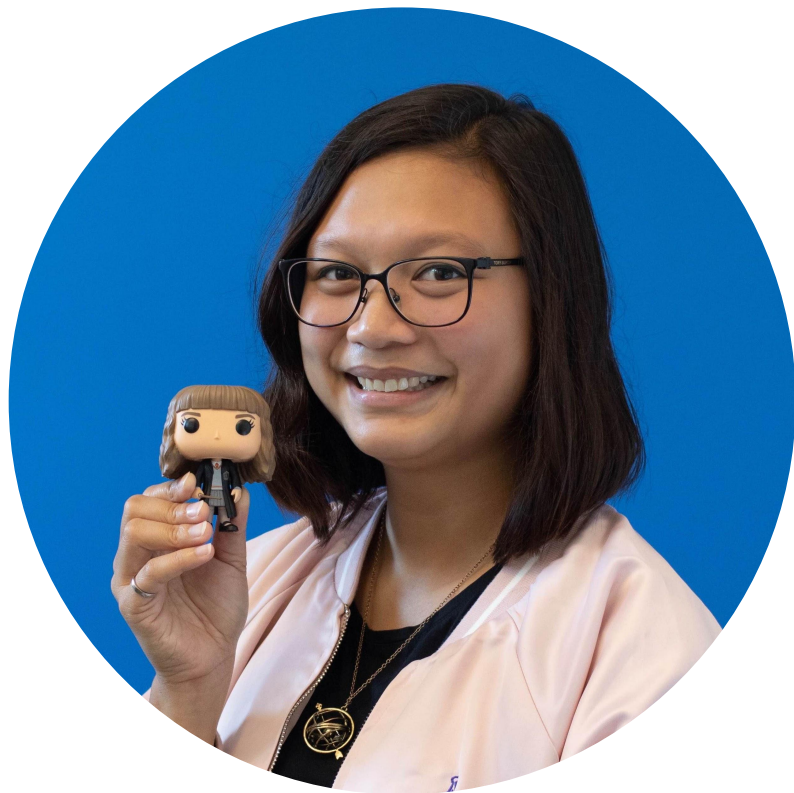


INTRO TO WEB DEVELOPMENT

StarterHacks 2020

Michelle Mabuyo | [@MichelleMabuyo](#)



