# the Master Course

{CODENATION}

# React Recap

{CODENATION}

{CN}®

React.js

## What is a component?

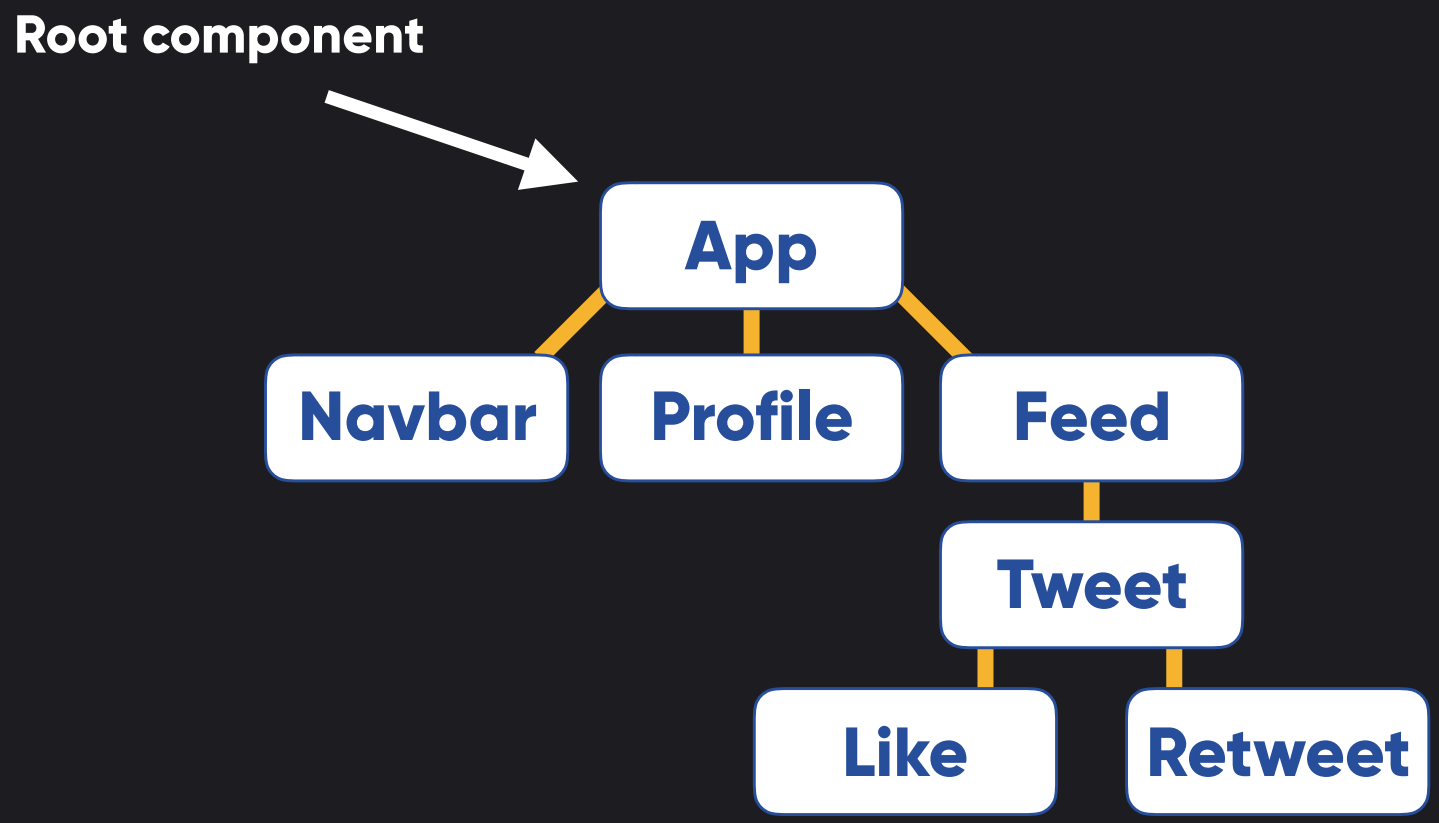
In simple terms, it's either a javascript function or class which returns a piece of the user interface.

