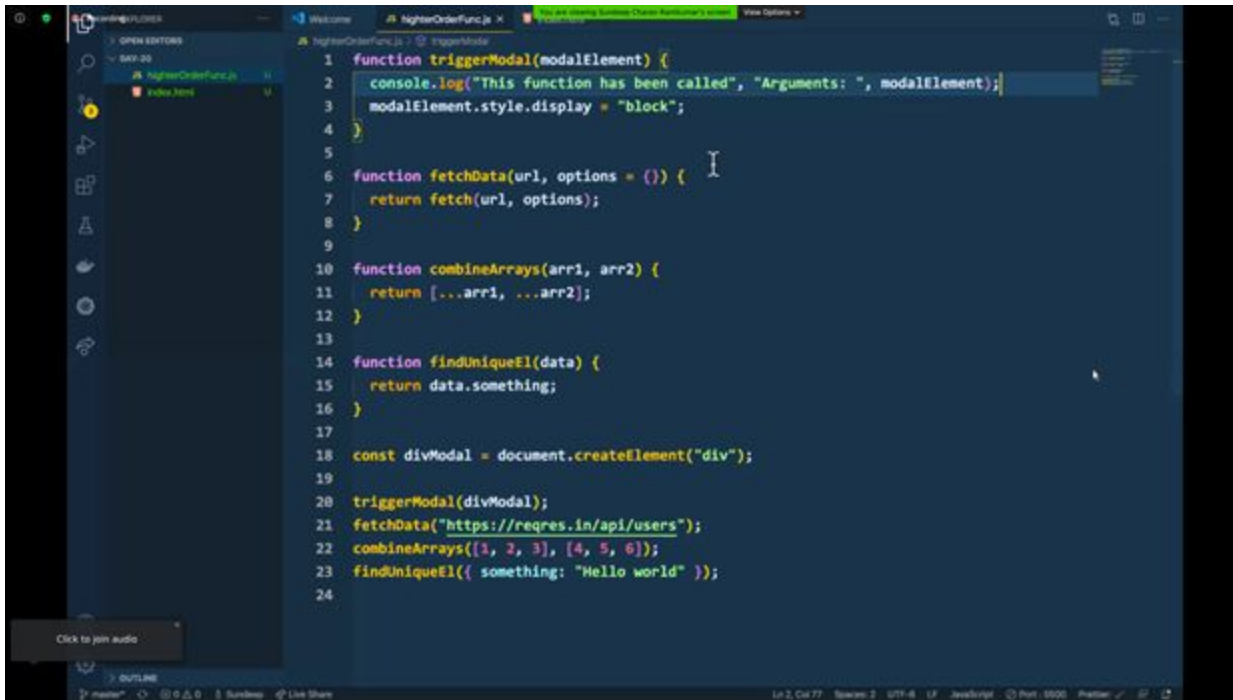


Date : 24 - 07 - 2020

Morning Session : 9 am – 11.00 PM

By ~ Sundeep Charan Ramkumar Today

Topics: Higher Order Components

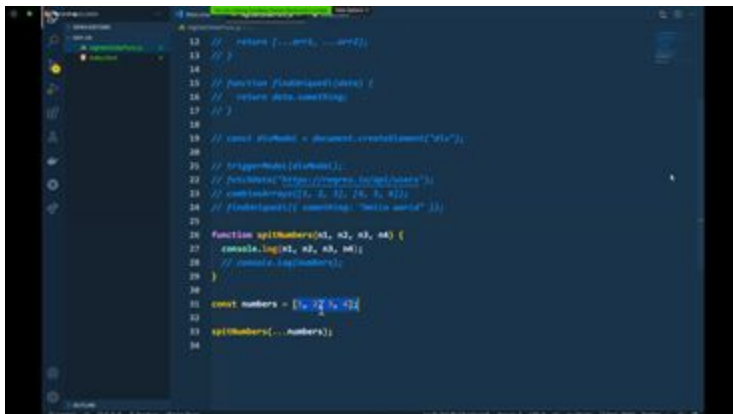


```
1 function triggerModal(modalElement) {
2   console.log("This function has been called", "Arguments: ", modalElement);
3   modalElement.style.display = "block";
4 }
5
6 function fetchData(url, options = {}) {
7   return fetch(url, options);
8 }
9
10 function combineArrays(arr1, arr2) {
11   return [...arr1, ...arr2];
12 }
13
14 function findUniqueEl(data) {
15   return data.something;
16 }
17
18 const divModal = document.createElement("div");
19
20 triggerModal(divModal);
21 fetchData("https://reqres.in/api/users");
22 combineArrays([1, 2, 3], [4, 5, 6]);
23 findUniqueEl({ something: "Hello world" });
24
```

