



React.Js Training

Get started with the best
Javascript Framework :p



Introduction to React

01.

What is a framework ?

02.

Frontend Javascript Frameworks

03.

Get to know React



React Fundamentals

01.

Create your first react app

02.

Components, JSX, Styling

03.

Props & State



React Fundamentals

04.

Conditional Rendering

05.

Lists and Keys

06.

Event Handling



React Hooks

01.

Core Hooks

02.

Stateful Functional Components with Hooks

03.

Create a Custom Hook



Routing with React Router

01.

React Router Concepts

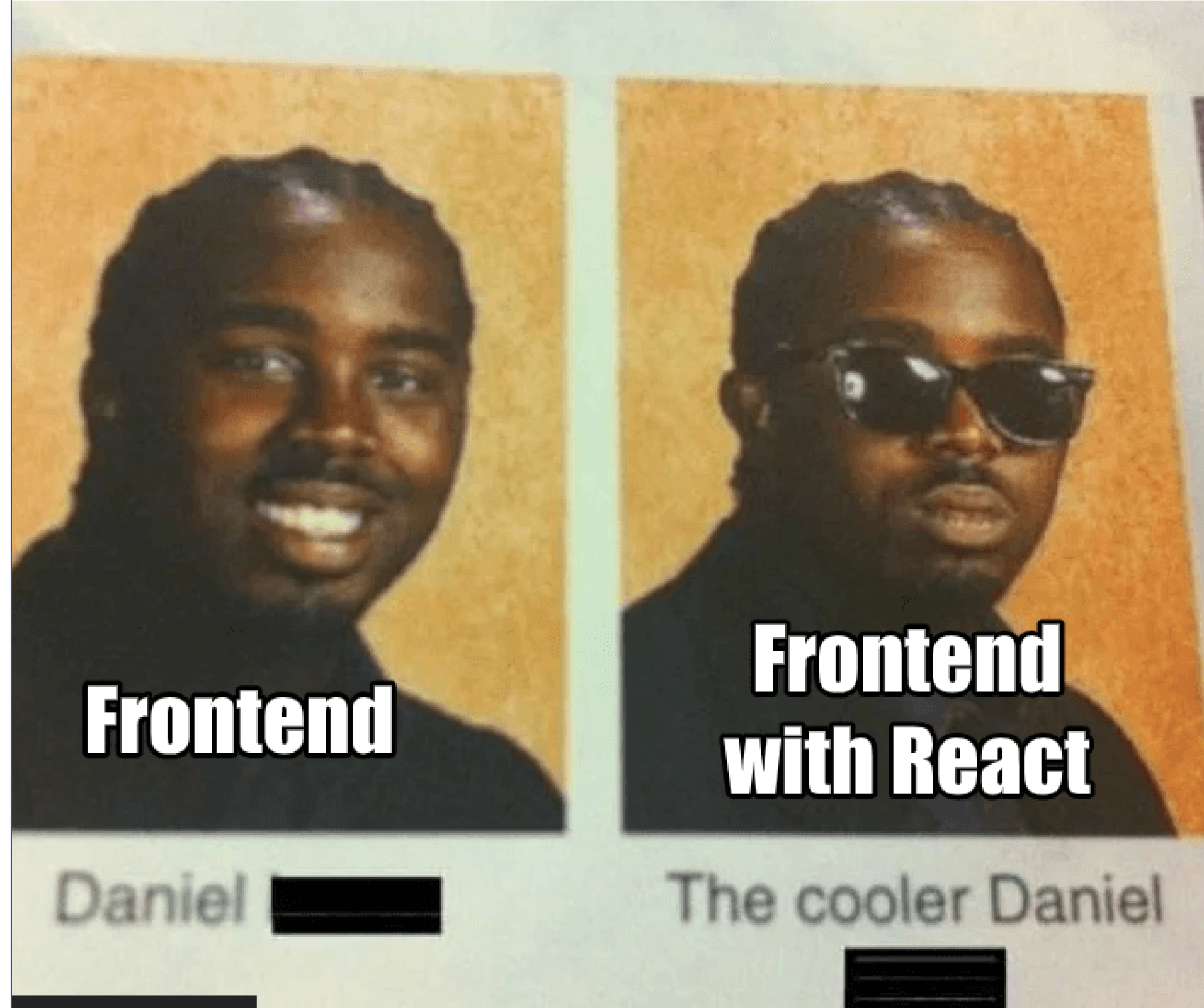
02.

Creating Application Routes

03.

React Router Hooks

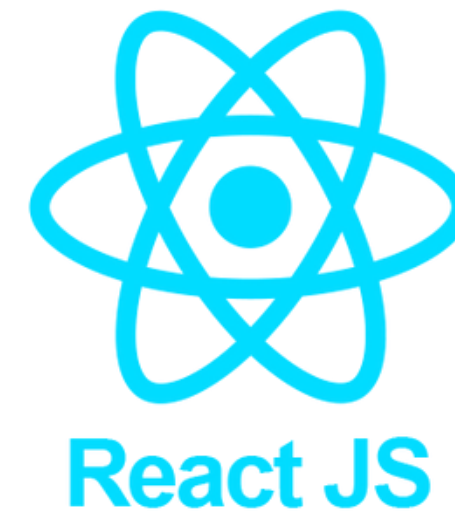
**Get ready to become
a cooler developer**



Introduction to React



- **What is a framework ?**
- **Frontend Javascript Frameworks**
- **Get to know React**



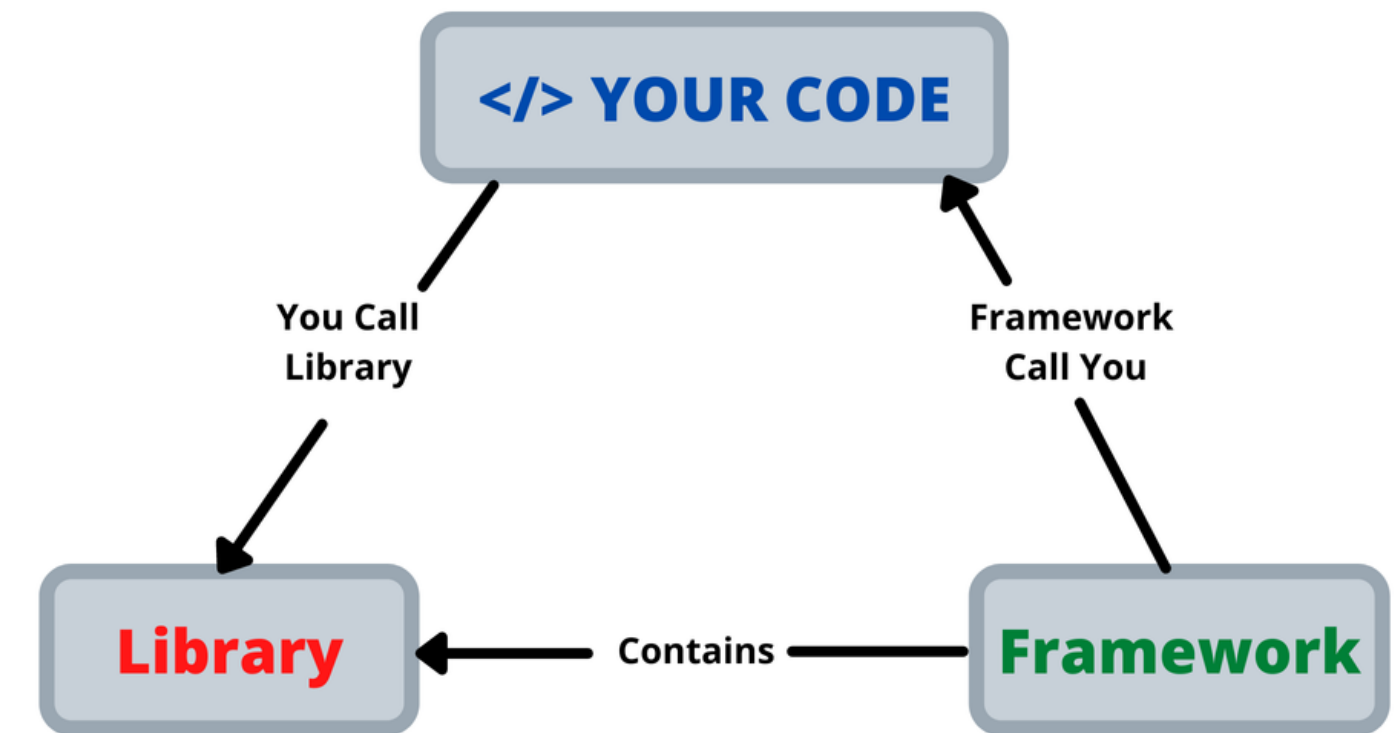
What is a framework?

