What is React Js?

React is an open-source JavaScript library for building user interfaces. It is declarative, efficient, and flexible. React makes it easy to create interactive UIs by using a component-based approach.

Components are self-contained pieces of code that can be reused throughout your application. This makes your code more modular and easier to maintain. React also uses a virtual DOM, which makes updates to the UI very fast.

Here are some of the benefits of using React:

Declarative:

React uses a declarative programming style, which means that you describe what you want the UI to look like, and React takes care of the rest. This makes your code more concise and easier to read.

Efficient:

React is very efficient at updating the UI. It uses a virtual DOM, which is a lightweight representation of the real DOM. React only updates the real DOM when necessary, which makes your application faster and more responsive.

Flexible:

React is very flexible. You can use it to build any type of UI, from simple websites to complex web applications.

If you are looking for a JavaScript library to help you build user interfaces, React is a great option. It is easy to learn, efficient, and flexible.

2. What is NPM in React Js?

The name **npm** (Node Package Manager) stems from when npm first was created as a package manager for Node.js.

All **npm** packages are defined in files called **package.json**.

The content of package.json must be written in **JSON**.

At least two fields must be present in the definition file: **name** and **version**.

npm can manage **dependencies**.

npm can (in one command line) install all the dependencies of a project.

Dependencies are also defined in **package.json**.

3. What is Role of Node Js in react Js?

Node.js is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. You need to remember that **NodeJS is not a framework and it's not a programming language**. Most people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App.

Features of Node.js: There are other programming languages also that we can use to build back-end services so what makes Node.js different I am going to explain.

- It's easy to get started and can be used for prototyping and agile development
- It provides fast and highly scalable services
- It uses JavaScript everywhere so it's easy for a JavaScript programmer to build back-end services using Node.js
- The source code is cleaner and consistent.
- Large ecosystem for open source library.
- It has an Asynchronous or Non-blocking nature.

4. What is CLI command In React Js?

In React.js, the CLI (Command Line Interface) command used for creating a new React application is `create-react-app`. This command is used to bootstrap a new React project with a basic directory structure and necessary configuration files.

Here's how you can use it:

In React.js, the CLI (Command Line Interface) command used for creating a new React application is create-react-app. This command is used to bootstrap a new React project with a basic directory structure and necessary configuration files.

Here's how you can use it:

First, you need to have Node.js installed on your system. You can download and install it from the official Node.js website if you haven't already.

Once Node.js is installed, open your terminal or command prompt.

To create a new React application, run the following command:

5. What is Components in React Js?

In React.js, components are the building blocks of user interfaces. They are reusable, self-contained pieces of code that encapsulate a specific functionality or UI element. Components can be thought of as custom HTML elements that allow you to split the UI into independent, reusable pieces, which makes the code more manageable, modular, and easier to maintain.

There are two main types of components in React:

1. Also known as stateless components or presentational components. They are defined as JavaScript functions and return JSX (JavaScript XML) to describe what should be rendered on the screen. Functional components are primarily used for simple UI elements without any internal state or lifecycle methods.

Also known as stateful components or container components. They are defined using ES6 classes and extend the React.Component class. Class components have access to React features such as lifecycle methods and internal state.

6. What is Header and Content Components in React Js?

In React.js, "Header" and "Content" components are typically used in the context of creating user interfaces. These components are often part of a broader layout or structure of a web application.

Header Component:

The Header component typically represents the top section of a web page or application. It often contains elements such as logos, navigation menus, search bars, user authentication controls, or any other elements that are meant to be consistently displayed at the top of the page across different views or pages within the application.

The Header component is usually designed to be static or semi-static, meaning its content remains largely unchanged as the user navigates through different parts of the application.

Content Component:

The Content component represents the main content area of a web page or application. It typically contains the dynamic content that changes based on user interactions, data fetched from a server, or any other factors.

The Content component is often designed to be flexible and responsive, adapting to various screen sizes and orientations. It may include components for displaying text, images, forms, interactive elements, or any other type of content relevant to the specific view or page within the application.

In React.js, these components are created as reusable building blocks that can be composed together to construct complex user interfaces. They can also be styled, customized, and configured according to the specific requirements of the application. Additionally, they can be combined with other components, such as Footer, Sidebar, or Modal components, to create more sophisticated layouts and user experiences.

7. How to install React Js on Windows, Linux Operating System? How to install NPM and How to check version of NPM?

1.sudo apt update

2.sudo apt install nodejs npm

3.node -v

4.sudo npm install -g create-react-app

5.npx create-react-app my-react-app

6.cd my-react-app

7.npm start

8.npm -v

8. How to check version of React Js?

To check the version of React.js in your project, you can use either npm or yarn, depending on which package manager you are using. Here's how you can do it:

1. npm list react/ yarn list react

9. How to change in components of React Js?

To make changes to components in a React.js application, you typically follow these steps:

Identify the Component: Determine which component(s) you want to modify.

Navigate to the Component File: Locate the file(s) where the component(s) are defined. In a typical React.js project, components are often defined in separate files within a components directory or similar structure.

Make Changes: Open the file corresponding to the component you want to modify. Then, make the necessary changes to the component's code. This might involve changing JSX markup, adding or modifying state or props, or implementing new methods.

Save Changes: Save the file after making your modifications.

Review Changes: Review the changes you've made to ensure they meet your requirements and haven't introduced any errors.

Test Changes: Run your React.js application locally to test the changes you've made. Make sure that the component behaves as expected and that your changes haven't introduced any bugs.

Commit Changes (if using version control): If you're using version control (e.g., Git), commit your changes with an appropriate commit message describing the modifications you've made.

Deploy Changes (if applicable): If your application is live and you've made changes that need to be deployed, follow your deployment process to push the changes to your production environment.

Monitor for Issues: Keep an eye on your application after deploying changes to ensure that everything is working correctly. If users report any issues, address them promptly.

Repeat as Necessary: If you need to make further changes or updates to your components, repeat the process outlined above.

Remember to follow best practices such as code readability, modularization, and writing reusable components to maintain a clean and maintainable codebase. Additionally, it's essential to thoroughly test any changes you make to ensure they don't negatively impact your application's functionality.

11. • Create Increment decrement state change by button click?

```
import React, { useState } from 'react'
import { Button } from 'reactstrap'

export default function Counter() {
  let [count, setcount] = useState(0)
  const inccount=() => {
    console.log('--inccount--', inccount);
    setcount(count+1)
  }
  return (
    <div className='text-black'>
      <h1>counter is {count}</h1>
      <Button className='bg-dark' onClick={inccount}>Inc Count</Button>
    </div>
 )
}
```