SNS COLLEGE OF ENGINEERING

An Autonomous Institution

Coimbatore-107

19TS601-FULL STACK DEVELOPMENT

UNIT-2

REACT

3. Components-Function components

Components

- Components are like functions that return HTML elements.
- Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.
- Components come in two types,
  - Class components and Function components

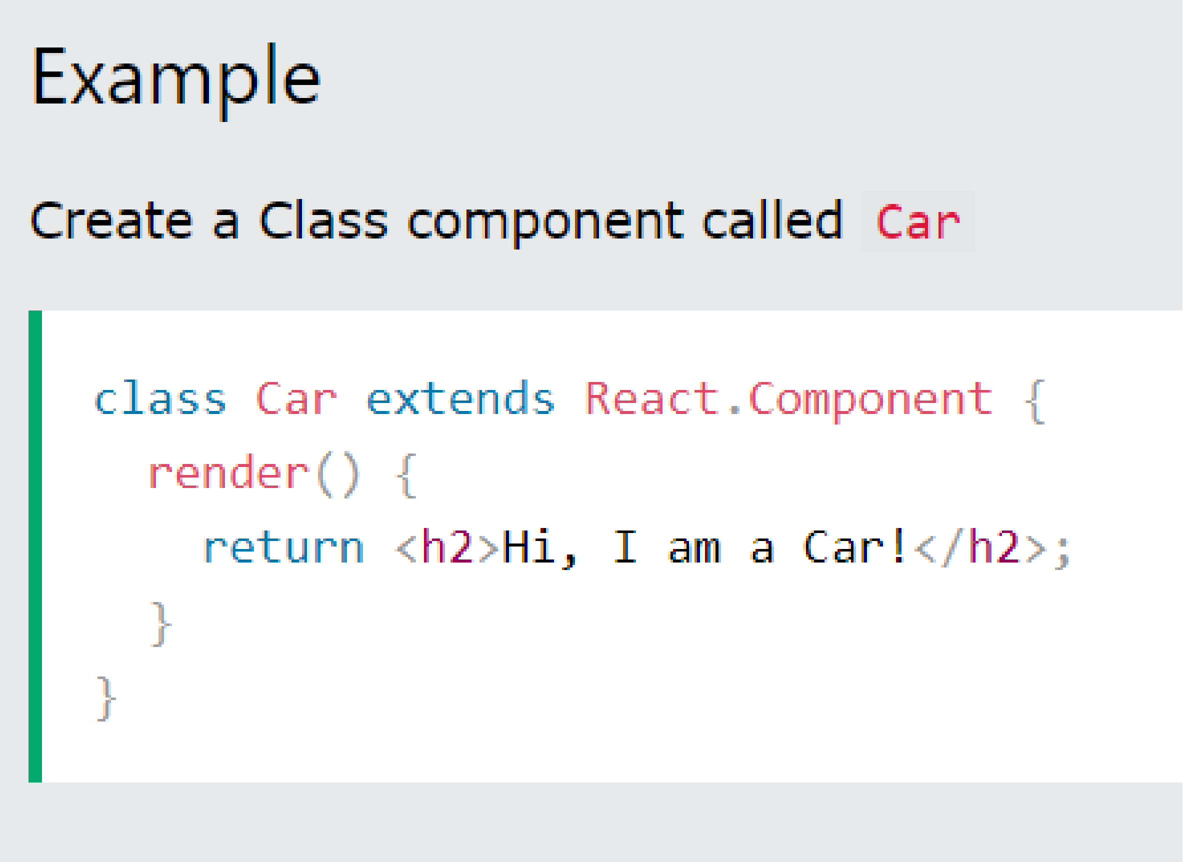
Create Your First Component

- When creating a React component, the component's name MUST start with an upper case letter.

Class Component

- A class component must include the extends React.Component statement.
- This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions.

- The component also requires a render() method, this method returns HTML.



