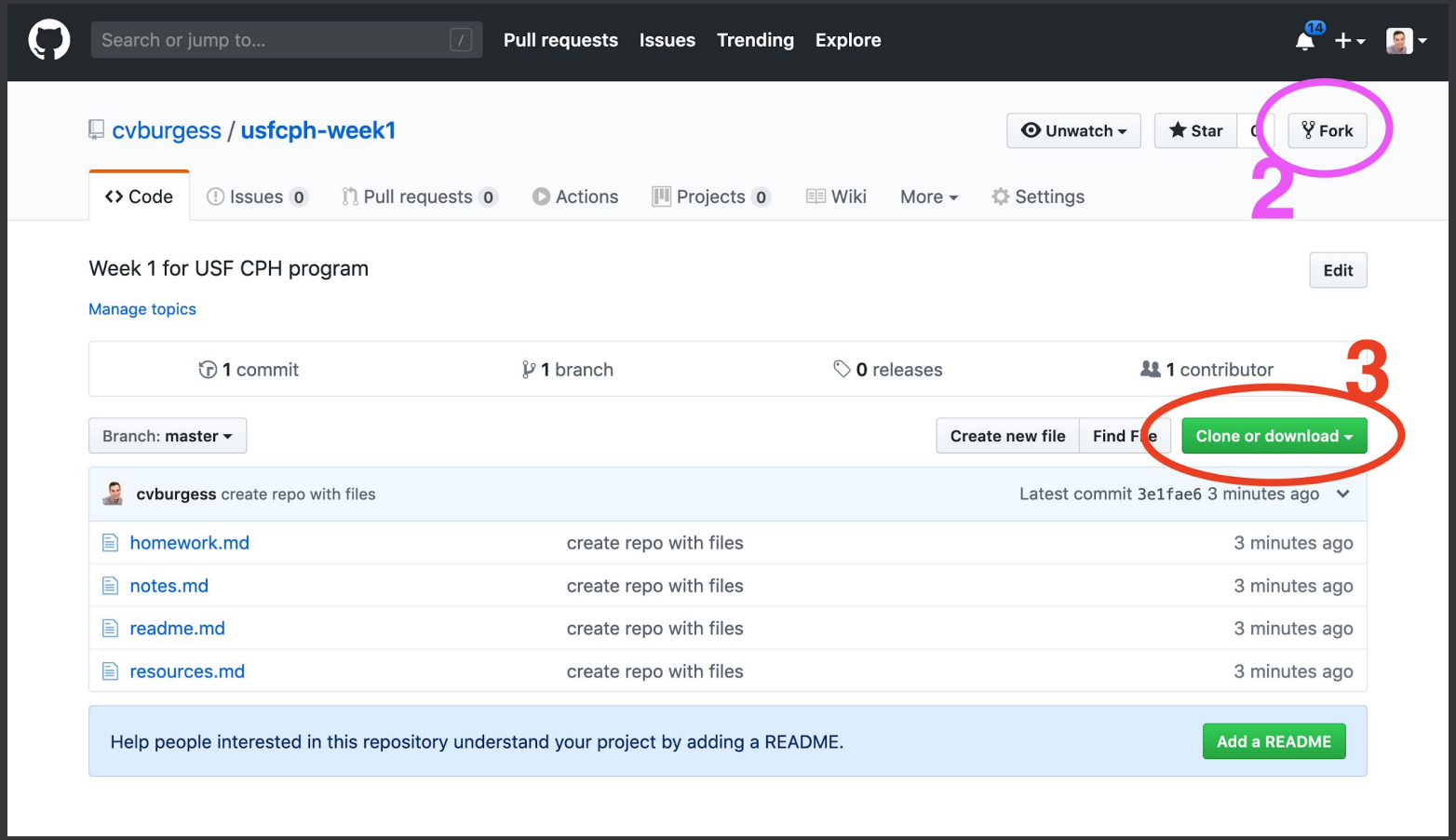
USF CPH - Week 3

Zero to prototype in 35 days

Getting Set Up

1. Go to github.com / cvburgess / usfcph-week3
2. Fork the repository to your account
3. Clone the repository to your computer
4. Open with VSCode