Michelle Mabuyo

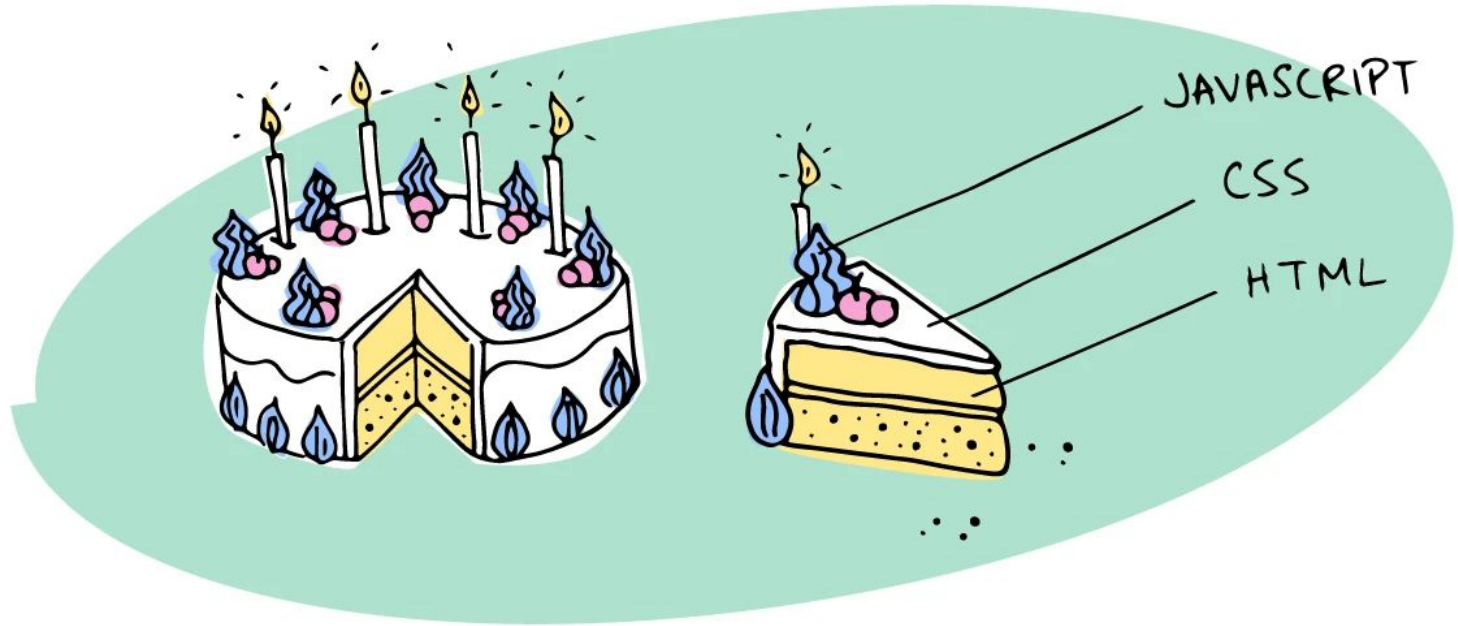
Software Engineer @
Manulife

KW Chapter Lead @ Canada
Learning Code

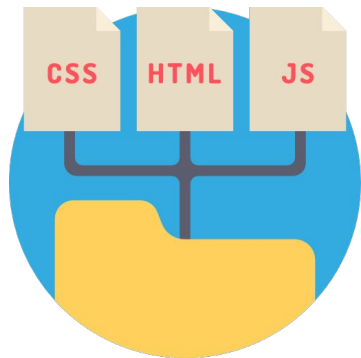
@MichelleMabuyo

INTRO TO WEB DEVELOPMENT

WEB DEVELOPMENT BUILDING BLOCKS



BY THE END OF THIS WORKSHOP, YOU WILL LEARN



**Basics of HTML,
CSS, JavaScript**



**Browser
Developer Tools**



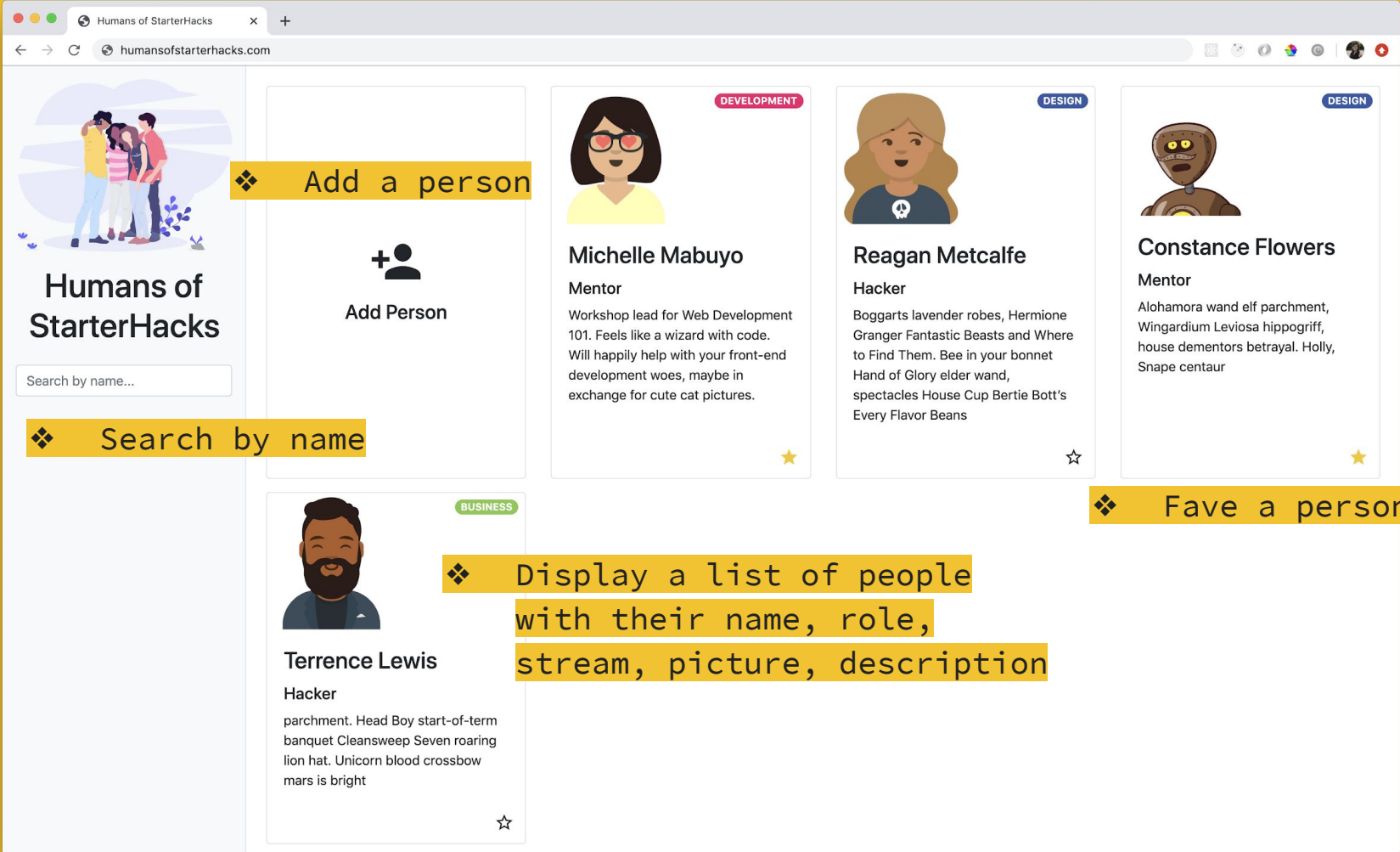
Online Resources



WHAT ARE WE
BUILDING?

Humans of StarterHacks

A website showcasing the
awesome people you'll meet
at StarterHacks!



FORMULATING A GAME PLAN

- What is the **smallest thing** I can build with a **low amount of effort** and **deliver a high value**?
- Try to do things in **iterations**. Each iteration has to have **value**.

