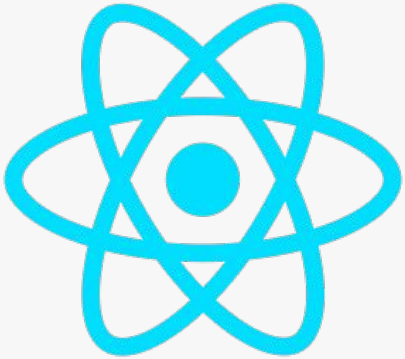
Newton School of Technology

The Ultimate React Course



@newtonschool

Lecture 15: State Management & Event Handling

-Narendra Kumar

JS

Prerequisites

- Setting up React dev environment
- Understanding of project structure in react
- React Functional Components
- Basic Understanding of JSX
- Launching React Application in browser using terminal

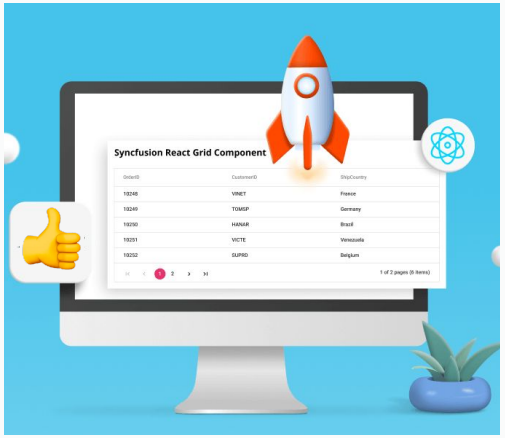
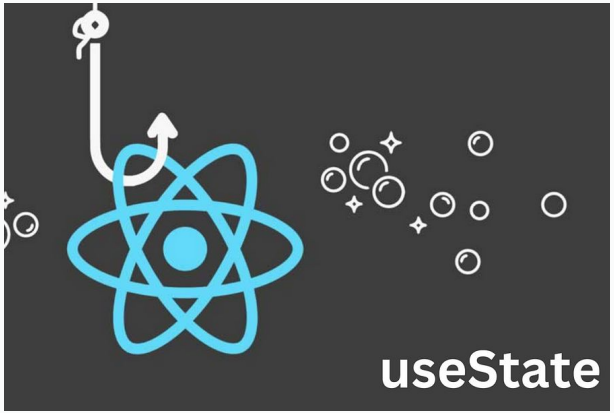
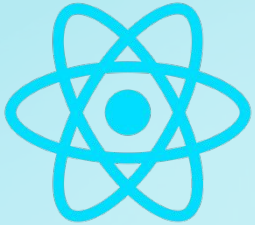


Table of Contents

- Quick Recap
- Problems in using javascript variables
- Understanding State with useState
  - What is State?
  - useState Hook syntax and usage
- Event Handling in React
- State and Events Together
- Workshop:To-do App
- Summary and Takeaways



Newton School of Technology



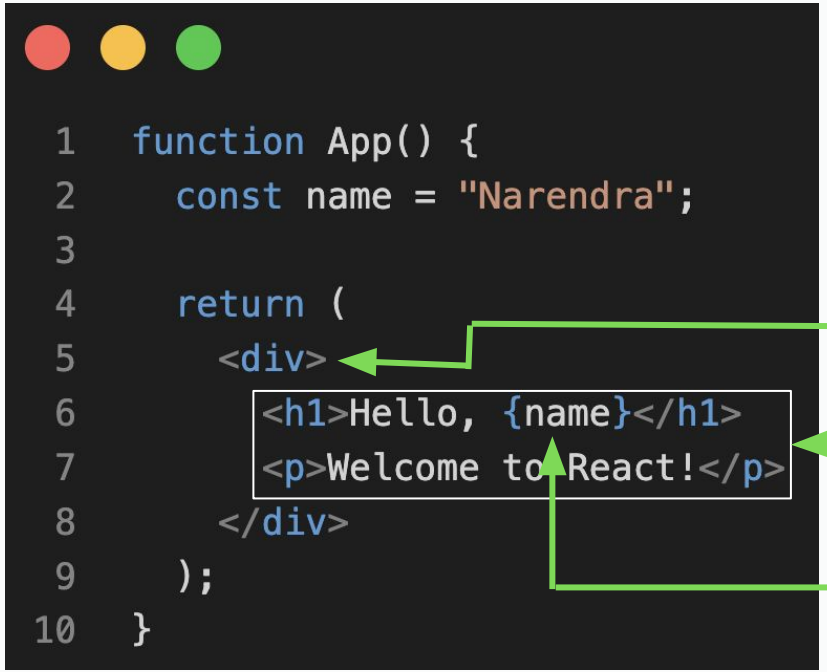
Quick Recap

A brief Overview

How to write JSX code?

Newton School of Technology

We have simple rules for writing JSX:-



```
1  function App() {
2    const name = "Narendra";
3
4    return (
5      <div>
6        <h1>Hello, {name}</h1>
7        <p>Welcome to React!</p>
8      </div>
9    );
10 }
```

Child elements inside root element.

Rules for writing JSX

JSX must have a **single root element**.

Use **curly braces {}** to embed Javascript.