How to Fork and Clone with the GitHub UI

Week 2

Looking back before we move forward

HTML + CSS

- HTML
  - Tag-based language used to structure the web
- CSS
  - Style-sheets that are used to theme the web
- Frameworks
  - We do not have to write everything ourselves
- Deployment with Netlify

Logic

How we explain things

Logic

- Can be described in any language
- How we define the interaction and chains of events
  - Conditional logic: if this is true, then do that
  - Chained logic: First do this, then do that
  - Iterative logic: For each of these, do that
- House analogy: directions for a person walking through the home

Exercise

Explain how to get a slice of pizza

Pizza Instructions

- Make a bulleted list
  - Be specific, only one action per step
- Note any conditions
  - What if the box is empty or closed?
  - What if the user is vegetarian?
  - What if the user wants more than one slice?

Flows

How we plan for human experiences

Concept vs Syntax

Words and their meaning in a digital context

Primitives and Objects

Primitives in JavaScript

- Boolean: True and False
- String: Words (or random letters)
- Number
- Null: An empty value
- Undefined: The absence of a value

```
// Boolean
const isAlive = true;
const isDead = !isAlive; // You can use "!" to negate things
const isNamed = Boolean(name); // You can convert something to a boolean

// String
const name = "Janet";
const country = 'Canada'; // Single-quote strings are just like double-quoted ones
const verb = String("born"); // You can convert something to a string
const combined = `${name} was ${verb} in ${country}`; // Template strings

// Number
const age = 37;
const weight = Number(124); // You can convert something to a number

// Null
const nothing = null;

// Undefined
const person = { name: "Janet", age: 37 };
person.height // undefined because person has no height property
```

Example usage of primitives in JavaScript

Objects in JavaScript

- Object: An item with nestable properties
- Array: A list of things
- Date

```
// Object
const person = { name: "Janet", age: 37 };

person.weight = 124; // Assigning values with dot notation
const name = person.name; // Reading values with dot notation

person["nickname"] = "Jan"; // Assigning values with bracket notation
const age = person["age"]; // Reading values with bracket notation

// Array
const hobbies = ["running", "skiing", "kayaking", "feeding cats"];
const lottoNumbers = [1, 4, 15, 24, 46];

hobbies.push("yoga"); // Adding an item to the end of the array
const third = lottoNumbers[2]; // Returns 15 because JS is a zero-based language

// Date
const now = new Date();
```

Example usage of objects in JavaScript

Functions and Conditionals

Functions

- The verbs of programming
  - getPizza, walkDog, onClick, requestData
- Called with optional parameters
  - getPizza(“extra cheese”, “veggie”, “thin crust”)
- Can call other functions
  - onClick => confirmDelete => processInput => closeWindow

```
console.log("Hello World"); // Calling a built-in function

const doNothing = () => {}; // Creating our own function
function doNothing() {} // Alternative syntax

// Creating our own function that calls a function
const sayHello = name => {
  console.log(`Hello ${name}`);
};
sayHello("Janet"); // Call the function with a single parameter

// Creating our own function that returns a value
const add = (number1, number2) => number1 + number2;
add(4, 6); // Call the function with two parameters

console.log(add(2, 7)); // Combine functions
console.log(`The sum of 5 and 1 is ${add(5, 1)}`); // Use functions in template strings
```