Alternatively... **what do you want to learn** this weekend?

WORKSHOP GAMEPLAN

Steps 1–3: HTML & CSS for a Static Website

Steps 4–5: JavaScript with Dynamic Data & Interactivity

Step 6 and beyond: Connecting to a database for persistent data

HOW TO FOLLOW ALONG TODAY

All steps are on **Github**:

<https://github.com/mabuyo/humans-of-starterhacks>

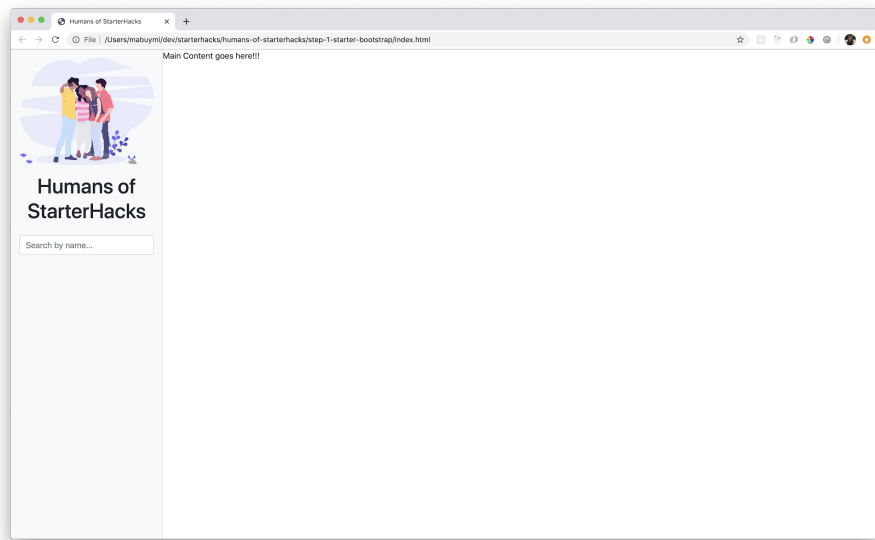
List of links are on the README.md file.

You can:

- ❖ Code along (highly encouraged!)
- ❖ Sit back, watch and learn
- ❖ Take notes
- ❖ Post on Slack / flag down a mentor if you need help or have questions

PROJECT SETUP
(3 MINS)

STEP 1



Create the HTML content for sidebar with website title, image and search input.

Use CSS to set up the page layout.