JSX: Displaying Dynamic Content

Newton School of Technology

We can store values in variables and used them in JSX. When these values change, JSX updates automatically. 🚀

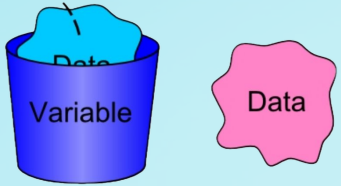
```jsx
function App() {
  const team_A_score = 45;
  const team_B_score = 38;

    return (
      <div>
        <h3>Team A Score: {team_A_score}</h3>
        <h3>Team B Score: {team_B_score}</h3>
      </div>
    );
}

export default App;
```

When these value changes

JSX updates automatically

However these updates are not guaranteed, so we use hooks.

Don't worry we will learn about hooks later.

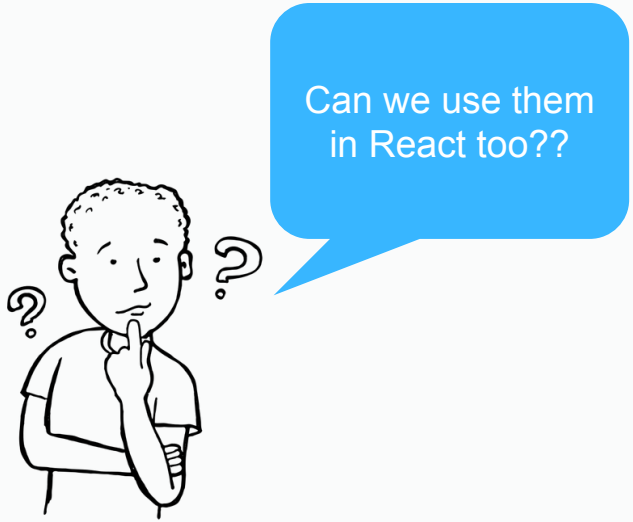


Problems in using Variables

Managing Data in React

In JavaScript, **we use variables to store data**. Can we use them in React to update the UI? Let's find out!

Javascript Variables to store values



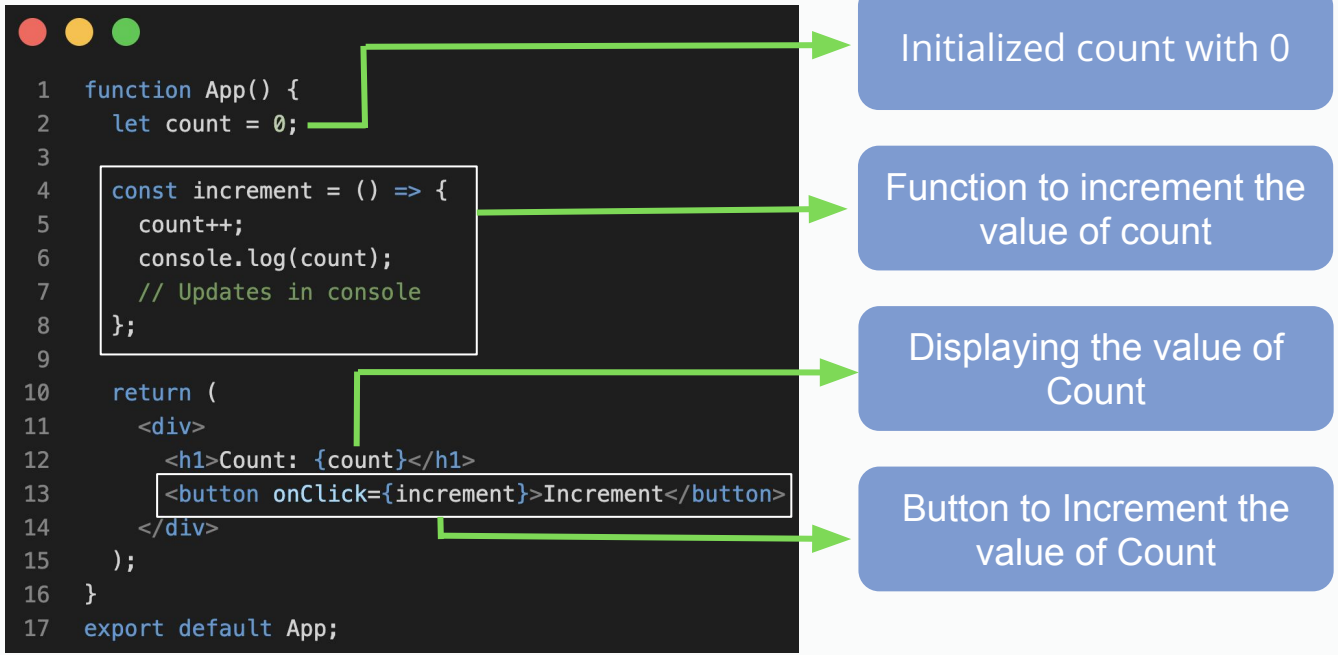
Managing Data in React

Here we have built an app where we can increment the value of count by clicking on *Increment* button.

```
1  function App() {
2    let count = 0;
3
4    const increment = () => {
5      count++;
6      console.log(count);
7      // Updates in console
8    };
9
10   return (
11     <div>
12       <h1>Count: {count}</h1>
13       <button onClick={increment}>Increment</button>
14     </div>
15   );
16 }
17 export default App;
```

Managing Data in React

Here we have built an app where we can increment the value of count by clicking on *Increment* button.