A framework is a structure that you can build software on. It serves as a foundation, so you're not starting entirely from scratch. Frameworks are typically associated with a specific programming language and are suited to different types of tasks.

A framework IS NOT a library

What is a framework?

When you use a library, you are in charge of the flow of the application. You are choosing when and where to call the library. When you use a framework, the framework is in charge of the flow. It provides some places for you to plug in your code, but it calls the code you plugged in as needed. This is called **inversion of control**.



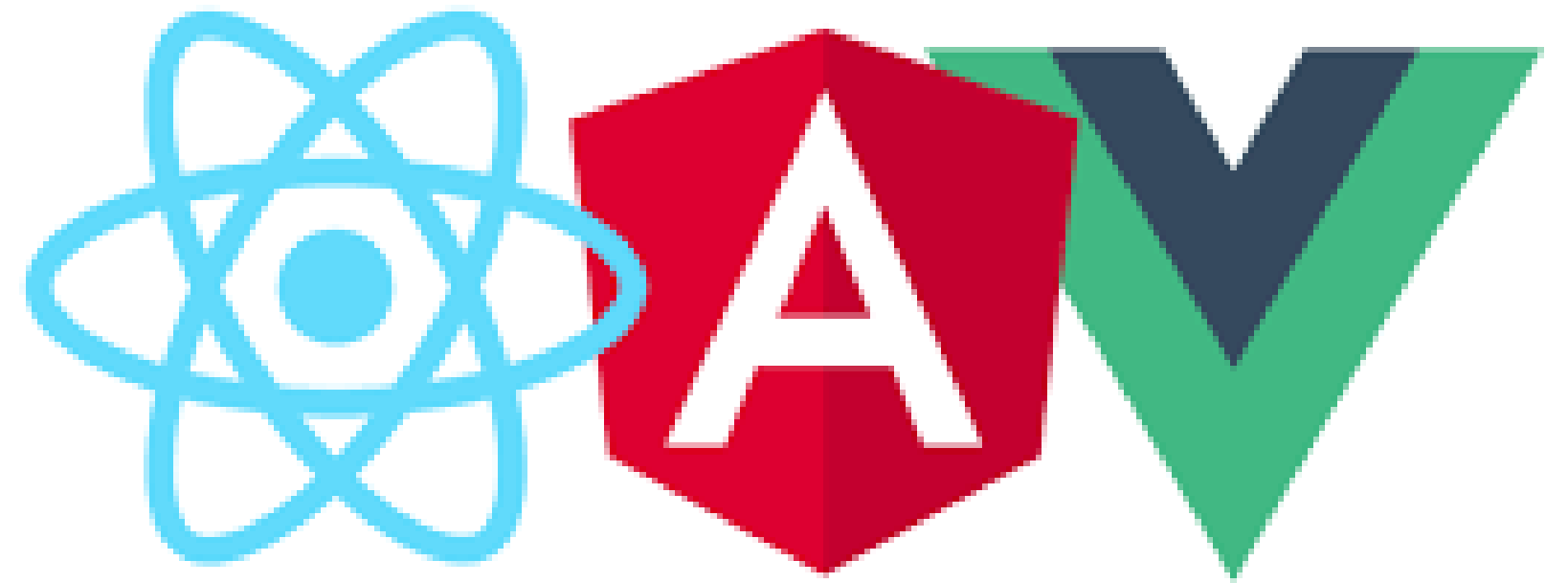
What is a framework?

Use a framework to:

- Save time and reduce the risk of getting errors
- More secure code
- Simpler testing and debugging
- Avoiding duplicate code
- Clean and easily adaptable code

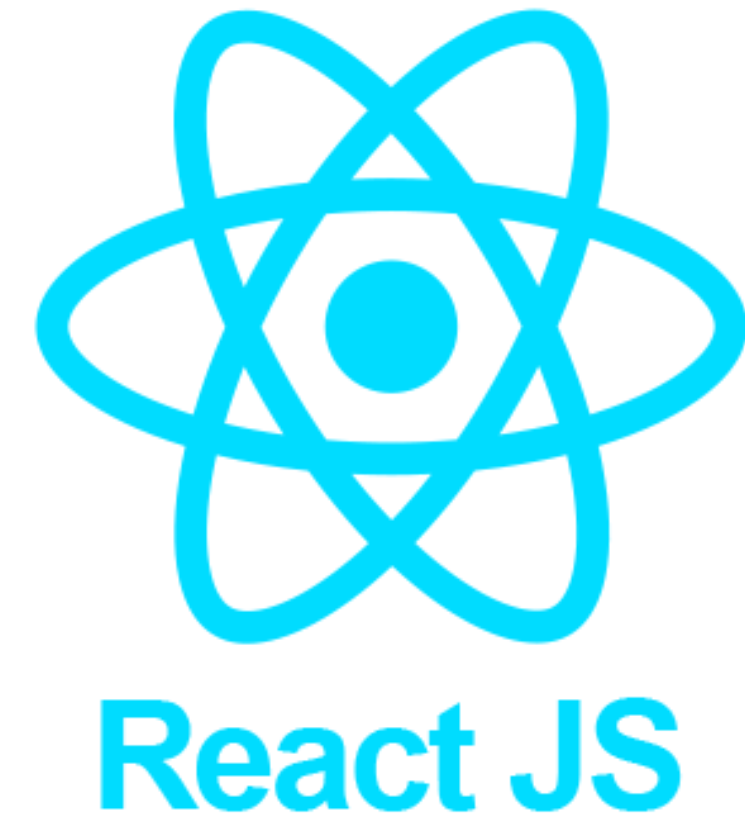
Frontend JavaScript Frameworks

Frontend JavaScript frameworks are an essential part of modern front-end web development, providing developers with tried and tested tools for building scalable, interactive web applications.



Get to know React

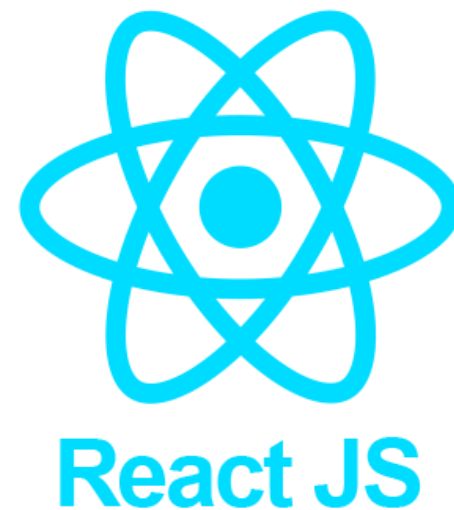
React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. ... React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple.