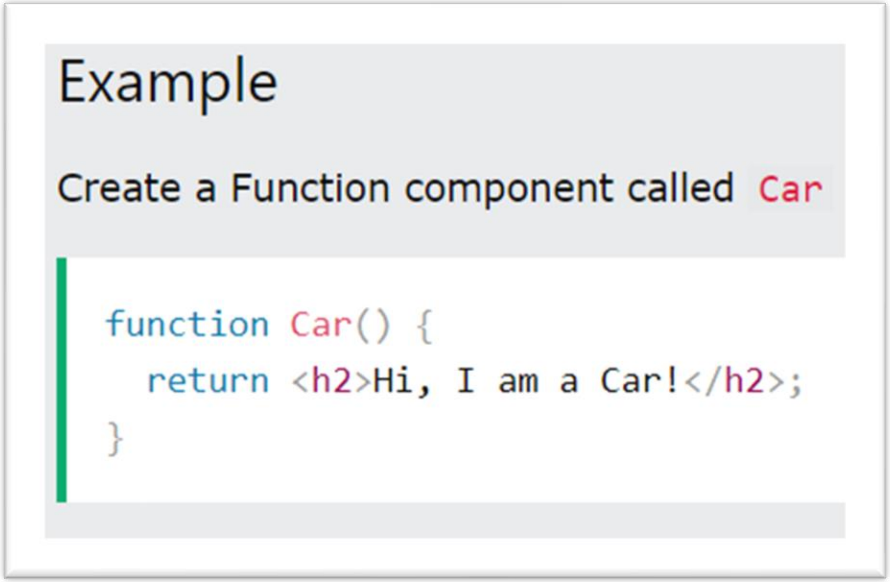
Example

Create a Class component called `Car`

```
class Car extends React.Component {
  render() {
    return <h2>Hi, I am a Car!</h2>;
  }
}
```

Function Component

- A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand.