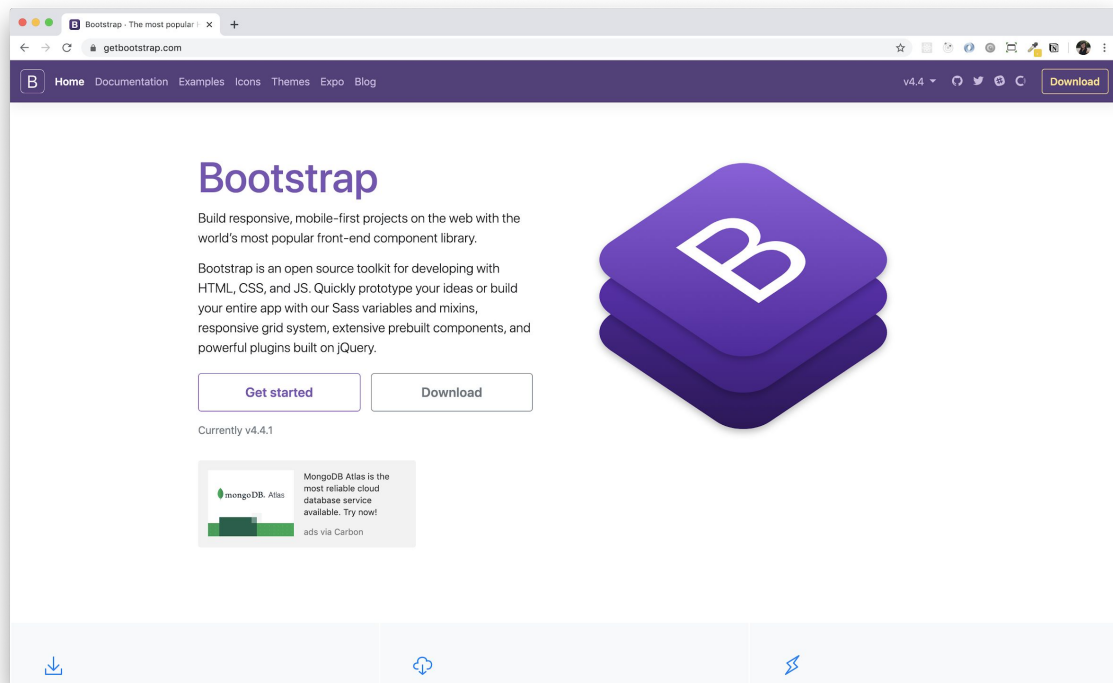
HTML

HTML BASICS

- ❖ Written with HTML tags (also known as elements)
- ❖ Tags are written with angled brackets `<> </>`
- ❖ Tags have opening and closing tags, with content in between `<tag>Content</tag>`
- ❖ Tags can be nested
`<parent>`
 `<child>Content</child>`
`</parent>`
- ❖ Tags can have attributes `<tag attribute="value">`

USING BOOTSTRAP

Bootstrap is a front-end component library that helps us quickly prototype and put our website together!



ADDING AN IMAGE WITH

- ❖ Image tags are **self-closing** (they don't have a closing tag).
- ❖ Needs a **src** attribute that links to the image
- ❖ Needs an **alt** attribute that describes the image

```

```

ADDING A SEARCH BOX WITH <INPUT>

- ❖ Input tags are self-closing
- ❖ Different input **types**: text, password, number, radio, checkbox, button...
- ❖ Can take a **placeholder** attribute to prompt the user what to type in

```
<input type="text" placeholder="Search by name..." />
```



CSS SYNTAX

- ❖ CSS follows a format like so:
target { styles }
- ❖ The target is HTML: **.class** or **#id** or the HTML tag
- ❖ Styles are written as **property: value;**
- ❖ Examples of styles are: background-color, font-size, width, border. Check out [css.cool](#)

LINKING CSS WITH HTML

HTML

```
<div class="aClassName"></div>  
<div id="idName"></div>  
<p></p>
```

CSS

```
.aClassName {}  
#idName {}  
p {}
```

// In your HTML file:

```
<link rel="stylesheet" href="styles.css" />
```

PAGE LAYOUT

Starter
Templates

page-container (parent container for sidebar and main content)

