

---

# Day 2 — JSX & React Components

## Goal:

Understand how React works using **JSX** and **Functional Components**, and learn how UI is built in React.

---

## 1. What is JSX?

JSX (JavaScript XML) allows us to write HTML-like code inside JavaScript.

Example:

```
const heading = <h1>Hello React</h1>;
```

JSX is **not HTML**, it gets converted into JavaScript by React.

---

## 2. Why JSX is Used?

- Makes UI code readable
  - Combines logic + UI in one place
  - Faster development
- 

## 3. React Functional Components

A component is just a **JavaScript function** that returns JSX.

Example:

```
function App() {
  return (
    <div>
      <h1>My First React Component</h1>
      <p>Welcome to React</p>
    </div>
  );
}
```

```
export default App;
```

Rules: - Component name must start with a **capital letter** - Must return **one parent element**

---

## 4. Using Components

You can reuse components like HTML tags.

Example:

```
function Header() {
  return <h2>This is Header</h2>;
}

function App() {
  return (
    <div>
      <Header />
      <p>Main Content</p>
    </div>
  );
}
```

---

## 5. JSX Expressions

You can use JavaScript inside JSX using `{}`.

Example:

```
const name = "Malik";

function App() {
  return <h1>Hello {name}</h1>;
}
```

## 6. File Structure (Day 2)

```
src/
├── components/
│   └── Header.jsx
├── App.jsx
└── main.jsx
```

## 7. Practice Tasks (Must Do)

### Task 1:

Create a `Header` component that shows: - Website name - Simple navigation text

### Task 2:

Create a `Footer` component with: - Current year - Simple copyright text

### Task 3:

Use `Header` and `Footer` inside `App.jsx`

## 8. Common Mistakes

- Forgetting to return JSX
- Using lowercase component names
- Missing parent element in JSX

## 9. End of Day Result

By the end of Day 2, you should: - Understand JSX - Create and reuse components - Structure React files properly

## Next Day Preview (Day 3)

- Props
- Passing data between components
- Dynamic components