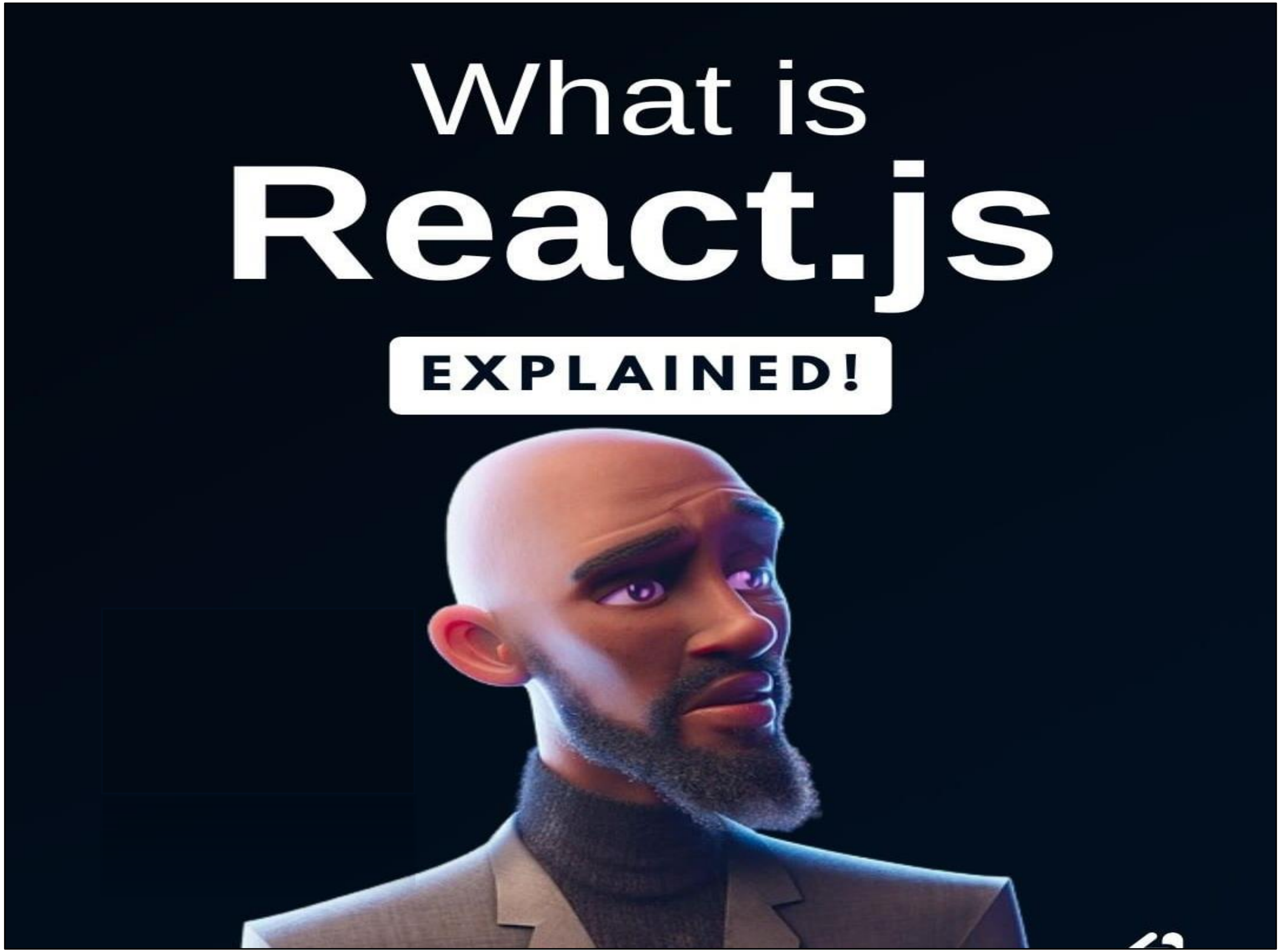
React JS

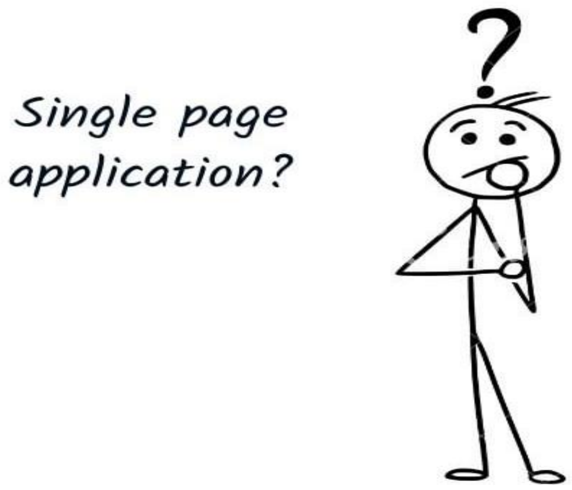
In learning you will teach, and in teaching you will learn.

