasshopper.codes
A free mobile app by Google that teaches Js in an easy and fun way. Requires mobile phone.

Learn JQuery
https://learn.jquery.com/
This site has a full tutorial on using jQuery. We'll be covering jQuery in our next lesson.

Bootstrap
https://Getbootstrap.com
Bootstrap's stie with reference documentation. We'll be covering Bootstrap in this course.

Codepen.io
https://codepen.io
The best place to build, test and discover front-end code. A social development network where you can see code examples from other developers.

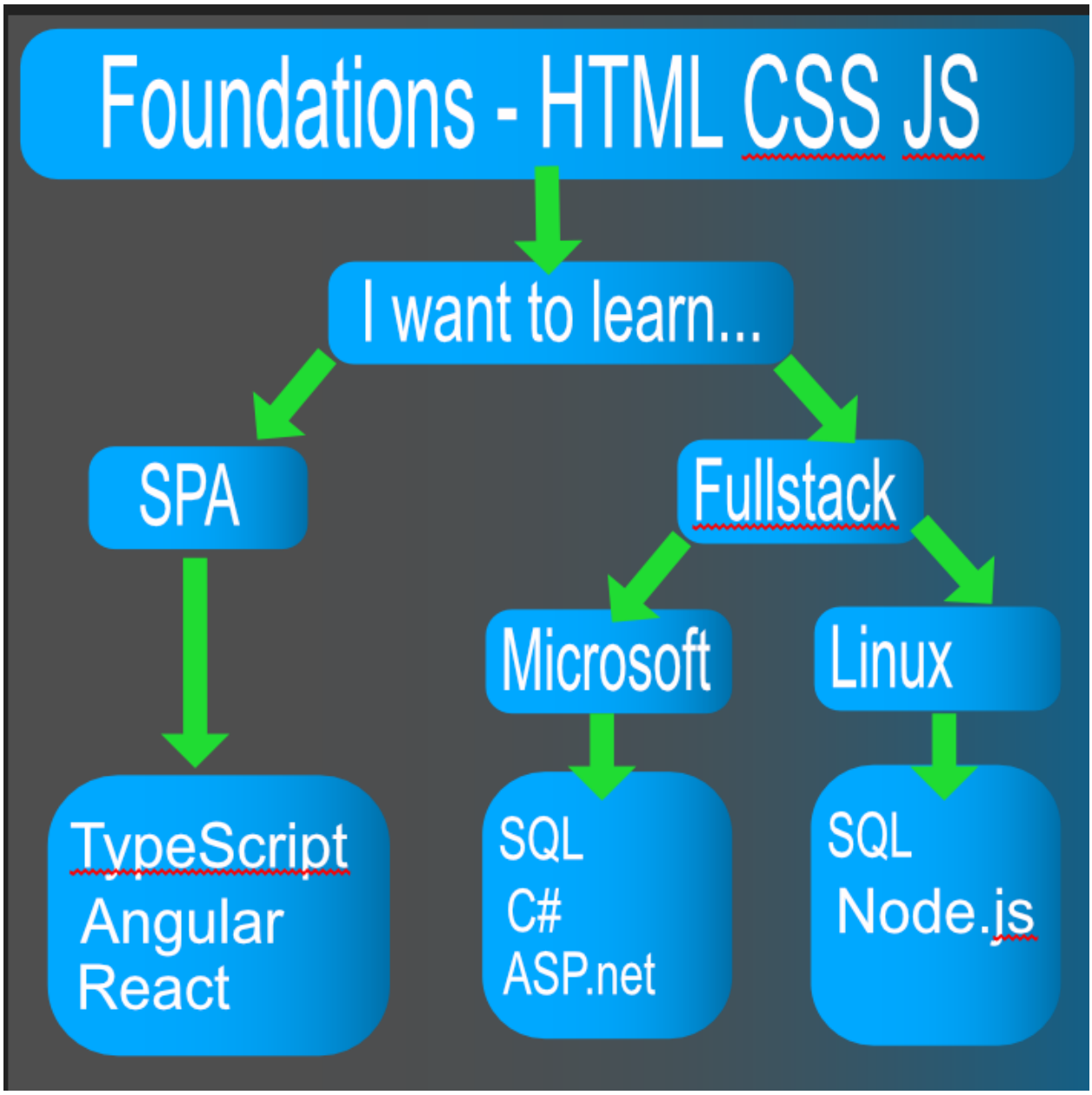
University of Helsinki - Full Stack Course
https://fullstackopen.com/en/

The University of Helsinki is offering a free full stack course in 2019-2020. This course is completely free and is eligible for college credit. There course focuses on SPA using React, Node.Js, and MongoDB, and this course is for learning an alternative stack after **finishing** our program.

Due to the advanced nature of the course, we do not recommend you start this until you've completed our complete full stack course.

TypeScript
https://www.typescriptlang.org/
After JS, TypeScript is generally the next step. The Typescript site has a handbook on learning. TypeScript is a superset of JS that compiles to standard JS code. Used to build large web sites and pre-req to Angular

Angular
https://Angular.io
Angular's site with a complete tutorial, for students who wish to learn angular and SPA. Recommended taking after completion of our program and learning TypeScript. Used to create SPA apps.


Foundations - HTML CSS JS
I want to learn...
SPA
Fullstack
Microsoft
Linux
TypeScript
Angular
React
SQL
C#
ASP.net
SQL
Node.js