

10/10/2024

Introduction to Unit Testing

- Testing is a process of checking whether our application is working the way it is intended to do. If we plan and execute testing properly, it ensures a good product delivery to the customer.
- Software Testing is defined as "A process of analysing a software item to detect the differences between existing and required conditions & to evaluate the features of the software item".
- Unit Testing is the process of testing each module or component of the application individually to check whether they are working as expected.
- Unit testing is the first testing performed by the developer & it is an integral part of development & it should be planned & executed in conjunction with development.

Software Product

module 1,

module 2,

module 3

test case 1

test case 2

test case 3

Testcase: set of instructions or ways to check the software product.

Testsuite: combination of multiple test cases

describe ("Testsuite title", function ()

```
{  
    if ("testcase1", function () { ... });  
    if ("testcase2", function () { ... });  
    if ("testcase3", function () { ... });  
}
```

- Q1) WAP using Js arrow function to check whether a no. is a prime no. or not.
 Q2) WAP using Js first class function to generate Fibonacci series of first 11 terms.
 Q3) WAP using Js first class function to find Lcm of n fine random Yenen numbers.
 Program files :- [30aps-Prime-using Arrow.html](#)
[30aps-fibonacci.html](#)
2 May 84
 Q1) WAP using Js to find the largest among 4 numbers using == comparison operator. [3may-largestNo.html](#)
 Q2) WAP using Js to convert the temperature from Fahrenheit to celsius $C = \frac{5}{9}(F - 32)$. [3may-temperature.html](#)
 Q3) WAP using Js to print all the prime numbers in a given interval. [3may-Prime.html](#)
 Q4) WAP using Js to find the number of even factors of a given input no. and display the even factor. [3may-EvenFactors.html](#)
 Q5) WAP using Js to implement the Async Await using arrow function. [3may-Await.html](#)
 Q6) WAP using Js to create a promise to log the address sent after a delay of 14 second using setTimeout. [3may-Promise.html](#)
 Q7) WAP using Js to implement method overriding. [3may-OVERRIDING.html](#)
 Q8) WAP using Js to implement the use of this keyword need in inheritance. [3may-this.html](#)
 Q9) WAP using Js to implement various coloring schemes. [3may-ColorScheme.html](#)
 Q10) WAP using Js to implement the navigation option for any web page. [3may-NavigationOption.html](#)

- Q11) WAP using Js to implement the use of filter() in filter the extict species. [3may-filter.html](#)
 Q12) WAP using Js to filter the extict species using Iterative programming style. [3may-filter-Iterative.html](#)
3 May 84
 Q1) WAP using Js to check whether a given number is a Neon Number or not. (ie a number where the sum of the digits is equal to the number). [3may-Q1-Neon.html](#)
 Q2) WAP using Js to convert a number from Decimal to Octal equivalent. [3may-Decimal-Octal.html](#)
 Q3) WAP using Js to convert a number from binary to Octal equivalent. [3may-Binary-Octal.html](#)
 Q4) WAP using Js to print the following pattern:

```

      A
     B A B
    C B A B C
  
```

Q5) WAP using Js Object oriented Programming Style to perform matrix multiplication. [3may-Matrix.html](#)

Q6) WAP using Js First Class Function to find the transpose of a 5×5 Matrix. [3may-Transpose.html](#)

7 May 84

React

Step 1 : Install node.js
 In terminal : `npm create-react-app myapp`
`cd myapp`
`npm start`

\$ node -v
v8.0.12.0

\$ npx create-react-app myapp
Happy hacking!
Open VS code
Open terminal
New terminal
cd myapp
npm start

Q) What is JSX? It is an extension of JS class function to convert any input string from lower to upper case letters & also do in pure function to convert a string from upper to lower. Also do in arrow function, map(), reduce() L:\may-1\1.html

Translated into regular JS as follows. In case of JSX, a method named as createRoot() which helps to display the react components inside the browser. 'render' is a fundamental part of class component & is used to display the component on UI.

