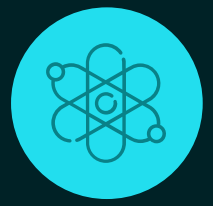
React JS Course

Information Technology Institute (ITI)

Course main parts



What is ReactJS?

ReactJS Basics

ReactJS Hooks

What is ReactJS?

Content:

- Introduction about ReactJS.
- Core concept and Features of ReactJS.
- Setup React on your machine.
- Folder structure of ReactJS.

ReactJS Basics

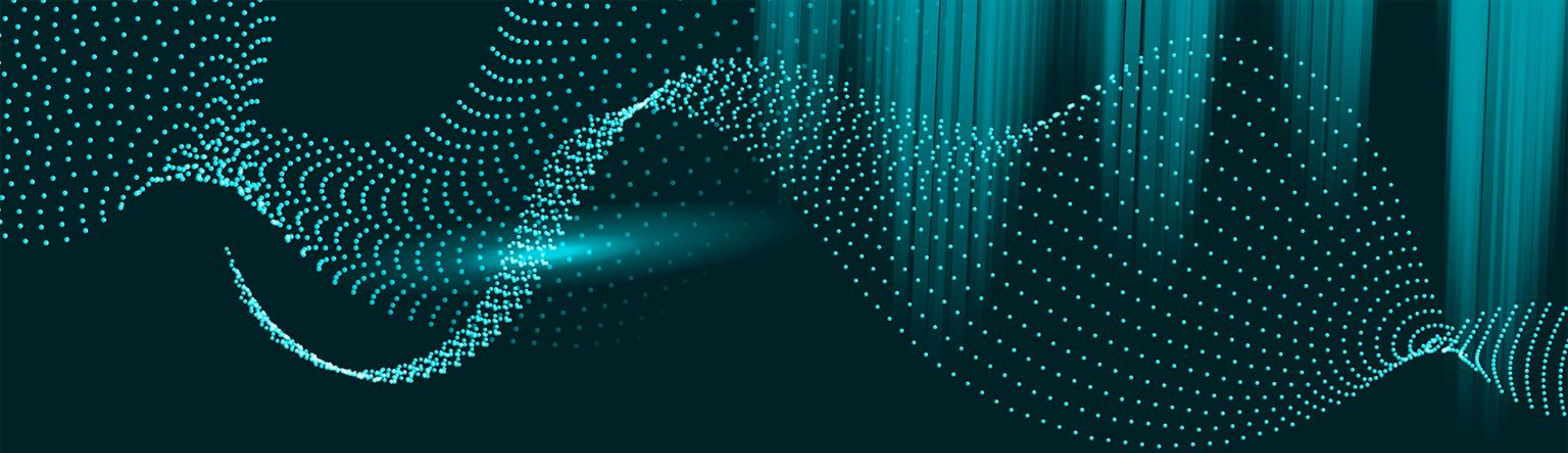
Content:

- Class based component.
- JSX, Props, and State.
- Destructing Prop, State.
- Event Handling, and Method as props.
- Conditional, List Rendering.
- Form Handling.
- Lifecycle Methods.
- Fragment, Refs, and Context.
- React with HTTP Requests.

ReactJS Hooks

Content:

- All about Hooks introduction and overview.
- useState Hook.
- useEffect Hook.
- Hooks Lifecycle.
- Fetch and add data using useEffect Hook.
- useContext Hook.
- useReducer Hook.
- useRef Hook.
- React memorization (Performance).
- Memo and useMemo Hook.
- useCallback Hook.
- Custom Hooks.



01 What is ReactJS?

Details and why using ReactJS?

Introduction about ReactJS

Definiation

Open-Source library for building user interfaces.

Key points you should notice:

1. React is a library not a framework.
2. React focus on UI nothing else.

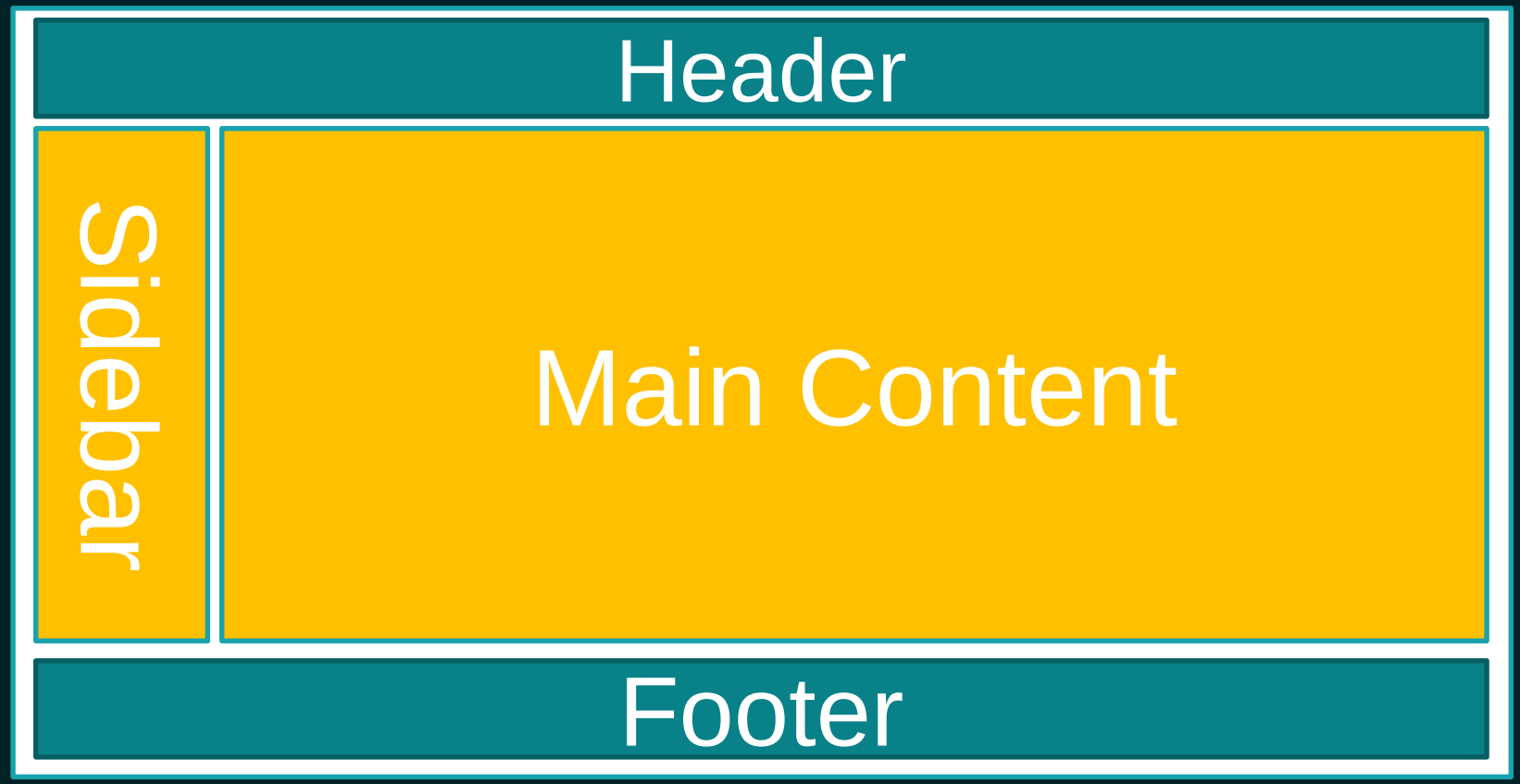
Hug Ecosystem and Community.

React created and maintained by Facebook.

Demand skillset in many companies.