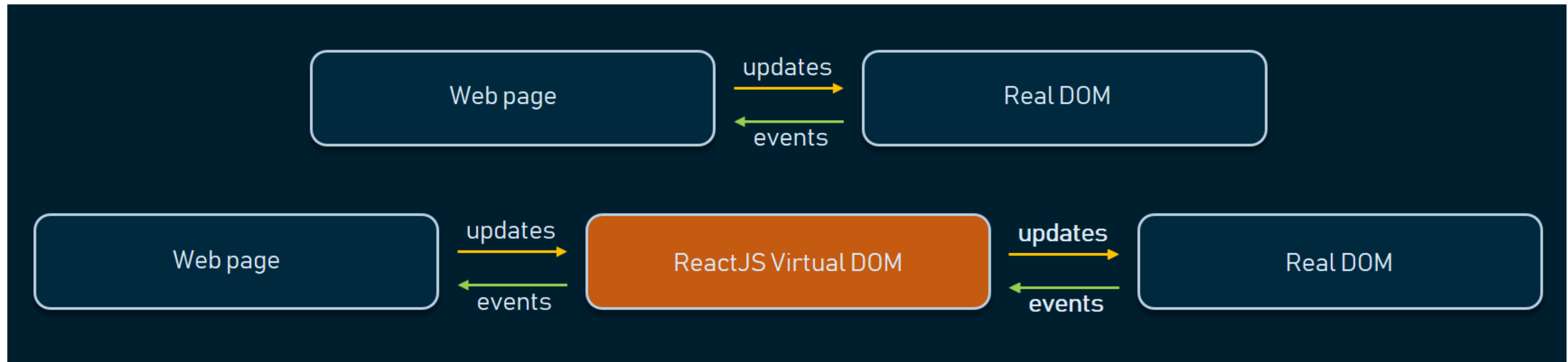
Get to know React

Here are the advantages of using React:

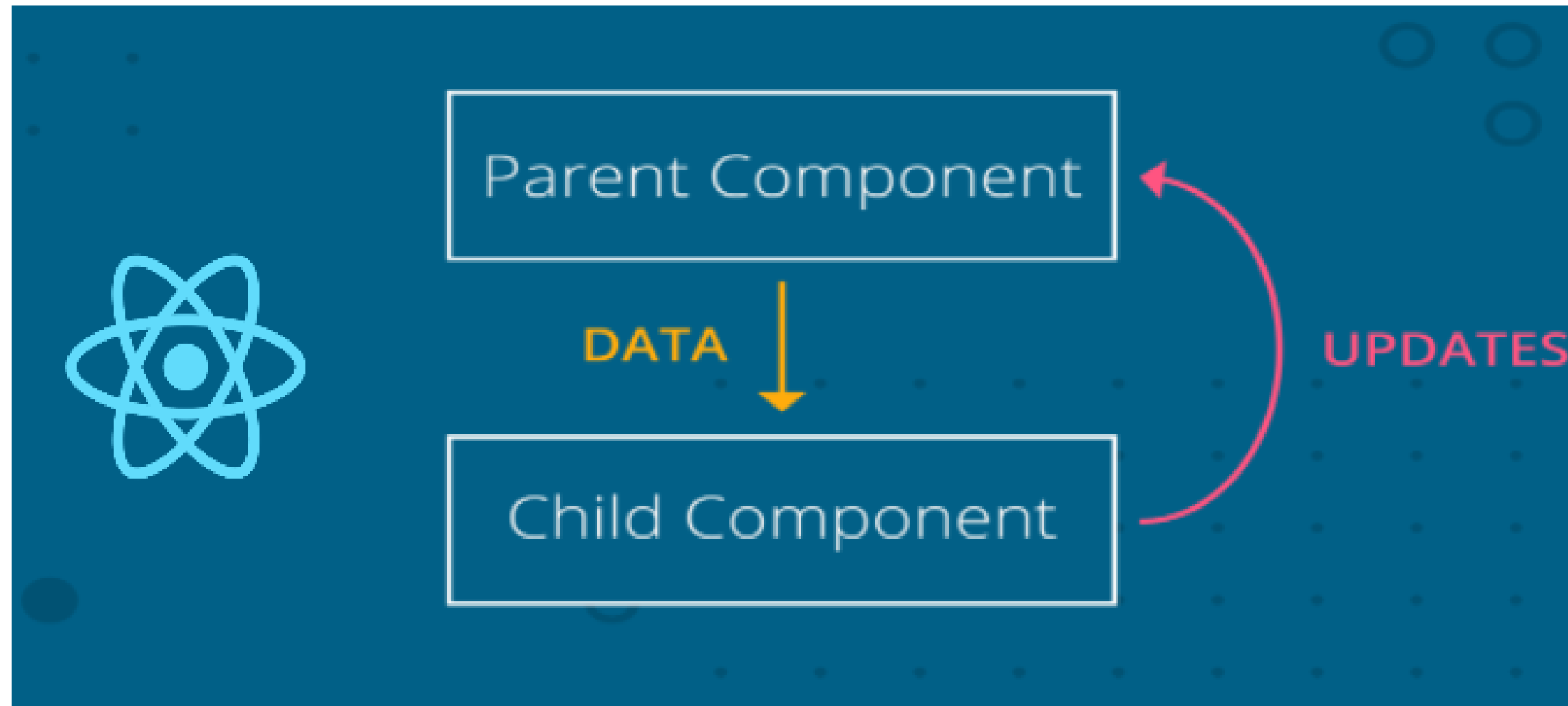
- Reusable Components
- Easy Integration
- One Direction Data Flow
- Simpler Syntax
- Virtual DOM
- SEO
- Continuous Improvement
- Huge Community Support



