what are the difference between variables and finctions give me in simple way

container to store a value variables i.e. =>x=1

functions are reuseable if we write code once we canreuse it no need of writing again and again

hosting :variables var() and functions move at the top of the scope during the execution

binding means to attach this keyword with function i.e bind() call()

closure : means the function remains in the environment where it is created it will assess the variables from outside if it finished executing

**React Virtual DOM and Tree Reconciliation:**

**Virtual dom is a light wait copy of actual dom react use it of better performance manipulatin with real dom**

**Tree reconciliation is a process to update the dom. It compares with previous dom a applies only necessary changes to real dom**

**Advantages of using hooks and custom hooks in React:**

**hooke allow to use states without class and it simplify the code to reuse**

**custom hooks :let us to extract component logic into reuseable functions and cleans the code**

**Differences between functional and class components in React:**

**Functional components it is light weight,uss hooke and give beterpreformance**

**class components :use this.state method and lifecycle method componentdidmount (boilarpaltes)**

**Managing state in a React application:**

**State can manage by use state it is functional component and this.state is managed by it is**

**what are the difference between variables and finctions**

* **container to store a value variable i.e. =>x=1**
* **functions are reuseable if we write code once, we can reuse it no need of writing again and again**
* **hosting :variables var() and functions move at the top of the scope during the execution**
* **binding means to attach this keyword with function i.e. bind() call()**
* **closure : means the function remains in the environment where it is created it would assess the variables from outside if it finished executing**

**React Virtual DOM and Tree Reconciliation:**

* **Virtual dom is a light wait copy of actual dom react use it of better performance manipulation with real dom**
* **Tree reconciliation is a process to update the dom. It compares with previous dom a applies only necessary changes to real Dom**

**Advantages of using hooks and custom hooks in React:**

* **hooke allow to use states without class and it simplify the code to reuse**
* **custom hooks :let us to extract component logic into reuseable functions and cleans the code**

**Differences between functional and class components in React:**

* **functional components: are simpler and more concise. They are pure function it takes props as**
* **input and return jsx to describe what component the render and it is perfect choice for presentational components**
* **class component is the blueprint to define how component renders and behaves it use class keywords and allows**
* **the functionality like state management and lifecycle method (boiler plates)**

**What is react**

* **A react is a java script library usef to create friendly user interface particularly for single page application by allows developers to create reusable components and manage the state of application efficiently**

**Why to use React over another framework ?**

* **Performance : react give the performance good and keep our app fast**
* **Com[possibility : the components are reuseable and easy to maintain**
* **And it has Jsx syntax means we can use html and JavaScript in the sane page which make the developer easier**

**Different between flex and grid**

* **Flex : one-dimensional layout .arrange elements in rows and column**
* **Perfect for simple layouts with horizontal and vertical alignment.**
* **Easily adapts to various screen sizes**
* **Greate for navigation bar and header footer**
* **grid : Two-dimensional layout creates rows and columns for complex layout**
* **maintain flexibility about device**
* **perfect for dashboard, product listing**
* **getelementby id:use to retrieve element by its unique id ; easy to find the element in exart id**
* **getelementbyclass:use for multiple elements with same class;userfull when we have multiple elements with same class**

**What are ES6 Features ?**

* **1 Arrow function : it give a clear information hoe to define functions ; const greet = name =>`hello , ${name}!`;**
* **2 Template litrals :const =`welcome ,${user.name}`**
* **3 Destructing const {firstname,secondname} = user**
* **4 calsses & modules**

**Slice : means it includes state itself.a bunch of actions which causes change of state**

**Trim : removes the white space from a both ends of the string**

**techical bug example**

1. **Situation: *"Users reported the app froze after submitting a form..."***
2. **Task: *"I needed to debug the infinite re-render issue..."***
3. **Action: *"Traced it to a missing dependency in useEffect, then..."***
4. **Result: *"Fixed it with a cleanup function, reducing crashes by 100%."***

**2nd example**

* **Situation: *"Users reported their profile information was disappearing when updating single fields..."***
* **Task: *"I needed to preserve existing data while allowing partial updates..."***
* **Action: *"Discovered we weren't merging state properly, then implemented a spread operator solution..."***
* **Result: *"Fixed all form updates and added a custom hook useMergeState to prevent future occurrences."***

**what are the difference between variables and finctions give me in simple way**

**container to store a value variables i.e. =>x=1**

**functions are reuseable if we write code once we canreuse it no need of writing again and again**

**hosting :variables var() and functions move at the top of the scope during the execution**

**binding means to attach this keyword with function i.e bind() call()**

**closure : means the function remains in the environment where it is created it will assess the variables from outside if it finished executing**

**React Virtual DOM and Tree Reconciliation:**

**Virtual dom is a light wait copy of actual dom react use it of better performance manipulatin with real dom**

**Tree reconciliation is a process to update the dom. It compares with previous dom a applies only necessary changes to real dom**

**Advantages of using hooks and custom hooks in React:**

**hooke allow to use states without class and it simplify the code to reuse**

**custom hooks :let us to extract component logic into reuseable functions and cleans the code**

**Differences between functional and class components in React:**

**Functional components it is light weight,uss hooke and give beterpreformance**

**class components :use this.state method and lifecycle method componentdidmount (boilarpaltes)**

**Managing state in a React application:**

**State can manage by use state it is functional component and this.state is managed by it is**

**Closure : is a function that remember its outer variable even after it finished running**

**function outer() {**

**let count = 0; // `count` persists due to closure**

**function inner() {**

**count++; // Modifies the `count` from `outer`**

**console.log(count);**

**return inner; // Returns the inner function (with access to `count`)**

**}**

**const counter = outer(); // `outer()` finishes, but `count` lives on**

**counter(); // Output: 1**

**counter(); // Output: 2 (remembers previous `count`)**

**--------------------------------------------------------------------------------------------------------------------------------------------**

**optimization**

***"First, I’d profile the app using React DevTools to identify slow renders. Then, I’d:***

1. **Memoize components with React.memo/useMemo.**
2. **Lazy-load routes/components to reduce initial bundle size.**
3. **Optimize state updates (avoid spreading state too deeply).**
4. **Virtualize long lists (e.g., with react-window).**
5. **Use keys effectively to prevent DOM reconciliation issues."\***