Web Applications

* Work through browser
* Client- phone, computer (front end)
* Server- takes info to give to client (back end)
* Database- where info is stored (back end)
* Front end: HTML/CSS/JavaScript/iOS
* Back end: Ruby, Python, “full stack” JavaScript

Front End

* HTML: Content

Text, headlines, images, links

How it is organized

* CSS: Presentation

Fonts, colors, placement

Responsiveness: reorganization of content based on client too (phone vs desktop)

* JavaScript Action

Event oriented: ex: page loads, click on button

*When something happens… what should happen?*

HTML

* “tags” tell the browser ehat the content is
* browsers displays the content according to the rules of the tags….. this is called “marking up” the content
* **H**yper **T**ext **M**arkup **L**anguage
* Tags surround content they describe
* Beginning tags <p> and ending tags </p>
* Browsers don’t care if you use upper or lower case…. Lower case is easier for humans to read
* Tabs aren’t necessary for computer to read, but it is easier to read and make code
* First plan website with the “skeleton” of tags: header, nav (links for navigation), section (big block of main content), article (chunks within section), footer

*Anatomy of a Web Page*

*<html>*

*<head>*

*<title> welcome title</title>*

*</head>*

*<body>*

*<content blah blah blah>*

*</body>*

*</html>*

* White space=tabs, spaces, returns
* All white space characters show up as one space
* Multiple white characters show up as one space
* We can use different tags to make double spaces lines <p> </p>
* Returns <br>
* Lists <ul> </ul> and <li> </li>
* Headings: tell the browser how important as line of text is
* <h1> the biggest heading </h1>
* <h6> the smallest heading </h6>
* Commonly are displayed in bold
* Attributes: give tags more info..... where to find an image, the URL of a link
* All have different names, but the syntax is the same
* <a href= “file name or URL”> </a>
* images: <img src= “file name or URL”> *no closing tag needed!*

CSS

* How to design and lay out you HTML elements
* Maintain consistent styles
* **C**ascading **S**tyle **S**heets
* Selects elements to apply a style to
* Defines what the style is
* Can be written inline next to the thing you what to style, in the header, or in an external file referenced in the header.
* Selector: p
* Opening bracket: {
* Property c: color:
* Value: “blue”;
* Closing bracket: }

*P {*

*Color: “blue”;*

*}*

* Selector:

HTML tag…any content area you want to style

CSS select content areas to style by tag

* Properties:

CSS almanac: <https://css-tricks.com/almanac/>

Color: <https://color.adobe.com/create/color-wheel/>

More color: <http://www.color-hex.com/color-wheel/>

* Connect CSS to HTML
  + <link rel=”stylesheet”
  + href=”filename”
  + type=”text/css”>
* Layout

The Box Model: the padding, border, and margin separating your content from other areas of content on your page. Use these properties wisely to create negative space to improve your layout and user experience.

* Square around text: border
* Space between border and font: padding
* Space outside border: margin

Job

* Web developer
* Software developer
* Front-end developer
* Software engineer
* UX Designer
* Differ by stack, company, and where you spend most of your time (front end, backend, or both)
* Identify bugs, plan out how to fix the bug or make new feature
* Write code
* Review code
* Make sure code gets tested
* Work with team to design and implement solutions

Related Jobs

* Testing
* Sales engineer: help potential buyers understand technical side
* Web producer/: good for 301 students potentially… manage content on a site using CMS/HTML/CSS/JavaScript
* Technical writer
* Program manager

Becoming a Dev

* Constant learning
* Learning a language that appeals to you
* Like finishing a problem
* Working with interesting people who will challenge you to think differently
* Like solving puzzles
* Want to help make things that change the world