Real vs Virtual DOM



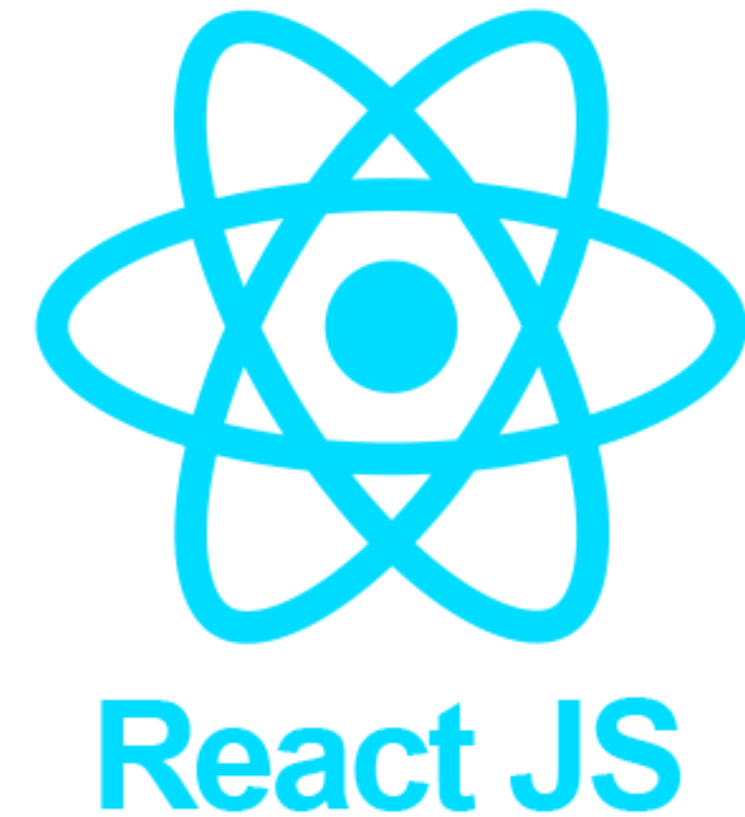
Unidirectional Data Flow



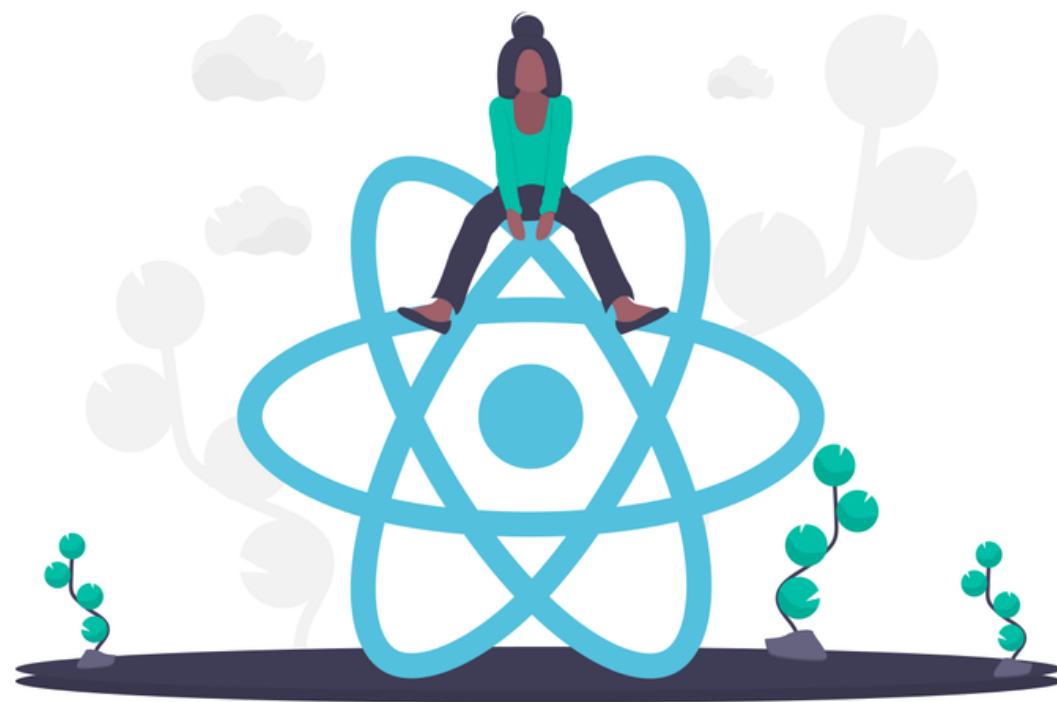
Get to know React

Here are the disadvantages of using React:

- The High Pace of Development
- View Part
- JSX as a barrier



React Fundamentals

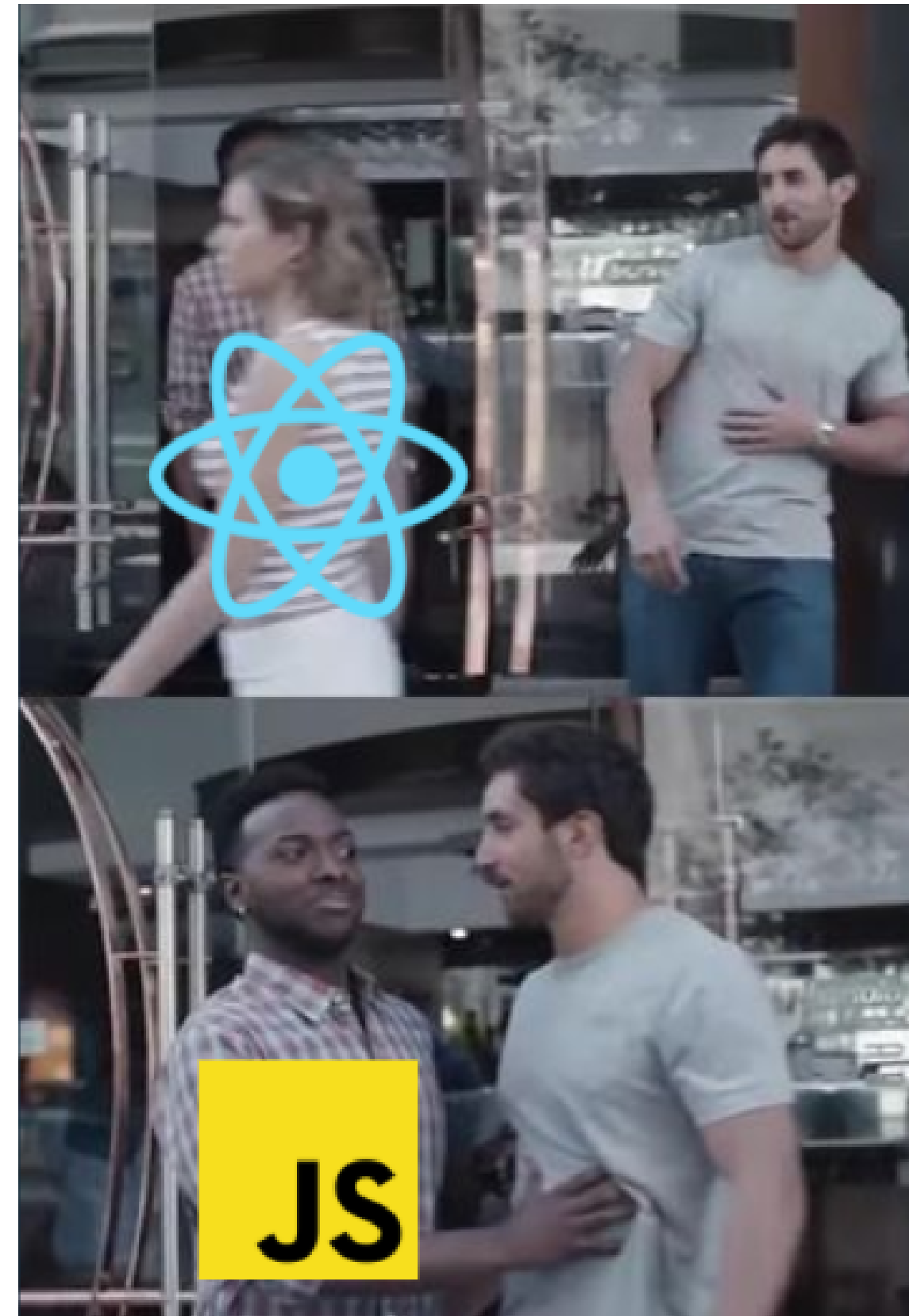


- **Create your first react app**
- **Components, JSX, Styling**
- **Props & State**
- **Conditional Rendering**
- **Lists and Keys**
- **Events Handling**