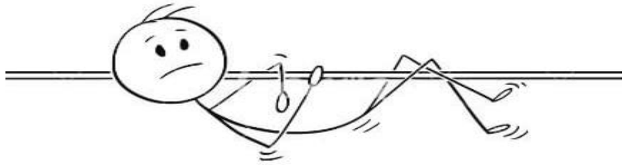
- Phil Collins



What is
React.js
EXPLAINED!

React.JS is an open source JavaScript library which is used for building user interfaces particularly for **single page applications**.



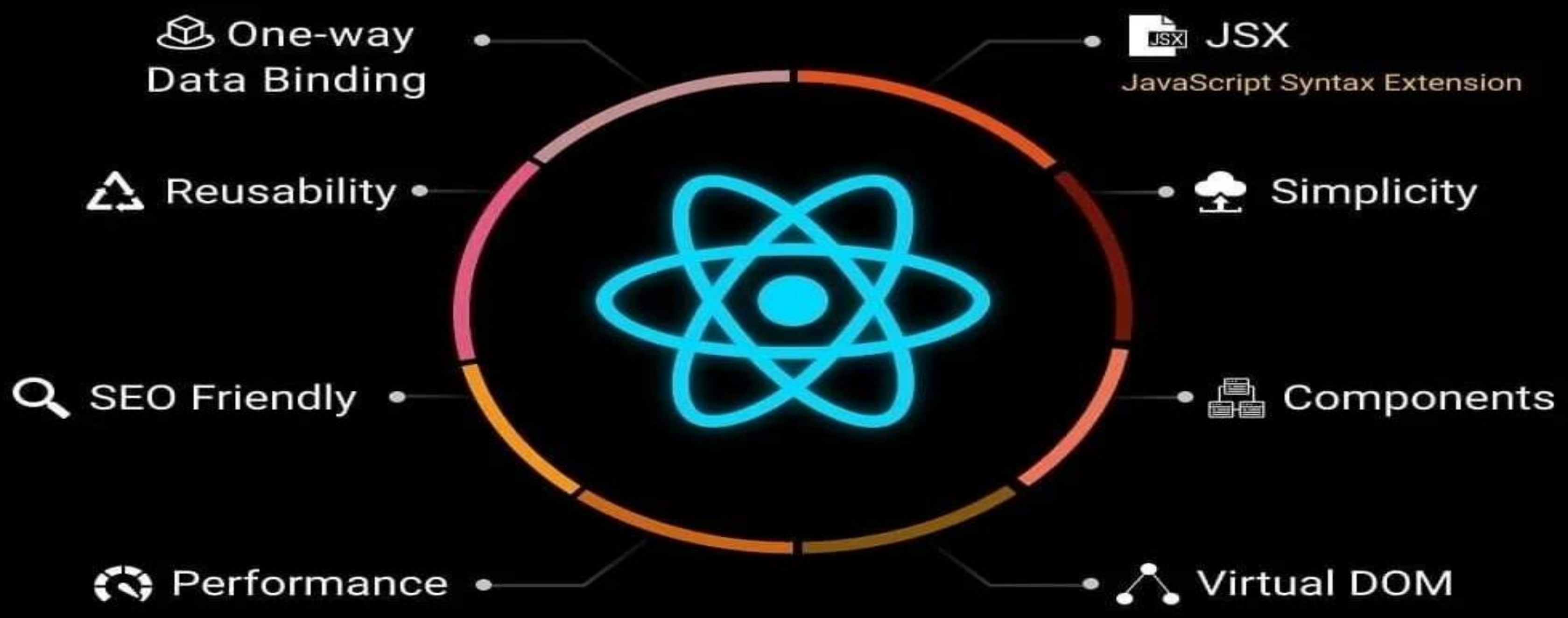


Single Page Application

It is a web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server.

