

Chapter 05 Assignment Theory

1. What is the difference between Named Export, Default export and * as export?

Named Export	Default Export	* Export
With named exports, one can have multiple named exports per file. Then import the specific exports they want surrounded in braces. The name of imported module has to be the same as the name of the exported module.	One can have only one default export per file. When we import we have to specify a name and import like:	Import all the named exports onto an object:
Example: <pre>import { MyComponent, MyComponent2 } from "./MyComponent";</pre>	<pre>// import import MyDefaultComponent from "./MyDefaultExport"; // export const MyComponent = () => {} export default MyComponent;</pre>	<pre>// export export const MyComponent = () => {} // import import * as MainComponents from "./MyComponent"; // use MainComponents.MyComponent</pre>

2. What is the importance of config.js file

Configuration files ("config files" for short) are important to modern computing. Configuration formats help developers store the data their applications need and help users store preferences for how they want applications to act. For example, you can store constants like API, API Key, Environment etc.

3. What are React Hooks?

React hooks are basically a JS function offering an approach to state management and lifecycle methods in the application. They let you “hook into” React state and lifecycle features from function components.

Before Hooks, React Functional components were only used for presentation purposes—to render data to the UI. They could only receive and render props from parent components, which were usually class components. Functional components did not keep track of an internal state and did not know the component lifecycle. Thus, they were referred to as “dumb components.”

4. Why do we need a useState Hook?

In react context we need useState to create state variable that are different from normal variables in their ability to cause re-render of component if state variable value is changed.

Syntax:

const [stateVar, setState] = useState(initialState)

where **stateVar** is the initial state variable, **setState** is the function to update the value of the variable and **initialState** is the initial default value you want to set in the **stateVar**.