Create your first react app

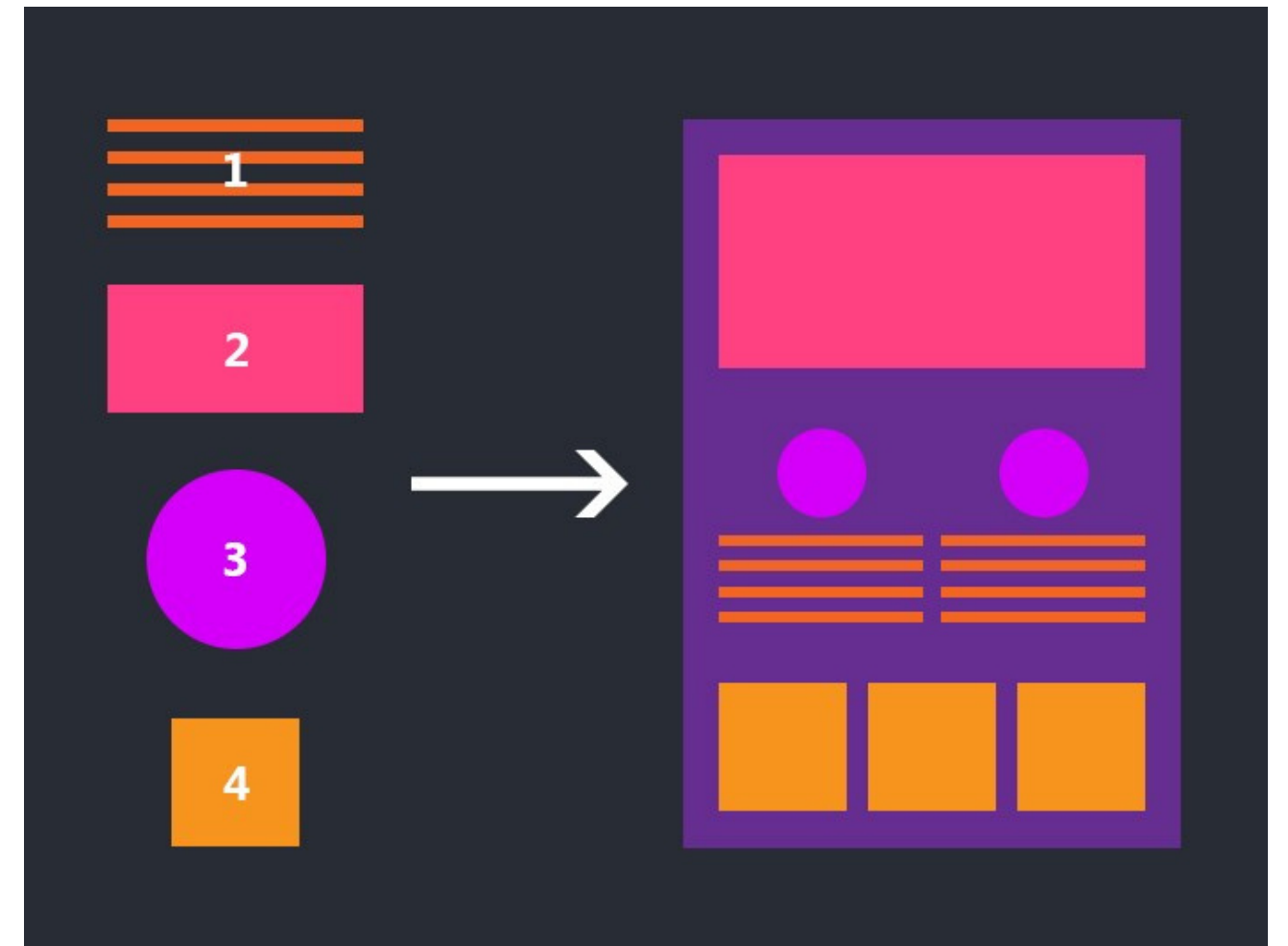
Before getting started with React, you need to have a basic knowledge in:

- HTML
- CSS
- JavaScript
- Package Manager (NPM/Yarn)



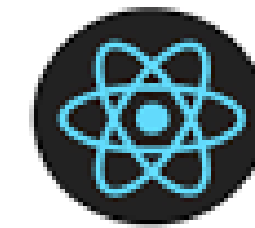
React Components

Components are the building blocks of any React app and a typical React app will have many of these. Simply put, a component is a JavaScript class or function that optionally accepts inputs i.e. properties (props) and returns a React element that describes how a section of the UI (User Interface) should appear alongside its logic.



React Components

If the component needs state or lifecycle methods then use class component otherwise use function component. However, from React 16.8 with the addition of Hooks, you could use state , lifecycle methods and other features that were only available in class component right in your function component.



React JS

**Class Component
Vs
Functional Component**

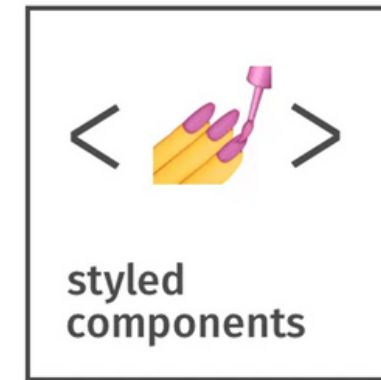
React JSX

JSX is an XML/HTML-like syntax used by React that extends ECMAScript so that XML/HTML-like text can co-exist with JavaScript/React code. ... Unlike the past, instead of putting JavaScript into HTML, JSX allows us to put HTML into JavaScript

