**1.What is React JS?**

React JS is a JavaScript library for building front end applications or UI. It allows to create reusable components. Components are building blocks of React app. It is developed by Facebook.

**2.Advantages of React JS:**

Reusable Component, Open Source, Efficient and Fast, Works in any browser, Large Community.

**3.What are the types of components?**

There are two types of components: **Class Component** and **Function Component**

**4.How to create a class component?**

To create a class component, we need to import React class and extend React.Components.

Import React from ‘react’;

Class Home extends React.Component{

render(){

return{

<h2>Hello World</h2>

}

}

}

Export default Home

**5.Why to use class components?**

Class components are stateful components. We can create life cycle methods, constructor and functions.

**6.What is functional component?**

It has no life cycle methods. Functional component is just a simple JavaScript function; it accepts the data in the form of props and returns the react element.

Component to make the class as a react component.

We need to write less code. It is stateless components.

No constructor, no render, no this, no need to extend React.Component & easy to write test cases. It’s just a simple JavaScript functions.

**7.How to create functional Component?**

import React from 'react';

function AboutUs(){

return (

<h3>React - Functional Component</h3>

)

}

export default AboutUs;

**8.Explain JSX in React.**

JSX stands for JavaScript XML. It makes feasibility to write HTML inside JavaScript code.

**With JSX:**

function AboutUs(){

    return (

        <h3>React - Functional Component</h3>

    )

}

**Without JSX:**

function AboutUs(){

    return (

        React.createElement(

            'h3',

            null,

            'React - Functional Component'

        )

    )

}

**9.Explain Props in React.**

Props stand for "Properties." They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes. It gives a way to pass data from one component to other components. It is similar to function arguments. We can pass any type of data in props like string, number, object etc. Declaration of prop in both class and functional component is different.

**10.How to use props in functional components?**

Passing value to the component from another component:

<About fullName="My name is Rakesh"/>

Accessing props values: using **prop**

import React from 'react';

function AboutUs(prop){

    return (

        <div>

            <h3>React - Functional Component</h3>

            <h3>{prop.fullName}</h3>

        </div>

    )

}

export default AboutUs;

**11.How to use props in class components?**

Passing value to the component from another component

<Home fullName="My name is Rakesh Swain"/>

Accessing the values: using **this.props :**

import React from 'react';

class Home extends React.Component{

    render(){

        return(

            <div>

            <h3>React - Class Component</h3>

            <h3>Name is: {this.props.fullName}</h3>

            </div>

        )

    }

}

export default Home;

**12.Explain state in React JS.**

State is a private internal object of a class. It works like a variable. Unlike props we can change the value of state. State can be used inside class component only. We can modify the value of state.

import React from 'react';

class Home extends React.Component{

    constructor(){

        super();

        this.state = {

            name: 'Nikun',

            email:'nikun@test.com'

        }

    }

    render(){

        return (

            <div>

            <h3>React - Class Component</h3>

            <h3>state: {this.state.email}</h3>

            </div>

        )

       }

}

export default Home;

**13.How can you change the values of a state.**

import React from 'react';

class Home extends React.Component{

    constructor(){

        super();

        this.state = {

            count:1

        }

    }

    updateState(){

        this.setState({

            count : this.state.count+1 or New Value

        })

    }

    render(){

        return(

            <div>

            <h3>click count: {this.state.count}</h3>

            <button onClick={()=>{this.updateState()}}>Click Count</button>

            </div>

**14. Event & Function with argument**

Class Component:

export default class FunClass extends React.Component{

     classFunction(arg){

     console.warn('Hi I am Class Function',arg);

    }

    render(){

        return(

            <div>

            <button onClick={()=>{this.classFunction(30)}}>Call Class</button>

            </div>

        )

    }

}

Functional Component:

export default function FunFun(){

     let functionalFunction = (args) =>{

        console.warn('Hi I am from Functional Component',args)

     }

    return(

        <div>

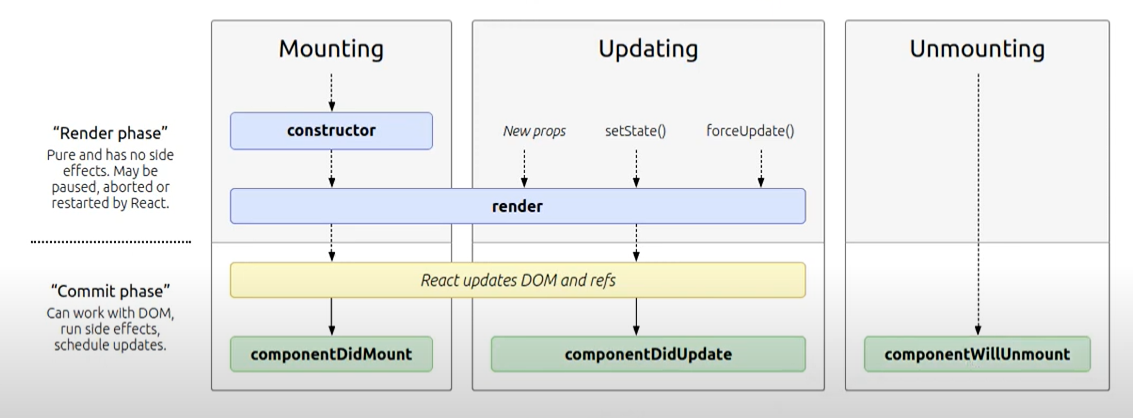
            <button onClick={()=>{functionalFunction(10)}} >Call</button>

        </div>

    )

}

**15. Life Cycle Method and Phases**



16.Component did mount life cycle method

For async process like API data bind we need to use it.

17.Component did update life cycle method

18.Compoenent did unmount life cycle method