16 May '24 Advantage of single page application:-
→ Here no page reload is required, no extra wait time is needed, and it is simple.

React: It is a framework which represents an open-source JS library for building the user interface.

Why to learn React?

The react based project was created & maintained by facebook and it has component based architecture.

Define JSX:

JSX stands as syntactic extension of JS written to be used in react. Its main aim is to embed user interfaces comprising of HTML, along with CSS & JS.

→ JSX element can be passed to a function as a parameter & it can be stored in an object of array.

→ JSX basically stands for XML which is basically used to allow us for writing HTML inside react. JSX allows us to write HTML elements in the JS & place them in the document object model.

Aim of JSX:-

→ JSX allows us to write HTML within the JS code.

→ If we want to create an element without using JSX following syntax is used.

without JSX :-

```
const myElement = React.createElement('h1', {}, 'h1');
```

With JSX :-

```
const myElement = <h1> Good Morning </h1>
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(myElement);
```

→ Here `createRoot()` is a method i.e. we need to create a root to display the react components inside a browser & 'render' is a fundamental part of class component to display component on user interface.

JSX Syntax :-

→ JSX comprises of singeliner elements using html tags with code embedded in expression enclosed within `{ }`

```
const element = <h1> hello, {name} </h1>
```

```
const e1 = <p> XYZ </p>
```

```
const e2 = <p> abc </p>
```

```
const e3 = <p> xyzabc </p> {5000 * 23} </p>
```

→ We can also write any kind of expression, numeric values, string manipulation within JSX.

→ A function named as 'convertToCaps(name)' is used in JSX to convert a string to upper case letter.

```
const element = <h1> hello Geeta your nickname is  
convertToCaps(geeta) and you have taken ₹10,000₹ rupees  
from me </h1>
```

[NOTE 1] If JSX expression extends to multiple lines at that time the multiple line need to be enclosed

within a pair of Parenthesis.

→ If multiple tags, const element = <h1> xyzabc </h1> for single tag.

```
const element = <div>
```

```
<h1> Good Morning </h1>
```

```
<h2> Good Night </h2>
```

```
<p> Hello ! Its me XYZ </p>
```

[NOTE 2]

If case of JSX you can write expression inside curly braces.

```
const element = <h1> {5 + 5} ? </h1>
```

{ }

```
return <div> className = 'app'
```

```
<h3> hello <b>xyzabc </b> type </h3>
```

```
<div className = 'box'>
```

```
<p className = "title" > click here to see magic</p>
```

```
</div>
```

```
</div>
```

→ JSX Fragments :- A function can return the value as follows:

```
const getobj = () => { let mystuff = {  
  name : 'Bishnu',  
  expertise : 'React';
```

→ Here the functional components in react are JS function that return JSX.

```
const header = () => {  
  return {<h1> ... </h1>  
        <h2> ... </h2>} ;  
}
```

→ Why to use fragments?

fragments in JS:

→ It allows us to wrap & group elements together where you want to treat as a single element.

→ The fragments do not introduce new elements to the document object model.

return (`<>` `<h1>` `hi</h1>`)
`</p> hello </p>` {fragment} {ifragment}

`using src = "path">`

11 May 24

11)

→ Types of Testing :-

- (1) Unit
- (2) Integration
- (3) E2E

Unit Testing :- It is a type of testing used to test the functionality of the individual component in isolation. It is used in React.

Integration :- It is the testing done to check the combine effect of all the modules at a time.

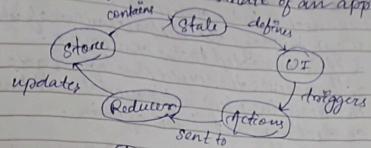
E2E testing :- It ensures that a proper user experience is received from the user perspective by testing an entire program or application.

Middleware in React :- These are the gatekeepers for the developers the chance to do extra things with the action like running some extra checkings & doing certain stuffs in the background.

Redux Architecture :-

→ It is used as a state managing library in case of JS application

for managing the data & state of an application.