React Styling

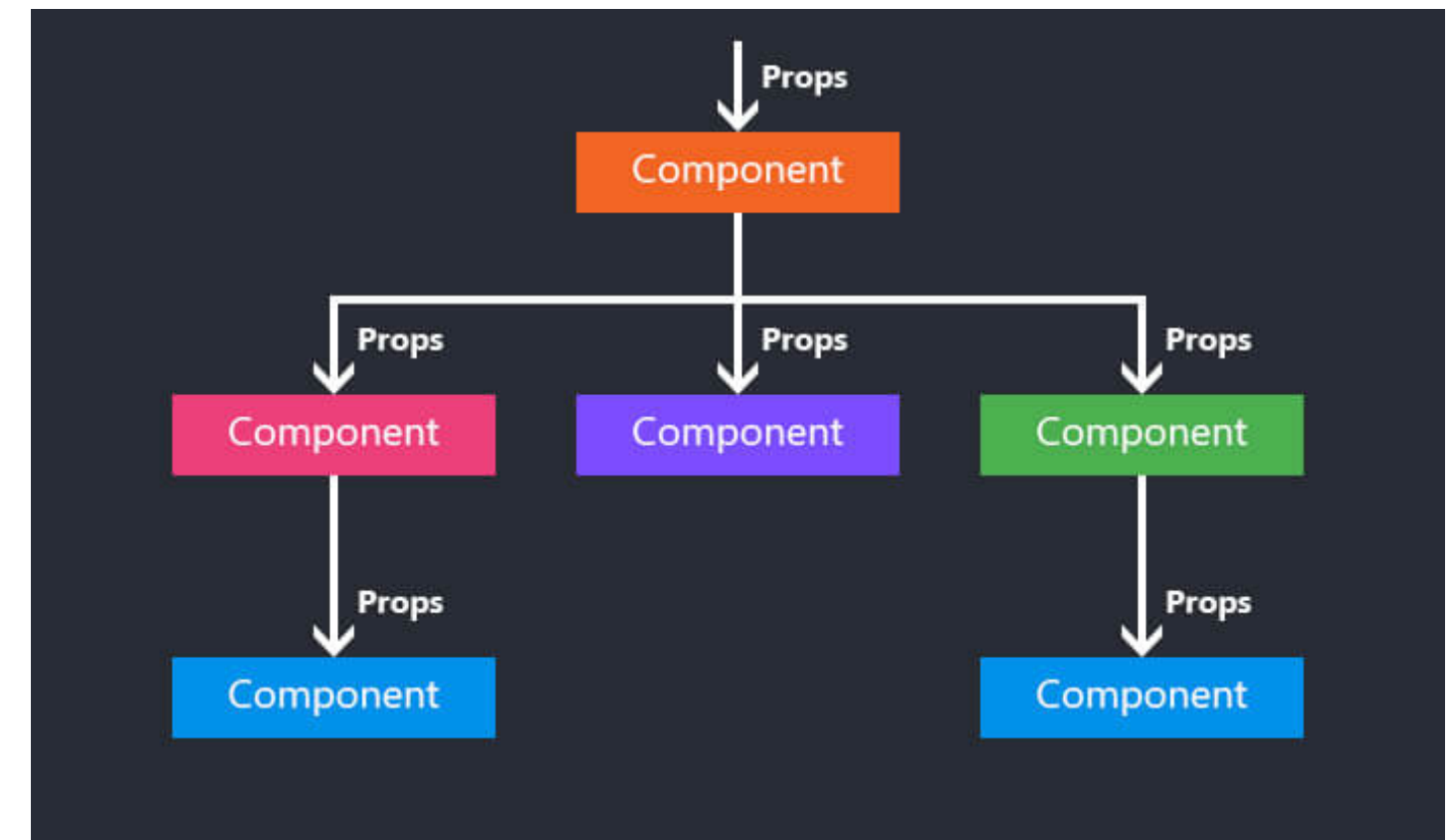
Inline
Styling

CSS
Modules



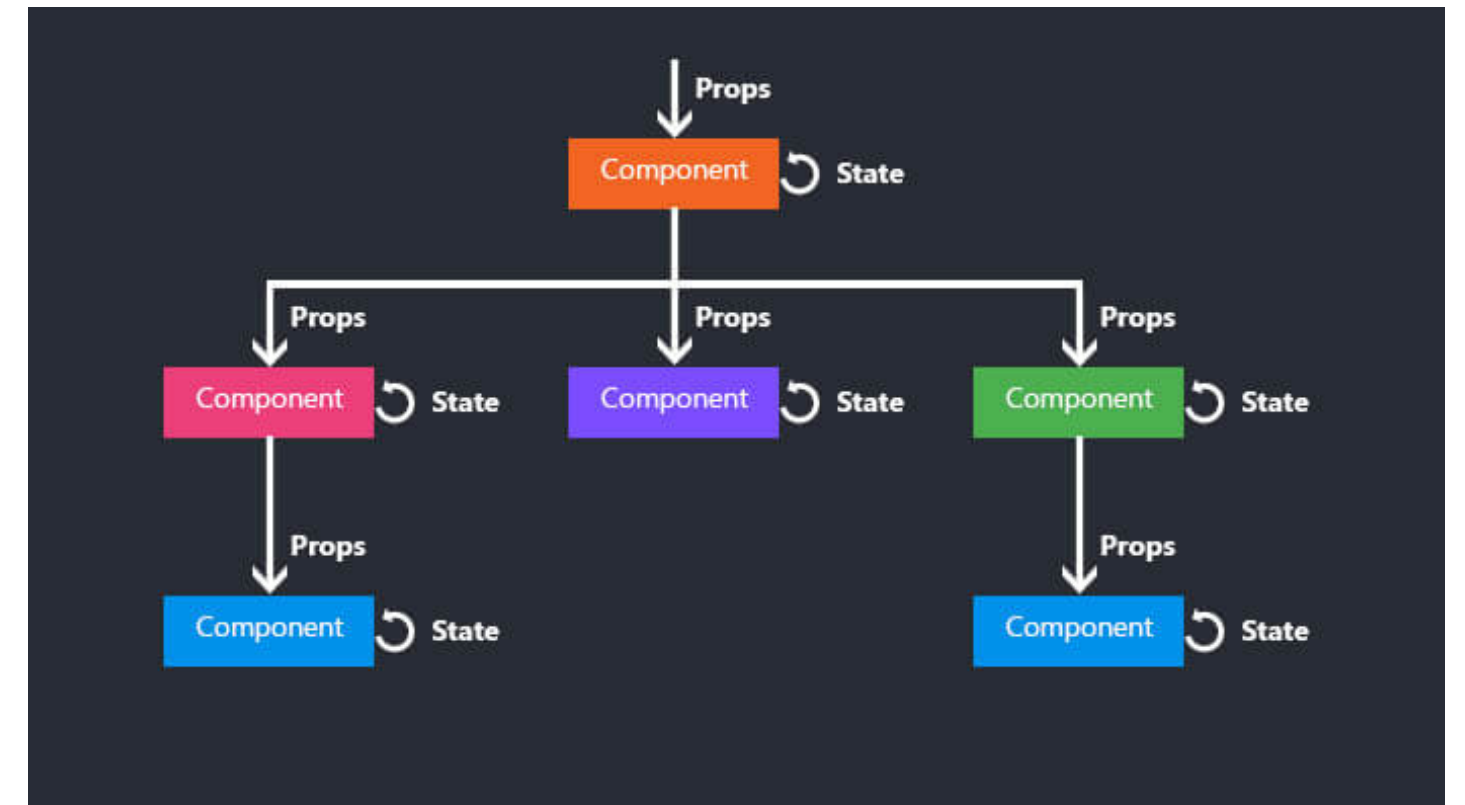
React Props

If you're already familiar with how arguments & functions work in JavaScript, understanding props is a piece of cake! In a nutshell, props are used to pass data from a parent component to a child component in React and they are the main mechanism for component communication.



React State

State is a plain JavaScript object used by React to represent an information about the component's current situation. It's managed in the component (just like any variable declared in a function).

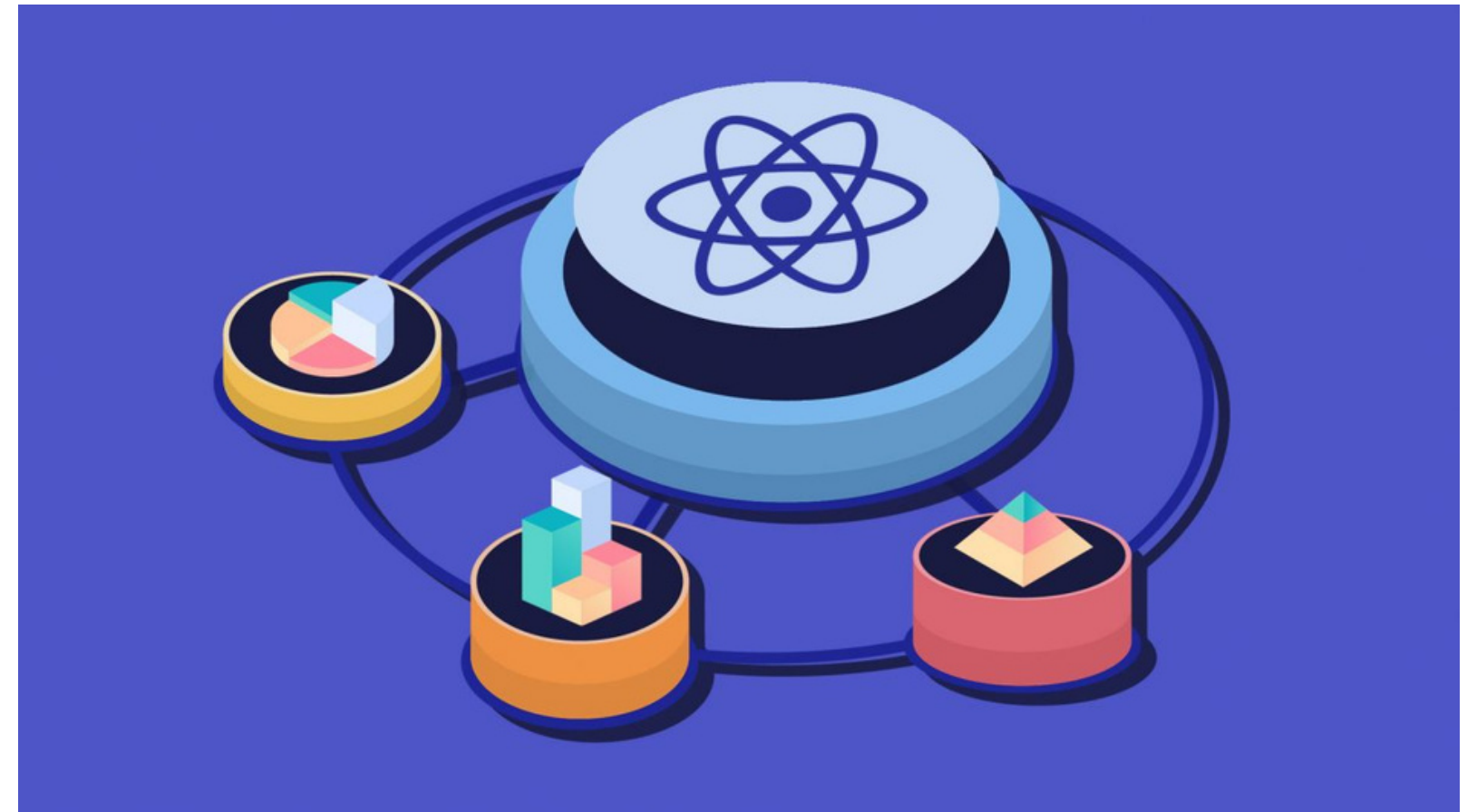


Props vs State

- Props are immutable i.e. once set the props cannot be changed, while State holds data that may change over time and to control the behaviour after each change.
- States can be used in Class Components, Functional components with the use of React Hooks (useState and other methods) while Props don't have this limitation.
- While Props are set by the parent component, State is generally updated by event handlers.