If we want to have a logger statement, it becomes redundant to insert it everywhere.

So, we need to take help of higher order fn.

Fundas:



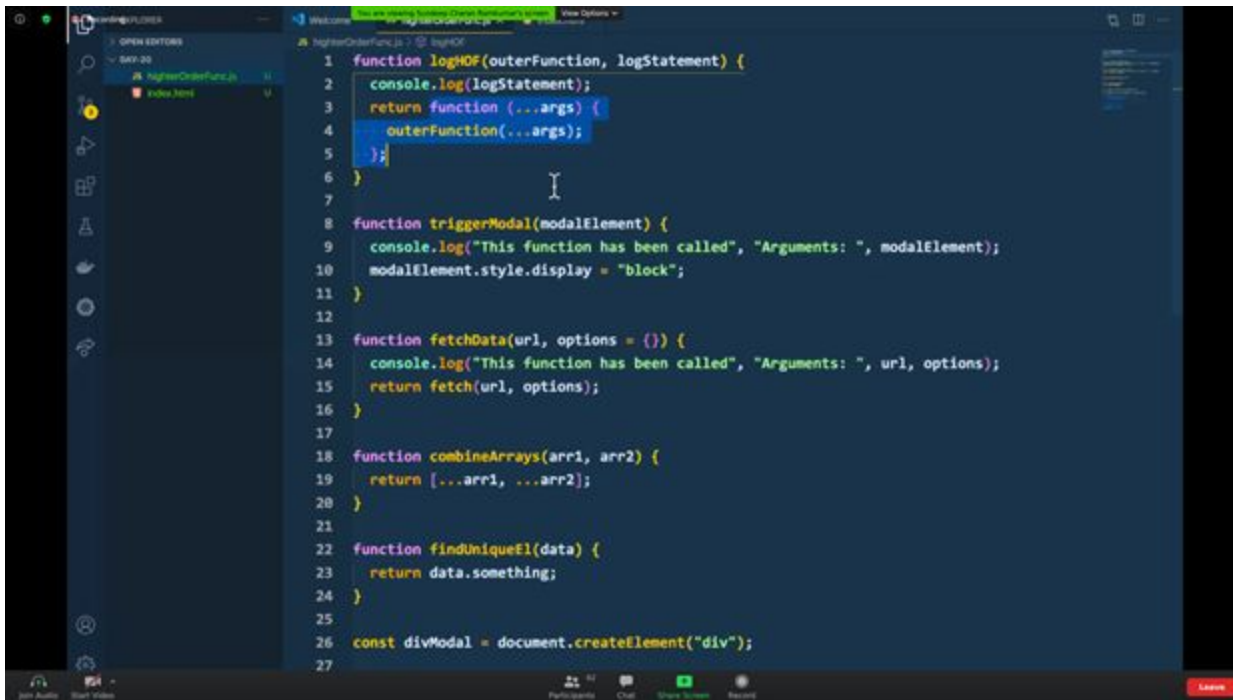
```
12 // return [...arr1, ...arr2];
13 // }
14
15 // function findUniqueEl(data) {
16 //   return data.something;
17 // }
18
19 // const divModal = document.createElement("div");
20
21 // triggerModal(divModal);
22 // fetchData("https://reqres.in/api/users");
23 // combineArrays([1, 2, 3], [4, 5, 6]);
24 // findUniqueEl({ something: "Hello world" });
25
26 function splitNumbers(arr, fn, arr2) {
27   console.log(arr, fn, arr2);
28   // console.log(arr2);
29 }
30
31 const numbers = [1, 2, 3, 4];
32
33 splitNumbers(...numbers);
34
```

