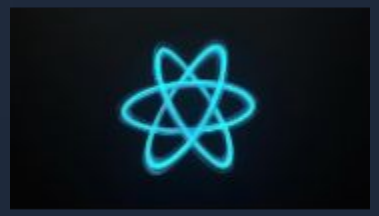


Getting Started with React

Day 2: Components, State, and Navigation

6-HOUR INTENSIVE

BOOTCAMP

Learning Goals

By the end of today, you will be able to build, style, and navigate modular React applications using modern best practices.

Core Learning Roadmap

Foundations

Understand components, props, and the power of JSX for reusable UI.

Reactivity

Master the useState and useEffect hooks to handle data and side effects.

Navigation

Implement routing and context to share state across complex apps.

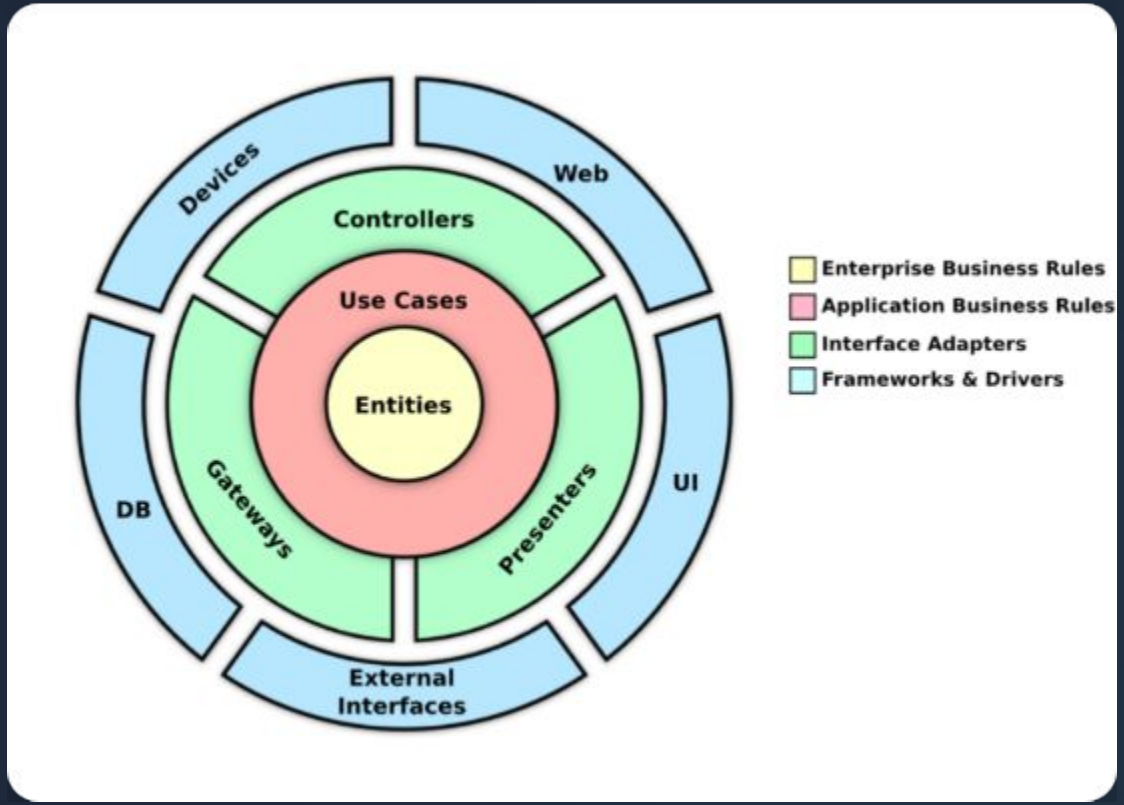
Understanding Components

What is a Component?

A component is an independent, reusable piece of UI. It allows you to split the UI into separate parts and think about each piece in isolation.

```
function Welcome() { return Welcome to React; }
```

Components can be used anywhere in your application, just like custom HTML tags.



JSX: JavaScript + HTML

The JSX Concept

JSX is a syntax extension that looks like HTML but works inside JavaScript. It makes writing UI much more intuitive.

```
const name = "Alex"; return Hello, {name};
```

Key Rules

Expressions must be wrapped in curly braces {}. Every JSX element must return a single root tag or a Fragment <>.

