The Complete Web Developer in 2025: Zero To Mastery

-notes S1-S7-

Section 1: Introduction

What this course goes into?

Web Foundations, HTML 5, CSS 3, Bootstrap, CSS Grid, Careers, JavaScript, Dev Environment, Git and Github, NPM / NPM Scripts, ReactJs, NodeJs, Databases, Production

ZTM Resources.

1) zerotomastery.io

2)https://www.youtube.com/@ZeroToMastery

3) https://www.linkedin.com/groups/12121940/

Section 2: How the internet works

We see the domain, google.com, but the servers see and work with the IP address.

1) We make a request: Give me google.com

2) The internet carrier gives the request further to the DNS

3) The DNS gives back the IP address to the carrier

4) The carrier loads that for you

5) The browser talks with Google Servers to get the files

6) The Google Servers provide the browser with the files it needs to show you google.com

The Internet is a huge world, that is actually physical not just digital. At the bottom of the sea there are Internet cables that connect each country and continent and that is the internet backbone.

Submarine Cable maps: https://www.submarinecablemap.com/

You can play around with traceroute and see how a browser pings and what are the IP addresses it travels to.

This course wants to explore the full-stack option, opening you to both front-end and back-end worlds.

Article that inspired this course: https://zerotomastery.io/blog/learn-to-code-for-free/

Section 3: History of the Web

Tim Berners-Lee = inventor of the World Wide Web (1989) - first web developer

World Wide Web is the 'www' we see in front of every link. The first website ever, made by Tim, is from 1991.

First website: https://info.cern.ch/hypertext/WWW/TheProject.html

-> Internet started off with Arpanet, a military project

-> Arpanet connected a few universities to discuss and share research

Internet maps: https://www.vox.com/a/internet-maps

HTML is a way for us to build the skeleton of the website. With HTML you just add the content

Because you can't style with HTML, CSS and JS were built.

In 1995-1996, CSS and JS were created.

These technologies provided a way to style the websites and make them prettier.

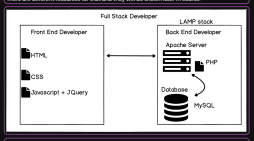
Back when the Internet started, there was something called 'the browser war'. Tim created a browser as a way to share files. During the browser wars, every company that built and owned a browser tried to do its own thing and launch more and more features.

But that gets to be the developer's pain point because when you build a website, you want to make sure it looks the same everywhere. In order to do that, you have to check if your code (your implementation) is the same (syntax) on every browser.

Opera might have one implementation. Google Chrome another one. Safari another type and Firefox another type of implementation. You have to match the styling on all of them if you want your website to look the same everywhere.

There are different resources for that and they will be shown later in course.

Finally, Let's talk about the websites from the early 2000s.



-> JQuery was a library that allowed JS to be written in a simple, clean way that allowed us not to worry too much about working in one browser, then the other. It created an environment for JS to flourish, but it is outdated and there are better methods to work now.

-> Less and less job offers appear with JQuery because it is no longer as demanded as in the beginning.

-> On the back-end we have LAMP stack.
Linux Apache MySQL PHP
In the 2000s, developers used to write PHP files that allowed there to be logic on the servers. Then, Apache. Apache is a software that really efficiently serves up files, and PHP is a programming language. Then, there is MySQL which is the database.

This course will look into HTML5, CSS3, JS, React, Node servers, NodeJs, Express.js, PostgreSQL, and MongoDB.

These technologies will allow us to make really good websites. Mastering these technologies makes you a full-stack.

Extra Resources.

1) https://www.youtube.com/watch?v=SQHzLVfpdUS&list=PLbGPuuaLjXNfUrzyH5r6jN9uIlgZ3ppdio&index=29

2) https://www.youtube.com/watch?v=AEwKnq3XpW8&index=30&list=PLbGPuuaLjXNfUrzyH5r6jN9uIlgZ3ppdo

3) https://www.youtube.com/watch?v=guest5CPUcE&index=31&list=PLbGPuuaLjXNfUrzyH5r6jN9uIlgZ3ppdo

Section 4: HTML 5

Andrei uses Sublime Text as his dev environment. I will use WebStorm.