React.js

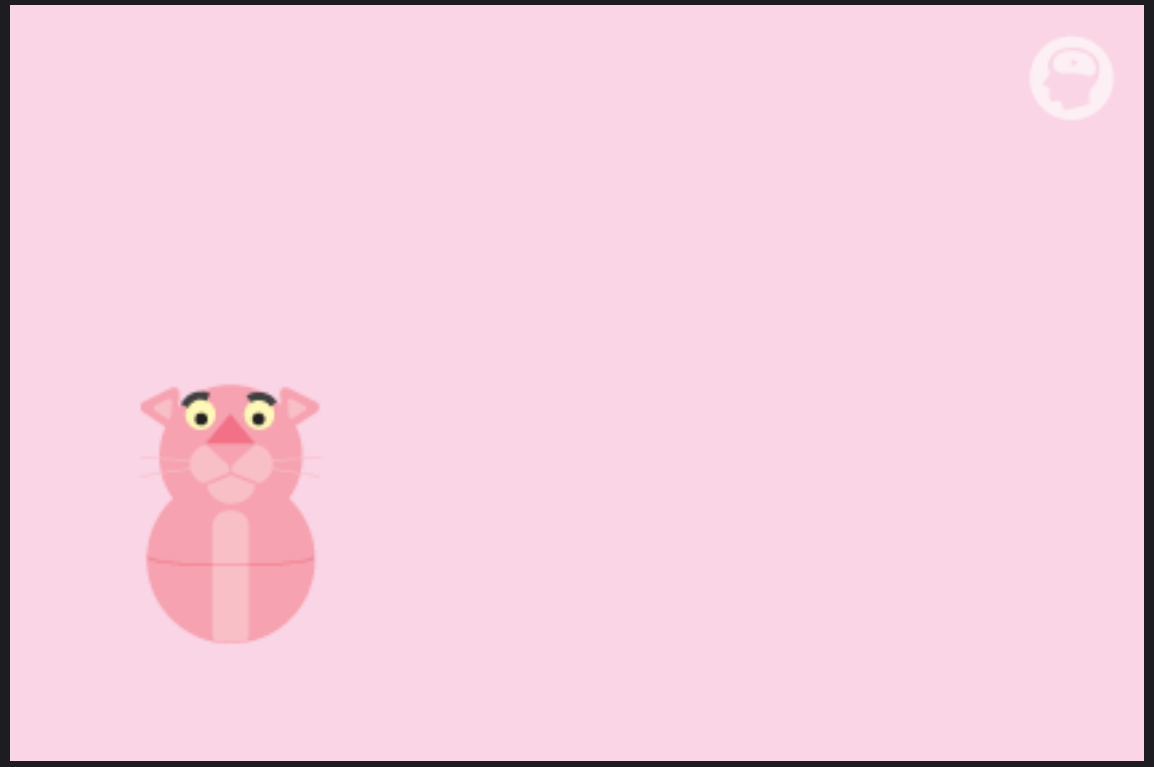
**Remember we can build our components in isolation and then put them all together.**

Our components form a tree structure or hierarchy, with one main (root) component.

# Component tree



# Think of it like this:



```
//functional component

const Person = () => {
  return (
      <div>
        <h1>I'm a functional component</h1>
      </div>
  )
}
```

```
//functional component

const Person = () => {
  return (
      <div>
        <h1>I'm a functional component</h1>
      </div>
  )
}
```

We use capital letters when naming our components.

{CODENATION}

```
//functional component

const Person = () => {
  return (
      <div>
        <h1>I'm a functional component</h1>
      </div>
  )
}
```

When returning JSX, there must be ONE parent element. In this case it's a div element. (Russian dolls!)