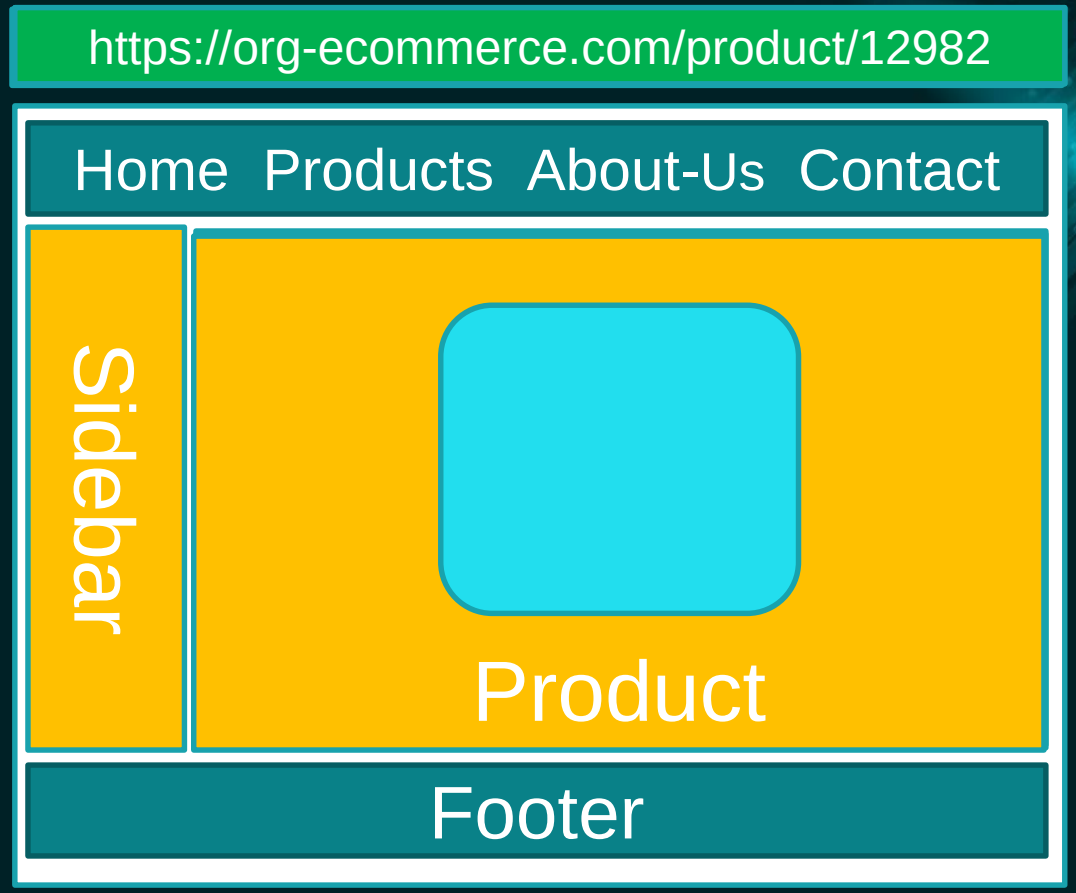
Introduction about ReactJS

Component based Architecture



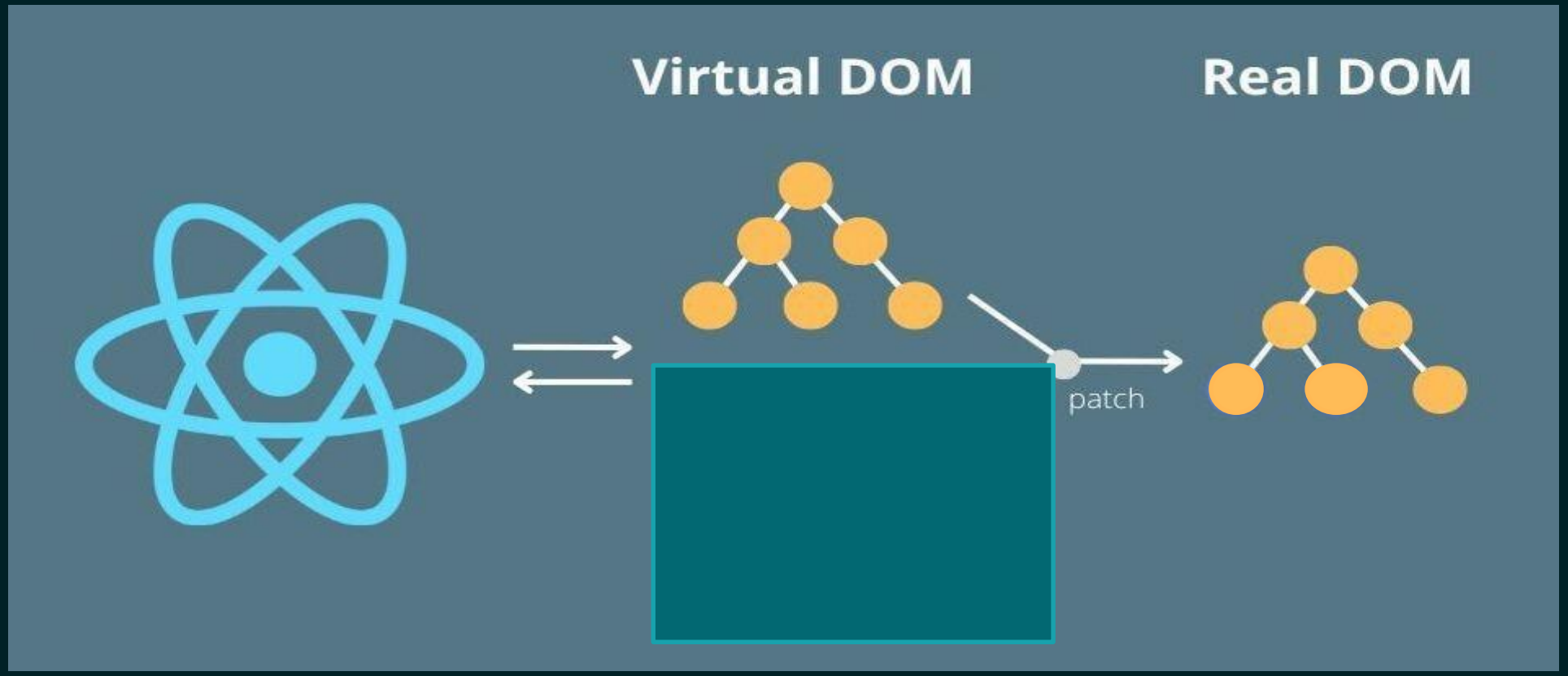
Core Concepts and features of ReactJS

- Single Page Application.
- Component based architecture
- JSX
- Virtual DOM



Core concepts and features of ReactJS

Virtual DOM



Setup React on your machine