React JS is not a framework. It is just a library developed by Facebook to solve some problems that we are facing in making of UI components.

Features of ReactJs



One-way Data Binding

Reusability

SEO Friendly

Performance

JSX

JavaScript Syntax Extension

Simplicity

Components

Virtual DOM

Introduction

- React is a JavaScript library for building user interfaces.
- It's 'V' in MVC. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application.
- React is created by Facebook in 2011.
- React is used to build single page applications.

ReactJS Virtual DOM

- React creates a VIRTUAL DOM in memory.
- Manipulating Real DOM is much slower than manipulating virtual DOM because nothing gets drawn on the screen.
- When the state of an object changes, Virtual DOM changes only that object in the real DOM instead of updating all the objects. Virtual DOM makes React extremely powerful and efficient.
- React only changes what needs to be changed!.

WHY IS REACT SO POPULAR?

Reusability

components can be used across multiple UIs

HTML + JavaScript = JSX

The Props: easy data flow

The State

Virtual DOM

Boosts performance

eliminates other code-heavy frameworks and libraries

Simple to migrate

... and meshes well with other technologies

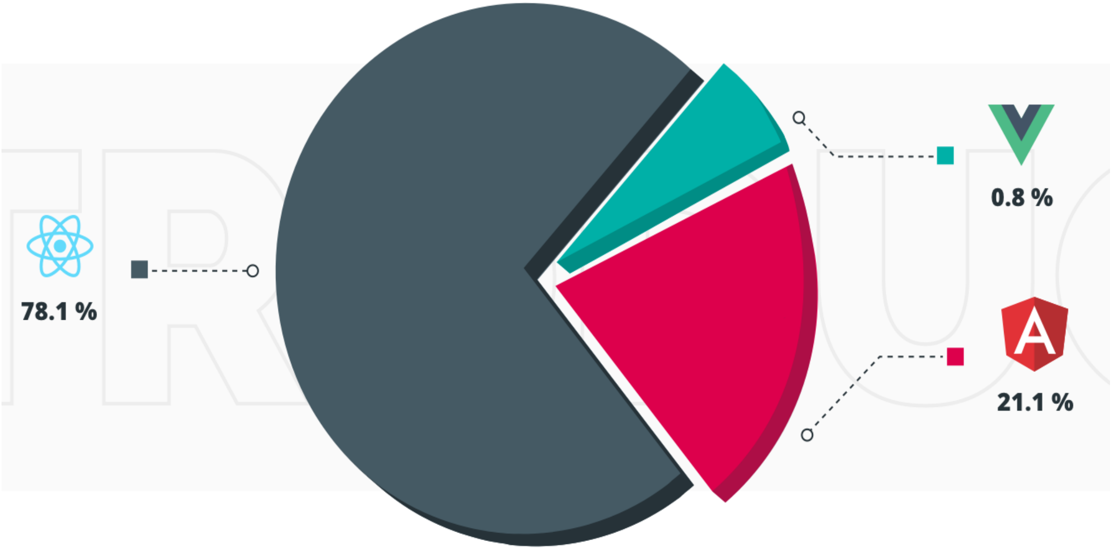
SEO-friendly