Example usage of functions in JavaScript

Conditionals

- If + else if + else: “ if this, then that...”
  - Simplest way to define conditional logic
- Switches: Test a bunch of values for a given input
  - A stricter if + else if
- Ternaries: Shorthand for if + else
- Short circuit: An emergency brake for code

```
const person = { name: "Janet", nickname: null, age: 37, isAlive: true, weight: 124 };

if (isAlive) {
  const name = person.nickname || person.name; // Prefer the nickname
  const age = person.age >= 18 ? "an adult" : "a child"; // Use a ternary
  console.log(`${name} is ${age} and ${person.weight}lbs`);
} else {
  console.log(`Sorry, ${person.name} has died`);
}

// Same thing but with a short circuit instead of an if / else

if (!isAlive) return console.log(`Sorry, ${person.name} has died`);
// We can assume if we got this far, the person is alive and well
// Return would be used inside of a function
// You can also throw an error instead of returning
const name = person.nickname || person.name; // Prefer the nickname
const age = person.age >= 18 ? "an adult" : "a child"; // Use a ternary
console.log(`${name} is ${age} and ${person.weight}lbs`);
```

Example usage of conditionals in JavaScript

Iteration and Classes

Iteration

- For / ForEach
  - Loops through an array and calls a function for each item in the array
- Map
  - Loops through an array and returns a new array with any given modifications
- Filter
  - Loops through an array and returns a new array with all values that pass a given test

```
const hobbies = ["running", "skiing", "kayaking", "feeding cats"];
const lottoNumbers = [1, 4, 15, 24, 46];

// ForEach

hobbies.forEach(hobby => console.log(`I love ${hobby}!`)); // Print out each phrase
lottoNumbers.forEach(number => console.log(number)); // Print out each lottery number

// Map

const tense = hobbies.map(hobby => hobby.replace("ing", "s")); // Replace "ing" with "s"
const doubled = lottoNumbers.map(number => number * 2); // Double each number

// Filter

// If the hobby contains the letter A, add it to the new array
const hobbiesWithAs = hobbies.filter(hobby => hobby.includes("a"));
// If the number divided by two has no remainder, add it to the new array
const evenNumbers = lottoNumbers.filter(number => number % 2 === 0);
```

Example usage of iteration in JavaScript

Classes

- More common in languages like Python and Java
- Used to describe nouns
- Have helper methods
- Commonly used in React prior to 2019

Exercise

All hail the Magic Orb

Adding JavaScript to HTML

1. Open exercise-1 / index.html in VSCode
2. Tweak the code
3. Open exercise-1 / index.html in Google Chrome
4. Repeat as many times as you would like

React + npm

Leaning on the community

npm

- Library of open source software
  - Nearly 1 million of them at last count
  - Average of 30 billion downloads a year
- Command Line Interface (CLI) tool for managing dependencies
  - npm install my-package
- Build less of the nuts and bolts
  - Focus on finding good packages and adding value

React

- Open source project from Facebook
  - Used to build websites and mobile apps
  - Most popular front end framework (as of writing)
- Highly extensible and customizable
  - (Almost) everything is a component
  - Tag-based, very similar to HTML
  - Thousands of packages on npm

Exercise

Upgrading the Magic Orb

Using npm and React

1. Open exercise-2 in VSCode
2. Tweak the code
3. Open the built-in terminal in VSCode
    a. Type cd exercise-2 and hit enter
    b. Type npm install and hit enter
    c. Type npm start and hit enter
4. Repeat as many times as you would like

Exercise

Multiple magic orbs

Complex interactions

1. Open exercise-3 in VSCode
2. Tweak the code
3. Open the built-in terminal in VSCode
    a. Type cd ../exercise-3 and hit enter
    b. Type npm install and hit enter
    c. Type npm start and hit enter
4. Repeat as many times as you would like

Exercise

A simply smart weather app

Complex interactions

1. Open exercise-4 in VSCode
2. Tweak the code
3. Open the built-in terminal in VSCode
    a. Type cd ../exercise-4 and hit enter
    b. Type npm install and hit enter
    c. Type npm start and hit enter
4. Repeat as many times as you would like

Deploy these with Netlify

One app per folder, or just deploy your favorite

Homework

1. Review your notes
2. Play with the apps we created
    a. E1 + E2: The Magic Orb
    b. E3: Magic Orbs
    c. E4: Weather
3. Discuss what features your idea will need
    a. Look for libraries on npm that would help

USF CPH - Week 3

Zero to prototype in 35 days