Download Last version of node then in CMD use command node –v & npm -v

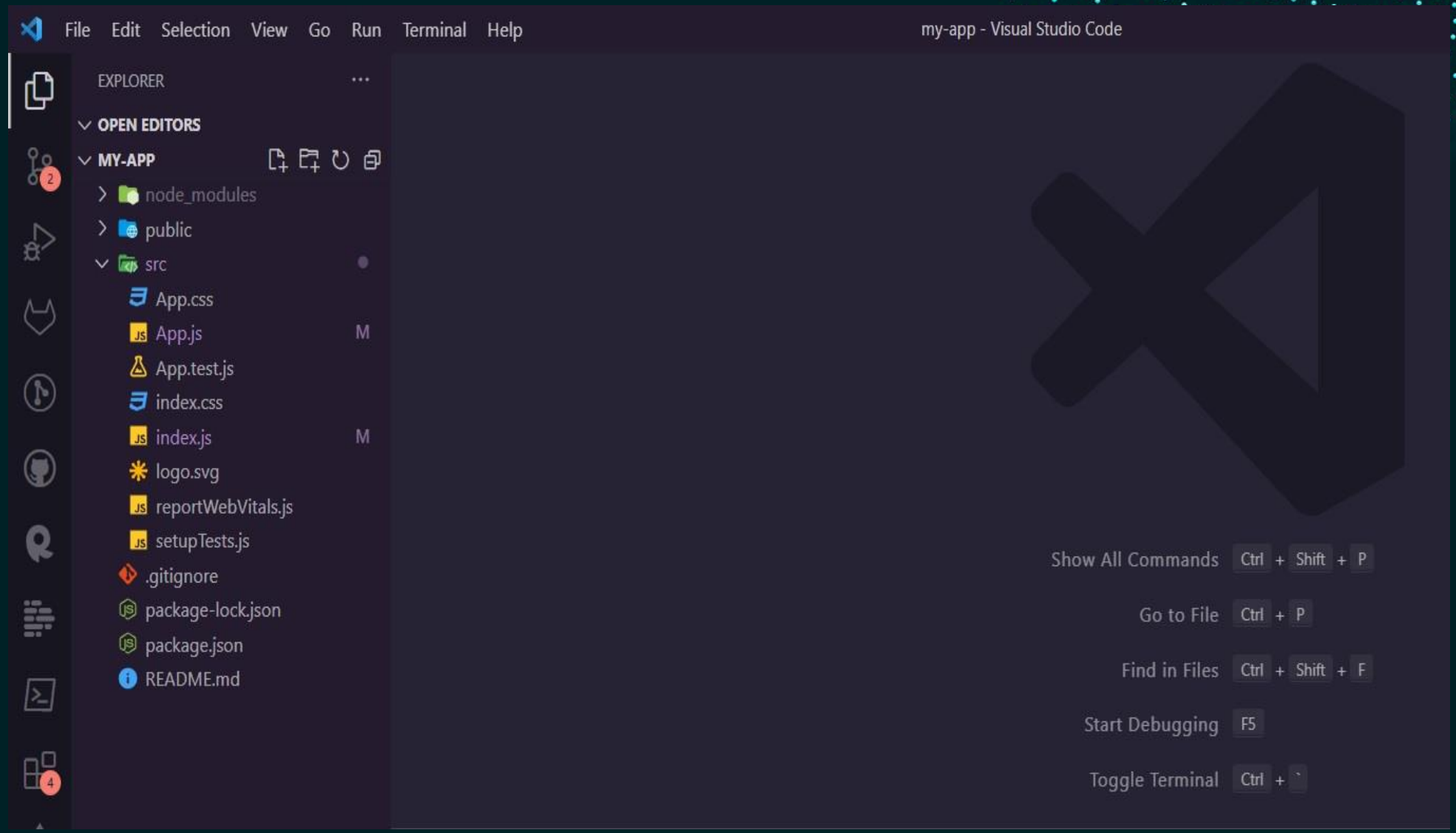
1. To install React

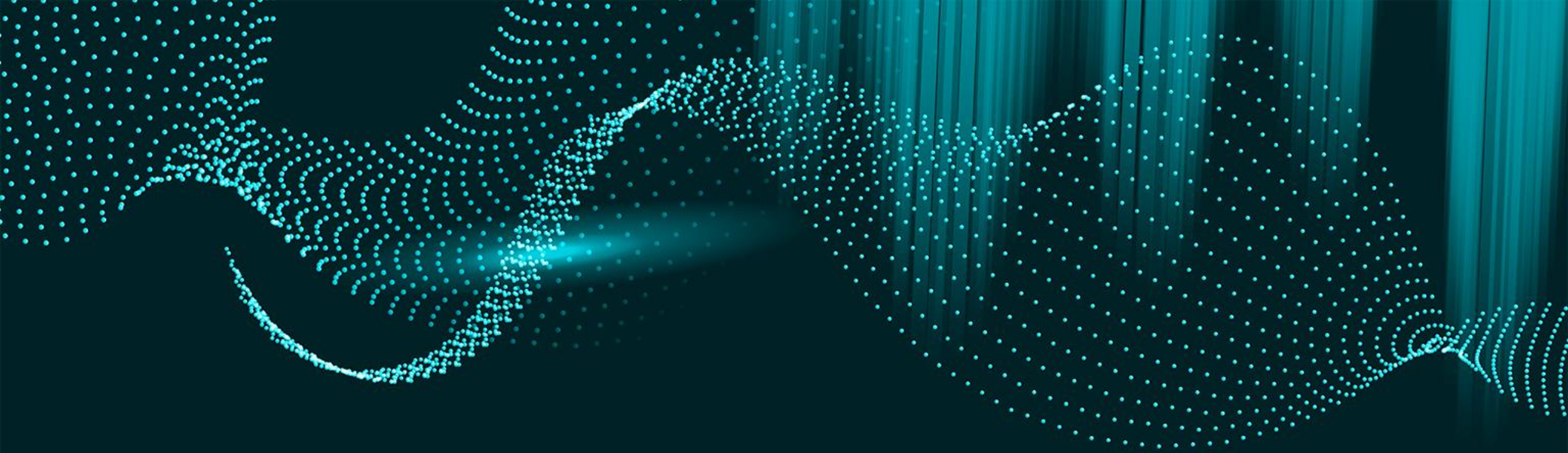
1. npx create-react-app {name of App}
2. cd {name of app}
3. npm start in command line

2. To install React

1. npm install create-react-app –g
2. create-react-app {name of App}
3. cd {name of app}
4. npm start in command line