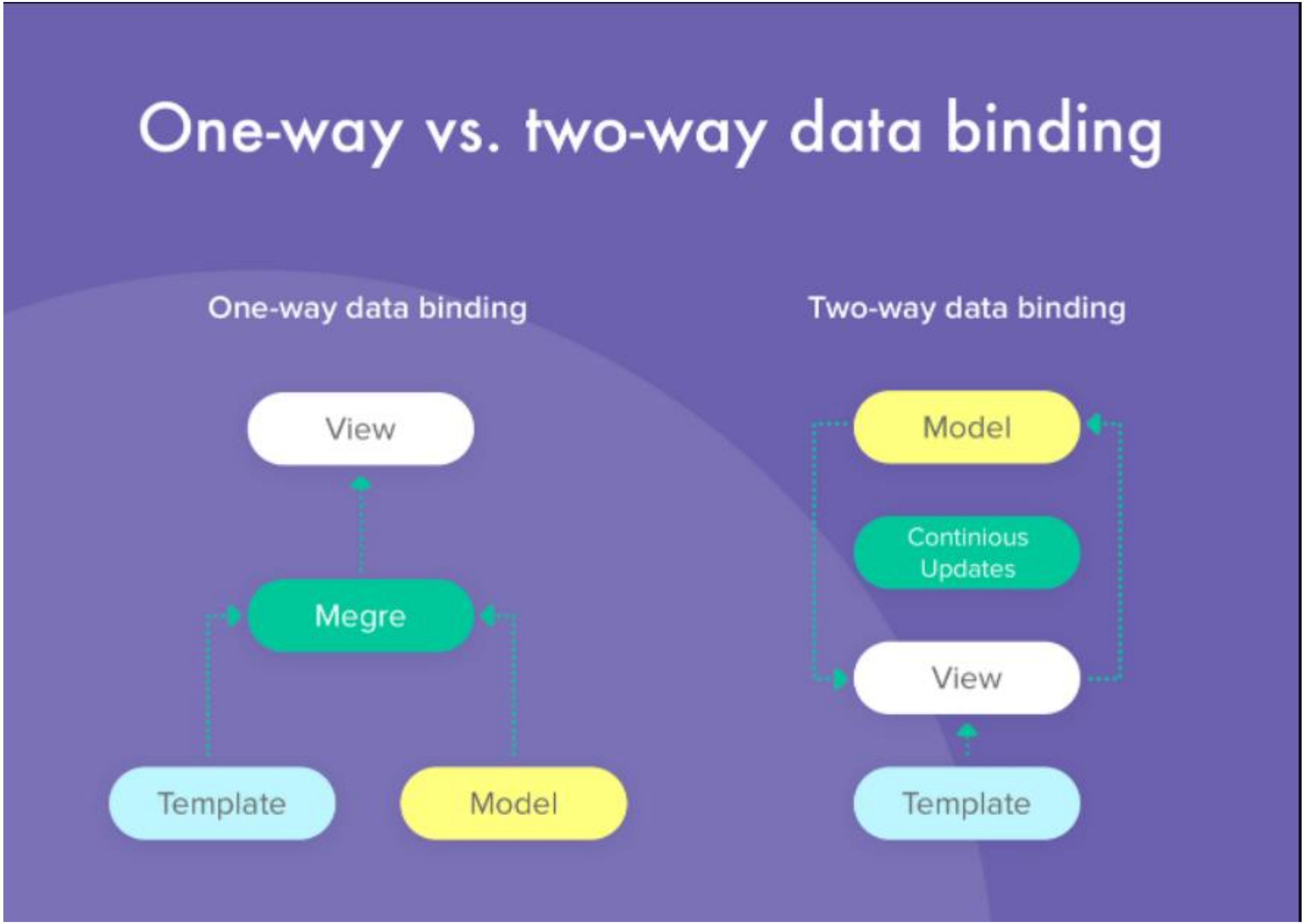
Easier debugging

Testing is fairly smooth

Why React is better than Angular?

- React beats Angular in terms of performance, full backward compatibility, and smaller bundle size.
- The component-driven architecture of React lets developers re-use components, saves development time.
- React.js apps use virtual DOM that allows developers to virtually update changes without rewriting the entire HTML doc. It renders updates faster and ensures faster performance.





One-way vs. two-way data binding

One-way data binding

View

Megre

Template

Model

Two-way data binding

Model

Continious Updates

View

Template

Bundle Size

- Angular solves this problem by not including the entire bundle of Angular framework in the user's app. As part of the build process, it gets rid of:
    - Development utilities
    - Unused modules
- Then, minification and compression take place allowing the framework to perform extra optimizations.
- As for ReactJS, it's the use of webpack (splits your code into smaller parts), tree shaking, and dynamic imports that reduce the bundle size.
- Tree shaking (or dead-code elimination) means that unused modules won't be included in the bundle during the build process.
- With dynamic imports, the app loads the code needed initially--and load the rest on demand.

