Example 1:

**Functional Component:**

import React from'react'

function Greet() {

return (

<h1>Welcome to React. this is the first functional component</h1>

);

}

export default Greet;

Arrow Function:

const Greet = () => <h1>Welcome to React. this is the first functional component</h1>

Any Name we can give while importing

Named Export

import {Greet} from './Greet'

export const Greet = () => <h1>Welcome to React. this is the first functional component</h1>

**Class Component**

import React, {Component} from 'react'

class Welcome extends Component{

render(){

return <h1>Welcome to React. this is the first class component which can maintain the state</h1>

}

}

export default Welcome;

**Passing Props to functional component:**

const Greet = props => {

return (

<div>

<h1>

Hello, {props. daypart } {props.batch }

</h1>

//{props.children}

</div>

)

}

export default Greet

App.js

<Greet daypart="Good Morning" batch="Batch" >

//<p>This is children props</p>

</Greet>

<Greet daypart="Good Afternoon" batch="Batch" >

//<button>Action</button>

</Greet>

<Greet daypart="Good Evening" batch="Batch" />

**Passing Props to class component:**

class Welcome extends Component {

render() {

return <h1>Welcome {this.props.batch} <h1>

// return <h1>Hello, {this.props. daypart } {this.props. batch} <h1>

}

}

export default Welcome

App.js

<Welcome batch ="batch" daypart ="Good Morning" />

<Welcome batch ="batch" daypart ="Good Afternoon" />

<Welcome batch ="batch" daypart ="Good Evening" />

Note : Props are immutable

**Difference between Functional and Class Components:**

|  |  |
| --- | --- |
| **Functional Components** | **Class Components** |
| A functional component is just a plain JavaScript function that accepts props as an argument and returns a React element. | A class component requires you to extend from React. Component and create a render function which returns a React element. |
| There is no render method used in functional components. | It must have the render() method returning HTML |
| Also known as Stateless components as they simply accept data and display them in some form that is they are mainly responsible for rendering UI. In React 16.8 hooks are added to maintain the state in functional component. | Also known as Stateful components because they implement logic and state. |
| React lifecycle methods cannot be used in functional components. | React lifecycle methods can be used inside class components. |
| There is no render method used in functional components | You must write render method. |
| These components have less code can be easy to debug, test. |  |

**JSX(Java Script XML)**

JSX stands for JavaScript XML, extension to Java Script.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.