Phase-1 JavaScript

1. Functions & Scope

NOTE: Make a copy of this document to edit: File > Make a copy

Link to Starter Code

*Coding along to live lectures is **NOT recommended**. It can actually make following the lecture more difficult. However, starter code is provided below if you wish to code along with the recorded video or want to practice the concepts on your own outside of lecture. Please hold all questions about errors or issues you have in your code until break or after lecture.*

Link to Warm-Up Questions

Warm-up questions are a great way to get your brain ready for lectures. They are not required but feel free to do them right before lecture or as you settle in during the first few minutes.

SWBAT

SWBATs are our learning goals for lecture. After each lecture review the SWBATs. If there are any SWBATs you are struggling with, stay to ask questions at the end of lecture (time permitting) or follow up with your Cohort Instructor or Technical Coach outside of lecture.

- [ ] Describe what functions are and their central importance in JS
- [ ] Review syntax differences between regular functions and arrow functions
- [ ] Explain the difference between:
    - [ ] Block scope
    - [ ] Function scope
    - [ ] Global scope
- [ ] Understand what it means that a function is a first-class object
- [ ] Explain what a higher-order function is
- [ ] Describe what a callback function and higher-order function is

What is a function?	
How do we declare a function?	
What's the difference between referencing a function and invoking a function?	
What are at least two different ways to declare a function?	
What are function parameters?	
What are arguments?	
What is Scope and how would you describe the difference between **Global Scope Function/local Scope and Block Scope?**	

What do we mean when we say functions are first class objects in JavaScript?	
What is a callback function?	
What is a higher-order function?	

2. DOM Manipulation

Warm-up

SWBAT

- [ ] Explain what the DOM is
- [ ] Observe how to traverse the DOM tree
- [ ] Observe how to select single dom elements with .querySelector() and getElementById()
- [ ] Observe how to select multiple elements with .querySelectorAll() and .getElementsByClassName()
- [ ] Observe how to add content with .textContent
- [ ] Observe how to create elements with .createElement
- [ ] Observe how to append elements to the dom with .appendChild and .append
- [ ] Explain the dangers of innerHTML and when it's safe to use
- [ ] Observe how to remove content with .remove

What is the DOM?	
What do we mean when we say the DOM is a tree?	

How do we select dom elements?	
How do we change DOM elements?	
What are some ways we can select multiple dom elements?	
How do we create DOM elements?	
Google Check: What is the danger of innerHTML	
How do we remove DOM elements?	

3. DOM Events

[Warm-up]

SWBAT

- ☐ Explain the importance of event handling in modern web applications
- ☐ Explain how callback functions are used with event listeners
- ☐ Observe how to add a form to a webpage using HTML and JavaScript
- ☐ Observe how onSubmit events are used to receive information from Users via forms
- ☐ Explain the purpose of .preventDefault() method
- ☐ Observe how to use MDN to discover and implement JavaScript events

What are JavaScript Events?	

List a few JavaScript events.	
What are the two arguments that .addEventListener() takes?	
What does .addEventListener() pass the callback function when triggered?	
What is a submit event?	
How do we prevent the page from refreshing when a form submits?	
How do we access the value from the form inputs?	
What is optimistic rendering?	

4. Communicating with the Server

Warm-up

SWBAT

- [ ] Describe the request-response cycle
- [ ] Explain the differences between a server and a client
- [ ] Name the different HTTP Verbs and describe their actions
- [ ] Observe how to send a GET request using .fetch()
- [ ] Explain what asynchronous means in JavaScript
- [ ] Explain why promises are important in JavaScript
- [ ] Observe how to handle promises and errors using .then() and .catch()

- [ ] Observe how json-server is used to create a local API
- [ ] Observe how to render data to the browser window after a fetch request

What is the request-response cycle?	
What is HTTP?	
What do the following HTTP methods do? GET POST PATCH PUT DELETE	
What is JSON?	
How do we install and run JSON server?	
What is .fetch()?	
What is a promise?	
What does .then() do?	

What happens when we return a fetch from a higher-order function? For example:
const foo => () {
  return fetch(url)
  .then()
}	

5. POST Requests

[Warm-up]

SWBAT

- [ ] Observe how to send a POST request using HTML forms and JavaScript
- [ ] Explain the difference between optimistic and pessimistic rendering

What is a POST request?	
Where does the data from a POST request come from?	
How does POST differ from PUT and PATCH	
What is the second argument a Fetch request takes when making any request besides a GET request?	
How is the request object formatted?	
What is 'Content-Type': 'application/json' for?	
Why do we need JSON.stringify()?	
What do we call the act of rendering content when a response returns from the server?	

6. PATCH and DELETE Requests

Warm-up

SWBAT

- [ ] Review how to send a PATCH request using HTML forms and JavaScript
- [ ] Review how to send a DELETE request using HTML buttons and JavaScript
- [ ] Explain the difference between optimistic and pessimistic rendering

What is a PATCH request?	
What are the major differences between PATCH and POST?	
How can we update data on the DOM after a PATCH?	
What is a DELETE request?	
What are the major differences between DELETE and POST?	
How do we update the dom after a DELETE request?	

7. Communicating with External APIs

Warm-up

SWBAT

- [ ] Explain what an API is
- [ ] Explain the limitations of working with an external API
- [ ] Observe how to parse API documentation

☐ Observe how to send a GET request to an external API with / without an API key

Define API.	
Give a few examples of APIs we've used.	
What can generally be found in API documentation?	
What is an API key?	
Where can we find instructions for getting and using an API key?	
Give an example of how an API key can be used.	