Use className instead of class for styling attributes.

Props: Passing Data

1-WAY

Data Flow Direction

Sending Data Downwards

Props (short for properties) are how we pass information from a Parent component to a Child component.

```
// Inside Greeting component: Hello {props.name}
```

This makes components highly reusable and customizable.

State Management: useState

The useState Hook

State is "memory" for a component. It stores data that changes over time, like counters or user input.

```
const [count, setCount] = useState(0); setCount(count
+ 1)}> Add
```

Handling Forms

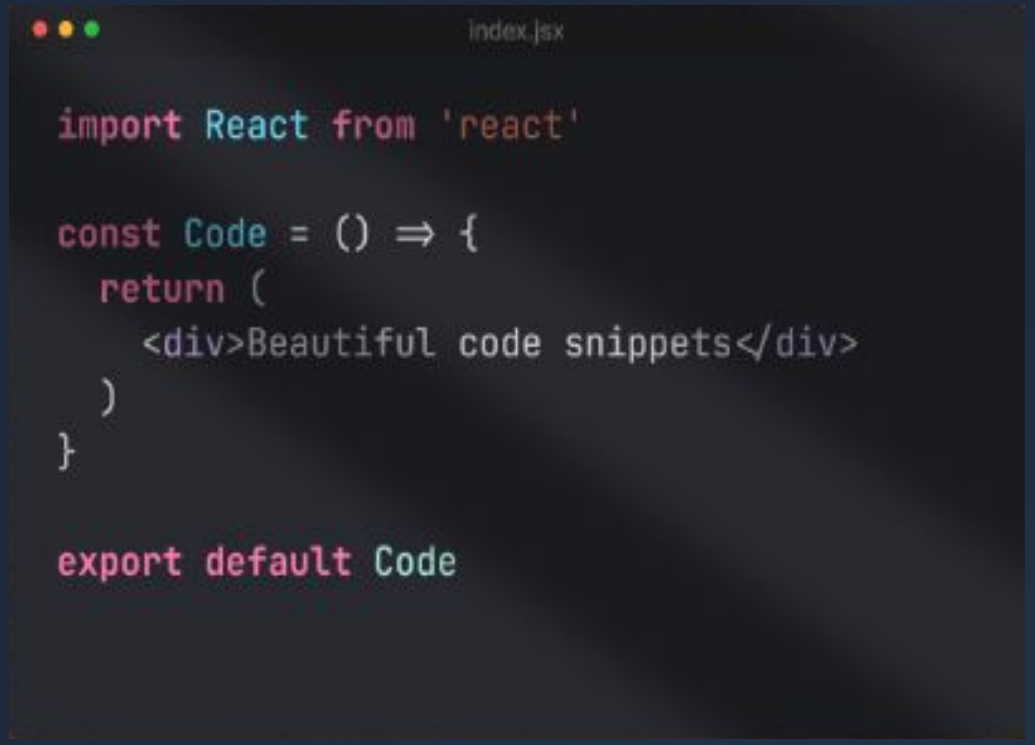
We use state to track input changes in real-time. This is called a "controlled component."

```
setName(e.target.value)} />
```

Side Effects: useEffect

- **Data Fetching:** Pull information from external APIs.
- **Mounting:** Run code as soon as the component loads.
- **Execution:** useEffect(() => { ... }, []) runs once.

It allows you to perform side effects in functional components that don't belong in the rendering logic.