React.js

```
ReactDOM.render(<App/>, document.getElementById('root'))
```

**This is the main component that will be rendered.**

```
//class component

class App extends React.Component {
  render(){
    return(
        <div>
          <h1>I'm a class component</h1>
        </div>
    )
  }
}
```

```jsx
const Person = () => {
  return (
      <div>
        <h1>I'm a functional component</h1>
      </div>
  )
}

class App extends React.Component {
  render(){
    return(
      <div>
        <h1>I'm a class component</h1>
        <Person />
      </div>
    )
  }
}

ReactDOM.render(<App />, document.getElementById('root'))
```

```
const Person = () => {
  return (
      <div>
        <h1>I'm a functional component</h1>
      </div>
  )
}

class App extends React.Component {
  render(){
    return(
        <div>
          <h1>I'm a class component</h1>
          <Person />
        </div>
    )
  }
}

ReactDOM.render(<App />, document.getElementById('root'))
```

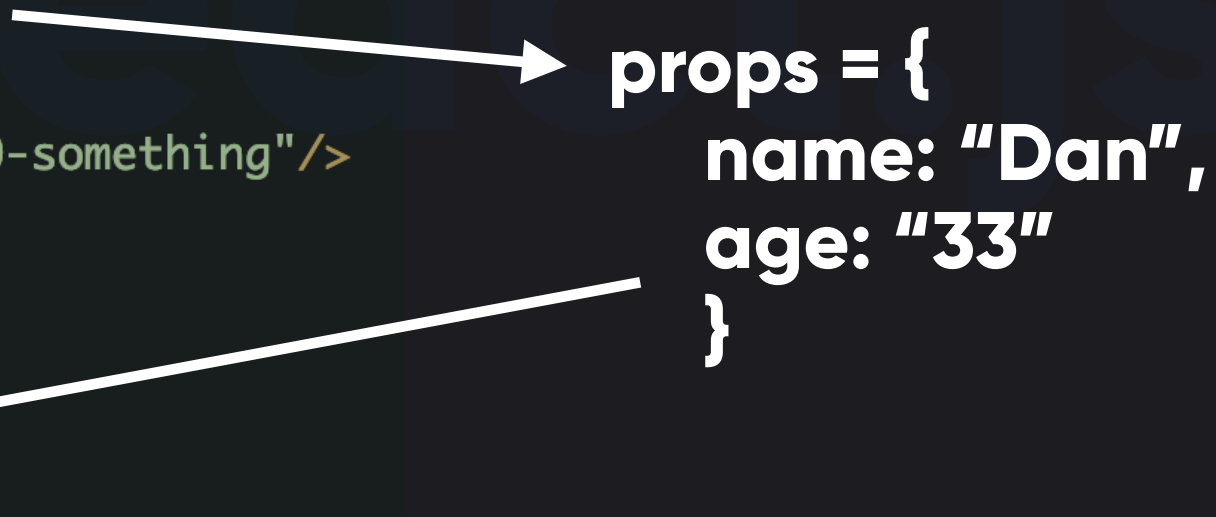
**Custom HTML elements**

React.js

**Task: Render a functional component 3 times inside a root class component.**

```
class App extends React.Component {
  render(){
    return(
        <div>
          <Person name="Dan" age = "33"/>
          <Person name ="Ben" age = "21"/>
          <Person name = "Stuar" age = "30-something"/>
        </div>
    )
  }
}

const Person = (props) => {
  return (
      <div>
        <h1>My name is something</h1>
      </div>
  )
}

ReactDOM.render(<App />, document.getElementById('root'))
```