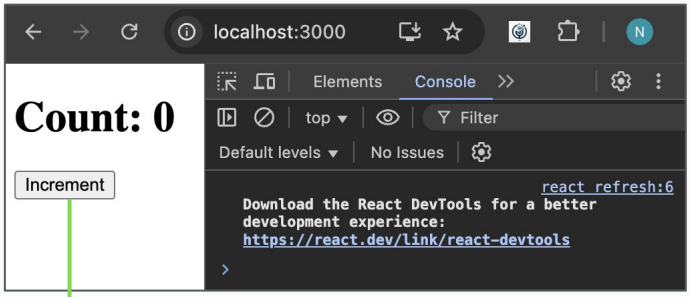


Managing Data in React

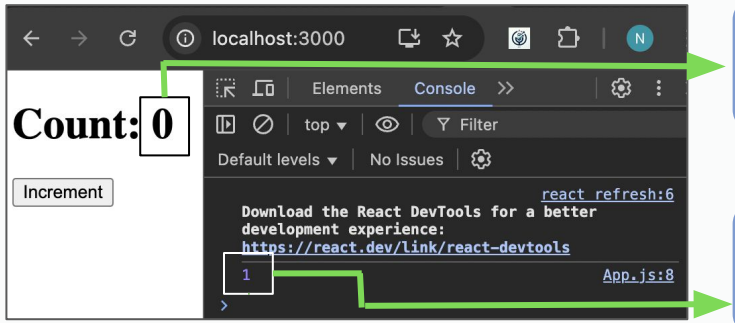
Let’s understand how it works:-

```jsx
function App() {
  let count = 0;

  const increment = () => {
    count++;
    console.log(count);
    // Updates in console
  };

  return (
    <div>
      <h1>Count: {count}</h1>
      <button onClick={increment}>Increment</button>
    </div>
  );
}
export default App;
```

Initial State



After first click in Increment

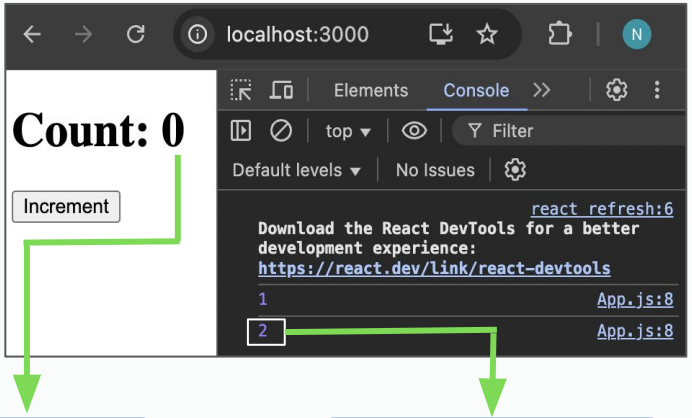


Managing Data in React

Newton School of Technology

Oh, let's try one more click:-

After second click in Increment



UI still remains unchanged

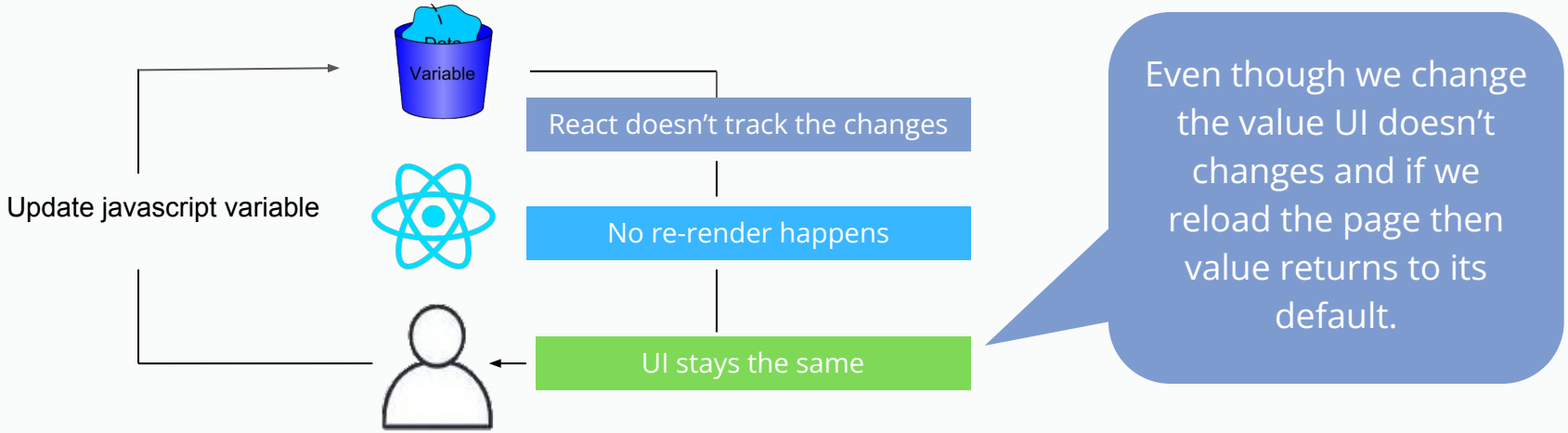
But

Value incremented from 1 to 2

Holly, molly, why is it not updating?

Why did it happen?

