MasseyHacks V | Alex Xie

Frontend Workshop: Outline

Details

Target Audience: Beginner coders, newcomers to programming

Recommended Prerequisites/Knowledge: None required, some basic knowledge of coding will be useful

Content Breakdown

1. **Introduction**
    - Short description of what we'll be doing in the workshop (bit of theory, then practical learning by building a personal site)
    - Background of myself
2. **Theory**
    - How the frontend of websites are built, deployed, served (basically the process a site goes through from developer to user)
    - Most commonly used technologies in frontend web dev (HTML, CSS, sometimes JS)
3. **Setup** (basically tell them to start getting their hands dirty)
4. **HTML**
    - Explanation of HTML tags and props, their functionalities, and behaviour
    - Designing our personal site with a sectional mindset (encourage them to think in components/split site into functionality or pieces)
5. **CSS**
    - Integrating CSS with HTML (`id` , `class` , selectors)
    - Integrating simple style changes to existing elements on your page (color, size, font-family, etc)
    - Designing a non-linear site with layout changes (`position` , `display` , `padding` / `margin` , possibly `zindex` or `flexbox` ?)
    - Animations & transitions (time permitting)
6. **Taking it one step further**
    - Frameworks for web development (React, Vue, Ruby on Rails)
    - Integrations with a backend
    - Dynamic sites/web apps
    - Mobile responsive design, PWAs

Frontend Workshop: Walkthrough

Introduction

- What we'll be doing:
    1. A bit of theory
        1. Technical stuff
        2. Technologies used on the web
        3. How everything works and fits together on the front-end
    2. Building and deploying a simple personal website
        - example on Github
- Alex, your host for the Intro to Frontend workshop
- Background about myself:
    - 2nd year CS @ UW
    - Previous co-op at Flipp, mostly backend
    - Other experience: Equithon, Hack the North, TEDxUW
    - Mentor at MH5, come talk to me about stuff
- Instructions, outline, final product on github as well
    - **https://github.com/alexieyizhe/mh5-frontend-workshop**

Theory

- What frontend web dev consists of:
    - Building the pages and interfaces that a user will see when they visit a website
    - Working with designers to make the web look nice
    - Creating dynamic apps and experiences (getting more prevalent)
- On the developer side:
    - Creating a website:
        - HTML
        - CSS
        - JS
    - Deploying and hosting a website
        - Web servers
        - Hosting services
        - Domain name & DNS
- On the user side:
    - Terminology
    - Viewing a website

        - Web browsers
        - URLs and navigation
        - Resolving DNS
        - Fetching assets from server
        - Displaying the site
        - Sending data back
    - Dynamic vs. static sites
        - Dynamic websites:
            - often respond to user input
            - built like apps
            - 2 way communication with a server to store data, perform actions, etc
            - Requires development of a backend
        - Static sites
            - You ask for a website by typing in the URL, server on host provides a preloaded page, sends it to your web browser to be displayed
            - No backend even needed most of the time
            - Often won't need to communicate back to the host
- Since this is a frontend workshop, we'll be making a static site
    - Going through the entire process, from coding, styling, deploying, and hosting (if you have a GitHub account)

Setup

1. Create a folder, call it whatever you want.
2. [Optional] If you know how to use git and want to keep track of what we build in this workshop, you might want to `git init` and then `git commit` as appropriate throughout.
3. Create the following files in the folder:
    - `index.html`
    - `index.css`
4. You're ready to go!

Checkpoint: You should be able to start writing code for the final product.

HTML: The Foundation

- Overview of HTML
    - Website is broken up into small pieces, known formally as **components**
    - HTML allows you to define blocks and pieces that make up a page on your website as **elements**

    - Start thinking about websites as built from tiny components
        - Reusability
        - DRY
        - Single responsibility principle
    - Syntax for:
        - Elements
        - Props
- Start writing code!

Checkpoint: By this point, attendees should have the basic structure of the final product.

CSS: The Decorations

- Overview of CSS
    - Why do you need it?
    - Abilities
        - Changing layout, moving stuff around
        - Changing look (color, size, shape)
        - Selective application of styles (hover, pseudoelements)
- Applying styles
    - Selectors
    - Properties
- Start styling!
- Advanced CSS concepts
    - Animations
    - Transitions
    - Media queries
    - Pseudoelements
    - Advanced selectors
    - etc.

Checkpoint: By this point, attendees should have the layout and look of the final product.

JavaScript: The Functionality

- Similar to other programming languages, but native to the web
- Simple JS to modify an element on our website
- Can do more, but not today

Going Live [OPTIONAL]

- Host on Github Pages:
    1. Commit & push to repository
    2. Navigate to repository page > settings
    3. Enable Github pages
    4. See your website live!
- How does this connect the theory and the application?

Checkpoint: By this point, attendees should have a live version of the final product, fully interactive and viewable on the internet.

Taking It One Step Further

- Building on modifying one element with JS: why not the entire page?
- Frontend frameworks
    - JQuery
    - React
    - Vue.js
- Integrating with a backend
    - REST APIs
    - Backend development
- Mobile
    - Responsive design
    - PWAs
    - Native apps

Questions? Come find me!