# 3. React Basics

Saturday, 20 December 2025 5:34 PM

Steps to create:

- Install Node from https://nodejs.org/en/download (LTS)
    - Check version: node -v
- Use this website for helping with basic features - https://create-react-app.dev/docs/getting-started/
    - npm init react-app my-app
    - cd my-app
    - npm start
- Once its started it will run in some localhost, then edit the App.js file inside src:
    - Function App(){ return <div>Hello React</div>;}
    - Export default App;

Components: All components inside react is just a tree
ReactDOM.render() -> will render our application

- If you want to add one component inside another component: Quiz is a component, Question is a component
    - Then add this quiz component by replacing app inside ReactDOM.render()
    - Then inside quiz add a tag with the other component:

        -
          ```
          Const Quiz = () => {
          Return (
          <div>
                  <div>Quiz</div>
                  <Question/>
          <div>
          );}
          Export default Quiz;
          ```

- const [current, setCurrent] = useState(0);

    This does **two things**:

    1. Creates a **state variable** → current
    2. Gives you a **function to update it** → setCurrent
    3. Sets the **initial value** → 0 (used **only once**, on first mount)

    Think of it like:

    "React, please remember a value called current for this component, and start it at 0."

## useState survives re-renders

useState stores the value **outside the function**, in React's memory.
So even when the function runs again, React gives you the **previous value**.

## useEffect

👉 **Runs side effects AFTER render**

-
  ```
  useEffect(() => {
   console.log("Component rendered");
  }, []);
  ```
  Think:

  "After React paints the UI, run this extra logic."

## Why does useEffect exist?

React rendering must be **pure**:

- No API calls
- No subscriptions
- No DOM manipulation
- No timers

So React says:

“First I’ll render UI → then you can do side effects”

That’s exactly what useEffect is.