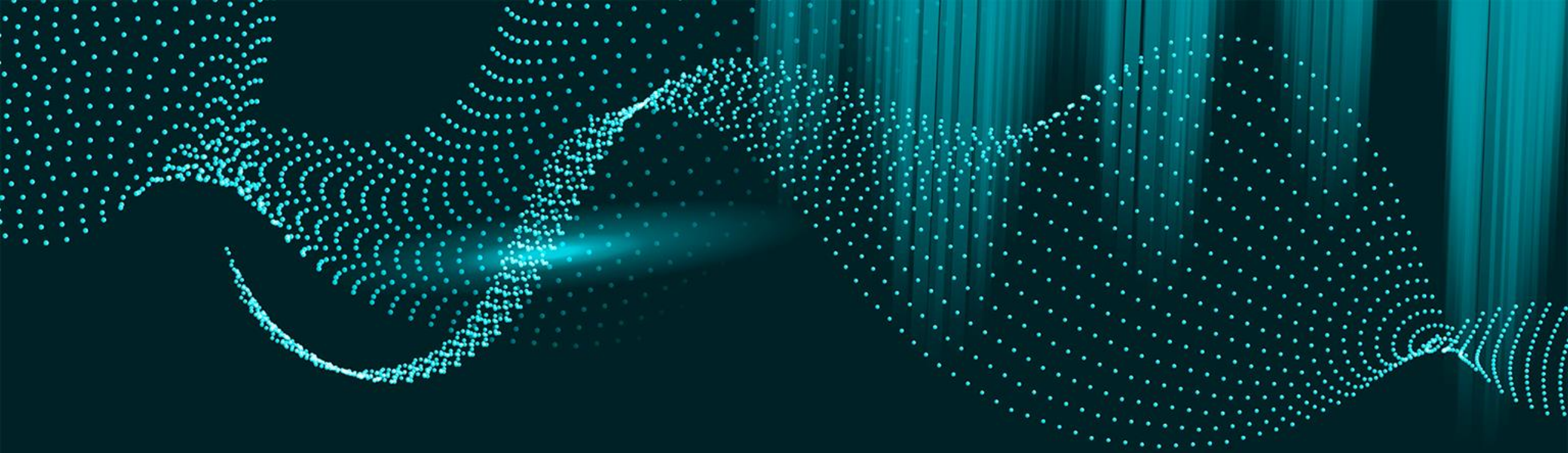
Folder Structure





Any Questions

Take a flyer to ask anything



02 ReactJS Basics

All you need to start your career

Class based compnent

There are two types of component:

Stateless functional component

```
1
2  function Greating(){
3      return (
4          <h1>Hello, IOT Track</h1>
5      )
6  }
```

Stateful Class component

```
1  import {Component} from "react";
2
3  class Greating extends Component {
4      render () {
5          return (
6              <h1>Hello, IOT Track </h1>
7          )
8      }
9  }
```

JSX

JavaScript XML- Extension to the JavaScript language syntax.

JSX tags like HTML tags have a tag name, attributes, and children.

JSX makes your react code simpler and elegant.

Props

Props is just a shorter way of saying properties.

We use props in React to pass data from one component to another (from a parent component to a child component(s)).

They are useful when you want the flow of data in your app to be dynamic.

State

State is a built-in React object that is used to contain data or information about the component.

component’s state can change over time; whenever it changes, the component re-renders.

The change in state can happen as a response to user action or system-generated events.

Destructing Props and State

is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.

We have two ways to Destruct props:

1. In parentheses of function
2. In function body

Events Handling

Handling events with React elements is very similar to handling events on DOM elements. There are some syntax differences:

- React events are named using camelCase, rather than lowercase.
- With JSX you pass a function as the event handler, rather than a string.

HTML Example :

```
<button onclick="activateLasers()">
  Activate Lasers
</button>
```

React Example :

```
<button onClick={activateLasers}>
  Activate Lasers
</button>
```

Conditional Rendering

We have 4 way to make Conditional Rendering in React:

1. If/else
2. Element variable
3. Ternary operator
4. Short circuit

List Rendering

You will often want to display multiple similar components from a collection of data. You can use the JavaScript array methods to manipulate an array of data. On this page, you'll use **filter()** and **map()** with React to filter and transform your array of data into an array of components.

Why You Need Keys in Lists?

React assigns every item a unique key attribute and so is able to keep track of them despite any changes. This helps in ensuring that you do not end up messing up your code when changes occur in your lists.

Form Handling

HTML form elements work a bit differently from other DOM elements in React, because form elements naturally keep some internal state.

1. Handling forms is about how you handle the data when it changes value or gets submitted.
2. In HTML, form data is usually handled by the DOM.
3. In React, form data is usually handled by the components.
4. When the data is handled by the components, all the data is stored in the component state.
5. You can control changes by adding event handlers in the onChange attribute.

Class component lifeCycle methods

Each component in React has a lifecycle which you can monitor and manipulate during its four main phases. The four phases are: **Mounting**, **Updating**, **Unmounting**, and **Error Handling**.

1. **Mounting**: Mounting means putting elements into the DOM.

   React has four built-in methods that gets called, in this order, when mounting a component:

   A. constructor()
   B. getDerivedStateFromProps()
   C. render()
   D. componentDidMount()

The **render()** method is required and will always be called, the others are optional and will be called if you define them

Class component lifeCycle methods

2. **Updating**: The next phase in the lifecycle is when a component is updated.

   A component is updated whenever there is a change in the component's state or props.

   React has five built-in methods that gets called, in this order, when a component is updated:

   A. getDerivedStateFromProps()
   B. shouldComponentUpdate()
   C. render()
   D. getSnapshotBeforeUpdate()
   E. componentDidUpdate()

   The **render()** method is required and will always be called, the others are optional and will be called if you define them.

Class component lifeCycle methods

3. **Unmounting**: The next phase in the lifecycle is when a component is removed from the DOM, or unmounting as React likes to call it.
   React has only one built-in method that gets called when a component is unmounted:

   A. componentWillUnmount()

4. **Error Handling**: There are two main methods in error handling. These method are used in the error boundary mechanism in React

   A. Static getDerivedStateFromError()
   B. componentDidCatch()

Refs

Refs provide a way to access DOM nodes or React elements created in the render method.

