# Q1: What is React JS?

Ans : React, also known as React.js, is an open-source javascript library for building user interfaces. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React is used to build user interfaces for single-page applications, mobile apps, and web applications.

# Q2: What is NPM in React JS?

Ans: NPM, or the Node Package Manager, is a package manager manager for JavaScript that is used to install and manage React packages. NPM is used to install React and its dependencies, as well as third-party packages that can be used in React applications.

# Q3: What is Role of Node JS in react JS?

Ans: Node.js is a JavaScript server-side environment that is used to create scalable and high-performing web applications. React is a JavaScript library that is used to create user interfaces.

The two technologies can be used together to create full-stack web applications, where React is used for the front-end and Node.js is used for the back-end.

# Q4: What is CLI command In React JS?

Ans: CLI (Command Line Interface) tools in React are command-line tools that help developers automate repetitive tasks and streamline their development workflow. They are typically used to create, build, test, and deploy React applications from the command line.

# Q5: What is Components in React JS?

Ans: A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

# Q6: What is Header and Content Components in React Js?

Ans:1) Header component :-

The Header component is a top section of a web page or application. Header component is used to display the essential elements of website that is needed to be displayed, and can be accessed from any page of website. It often contents elements like:

1. Logo of website.
2. Navigation menus.
3. Search box.
4. Logout/Login Buttons.
5. Or any other special element.

2) Content component :-

The Content component represents the main section of webpage or application. It’s where the main content of a website is shown. There can be many content component for different web pages for displaying different data related to website. The content varies on the purpose of the application.

Content components are often designed to be flexible and reusable, allowing you to swap out different content while keeping the same layout and structure. The content components includes of the following elements:-

1. Articles.
2. Products .
3. User dashboard.
4. And many other essential elements.

# Example of a Header component in React.js

## Import React from ‘react’;

## export default const Header = () => {

## return (

## <header>

## <nav>

## <ul>

## <li><a href=”#”> Home </a></li>

## <li><a href=”#”> About </a></li>

## <li><a href=”#”> Services </a></li>

## <li><a href=”#”> Contact Us </a></li>

## </ul>

## </nav>

## </header>

## ) };

## 

# Example of a Content component in React.js

## Import React from ‘react’;

## export default const Content = () => {

## return (

## <div className=”content”>

## <h1> Welcome to Our Website. </h1>

## <p> Lorem ipsum dolor sit amet, consecterur adipiscing elitinng … </p>

## { /\* Additional Content Elements. \*/ }

## </div>

## ) };

# Q7: How to install React Js on Windows, Linux Operating System? How to install NPM and How to check version of NPM?

Ans:

|  |
| --- |
| Installation of React JS and npm on Windows : |

1. Install Node.js : React.js requires Node.js, which includes npm. You can download the windows installer from the official Node.js website : <https://nodejs.org/>
2. Visit the website and download the LTS (Long Term Support) version for Windows.
3. Run the installer and follow the installation instructions.

|  |
| --- |
| Verify Installation checking version |

The installation was completed, then open a Command Prompt or Powershell window and enter the following commands to verify that Node.js and npm have been installed successfully :

* Node –v
* npm –v

|  |
| --- |
| Install Create React App |

Command for installing the Create React App:-

# npm install –g create-react-app.

|  |
| --- |
| Create A New React Project |

Command for creating a new React project:-

# create-react-app app-name.

|  |
| --- |
| Installing React and npm |

* We have to install Node.js and npm in macOs system as well. Steps to install Node.js and npm in macOS are given below:-

1. Visit the official Node.js website.
2. Download the macOs installer.
3. Once downloaded, click the **.pkg** file in your Downloads folder to run the installer and then follow the steps appear on your screen.