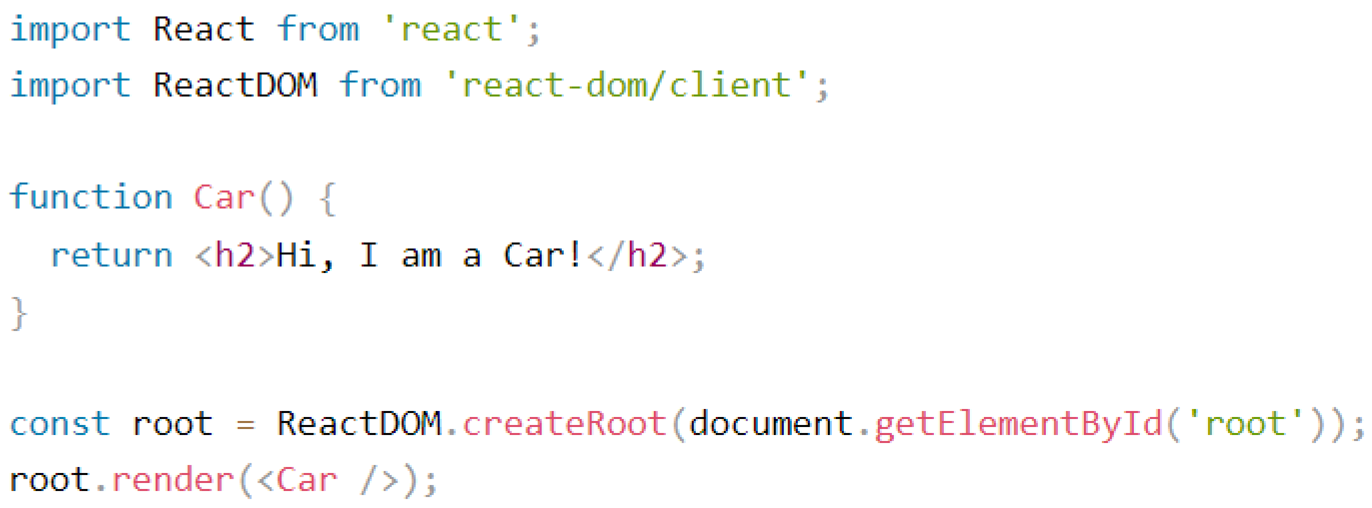
Example

Create a Function component called `Car`

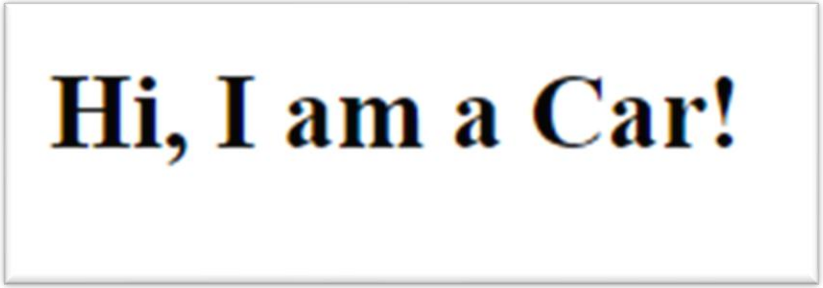
```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}
```

Rendering a Component

- Now your React application has a component called Car, which returns an <h2> element.
- To use this component in your application, use similar syntax as normal HTML: <Car />



```
import React from 'react';
import ReactDOM from 'react-dom/client';