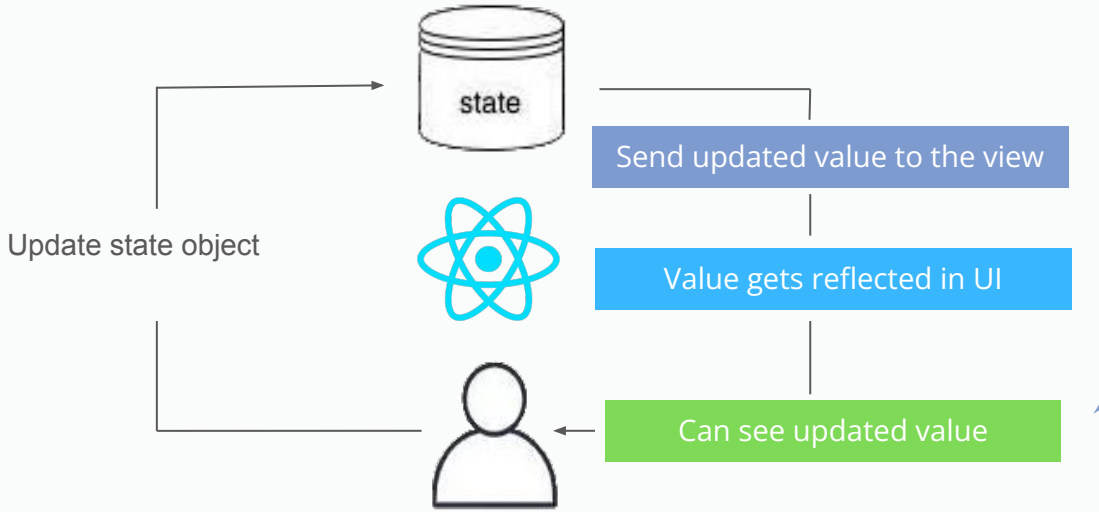
It happened because **React doesn't track normal variable changes**. React only re-renders when state or props change.



Why did it happen?

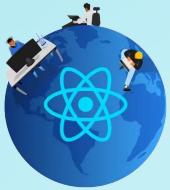
Newton School of Technology

React re-renders only when state or props change, ensuring the UI always reflects the latest state accurately. 🚀



Use state to re-render the UI when a value changes; avoid it for values that don't affect rendering.

Newton School of Technology



Understanding React State

Introduction - What is State?

Newton School of Technology

State is like a theater play 🎭—the stage (UI) must always reflect the latest scene.

Actors (state values) change their poses.

Actors in different dance poses(state values)

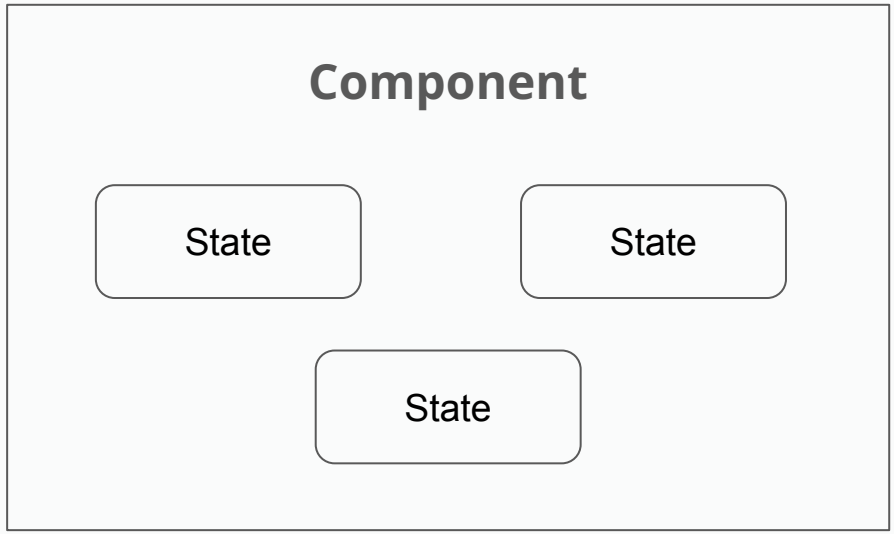
The director (React) ensures the audience sees the right scene by updating the stage when needed.

Stage (UI)

Introduction - What is State?

Newton School of Technology

Different states in react gets stored in the memory. It stores values that change over time and updates the UI accordingly.

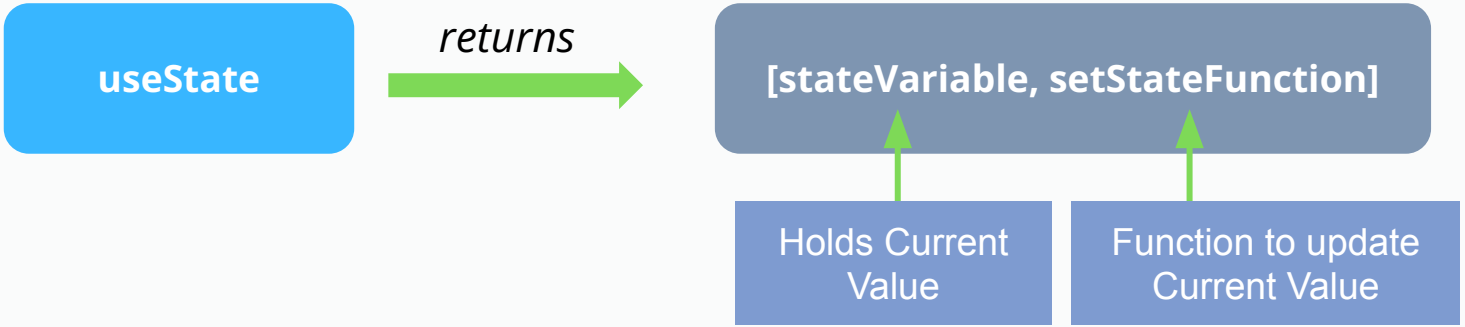


Just like each play has several actors having different poses, each component in React can maintain multiple states.

