# React State

React Tutorials
By:
Homeira Haddad

# What's the React State

State is the most complex thing in React, and it's something both beginners and experienced developers struggle to understand. So in this tutorial, we'll explore all the basics of state in React.

React components has a built-in state object.

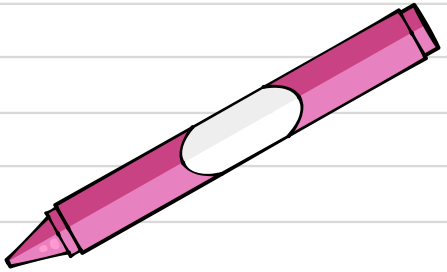
The state object is where you store property values that belong to the component.

When the state object changes, the component re-renders.

State allows us to manage changing data in an application. It's defined as an object where we define key-value pairs specifying various data we want to track in the application.

# Creating the state Object

## React useState Hook



# import useState

useState is a React Hook that lets you add a state variable to your component.

The React useState Hook allows us to track state in a function component.

State generally refers to data or properties that need to be tracking in an application.

To use the useState Hook, we first need to import it into our component.

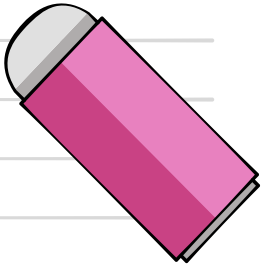
At the top of your component, import the useState Hook.

```
import React, {useState} from "react";
```

Notice that we are destructuring useState from react as it is a named export.

# Anatomy of the useState() Hook

Hooks are a React feature which allow you to “hook” functionality into functional components. As functions are pure and don’t have instances, capabilities which were originally implemented as React.Component class methods can’t be used directly. Hooks let you add these features to components without having to convert to classes.



The useState() hook sets up an individual state property. It returns an array containing two elements: the current state value, and a function you can call with a new value to update the state.

# The syntax of useState



```
const [state, setState] =
useState(initialState);
```

## state

The first value, state, is our current state.

## setState

The second value, setState, is the function that is used to update our state.

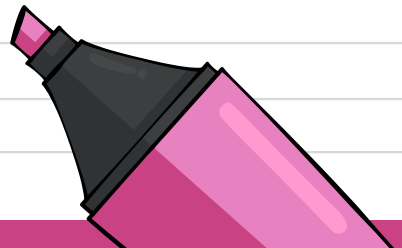
These names are variables that can be named anything you would like.

# initialState Is the default value of useState

You can use numbers, string, or just an empty string “”

# Recap

Let's now recap what we learned line by line and check our understanding.



```jsx
import React, { useState } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
       Click me
      </button>
    </div>
  );
}
```

Anonymous Function

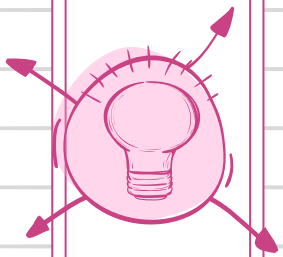
# Explain Example

## Line1

We import the useState Hook from React. It lets us keep local state in a function component.

## Line4

Inside the Example component, we declare a new state variable by calling the useState Hook. It returns a pair of values, to which we give names. We're calling our variable count because it holds the number of button clicks. We initialize it to zero by passing 0 as the only useState argument. The second returned item is itself a function. It lets us update the count so we'll name it setCount.



## Line9

When the user clicks, we call setCount with a new value. React will then re-render the Example component, passing the new count value to it.

# You can also use a nested function

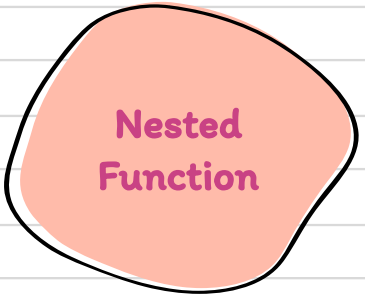
In Line 9:

```
<Button onClick = { onClickHandler } > Click me </button>
```