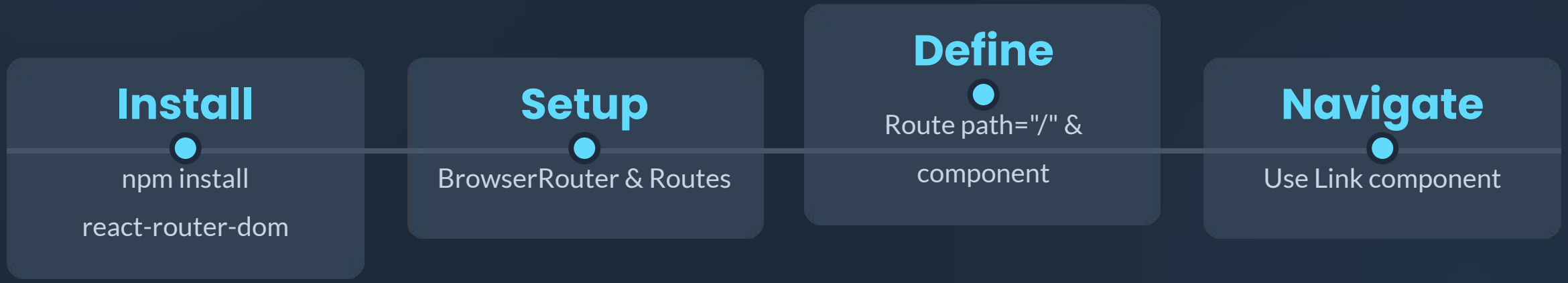


Styling the Modern Web

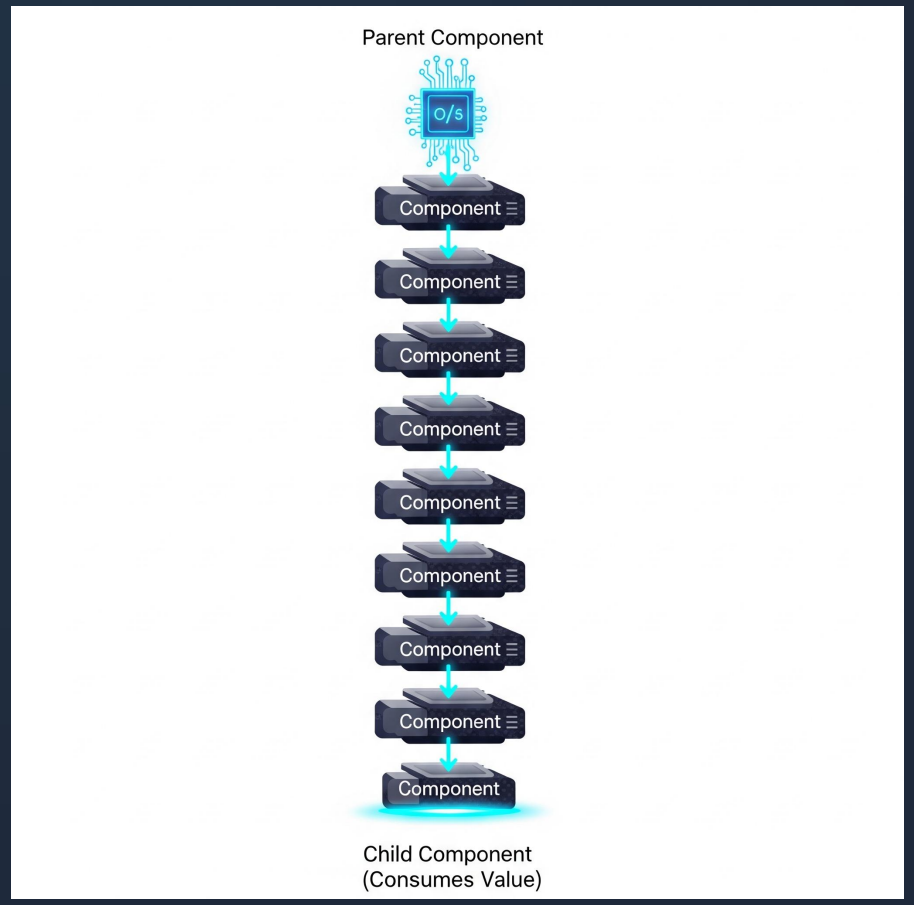
Styling Method	Standard CSS	Tailwind CSS
Syntax	Separate .css files	Utility classes in JSX
Customization	Highly manual	Pre-defined design system
Example	className="title"	className="text-2xl font-bold"
Best For	Legacy or strictly unique designs	Rapid, consistent development

Page Navigation Flow

Implement Multi-Page behavior using **react-router-dom**.



State Sharing: useContext



Avoid Prop Drilling

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

```
const user = useContext(UserContext); return User: {user};
```

Ideal for themes, user auth, and global settings.

Questions?

Ready to build your first React app!

help@reactbootcamp.i

docs.reactjs.org

o