* Commands for verification:-

1. node –v
2. npm -v

* First, open the terminal and create a folder named react-app by typing the following command:

-mkdir react-app

|  |
| --- |
| Installing React and npm on Linux |

|  |
| --- |
| Verify installation and checking version |

* sudo apt install npm
* npm –version
* node -version

1. create-react-app is a utility that allows you to set up all the tools required for a React application.
2. To install the tool , run the following npm command:-

* sudo npm –g install create-react-app

|  |
| --- |
| Verifying version of create-react-app |

Command:- create-react-app --version

|  |
| --- |
| Install Create React App utility |

Now you’re ready to create your React app by using command given below:-

* create-react-app app-name

|  |
| --- |
| Command for running the project |

Use below command to run the project in above operating storage:

* npm start.

### Q8: How to check version of React Js?

Ans: Three ways to find out version of React:-

1. Using package.json file.
2. Using command line.
3. Using the version property of default import from React.

1.Using package.json file :-

* The package.json contains metadata about our project.
* It is created by default when we create our React project.
* We can create a react app using the command mentioned below.
* npm create-react-app aap-name.

2.Using the command line :-

* We can easily check the React version by using the command mentioned below on our command line.
* npm view react version.

3.Using the version property of default import from React :

* The default import from React library is an object that has a version property on it.
* We can use this property inside out JSX elements in our desired manner.
* Import React from ‘react’;
* Let a = React.version

### Q9: How to change in components of React Js?

Ans:- Three ways to make a change in component:-

# Update in State

* The state change can be from a prop or setState change to update a variable.
* The component gets the updated state and React re-renders the component to reflect the change on the app.

# Update in prop :

* Likewise the change in prop leads to state change and state change leads to re-rendering of the component by React.

1. Re-rendering of parent component **:**

* Whenever the components render function is called, all its subsequent child components will re-render, regardless of whether their props have changed or not.

### Q10: How to Create a List View in React JS ?

Ans:

* **Step 1 :**
* Create a list of elements in React in the form of an array ans store it in a variable.
* We will render this list as an unordered list element in the browser.
* **Step 2 :**
* We will then traverse the list using the JavaScript map() function and updates elements to be enclosed between <li> </li> elements.
* **Step 3 :**
* Finally we will wrap this new list within <ul> </ul> elements and render it to the DOM.

## Example :

## Import React from ‘react’;

## Import ReactDOM from ‘react-dom’;

## const numbers = [ 1,2,3,4,5 ];

## const updateNums = number.map((number) => {

## return <li> {number} </li>;

## });

## ReactDOM.render(

## <ul>

## { updateNums };

## </ul>,

## document.getElementByID(‘root’)

## );

## Q11: Create Increment decrement state change by button click?

Ans:

import React, { useState } from 'react'

export default function Counter() {

    const [counter, setcounter] = useState(0)

    //declared a state with vlaue 0

    const increment = () => {

        setcounter(counter + 1)

    };

    // make an arrow function for increment

    const decrement = () => {

        if (counter > 0) {

            setcounter(counter - 1)

        }

    };

    // make an arrow function for decrement

    const reset = () => {

        setcounter(0)

    };

    // make an arrow function for reset

    return (

        <>

            <div className='bg-secondary d-flex flex-column justify-content-center align-items-center' style={{ height: '100vh', width: '100vw' }}>

                <div className="w-100 d-flex align-items-center justify-content-center">

                    <h1 className='border px-4'>{counter}</h1>

                </div>

                {/\* displayed counting  in h1 tag \*/}

                <div className="px-5 w-25 mb-3 d-flex align-items-center justify-content-around">

                    <button className='btn btn-success' onClick={decrement}>Decrement</button>

                    <button className='btn btn-success' onClick={increment}>Increment</button>

                </div>

                {/\* displayed increment and decrement button in seperate div \*/}

                <div className="w-100 d-flex align-items-center justify-content-center">

                    <button onClick={reset} className='btn btn-danger'>Reset</button>

                </div>

                {/\* displayed reset button in seperate div \*/}

            </div>

        </>

    )

}

# Output:-