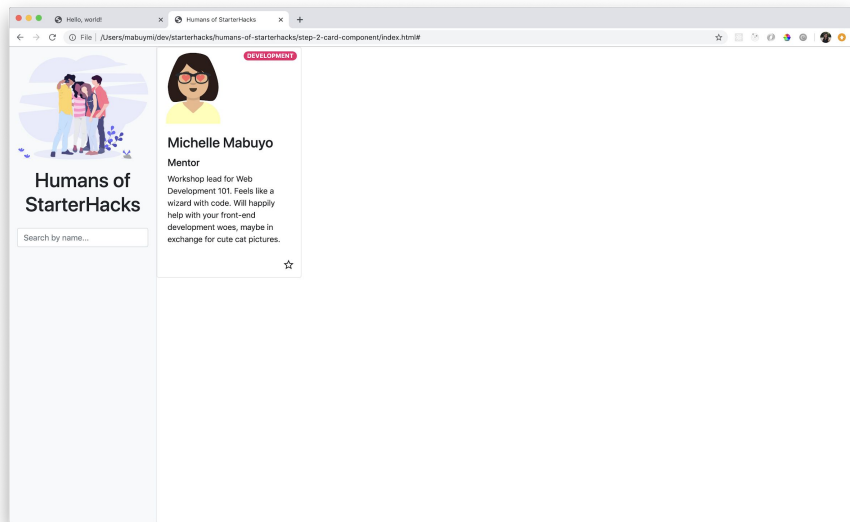
sidebar

- fixed width
- does not scroll with the page

main content

- takes up the rest of the browser width
- scrolls within its container

STEP 2 GOAL



Create the HTML content for the Card component that displays a person's information for: name, role, stream, description, image, and a favoured indicator.

BOOTSTRAP COMPONENTS

- ❖ Quick prototyping
- ❖ Documentation includes code samples for easy copy-paste

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

Navbar

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

5 MIN CHALLENGE

- Replace card content with relevant content: name, role, description
- Find an avatar and replace the `` tag.
 - Try using [avataaars](#)!

BOOTSTRAP - BADGE COMPONENT

- ❖ Use the Badge component for the person's role (Hacker, Mentor, Organizer, etc)
- ❖ We'll set up our own colors!
 - <http://css.cool/#/colorpicker>

Primary Secondary Success Danger Warning Info Light Dark

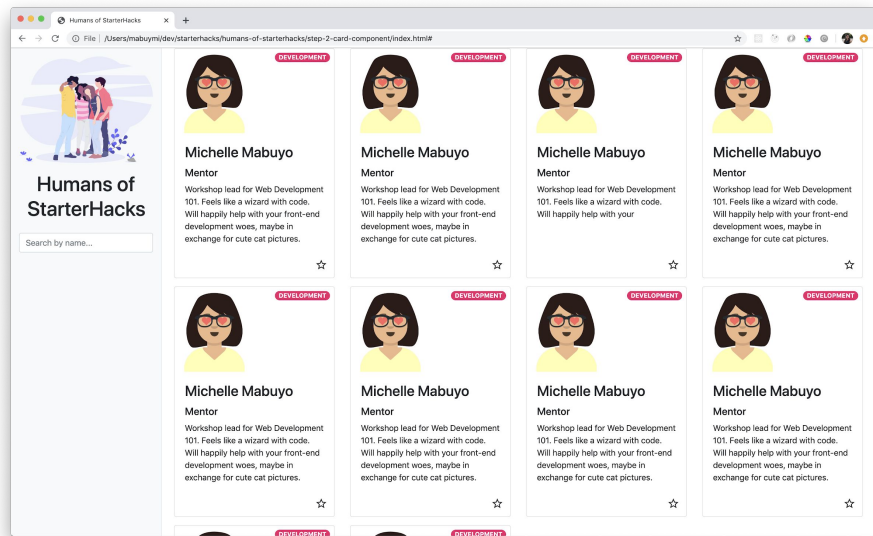
```
<span class="badge badge-pill badge-primary">Primary</span>  
<span class="badge badge-pill badge-secondary">Secondary</span>  
<span class="badge badge-pill badge-success">Success</span>  
<span class="badge badge-pill badge-danger">Danger</span>  
<span class="badge badge-pill badge-warning">Warning</span>  
<span class="badge badge-pill badge-info">Info</span>  
<span class="badge badge-pill badge-light">Light</span>  
<span class="badge badge-pill badge-dark">Dark</span>
```

