function Test

{

  return(

    <h1> This is a test </h1>

  );

}

export default Test

in index.js file

import Test

from './test';

ReactDOM.render(

<Test />,

  document.getElementById('root')

);

var name="Ajay";

function Test()

{

  return(

    <h1> This is a test {name}  </h1>

  );

}

export default Test

function Test(props)

{

  return(

    <h1> This is a test {props.name}  </h1>

  );

}

export default Test

ReactDOM.render(

<Test name="Deepak"/>,

  document.getElementById('root')

);

<https://www.freecodecamp.org/news/react-components-jsx-props-for-beginners/>

<https://reactjs.org/docs/components-and-props.html>

Can we use props in functional component?

Functional components do not have access to dedicated state variables like class-based components. **The only “state” that a functional component effectively has access to are the props passed to it from its parent component**.

What is component properties in React?

Function and Class Components  
  
This function is a valid React component because it accepts a single “props” (which stands for properties) object argument with data and returns **a React** element. We call such components “function components” because they are literally JavaScript functions.