If you use ... in invocation \Rightarrow scattering

If you use ... in declaration \Rightarrow batching

Higher-Order Functions:

- They accept fn as an argument or return the fn or do both.
- The below-mentioned example would help you to understand

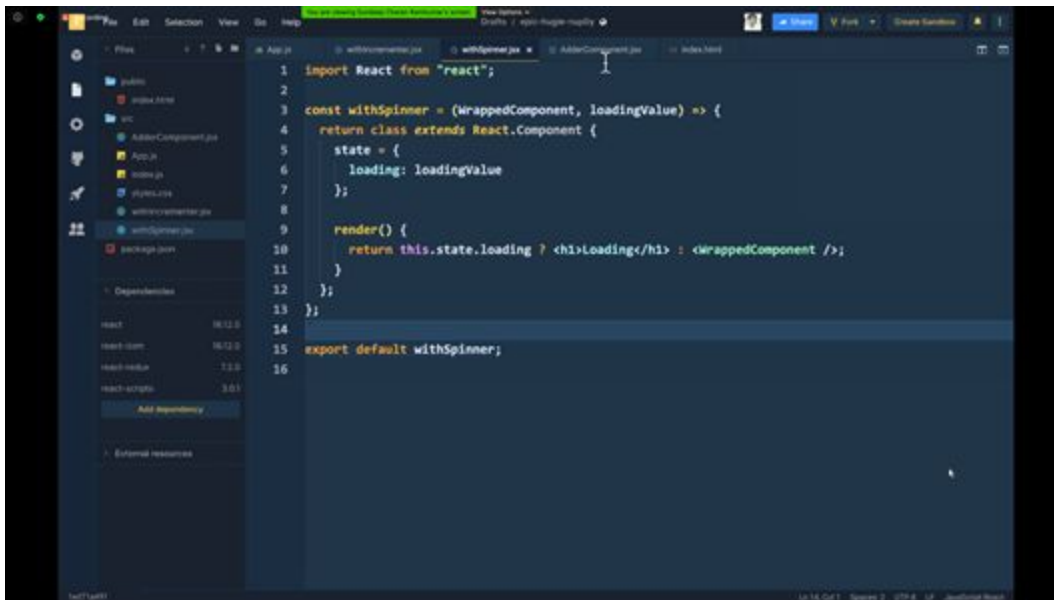


```
1 function logHOF(outerFunction, logStatement) {
2   console.log(logStatement);
3   return function (...args) {
4     outerFunction(...args);
5   };
6 }
7
8 function triggerModal(modalElement) {
9   console.log("This function has been called", "Arguments: ", modalElement);
10  modalElement.style.display = "block";
11 }
12
13 function fetchData(url, options = {}) {
14   console.log("This function has been called", "Arguments: ", url, options);
15   return fetch(url, options);
16 }
17
18 function combineArrays(arr1, arr2) {
19   return [...arr1, ...arr2];
20 }
21
22 function findUniqueEl(data) {
23   return data.something;
24 }
25
26 const divModal = document.createElement("div");
27
```

A simple way to see it: generalised

Usage:

- Debouncing \rightarrow waiting for the search to complete (delaying bouncing)
- To add a Spinner Component (with spinner)



```
1 import React from "react";
2
3 const withSpinner = (WrappedComponent, loadingValue) => {
4   return class extends React.Component {
5     state = {
6       loading: loadingValue
7     };
8
9     render() {
10      return this.state.loading ? <h1>Loading</h1> : <WrappedComponent />;
11    }
12  };
13 }
14
15 export default withSpinner;
```

Debouncing:

Debouncing is nothing but reducing unnecessary time-consuming computations so as to increase browser performance. There are some scenarios in which some functionalities take more time to execute a certain operation. For instance, take an example of a search bar in an e-commerce website.

Note:

- A higher-order component (HOC) is an advanced technique in React for reusing component logic. HOCs are not part of the React API, per se.

They are a pattern that emerges from React's compositional nature

Resources:

<https://reactjs.org/docs/higher-order-components.html>

<https://www.geeksforgeeks.org/debouncing-in-javascript/>