USING MATERIAL ICONS

- ❖ Material Icons is an easy-to-use Icon Font Library.
- ❖ Follow the instructions for setup [here](#).

```
<i class="material-icons">star_border</i>
```

STEP 3

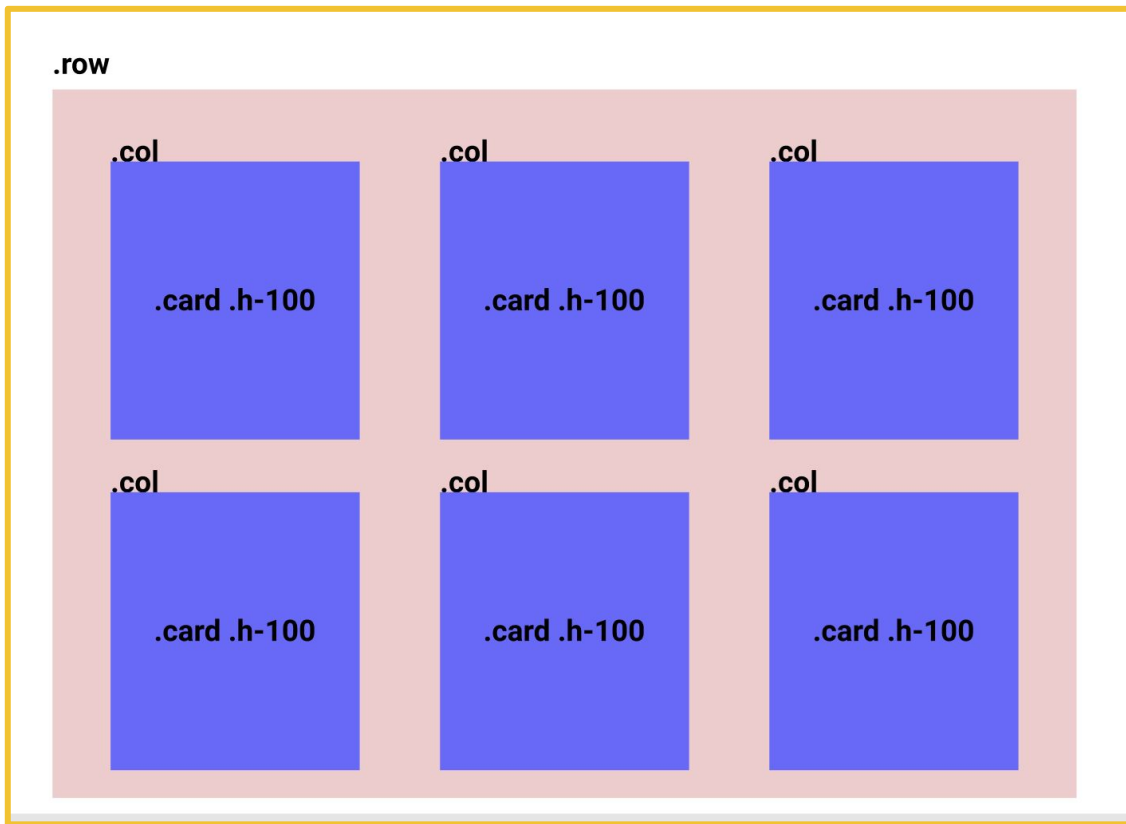


Create more cards and
style the layout.