When to Use Refs ?

There are a few good use cases for refs:

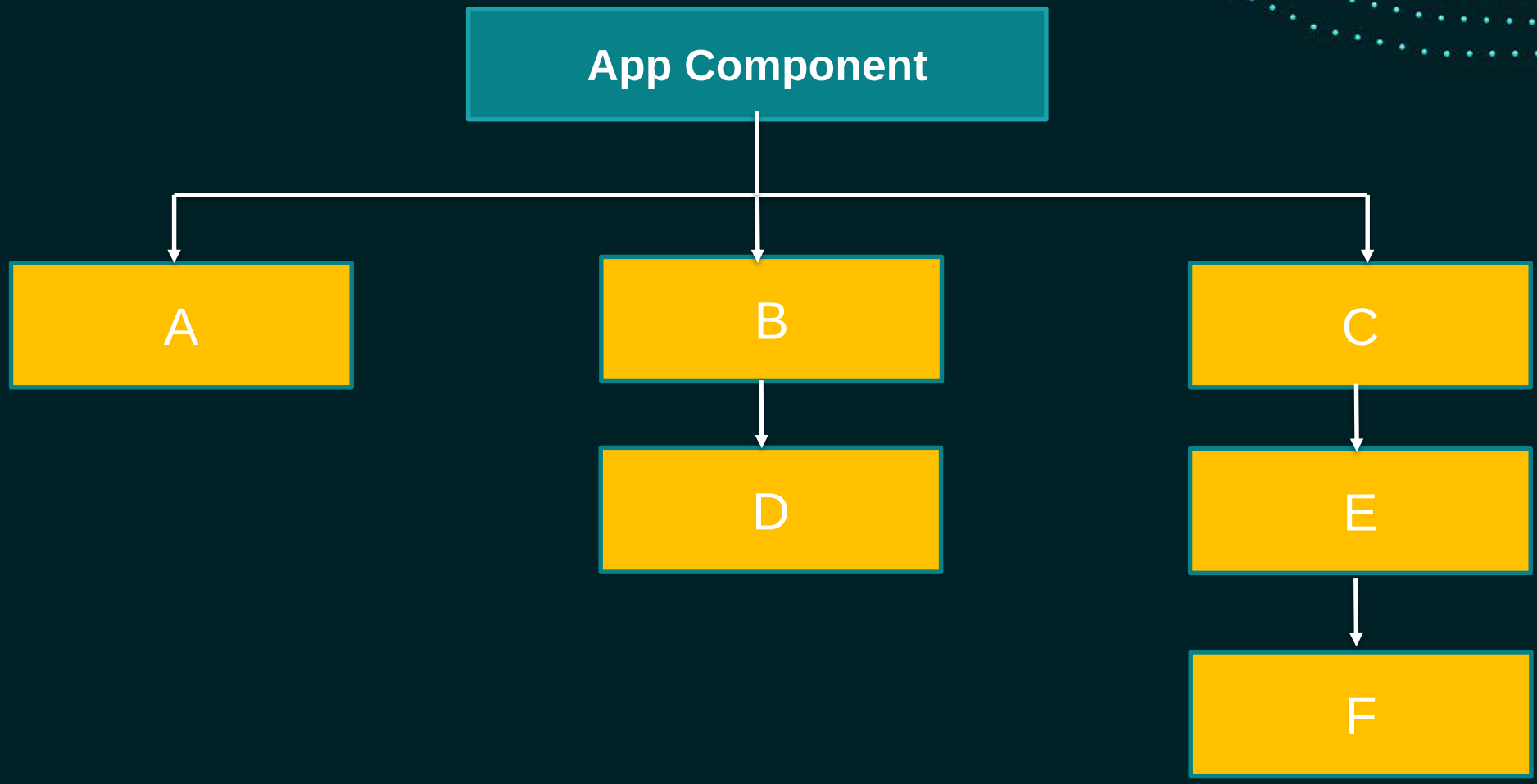
- Managing focus, text selection, or media playback.
- Triggering imperative animations.
- Integrating with third-party DOM libraries

