**\*\*\* Differece b/w functional component and class component over function calling\*\*\***

**Functional component**

import React from 'react'

export default function Home()

{

    function get()

{

        alert('get function has called')

    }

    return(

        <div>

            <h1>this is functinal component</h1>

            <button onClick={get}>click me</button>

        </div>

    )

}

**Class component**

import React from 'react'

 export default class Home extends React.Component

 {

    get()

    {

        alert("get function called")

    }

    render()

    {

        return(

            <div>

                <h2>this is class component</h2>

                <button onClick={()=>this.get()}>Click me</button>

            </div>

        )

    }

 }

**\*\*\* state is working only with class component ? \*\*\***

**\*\*\* LifeCycle Method \*\*\***

constructor() =>render()=>componentDidMount()=>componentDidUpdate()=>componentWillUnmount()

**Note:-**

(1).LifeCycle Method me pahle constructor() than render() than componentDidMount() than componentDidUpdate() than componentWillUnmount() call hota hai.

(2) Agar componentDidMount() ke andar kuch update hota hai to uske baad ek baar fir render() call hota hai.

(3) componentDidUpdate() method me maximum 3 parameter pass kar sakte hai, pahla **prevProps**, dusra **prevState** aur tisra **snapshot**

jarurat ke according 1,2 ya 3 parameter use kar sakte hai koi mandatory nahi hai. jab koi state update hota hai to ye method automatically call ho jata hai.

(4) jab component destroy OR unmount OR remove ho jata hai tab componentWillUnmount() ye method call hota hai. Is method ke andar state update nahi karni chahiye kyoki component destroy hone ke baad render() method call/execute nahi hota hai.

(5) Lifecycle methods sirf class component me hi work karta hai.

**\*\*\* Note \*\*\***

1. Lifecycle methods sirf class component me hi work karta hai.
2. Hooks sirf functional component me hi work karta hai.
3. Functional component ke andar render() method use nahi hota hai.
4. Function ko ham do tarike se define kar sakte hai:

function get()

{

alert(“get function called”);

}

**OR**

const get=()=>

{

alert(“get function called”);

}

1. fetch and display API data:

Pahle API ka data fetch karke state variable me store karte hai, fir condition ke basis pe display karte hai, agar state variable me data hai to display hoga (listing ke through) other wise nahi hoga.

**\*\*\* Practice \*\*\***

**Class component:** Simple program, onClick button, Calling with single Parameter, Calling with Multiple parameter, Calling multiple time, state - creating and updating, Hide-Show property, Life cycle method, Form submission , Form submission with validation, Listing, fetch API data, Create portal, Ref, PureComponent (it is same as another component so find differance), Uncontrolled component(try to display form data by using Ref ), controlled component (display form data by using state), lazy loading (import component by lazy method, calling component with in Suspense tag with message )

**Functional component:** Simple program, onClick button, Calling with single Parameter, Calling with Multiple parameter, Calling multiple time, Hooks-creating and updating, Create portal, memo(sending dynamic data using hooks, watch another exp.),

**Note:**

1. netlify - free uploading react.js project
2. For using dummy API: https://reqres.in/ Exp: <https://reqres.in/api/user/2>
3. https://www.youtube.com/watch?v=5PfvoAj-SMk //CRUD Operation , creating api by using json
4. almabetter.com //for online course and placement