Conditional Rendering

- Ternary Operator
- Logical && Operator



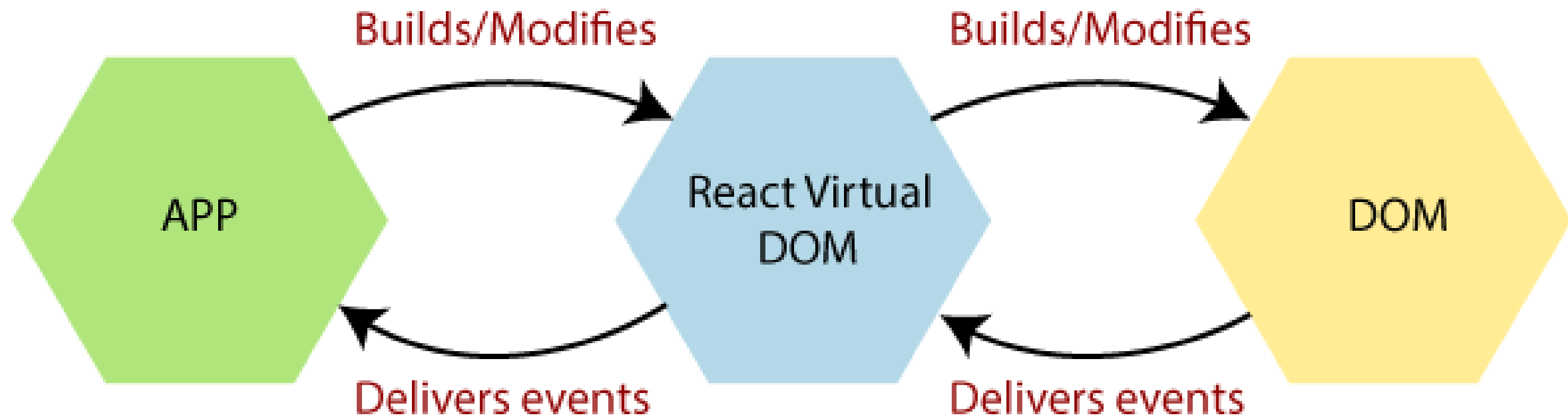
Lists and Keys

- Use Array.map method
- Not a for loop
- Give each item a unique key
- Avoid using array index as the key



Lists and Keys

Events Handling



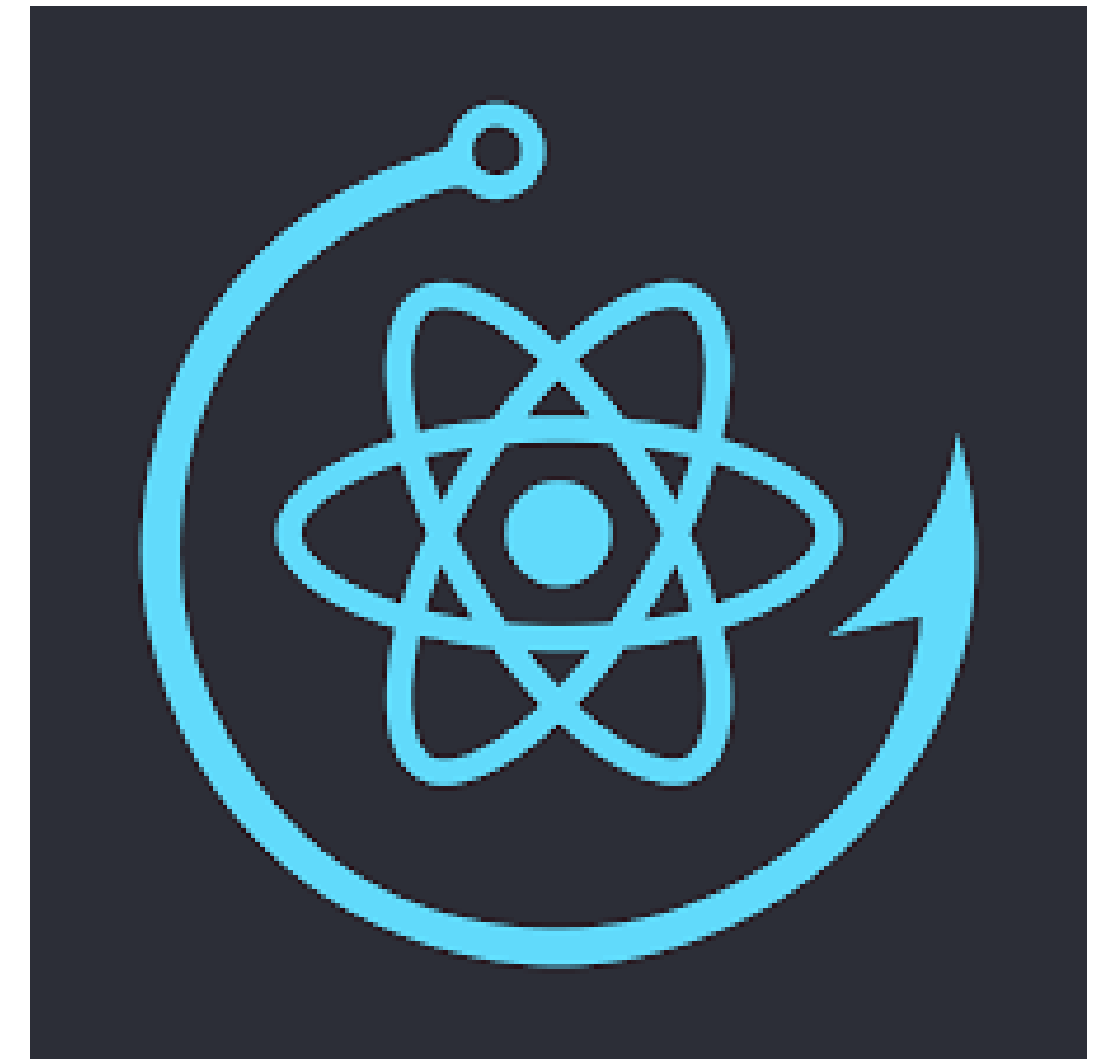
React Hooks



- **Core Hooks**
- **Stateful Functional Components with Hooks**
- **Custom Hooks**

What are Hooks ?