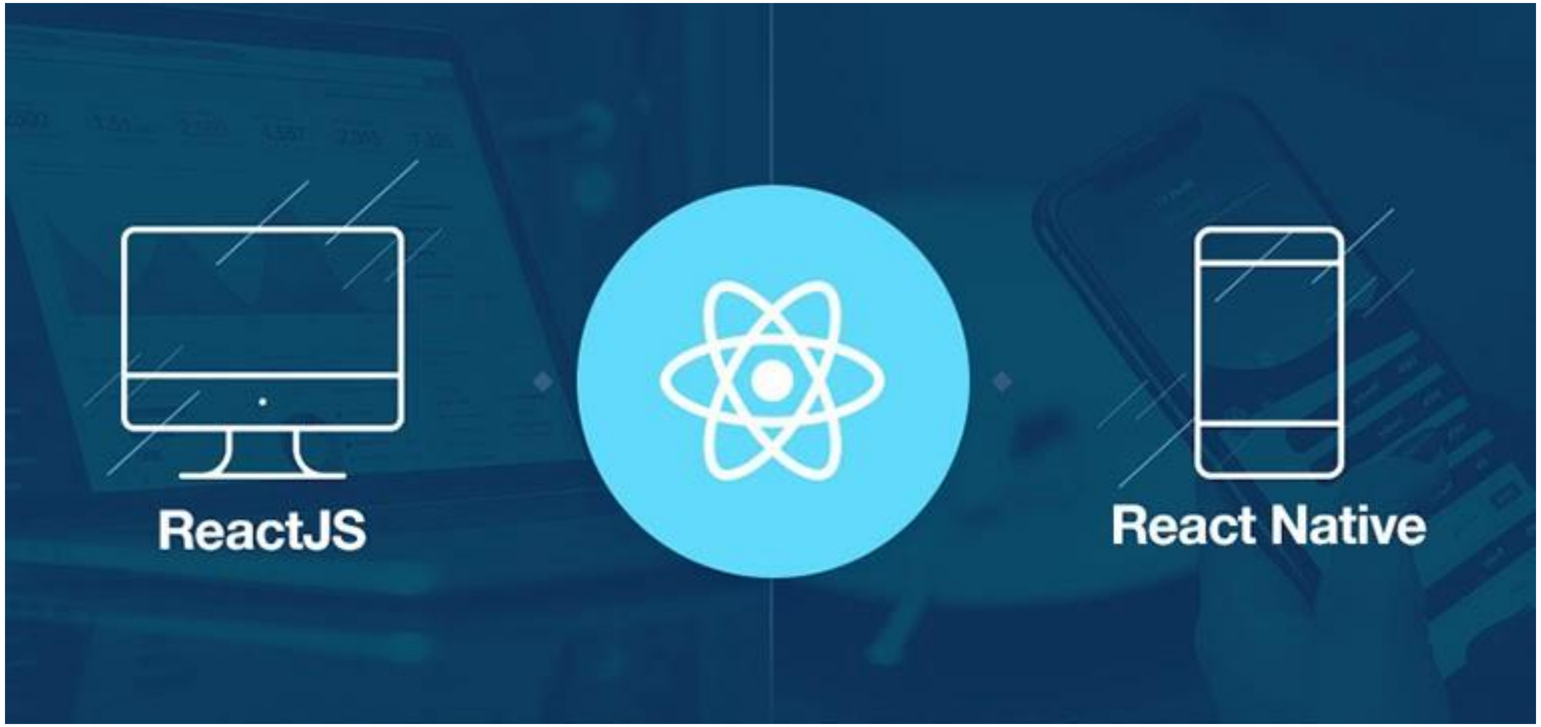
Angular vs. React performance tests

Performance time	Angular	React
Loading	10ms	7ms
Scripting	173ms	102ms
Rendering	3ms	6ms
Painting	2ms	4ms
System	73ms	129ms
Idle	3034ms	3042ms
Total	3295ms	3289ms

Angular 2 vs React: full comparison

Parameter	Angular	React
Type	Framework	Library
Company	Google	Facebook
Language	TypeScript	JavaScript
Architecture	Component-based	Component-based
Data Binding	Two-way	One-way
DOM	Real DOM	Virtual DOM
Scalability	Scalable	Scalable
Compatibility	Need to install updates	Full backward compatibility

React Js Vs React Native



React = Vanilla JS + ES6 + HTML + CSS = JSX = Web Apps(Front end)

React-Native = React (Vanilla JS + ES6 + Bridge between JS and Native code) + Native(iOS, Android) = Mobile Apps(Android, iOS, also supported web but have some limitations)

- React
- React Features
- Why React
- Features Explained
- Angular vs React
- React JS vs React Native

Upcoming Topics

- **How to Create React App**
- **Environment Set Up**
- **REACT ES6 & its Features**
- **React JSX**
- **React Components**
- **Props**
- **State**
- **React Life Cycle**
- **React Phase's**
- **React Event Handling**
- **React CSS**
- **React Fragments**
- **React Sass**
- **React Keys**
- **Hooks Concept**

QUERIES