Understanding useState Hook

Newton School of Technology

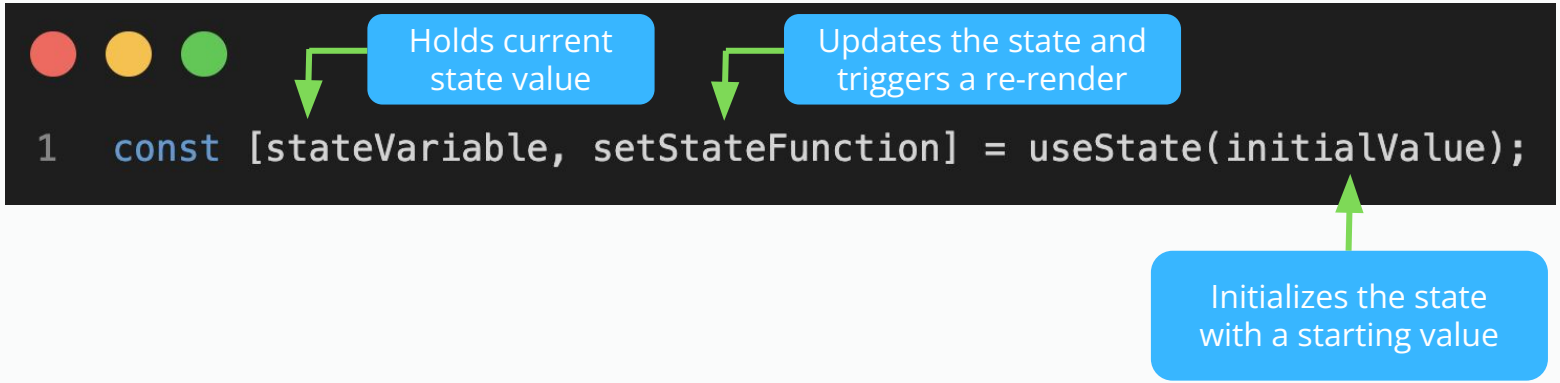
useState is a React Hook that lets us add state to functional components.It returns an array with two elements:



Declaring state with useState hook

Newton School of Technology

We use the useState hook to **initialize a React state variable**, allowing components to track and update values dynamically.



useState hook usage

Let’s create a counter where we can increment or decrement the counter by clicking buttons, for storing count variable, here we will use state variable.

```jsx
import { useState } from "react";

function App() {
  // Initializing state with useState
  const [count, setCount] = useState(0);

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button>Increment</button>
      <button>Decrement</button>
    </div>
  );
}

export default App;
```

State variable

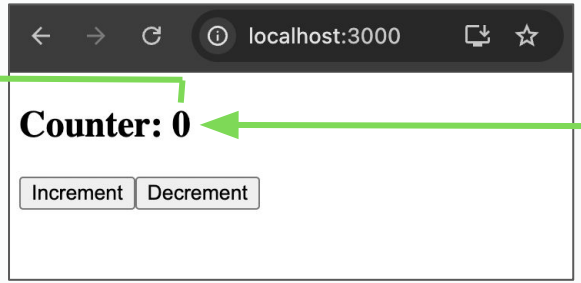
At first we must import the useState hook from the React library

We initialized the state variable with 0

useState hook usage

Let’s create a counter where we can increment or decrement the counter by clicking buttons, for storing count variable, here we will use state variable.

```jsx
import { useState } from "react";

function App() {
  // Initializing state with useState
  const [count, setCount] = useState(0);

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button>Increment</button>
      <button>Decrement</button>
    </div>
  );
}

export default App;
```



Accordingly value in UI should change

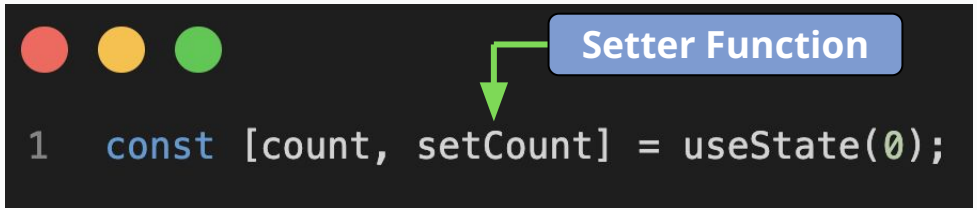
Now we need to add an event handler to handle events:-

1. User Clicks on Increment button
2. User Clicks on Decrement button

useState: setter function

Newton School of Technology

The **setter function** returned by useState is used to update the state value and trigger a re-render.




```
setCount(count + 1);
```

Increments the value of count variable state

```
setCount(count - 1);
```

Decrements the value of count variable state

useState: setter function

Newton School of Technology

Let’s write complete code for Counter app using useState and it’s setter function:-

```jsx
import { useState } from "react";

function App() {
  // Initializing state with useState
  const [count, setCount] = useState(0);

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button onClick={() => setCount(count + 1)}>
        Increment
      </button>
      <button onClick={() => setCount(count - 1)}>
        Decrement
      </button>
    </div>
  );
}

export default App;
```

Event Handlers

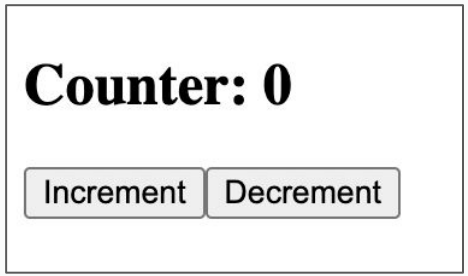
Incrementing and decrementing using setCount() setter function when user clicks the button.

Testing our app: Increment Button

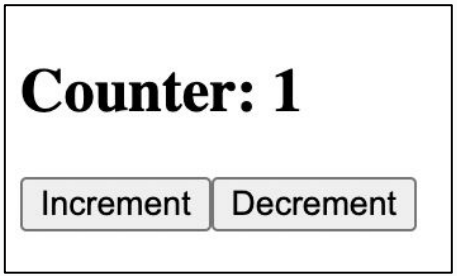
Newton School of Technology

Let’s try clicking on the ‘Increment’ button:

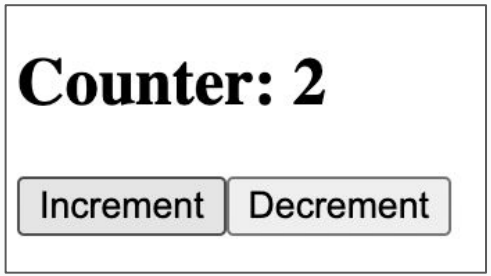
Before Click:-



2nd Click:-



3rd Click:-



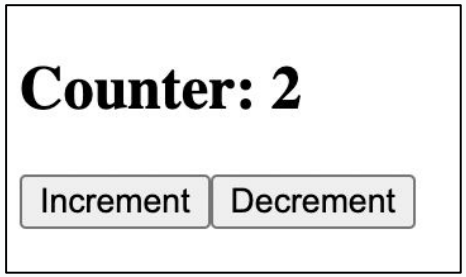
Increment button is working as expected

Testing our app: Decrement Button

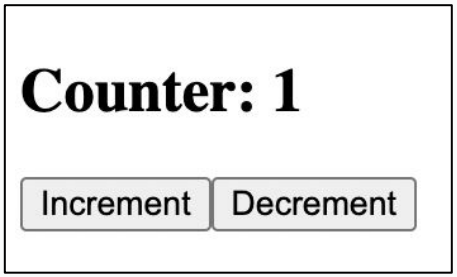
Newton School of Technology

Let’s try clicking on the ‘Decrement’ button:

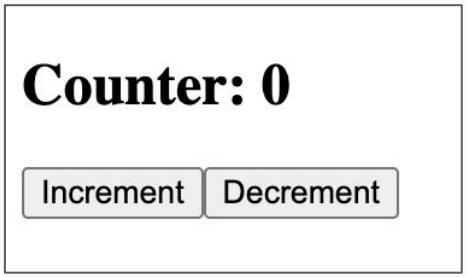
Before Click:-



2nd Click:-

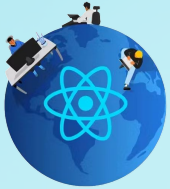


3rd Click:-



Decrement button too is working as expected

Newton School of Technology

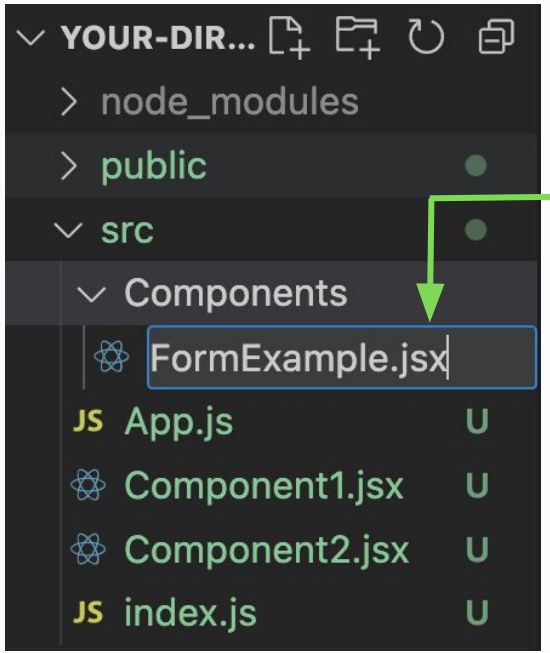


Event Handling in React

Forms: Creating the file

Newton School of Technology

Forms in React are different from regular HTML because React **manages form elements using state**. Let's create a *FormExample.jsx* component and include it in our App.js



Go to Explorer and create a folder Components and inside it create a component named FormExample.jsx

Include a form in that file.

```jsx
function FormExample() {

  return (
    <form>
      <label>
        Name:
        <input type="text" placeholder="Name" />
      </label>
      <button type="submit">Submit</button>
    </form>
  );
}

export default FormExample;
```

Forms: Creating and including in App.js

Newton School of Technology

Let’s add FormExample in our App.js as a component:

```jsx
function FormExample() {

  return (
    <form>
      <label>
        Name:
        <input type="text" placeholder="Name" />
      </label>
      <button type="submit">Submit</button>
    </form>
  );
}

export default FormExample;
```

Importing FormExample using a Relative path

```jsx
import FormExample from "./Components/FormExample";

function App() {
  return (
    <div>
      <FormExample />
    </div>
  );
}

export default App;
```

Including FormExample in App.js

Forms: Viewing in browser

Newton School of Technology

Let’s have a look at it in the browser:-

```
1  import FormExample from "./Components/FormExample";
2
3  function App() {
4    return (
5      <div>
6        <FormExample />
7      </div>
8    );
9  }
10
11 export default App;
```

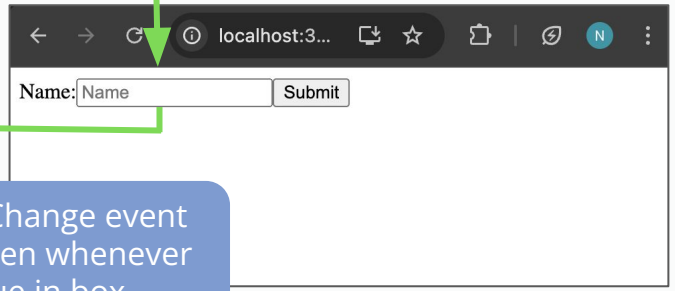
Forms in React

Create a state variable name and its setter function setName. And add it to the value attribute to our input box.

```
1  function FormExample() {
2      const [name, setName] = useState("");
3
4      return (
5        <form>
6          <label>
7            Name:
8            <input
9              type="text"
10             value={name}
11             placeholder="Name"
12         />
13         </label>
14         <button type="submit">Submit</button>
15       </form>
16     );
17 }
18
19 export default FormExample;
```

When name state variable changes, value in the input box also changes too

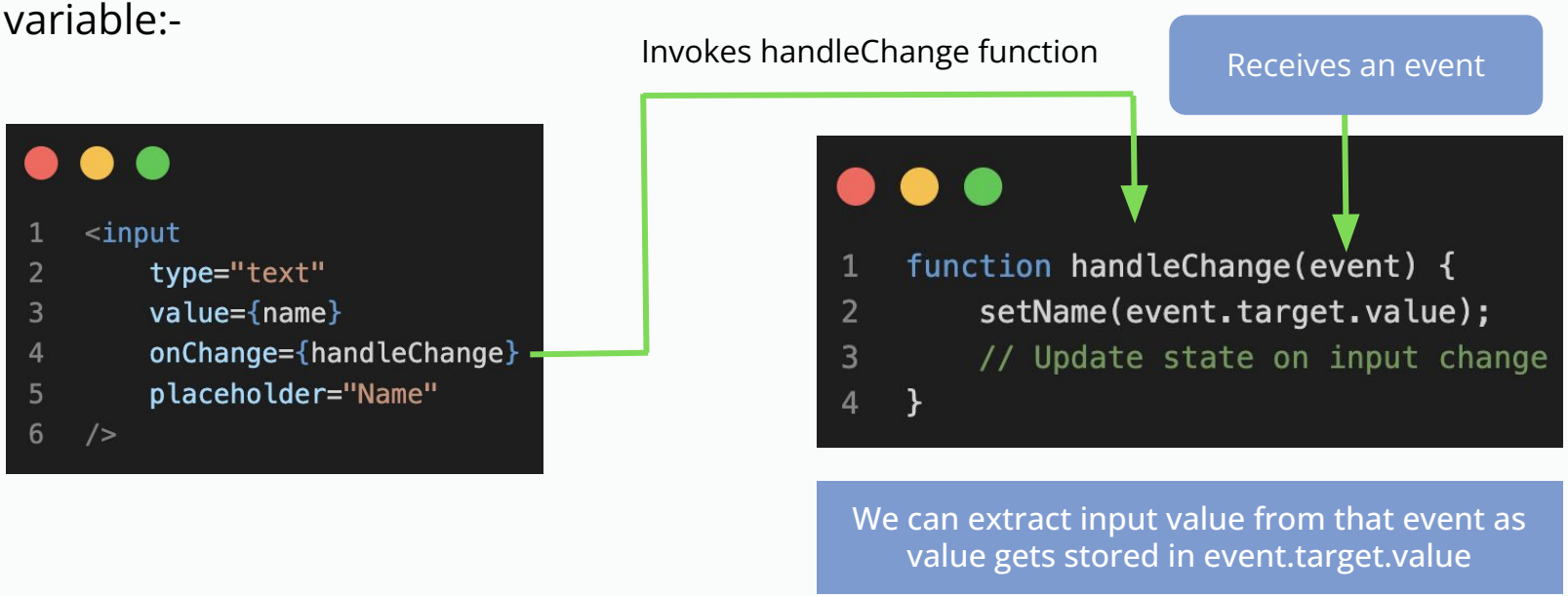
Now we need to change the name state variable whenever user types a value in input box



We can add an onChange event which throws an even whenever we change value in box

Forms in React

Add onChange event and associate a function which changes the value of name state variable:-



Forms in React

Let’s also add an event on form submission:-

```
function FormExample() {
    const [name, setName] = useState("");

    const handleSubmit = (event) => {
        event.preventDefault();
        console.log('Submitted:',    name );
    };

    function handleChange(event) {
        setName(event.target.value);
        // Update state on input change
      }

    return (
      <form onSubmit={handleSubmit}>
        <label>
          Name:
          <input type="text" value={name} onChange={handleChange} />
        </label>
        <button type="submit">Submit</button>
      </form>
    );
}
```

We need to prevent the default behaviour of submit using event.preventDefault()

When we submit the form handleSubmit function is invoked and by default an event is passed to it.

Workshop

Workshop: Build a ToDo App

Create a **To-Do App** using React that allows users to **add, and display tasks**.

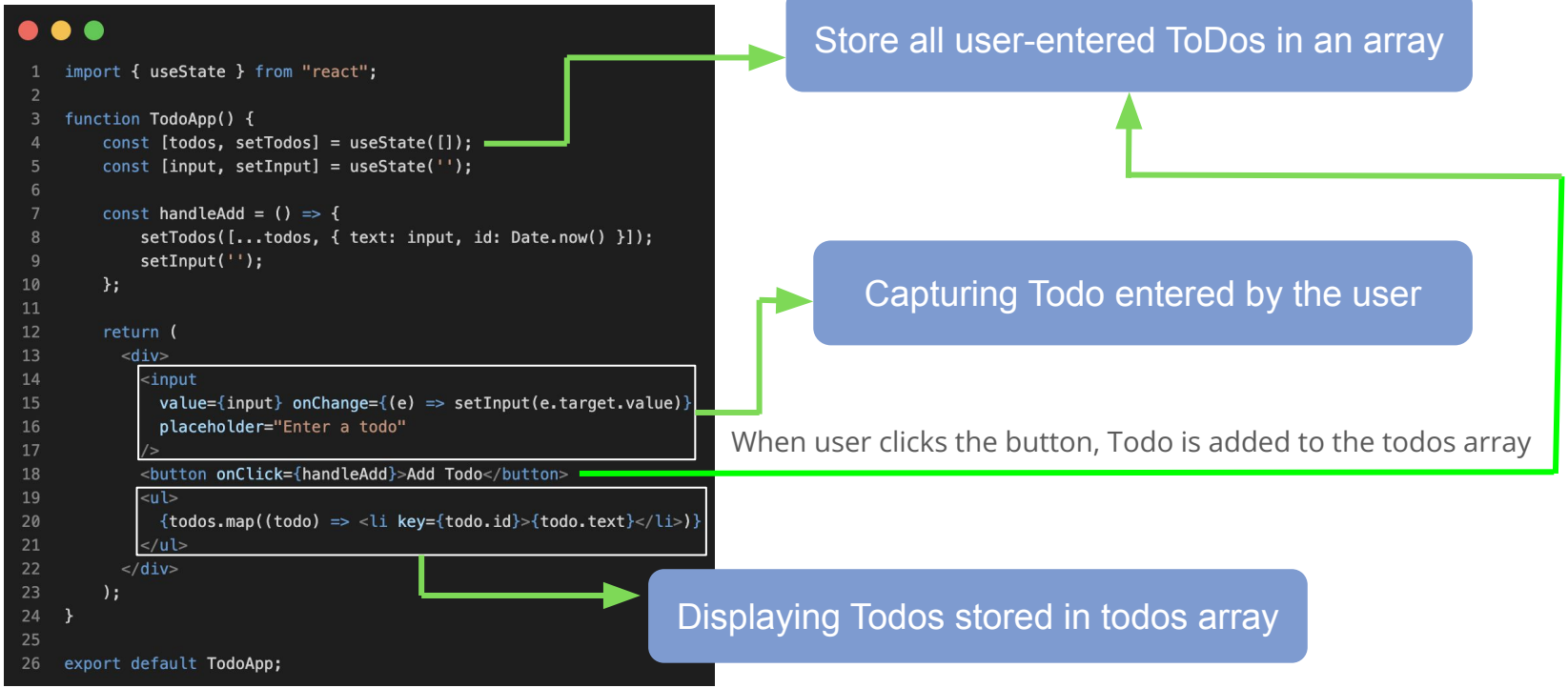
- Use the useState hook to manage the list of tasks.
- Provide an input field where users can enter a task.
- Add a button to submit and add the task to the list.

After Entering a task and pressing Enter

Workshop: Solution

Newton School of Technology

Let’s have a look at the solution:-



Workshop: Solution

Newton School of Technology

Let’s have a look at the solution:-

```
7       const handleAdd = () => {
8           setTodos([...todos, { text: input, id: Date.now() }]);
9           setInput('');
10      };
11
```

Here:-

1. **...todos**: fetches all the previous todos and
2. **{ text: input, id: Date.now() }**: New todo

We are using timestamps as id

In Class Questions

References

1. **MDN React Guide**: Comprehensive and beginner-friendly documentation for React https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started
2. **Websites and Tutorials**:
   a. 📌 w3schools: https://www.w3schools.com/REACT/DEFAULT.ASP
   b. 📌codecademy: https://www.codecademy.com/learn/react-101
3. **Other Useful Resources**
   a. 📌 React GitHub Repository: https://github.com/facebook/react
   b. 📌 React Patterns & Best Practices: https://reactpatterns.com/

Newton School of Technology

Thanks for watching!