# React.js

Question : 1

What is react.js ? How is it different from other javascript framwork andlibraies ?

→ React is an **open.souece** javascript library used building user interfaces (UIS), primarily for - single -page application. (SPAS),
→ React.js differs from other javascript framework and libraries in several key ways.

Question : 2

Explain the core prainciples of react such as the virtual DOM and component baced architecture.

→ React.js is built on a set of core principles that make it powerful efficient and developer friendly two of the most significant principles are the virtual DOM and component – Based architecture.
DOM (Document object model )

**1. Virtual DOM**

The **Virtual DOM** is a lightweight representation of the real DOM (Document Object Model) used by React to optimize UI rendering.

**How it Works:**

1. **Virtual DOM Creation**:
   React creates a Virtual DOM whenever there is a change in the application's state or UI.
2. **Diffing Algorithm**:
   React compares (or "diffs") the new Virtual DOM with the previous version to identify the exact changes.
3. **Efficient Updates**:
   React updates only the specific parts of the real DOM that have changed, instead of re-rendering the entire UI. This process is called **reconciliation**.

**Benefits of the Virtual DOM:**

- **Improved Performance**:
  DOM manipulations are expensive; by updating only the parts that change, React reduces the number of interactions with the real DOM, making updates faster.
- **Better User Experience**:
  The quick rendering of updates ensures smooth interactions, even in complex applications.
- **Declarative UI**:
  Developers can focus on *what* the UI should look like rather than worrying about *how* to update the DOM.

**2. Component-Based Architecture**

React applications are built using **components**, which are reusable, self-contained units of the UI. Each component is responsible for rendering a small, specific part of the interface.

**Key Features of Components:**

1. **Reusability**:
   Components can be reused across the application, reducing code duplication and making development faster.
2. **Encapsulation**:
   Each component has its own logic, state, and styles, making it easy to manage and maintain.
3. **Hierarchy**:
   Components can be nested to create a hierarchy, forming a tree-like structure where parent components pass data to child components.
4. **Declarative Syntax**:
   Components use JSX (JavaScript XML) to describe how the UI should look, blending HTML-like syntax with JavaScript logic.

Question : 3

What are the advantages of using React.js in web development?

→ React.js offers several advantages that make it a popular choice for web development, particularly for building dynamic and responsive user interfaces.

# JSX (JavaScript XML)

Queston : 1

What is JSX in React.js? Why is it used?

→ **HTML-like Syntax in JavaScript**: JSX combines the power of JavaScript with the readability of HTML, enabling developers to describe the UI structure within their JavaScript code.

→ **Preprocessed Code**:
Browsers do not understand JSX directly. A **transpiler** like Babel converts JSX into regular JavaScript function calls (using React.createElement) before it is executed by the browser.

→ **JSX (JavaScript XML)** is a syntax extension for JavaScript used in React.js. It allows developers to write HTML-like code directly within JavaScript, making it easier to create and manage the structure of React components.

Question : 2

How is JSX different from regular JavaScript? Can you write JavaScript insideJSX?

→ **JSX (JavaScript XML)** differs from regular JavaScript in its syntax and purpose but is closely tied to JavaScript, as it compiles into standard JavaScript code. Here's a detailed comparison and an explanation of how JavaScript can be used within JSX.

**HTML-Like Syntax :**

→ JSX allows you to write HTML-like tags within JavaScript code.

→ Regular JavaScript cannot include such tags natively. JSX makes it easier to describe the structure of user interfaces.

Question : 3

Discuss the importance of using curly braces {}in JSX expressions.

→ In **JSX (JavaScript XML)**, curly braces {} play a crucial role in embedding **JavaScript expressions** within the JSX code. These curly braces allow developers to mix **JavaScript logic** with **HTML-like syntax**, enabling dynamic rendering of content in React components.

# Components (Functional & Class Components)

Question : 1

What are components in React? Explain the difference between functionalcomponents and class components.

→ In React, **components** are the building blocks of a React application. They are reusable, self-contained pieces of code that represent a part of the user interface (UI). Components allow developers to split the UI into smaller, independent, and manageable parts, making it easier to build, maintain, and scale applications.

→ In React, **functional components** and **class components** are the two main types of components, and while they serve the same purpose—defining the UI and behavior of parts of an application—they differ in how they are written, their features, and how they handle state and lifecycle methods.

**1. Functional Components**
**Definition:**

- Functional components are JavaScript functions that accept **props** as input and return JSX to define the UI.
- They are simpler and are primarily used for rendering static or presentational content.

**2. Class Components**
**Definition:**

- Class components are ES6 classes that extend React.Component and include a render() method that returns JSX.
- They were traditionally used for stateful and complex logic but are now less common due to the rise of functional components with hooks.

Question : 3

How do you pass data to a component using props?

→ In React, **props** (short for "properties") are used to pass data from a parent component to a child component. Props are like arguments for a function—they allow components to be **reusable** and **dynamic** by customizing their behavior or appearance based on the passed data.

**How to Pass Data Using Props**

1. **Define the Props in the Parent Component**
   When rendering a child component, you can pass data as attributes in the JSX tag of the child component.
2. **Access Props in the Child Component**
   a. In **functional components**, props are accessed as the argument of the component function.
   b. In **class components**, props are accessed using this.props.

Question : 3

What is the role of render()in class components?

→ In React, the render() method is a **required lifecycle method** in class components. Its primary role is to describe what the UI of the component should

look like by returning JSX, which React then converts into DOM elements to be displayed in the browser.

### Key Roles of `render()` in Class Components

→ **Defining the UI**:

- The `render()` method returns the JSX that specifies the component's structure and appearance.
- This JSX is then rendered into the DOM by React.