Fragment

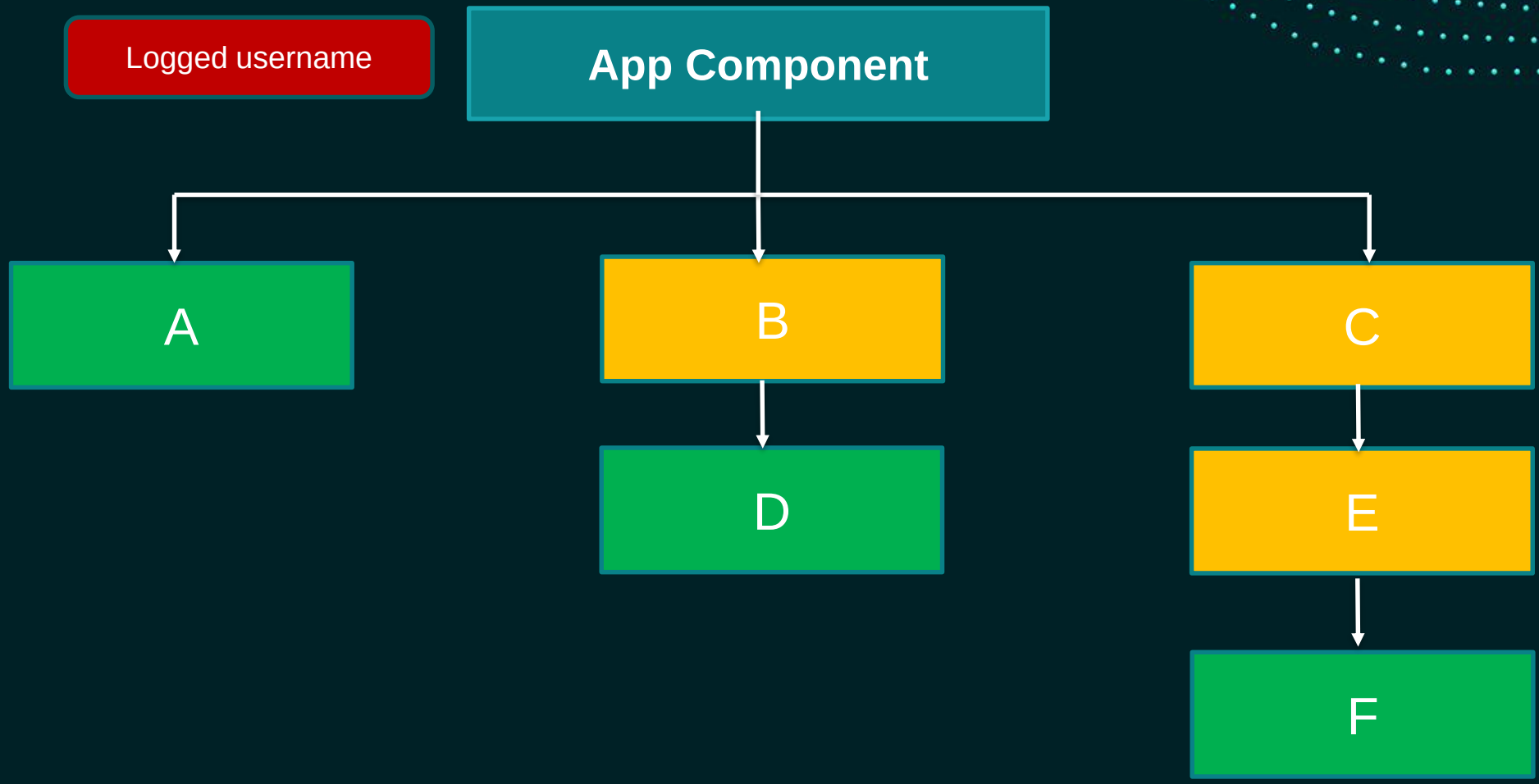
Wrap elements in <Fragment> to group them together in situations where you need a single element. Grouping elements in Fragment has no effect on the resulting DOM; it is the same as if the elements were not grouped. The empty JSX tag <></> is shorthand for <Fragment></Fragment> in most cases.

Fragments are useful because grouping elements with a Fragment has no effect on layout or styles, unlike if you wrapped the elements in another container like a DOM element.