function Car() {
  return <h2>Hi, I am a Car!</h2>;
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```

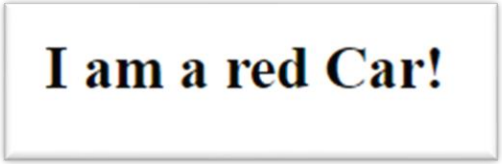


Hi, I am a Car!

Props

- Components can be passed as props, which stands for properties.
- Props are like function arguments, and you send them into the component as attributes.

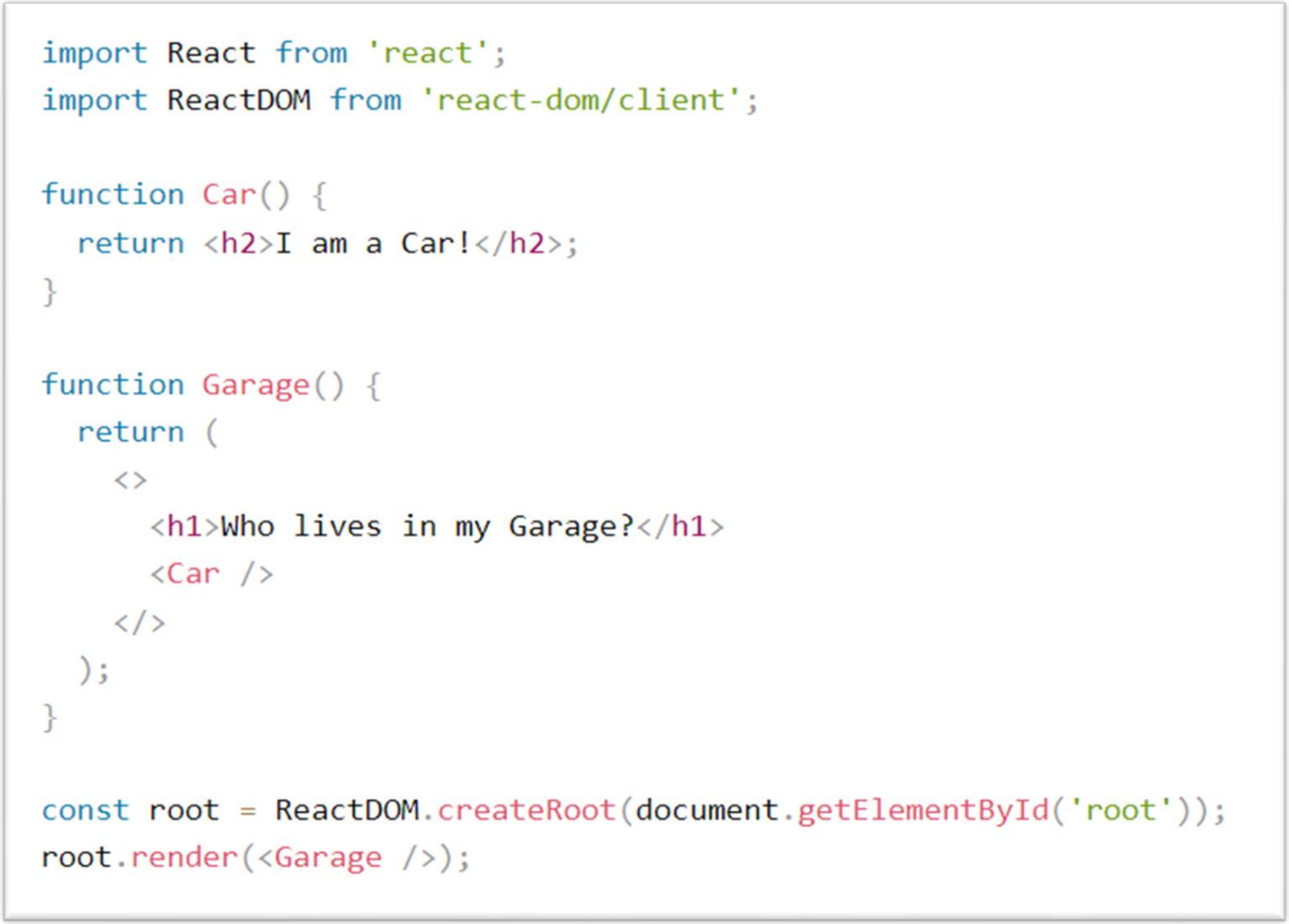
```
import React from 'react';
import ReactDOM from 'react-dom/client';