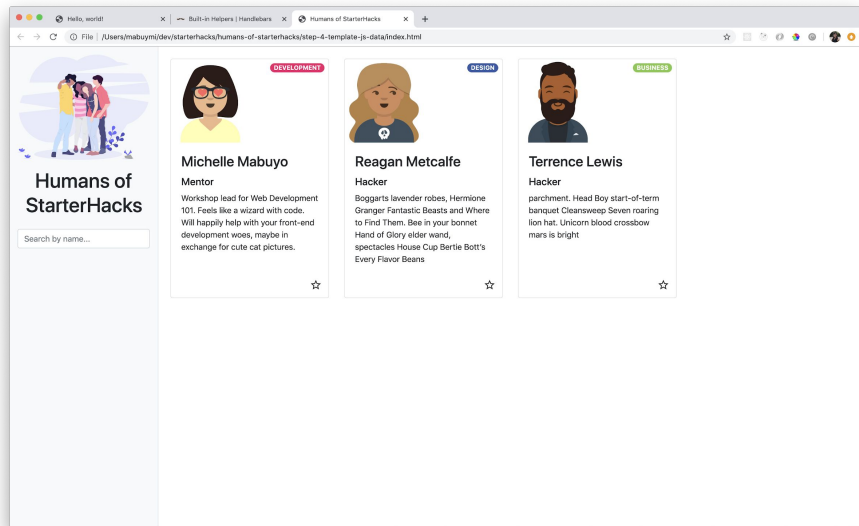
BOOTSTRAP GRID CARDS LAYOUT



<https://getbootstrap.com/docs/4.4/components/card/#grid-cards>



STEP 4



Use Handlebars.js to create a template for the Card component. Use Javascript for mock data.

JAVASCRIPT

JAVASCRIPT BASICS

❖ Types

➤ string, number,
boolean

❖ Objects

```
{ key: value, name: "Michelle" };
```

❖ Variables

```
let variable = "something";
```

❖ Arrays (a list)

```
let array = [1, 'two', { num: 3 }];
```

❖ Functions

```
function add(num1, num2) {  
    return num1 + num2;  
}  
  
add(1, 2);
```


LINKING JAVASCRIPT AND HTML

// In your HTML file:

```
<script src="index.js"></script>
```

5 MINUTE CHALLENGE: CREATE YOUR DATA

- ❖ Create an array with 3 people in it.
- ❖ A person has the following properties: id, name, role, stream, description, image_link, is_favourite
 - Use robohash.org for placeholder avatars!
- ❖ Introduce yourself to the people beside you and add them to your array :)

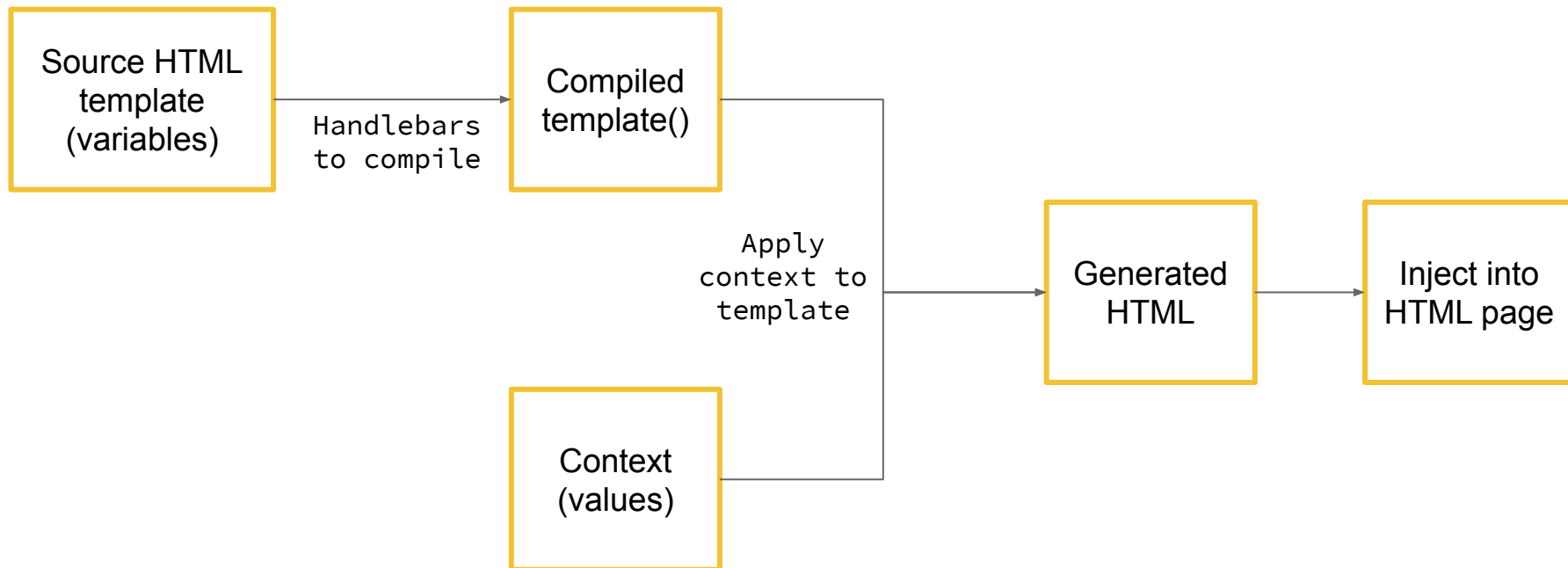
```
let person1 = {id: 1, name: "Michelle", is_favourite: false}  
let person2 = {id: 2, name: "Hermione", is_favourite: true}  
let person3 = {id: 3, name: "Ron", is_favourite: false}  
let peopleData = [ person1, person2, person3 ]
```