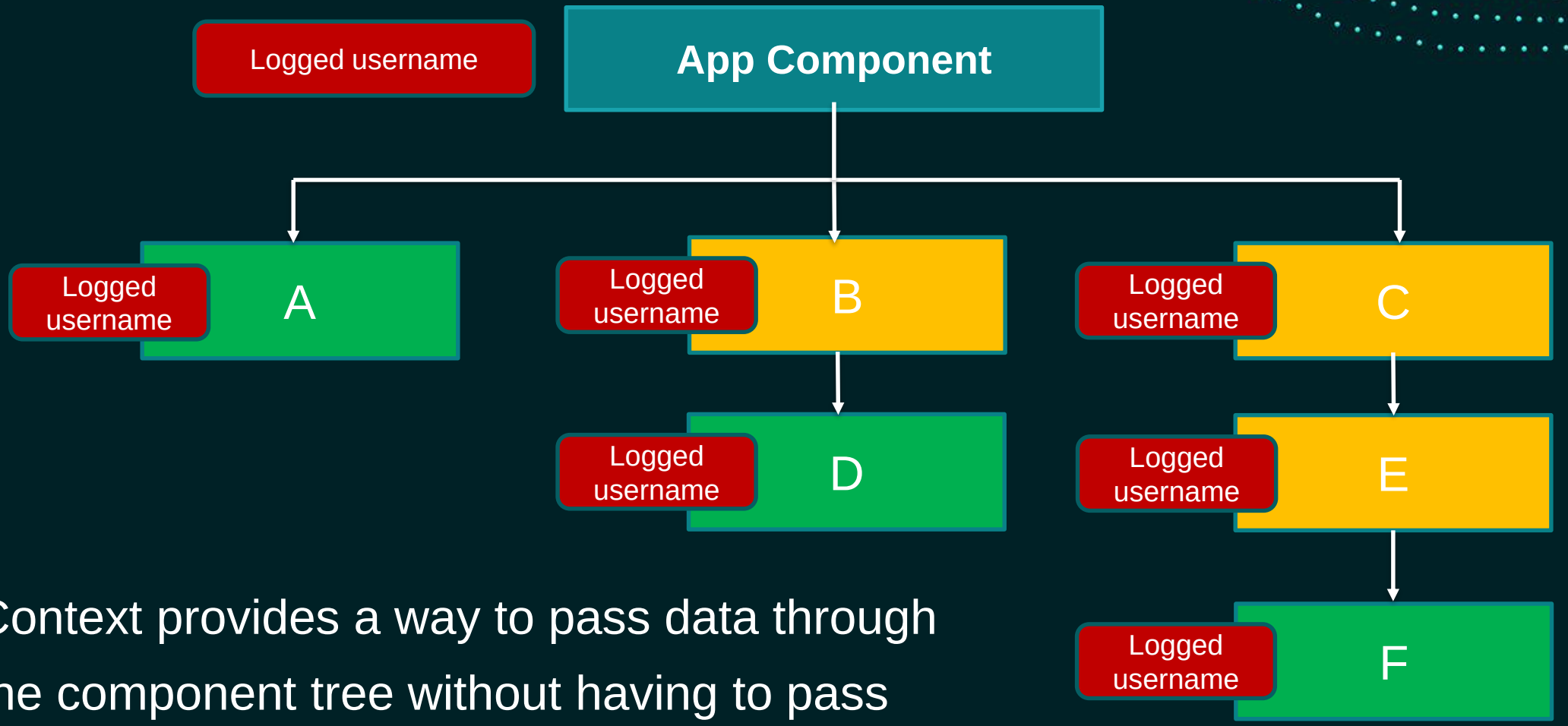
Context



Context

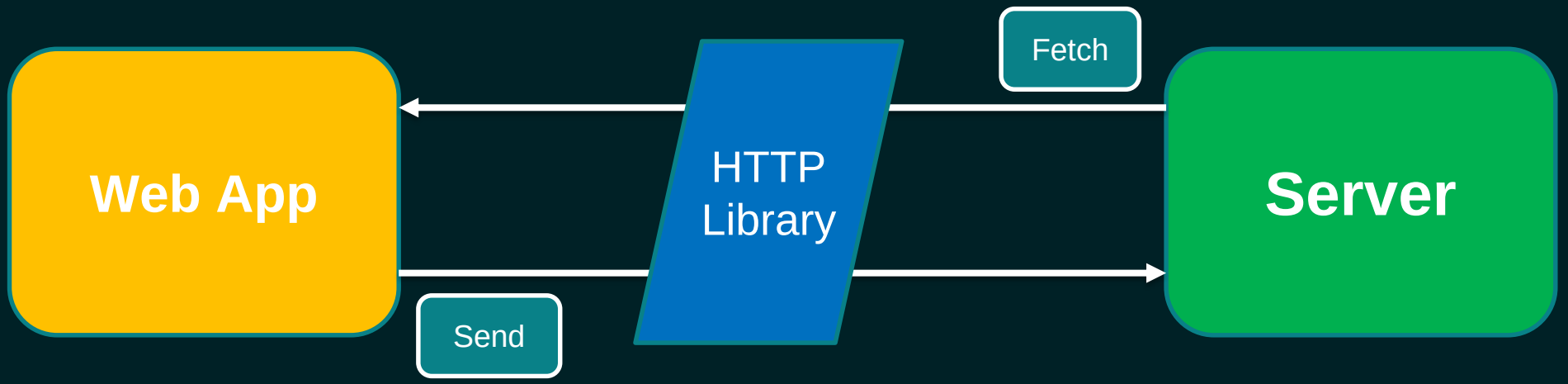


Context

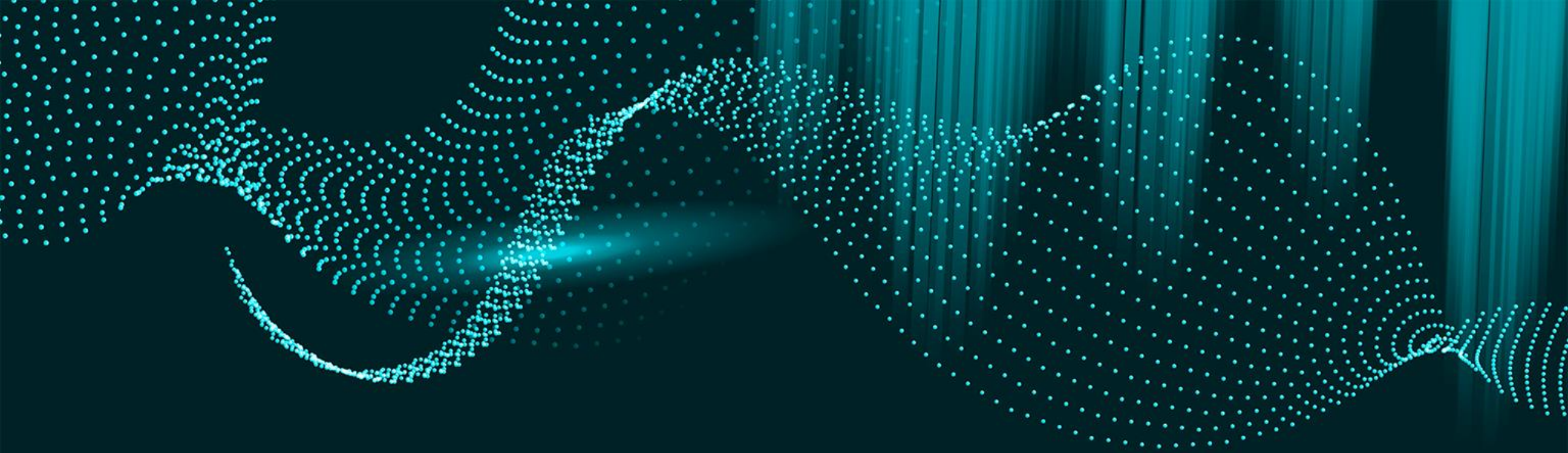


Context provides a way to pass data through the component tree without having to pass props down manually at every level.

HTTP Requests



Axios: `npm install axios`



Any Questions

Take a flyer to ask anything you want