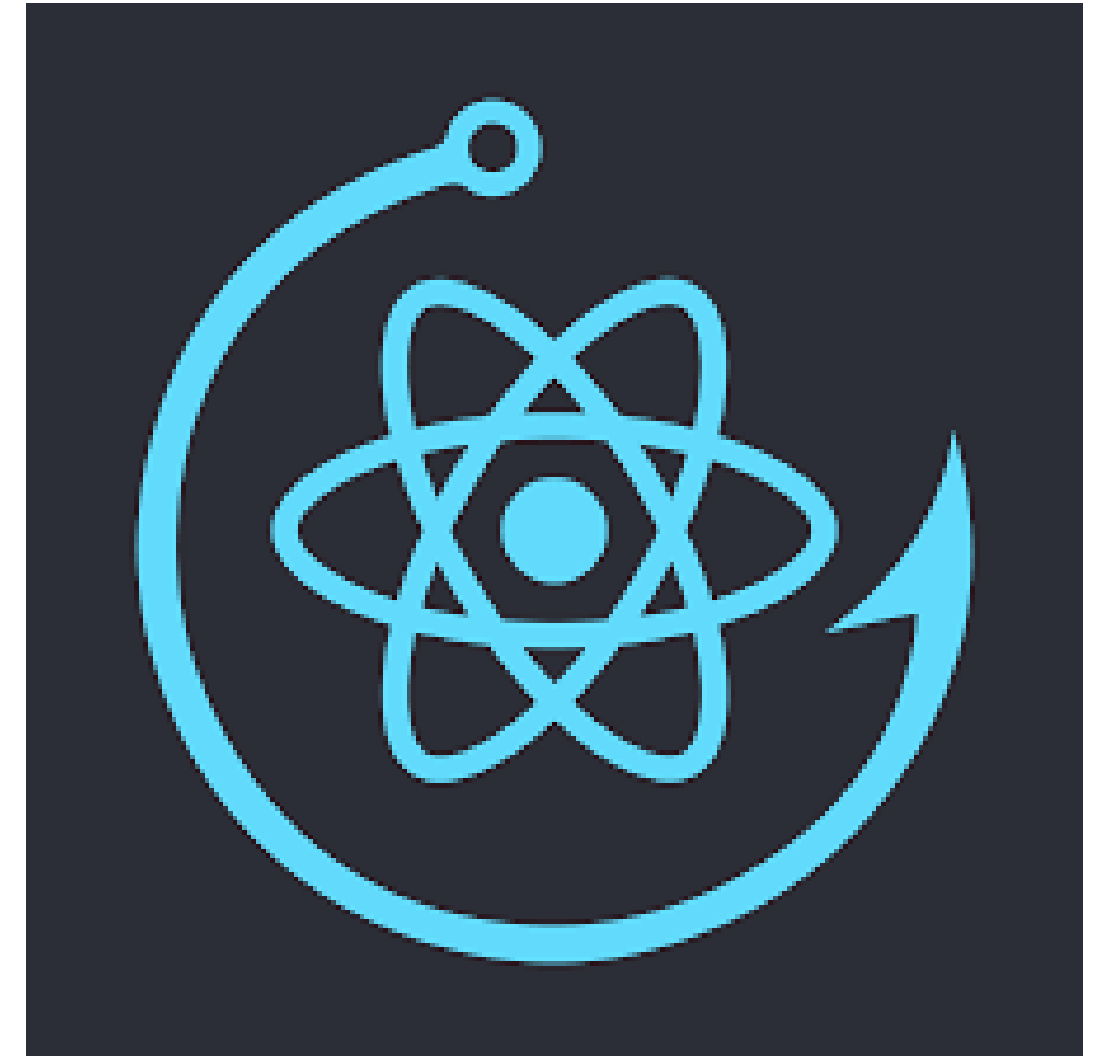
Hooks are a new feature introduced in the React 16.8 version. It allows you to use state and other React features without writing a class... Hooks are backward-compatible, which means it does not contain any breaking changes. Also, it does not replace your knowledge of React concepts.



Core Hooks

- useState
- useEffect

Let's build a stateful functional component using these two hooks!



Custom Hooks

useFetch, useDarkMode...

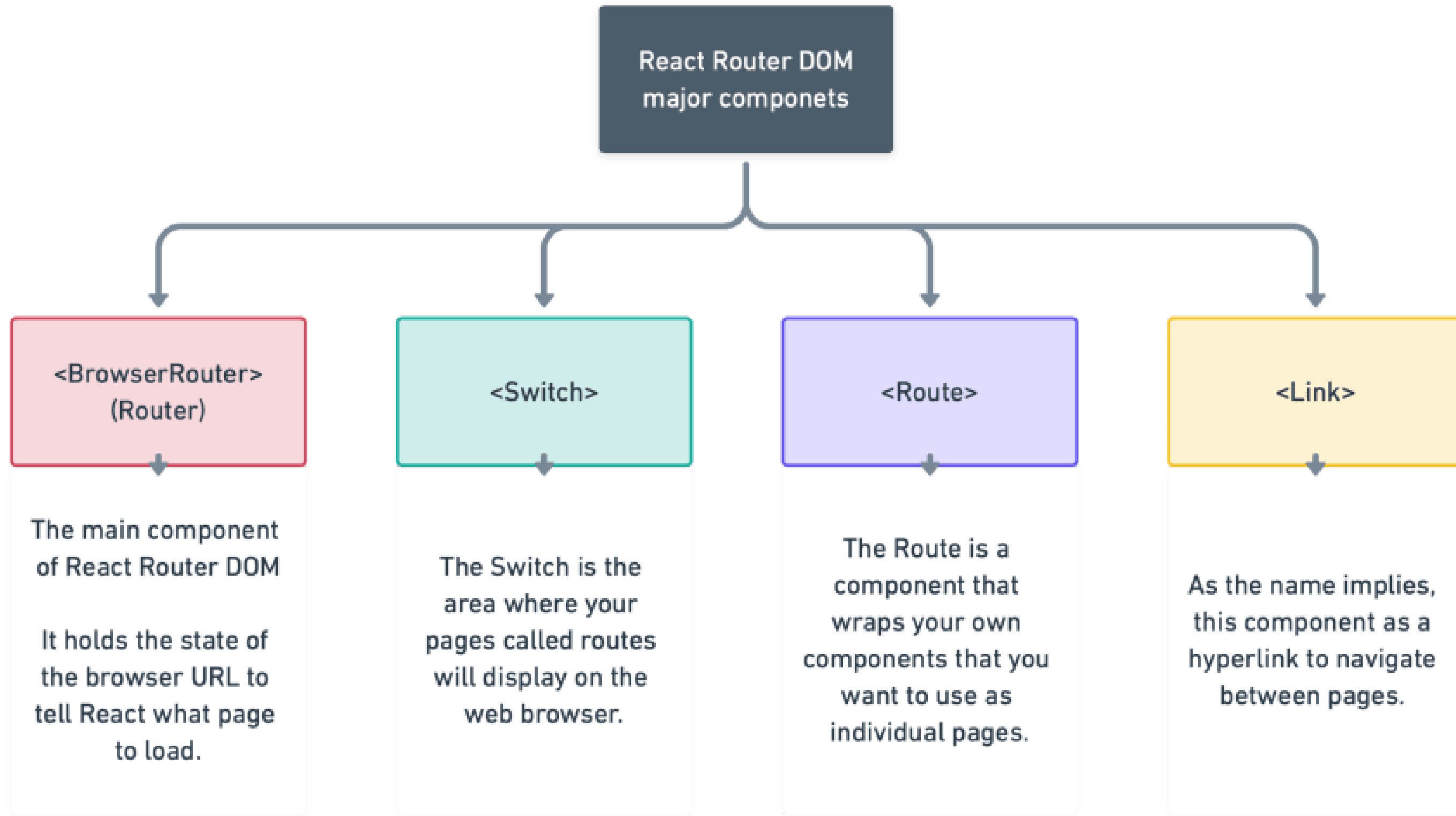


Routing with React Router



- **React Router Concepts**
- **Creating Application Routes**
- **React Router Hooks**

React Router Concepts



React Router Hooks

- `useNavigate`
- `useParams`
- `useLocation`