HANDLEBARS.JS

- ❖ Handlebars is a **templating language** that can **generate HTML** using JavaScript variables and expressions.
- ❖ This means less duplicated code in the HTML and one source of truth for our Card.
- ❖ Handlebars uses double braces: **{{variable}}**

```
<h3 class="card-title">{{name}}</h3>
```

HANDLEBARS PROCESS



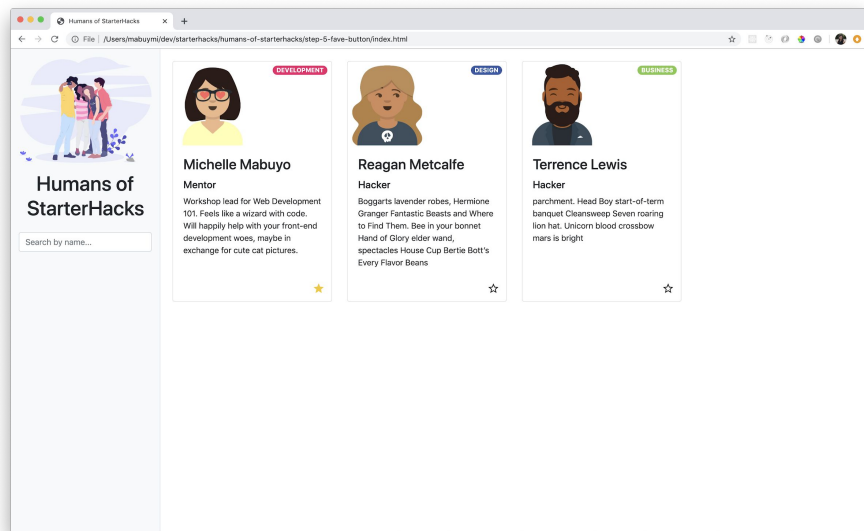
JAVASCRIPT & THE DOM

- ❖ DOM refers to **Document Object Model** (your HTML content)
- ❖ JavaScript can interact with the DOM to add things like event listeners or retrieve and replace the HTML content (what Handlebars does)

```
document.getElementById('card-template').innerHTML
```

```
document.getElementsByClassName('fave-btn')
```

STEP 5



Fave a person with the star button.

EVENT LISTENERS

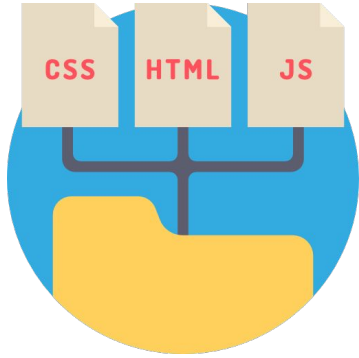
- ❖ JavaScript can add event listeners: functions that listen for events (like clicks) and then run code to deal with what should happen next

```
btn.addEventListener('click', function() {  
    console.log('I was clicked!');  
});
```

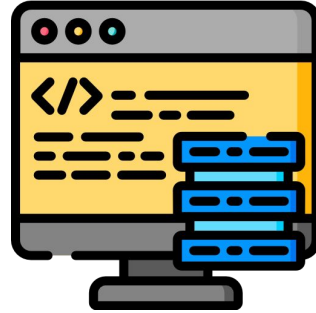
CONGRATULATIONS!



IT'S THE END OF THE WORKSHOP! DID YOU LEARN...



**Basics of HTML,
CSS, JavaScript**



**Browser
Developer Tools**



Online Resources

THANK YOU!

@MichelleMabuyo

