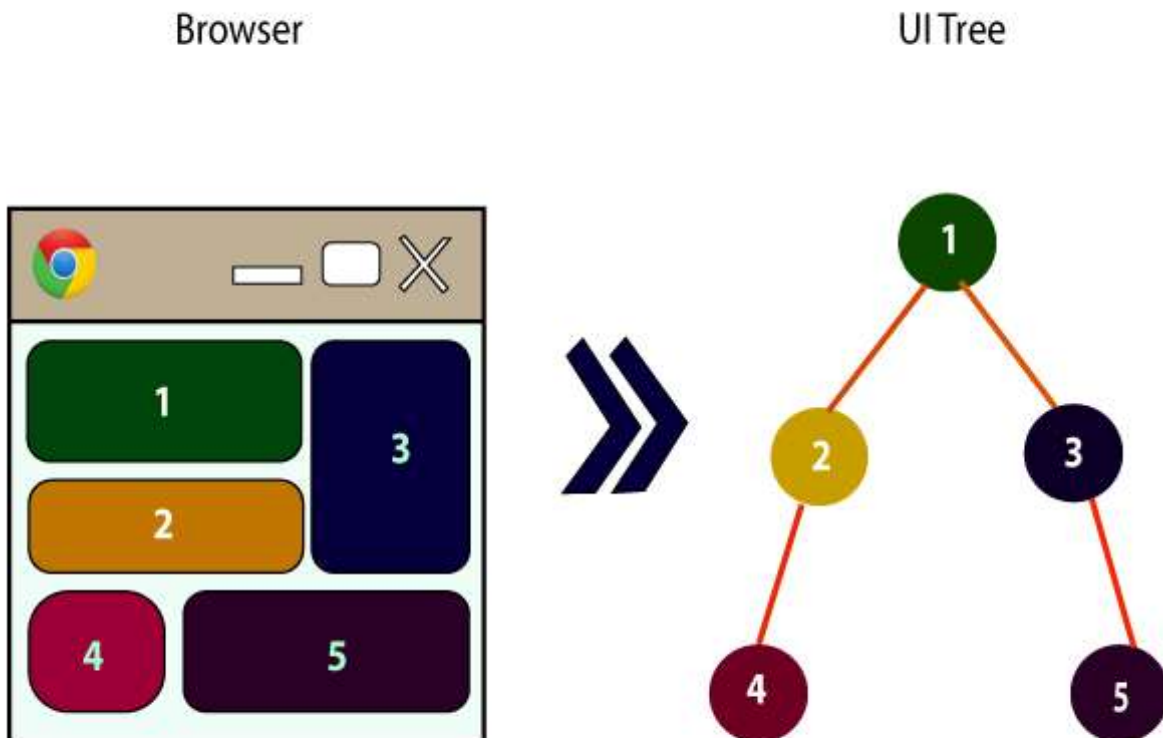


React Components

Earlier, the developers write more than thousands of lines of code for developing a single page application. These applications follow the traditional DOM structure, and making changes in them was a very challenging task. If any mistake found, it manually searches the entire application and update accordingly. The component-based approach was introduced to overcome an issue. In this approach, the entire application is divided into a small logical group of code, which is known as components.

A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

Every React component have their own structure, methods as well as APIs. They can be reusable as per your need. For better understanding, consider the entire UI as a tree. Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches.



In ReactJS, we have mainly two types of components. They are

1. Functional Components
2. Class Components

Functional Components

In React, function components are a way to write components that only contain a render method and don't have their own state. They are simply JavaScript functions that may or may not receive data as parameters. We can create a function that takes props(properties) as input and returns what should be rendered. A valid functional component can be shown in the below example.

```
function WelcomeMessage(props) {  
  return <h1>Welcome to the , {props.name}</h1>;  
}
```

The functional component is also known as a stateless component because they do not hold or manage state. It can be explained in the below example.

Example

```
import React, { Component } from 'react';  
class App extends React.Component {  
  render() {  
    return (  
      <div>  
        <First/>  
        <Second/>  
      </div>  
    );  
  }  
}  
  
class First extends React.Component {  
  render() {  
    return (  
      <div>  
        <h1>JavaTpoint</h1>  
      </div>  
    );  
  }  
}
```

```
class Second extends React.Component {
```

```
  render() {
```

```
    return (
```

```
      <div>
```

```
        <h2>www.javatpoint.com</h2>
```

```
        <p>This websites contains the great CS tutorial.</p>
```

```
      </div>
```

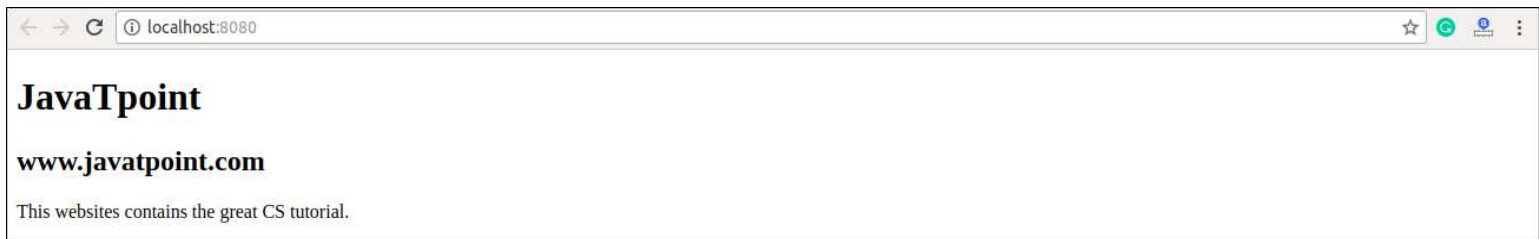
```
    );
```

```
  }
```

```
}
```

```
export default App;
```

Output:



Class Components

Class components are more complex than functional components. It requires you to extend from React.Component and create a render function which returns a React element. You can pass data from one class to other class components. You can create a class by defining a class that extends Component and has a render function. Valid class component is shown in the below example.

```
class MyComponent extends React.Component {
```

```
  render() {
```

```
    return (
```

```
      <div>This is main component.</div>
```

```
    );
```

```
  }
```

```
}
```

The class component is also known as a stateful component because they can hold or manage local state. It can be explained in the below example.

Example

In this example, we are creating the list of unordered elements, where we will dynamically insert StudentName for every object from the data array. Here, we are using ES6 arrow syntax (=>) which looks much cleaner than the old JavaScript syntax. It helps us to create our elements with fewer lines of code. It is especially useful when we need to create a list with a lot of items.

```
import React, { Component } from 'react';
class App extends React.Component {
  constructor() {
    super();
    this.state = {
      data:
      [
        {
          "name":"Abhishek"
        },
        {
          "name":"Saharsh"
        },
        {
          "name":"Ajay"
        }
      ]
    }
  }
  render() {
    return (
      <div>
        <StudentName/>
        <ul>
          {this.state.data.map((item) => <List data = {item} />)}
        </ul>
      </div>
    );
  }
}
```

```

}

class StudentName extends React.Component {
  render() {
    return (
      <div>
        <h1>Student Name Detail</h1>
      </div>
    );
  }
}

class List extends React.Component {
  render() {
    return (
      <ul>
        <li>{this.props.data.name}</li>
      </ul>
    );
  }
}

export default App;

```

Output:



← Prev

Next →

 Youtube For Videos Join Our Youtube Channel: [Join Now](#)