Redux Architecture

→ It is the architecture that helps in creating applications that are predictable, testable & can run in different environments.

→ Here, in the diagram,

Store :- It represents an object that brings actions and the reducer all together. It holds the application state & handles the event listeners.

Actions :- Actions are payload which send data from application to redux store & they are the only source of information for the store.

Reducers :- It specifies how the state of the application changes in response to actions sent to the store. Here redux store is the main control bracket which contains all the states of application.

MVC Architecture (Model View Controller)

Model - Database

view - client

controller - server

→ It is used in react for implementing user interface & it is used as a software design pattern.

→ 3 major components of MVC :-

Model :- The model stands for database & it is used for storing the data & is connected to the controller.

View :- The view displays the visualization of data to be the user & it is connected to the controller.

Controller :- It processes the server side logic & acts as a middleware between the view & the model and basically it is used for controlling the flow of data.

- Controlled vs Uncontrolled component in React
- In React, controlled component refers to components where the state & the behaviour are controlled by the parent component.
 - Uncontrolled component are ones having control of their own state & they manage the behaviour on themselves.

Inline Style in React :-

Class Component in React :- It must have the following features

- (i) It must be a JS class
- (ii) It must extend the React component
- (iii) It must define a react method to return.

class Welcome extends React.Component

{

 render()

{

 return <h1>Hello Guys! </h1>

}

Inline Styling in React :-

class MyHeading extends React.Component

{

 render()

{

 return (<div>

 style={ { color: 'red' } }

 >Hello

 </div>)

}

Using External Styling sheet in React :- For this, we can write our class styling in a separate file with .css extension App.css

body {

 background-color: lightblue;

 color: white;

 padding: 40px;

 font-family: Arial;

 text-align: center;

}

→ Now we can import the stylesheet in our application as follows

import React from 'react';

import ReactDOM from 'react-dom/client';

import './App.css';

class Myclass extends React.Component

{

 render()

{

 return (<div> My first React application </div>);

}

Props:- It represents attribute of components used in react.

The components can be of 2 types :-

(i) Statefull component

(ii) Stateless component

Statefull components :- These are the class components that can handle state.

Stateless Components :- These are the components that do not have state associated with them.

React Reconciliation Algorithm :- It is used for bringing harmony between performance & user experience for betterment in web development. It is the process by which the React determines the changes made to the virtual document object model & applies those changes to the actual document model. It uses concurrent mode that allows react to work on multiple task concurrently.

→ The new reconciliation algorithm in React 18 known as concurrent React which divides the reconciliation work into smaller units called fibers & priorities them based on their importance.

Fiber reconciliation :- A fiber represents a components and its corresponding work. By dividing the reconciliation work into smaller fiber, react can manage and prioritize the fiber effectively.

→ Two important phases involved in react reconciliation :-

(1) Render phase :- In it, react traverses the component tree & creates & update the fiber.

(2) Commit phase :- In it, react applies the changes to the

actual document model. It is divided into multiple priority level & various & critical updates are made inside phase..

Function component :- It represents simple JS functions we can create a functional component to react by writing a simple JS function inside it.

Syntax :- const car = () =>

React LifeCycle :- return <h1>Hello</h1>?
Every component in react has a life cycle that we can monitor & manipulate under three phases :-

(i) Mounting phase

(ii) Updating phase

(iii) OnMounting phase

Mounting Phase :-

It means putting the elements into document object model 4 built in methods are need for React for mounting :-

(1) constructor

(2) getDerivedStateFromProps

(3) render

(4) ComponentDidMount

Updating Phase :-

It is the next phase after mounting. A component is updated when there is a change in state of the component.

update :-

↳ getDerivedStateFromProps()

↳ shouldComponentUpdate()

↳ render()

↳ getSnapshotBeforeUpdate()

↳ ComponentDidUpdated()

Unmounting Phase :- It is used when a component is removed the component
Unmounting → Component Will Unmount (C)