function Car(props) {
  return <h2>I am a {props.color} Car!</h2>;
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car color="red"/>);
```



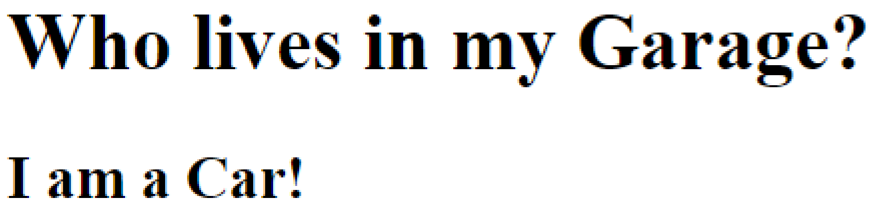
I am a red Car!

Components in Components

- Component inside another component



```
import React from 'react';
import ReactDOM from 'react-dom/client';

function Car() {
  return <h2>I am a Car!</h2>;
}

function Garage() {
  return (
    <>
      <h1>Who lives in my Garage?</h1>
      <Car />
    </>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Garage />);
```



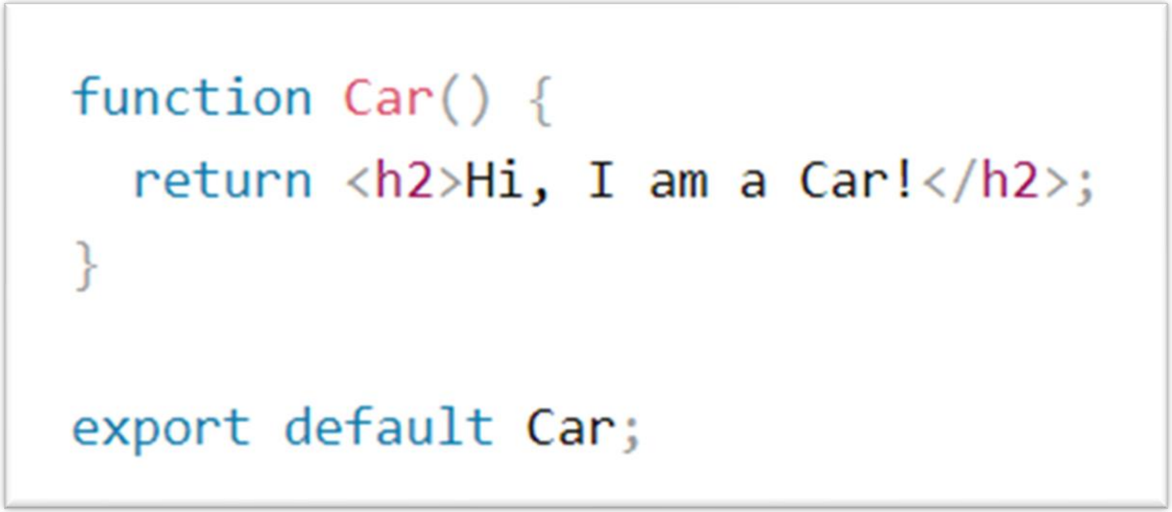
Who lives in my Garage?

I am a Car!

Components in Files (import & export)

- React is all about re-using code, and it is recommended to split your components into separate files.
- To do that, create a new file with a .js file extension and put the code inside it:
- The filename must start with an uppercase character.

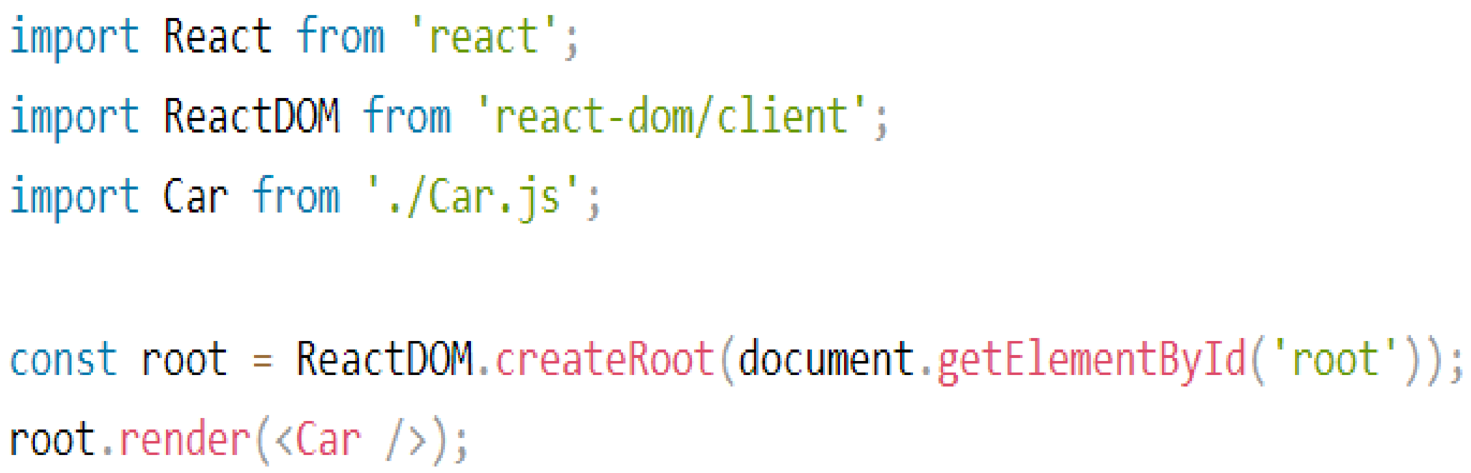
Car.js



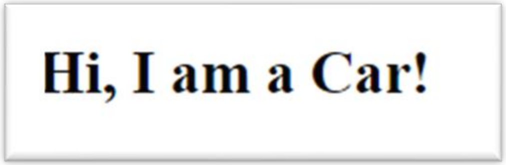
```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}

export default Car;
```

- To be able to use the Car component, you have to import the file in your application.

- import the "Car.js" file in the application, and we can use the Car component.



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import Car from './Car.js';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```



Hi, I am a Car!

Thank You