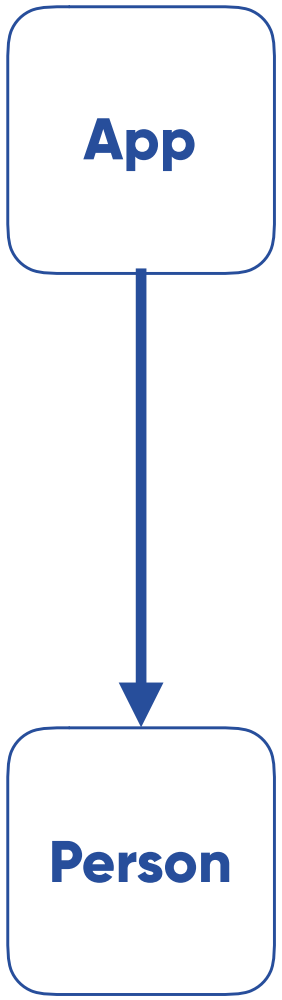
```
class App extends React.Component {
  render(){
    return(
        <div>
          <Person name="Dan" age = "33"/>
          <Person name ="Ben" age = "21"/>
          <Person name = "Stuart" age = "30-something"/>
        </div>
    )
  }
}

const Person = (props) => {
  return (
      <div>
        <h1>My name is {props.name}</h1>
      </div>
  )
}

ReactDOM.render(<App />, document.getElementById('root'))
```



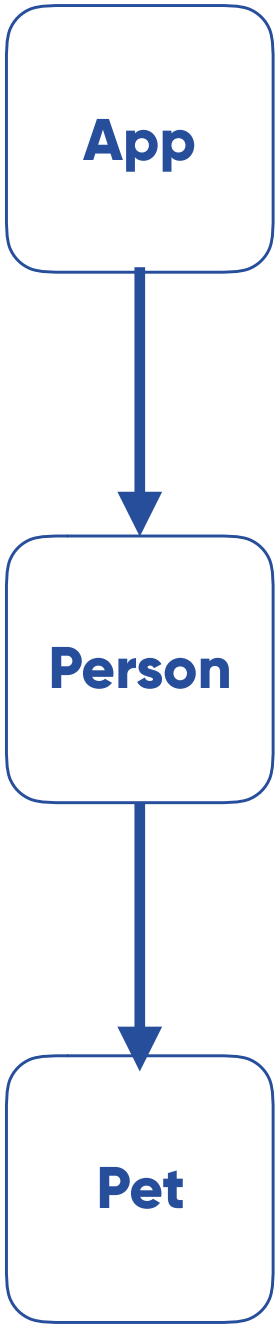
React.js

**Passing props is how we pass data down the hierarchy of components.**

{CODENATION}




**Data flows down the component tree through props**




## Pet's name property



**Data flows down the component hierarchy, and we can keep passing props along.**

```
        return(
          <div>
            <Person name="Dan" age = "33" pet = "Polly"/>
            <Person name ="Ben" age = "21" pet = "john"/>
            <Person name = "Stuart" age = "30-something" pet = "sam"/>
          </div>
      )
    }
}

const Person = (props) => {
  return (
      <div>
        <h1>My name is {props.name}</h1>
        <Pet petsName = {props.pet} />
      </div>
  )
}

const Pet = (props) => {
  return (
      <div>
        <h6>My pet's name is {props.petsName}</h6>
      </div>
  )
```

```
        return(
            <div>
              <Person name="Dan" age = "33" pet = "Polly"/>
              <Person name ="Ben" age = "21" pet = "john"/>
              <Person name = "Stuart" age = "30-something" pet = "sam"/>
            </div>
        )
    }
}

const Person = (props) => {
    return (
        <div>
          <h1>My name is {props.name}</h1>
          <Pet petsName = {props.pet} />
        </div>
    )
}

const Pet = (props) => {
    return (
        <div>
          <h6>My pet's name is {props.petsName}</h6>
        </div>
    )
```

React.js

**Task**: Pass a prop called ‘favouriteDrink’ from a class component to a functional component and render it on screen

{CN}®

React.js

**Now render another functional component within the original functional component.**

React.js

**Task**: Pass a prop called 'extras' from our **class component** to our **1st functional component** and then to our **new** functional component and render it on screen

{CN}®

React.js

Here's what you should've got:

# Task: Create a card view that displays the data using 4 components.

```
ben: {
    name: 'Ben',
    age: 21,
    hobbies: ['Cycling', 'Hockey', 'F1']
}
```

{CODENATION}

React.js

Here's what you should've got:

# Task: Now I want 3 of them. With different data in each.

React.js

Here's what you should've got: