

```

import { useState } from "react"

export default function Counter() {
  let [count, setCount] = useState(0); // to introduce state variable in functional
  let [name, setName] = useState("Roger");

  return <div>
    Name: {name} <br />
    Count : {count} <br />
    <button type="button" onClick={() => setCount(count + 1)}>Increment</button>
  </div>
}

```

```

class Counter extends Component {
  state = {
    name: "Roger",
    count: 0
  }

  updateCount() {
    this.setState({
      count: this.state.count + 1
    })
  }

  render() {
    return <div>
      Name: {this.state.name} <br />
      Count: {this.state.count} <br />
      <button type="button" onClick={() => updateCount()}>Increment</button>
    </div>
  }
}

```

```

const initialState = {
  count: 0
}

export default function ReducerComponent() {
  let [state, dispatch] = useReducer(countReducer, initialState);

  return <div>
    Count: {state.count} <br />
    <button type="button" onClick={() => dispatch({type: 'INCREMENT', payload: 10})}>Increment</button>
    <button type="button" onClick={() => dispatch({type: 'DECREMENT'})}>Decrement</button>
    <button type="button" onClick={() => dispatch({type: 'RESET'})}>CLEAR</button>
  </div>
}

export default function counterReducer(state, action) {
  switch(action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
      }
    default:
      return state;
  }
}

```

```

const initialState = {
  count: 0
}

export default function ReducerComponent() {
  let [state, dispatch] = useReducer(countReducer, initialState);

  return <div>
    Count: {state.count} <br />
    <button type="button" onClick={() => dispatch({type: 'INCREMENT', payload: 10})}>Increment</button>
    <button type="button" onClick={() => dispatch({type: 'DECREMENT'})}>Decrement</button>
    <button type="button" onClick={() => dispatch({type: 'RESET'})}>CLEAR</button>
  </div>
}

export default function counterReducer(state, action) {
  switch(action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
      }
    default:
      return state;
  }
}

```

Handwritten annotations: A box labeled '1B' contains the state {count: 10}. An arrow labeled '2' points from this box to the 'INCREMENT' case in the reducer. An arrow labeled '1A' points from the 'INCREMENT' case back to the state box. An arrow labeled 'action' points to the dispatch function call in the UI component.