He starts off with a basic doctype. Explains what the html, head, title, and body tags are.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>My first website!</title>
</head>
<body>
    hellooooooooooo there!
</body>
</html>
```

basic HTML in WebStorm

Resources to keep in mind

MDN, W3Schools

HOW TO ASK QUESTIONS

https://rubberduckdebugging.com

```
<ol>
    <li>banana</li>
    <li>apple</li>
    <li>orange</li>
</ol>
```

Ordered list (1, 2, 3 list)

```
<ul>
    <li>banana</li>
    <li>apple</li>
    <li>orange</li>
</ul>
```

unordered list (bullet point list)

```
<ul>
    <li>banana</li>
    <li>apple</li>
    <ul>
        <li>fuji</li>
        <li>honey crisp</li>
    </ul>
    <li>orange</li>
</ul>
```

list in list

- banana
- apple
  - fuji
  - honey crisp
- orange

result

```
<body>
    <h1>Boo</h1>
    <h2>Boo</h2>
    <h3>Boo</h3>
    <h4>Boo</h4>
    <h5>Boo</h5>
    <h6>Boo</h6>
</body>
```

All headings

self-closing tags

examples of tags like this:
<img>

<hr>
<in>

LS3: he talked about the anchor tag and how you link 2 pages

Section 5: Advanced HTML 5

LS3: HTML Forms

simple form. includes

-> first name input
-> last name input
-> email input
-> password input
-> birthdate input
-> gender radios
-> pets radios
-> cars radios
-> register button
-> reset button

```
<form method="GET">
    First Name: <input type="text"><br>
    Last Name: <input type="text"><br>
    Email: <input type="text"><br>
    Email: <input type="text"><br>
    Password: <input type="password" minlength="5"><br>
    Birthday: <input type="date"><br>
    Gender: <br>
    <input type="radio" name="gender">Male<br>
    <input type="radio" name="gender">Female<br>
    <input type="radio" name="gender">Other<br>
    Pets: <br>
    <input type="checkbox">Dog<br>
    <input type="checkbox">Cat<br>
    <input type="checkbox">Other<br>
    Cars:<br>
    <select>
        <option value="volvo">Volvo</option>
        <option value="audi">Audi</option>
    </select>
    <input type="submit" value="Register!">
    <input type="reset">
</form>
```

code

result

https://www.w3schools.com/html/html_quiz.asp

Section 6: CSS

Use S6_project_files for a code along

L1 of this section covered text color and how to color the background of the text in some parts

Palette link: **https://palette.cc/#uid=1000u0kIlIlaFw0g0qFqFg0w0aF**

Almanac link: **https://css-tricks.com/almanac/**

how to add an image as a background

Resource:

Selectors game: **https://css-diner.netlify.app/**

talked about fonts styling and how you can use Google Fonts to choose your font family

Side note: It's better to import fonts and work in the cloud than to download fonts and use them locally, linking to the font files

resource for CSS sizes

https://developer.mozilla/en/docs/Learn/Tools_and_testing/Cross_browser_testing/Responsive_design_testing

CSS Quiz

https://www.w3schools.com/css/css_quiz.asp

Section 7: Advanced CSS

Resource 1: CSS Minify
https://www.cleancss.com/css-minify/

Resource 2: pictures-links.txt

Resource 3: **https://flexboxfroggy.com/**

Resource 4: **https://css-tricks.com/snippets/css/a-guide-to-flexbox/**

Resource 5: picture + links (1).file

Resource 6: **https://autoprefixer.github.io/**

Resource 7: **https://shouldiprefix.com/**

Resource 8: **https://caniuse.com/**

Resource 9:
https://www.w3schools.com/cssref/css3_browsersupport.php

Resource 10: **img_gallery.zip**

Resource 11: **robot.zip**

Resource 12: **robot(1).zip**

Resource 13: **robotfriend.zip**

Resource 14:
https://learn.freecodecamp.org/responsive-web-design/basic-css

Resource 15:
https://robots.thoughtbot.com/transitions-and-transforms

Presented with xmind