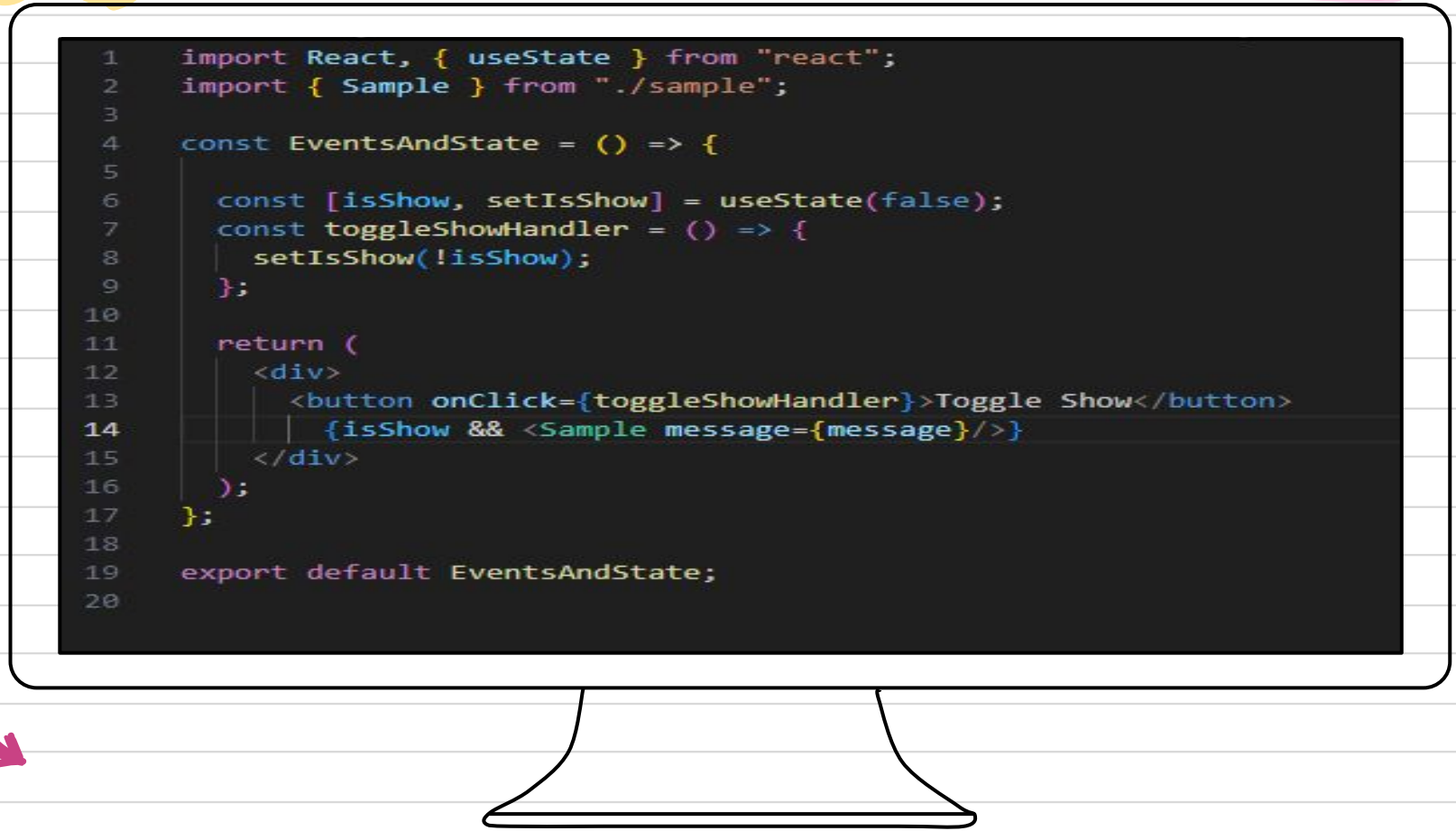
And before return, in Line 5:

```
Const onClickHandler = () => {
        setCount( count + 1 );
}
```





# Example: toggleShow



```jsx
import React, { useState } from "react";
import { Sample } from "./sample";

const EventsAndState = () => {

  const [isShow, setIsShow] = useState(false);
  const toggleShowHandler = () => {
    setIsShow(!isShow);
  };

  return (
    <div>
      <button onClick={toggleShowHandler}>Toggle Show</button>
      {isShow && <Sample message={message}/>}
    </div>
  );
};

export default EventsAndState;

```

# Explain Example

We have a component named Sample. We set a button to toggle show the content of this component by using useState hook. For this purpose it's better to use isShow for state value, and setIsShow for setState.

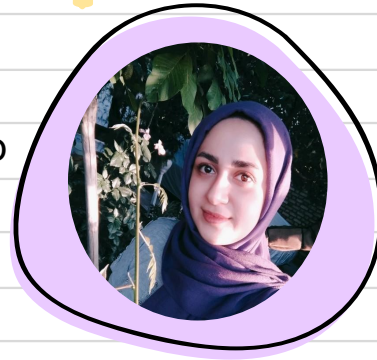
As you can see in the example, we considered false value for default. Then setState the !isShow means true value.
And this provides for our button, toggle action.
We can send props (like message in the example) to the component, then adjust it.

# Thanks!

Do you have any questions?

homeira2010@gmail.com

https://www.linkedin.com/in/homeira-haddad-792ab226b



CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon and infographics & images by Freepik

# Resources

- https://www.freecodecamp.org
- https://www.w3schools.com
- https://www.reactjs.org
- Sematec react.js class