COS30049 – Computing Technology Innovation Project

Week 8 - Introduction to Web Development & Front-End

(Lecture – 01)

Ningran Li (Icey)
ningranli@swin.edu.au

Acknowledgement of Country

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campuses are located in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures, and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.

Objectives of Today

Key Components of Web Application Architecture

- Presentation Layer
- Application Layer
- Data Layer

Core Foundations of Front End

- HTML
- CSS
- JavaScript

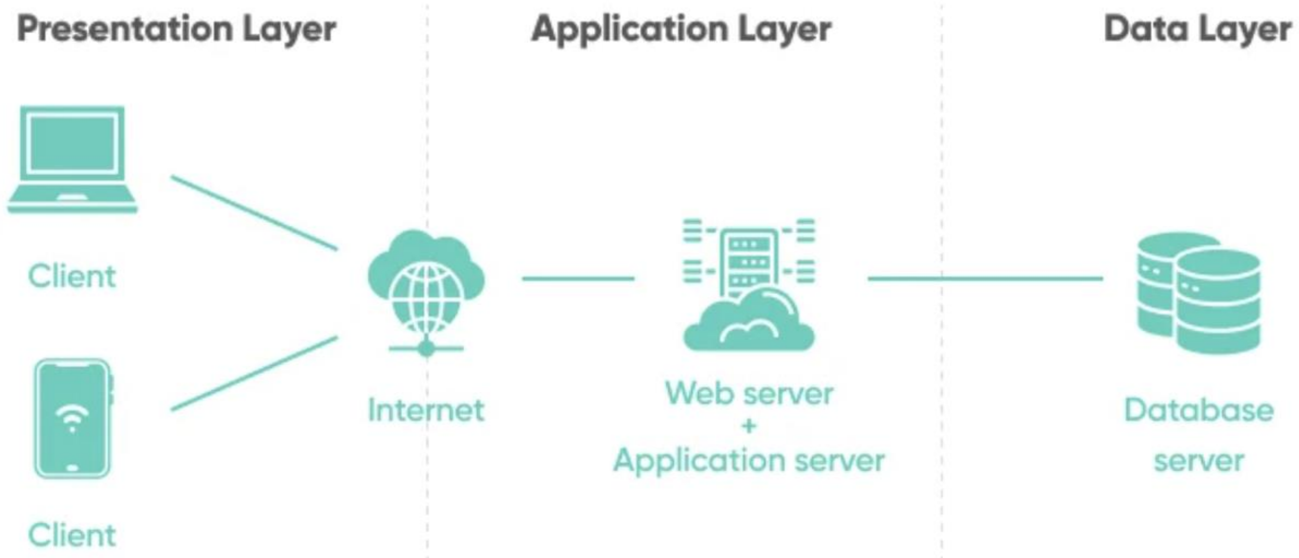
React and Node.js

- What’s this?
- React Basics (Seminar)

Key Components of Web Application Architecture

Web Application Architecture

- Presentation Layer
  - ***User Interface*** (UI) of the web application
  - Visual aspects of the web app
  - Such as the design of UI, the layout of the screens, and the navigation
- Application Layer
  - The business logic layer
  - ***Logic and processes*** that the web application needs to perform
  - Including the processing of user input, data manipulation, and the execution of business rules
  - One or multiple web servers live within the application layer
- Data Layer (not compulsory for this unit)
  - Persistent ***data Storage*** and the retrieval of data
  - Such as database (DB)
  - Relational/Non-Relational DB

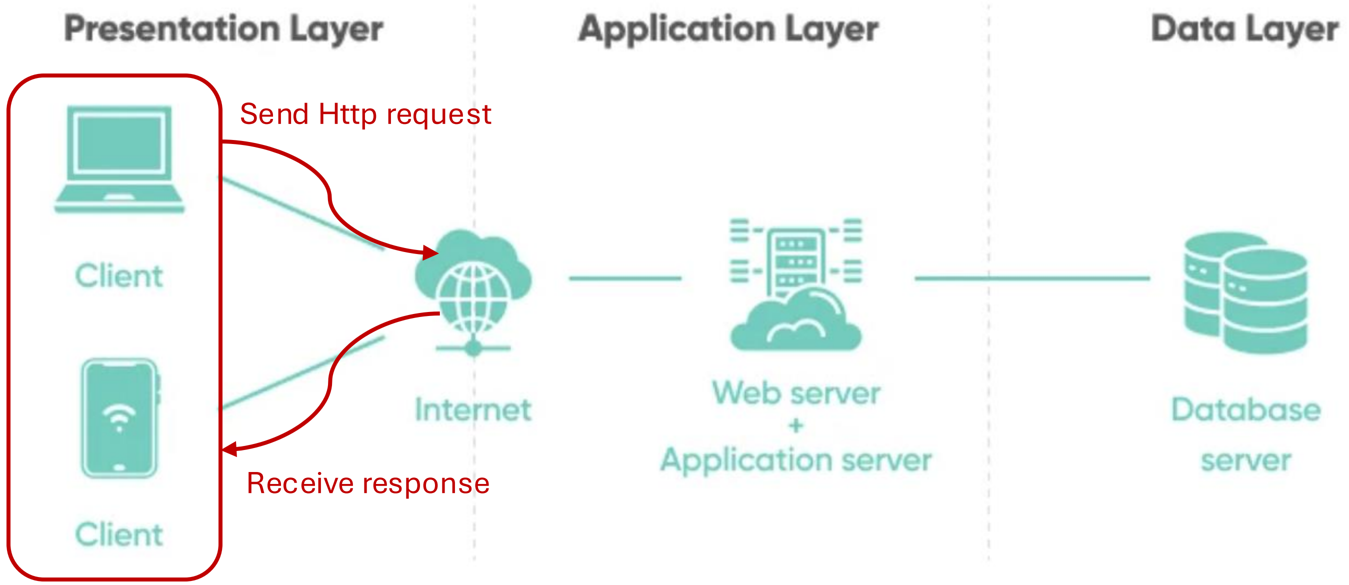


Web Application Architecture

- **Presentation Layer**

It is part of the application that interacts directly with the user. In modern web development, this is known as the **front end**. It is responsible for displaying the interface and collecting input from the user.

- **HTML, CSS, and JavaScript**: Core building blocks of the front end.
- **API Calls**: Communicating with the back end to fetch and display data.
- **Responsive Design**: Ensures that the user interface adapts to different screen sizes and devices
- **Performance Optimization**: Optimizes the speed and performance of the website, reducing load times
- **User Experience and Accessibility**: Ensures the UI is user-friendly, intuitive, and accessible to people of all abilities



Client-Side Rendering(CSR)

- User requests a webpage
  - ➢ The user requests a webpage from the browser
- A server receives the request
  - ➢ The user's action triggers a **GET** request to the server
- The server sends a minimal HTML page with links to CSS and JavaScript files
- The ***browser parses the HTML*** and constructs the Document Object Model (DOM) tree, which represents the webpage's structure
- Browser downloads CSS and JavaScript
  - ➢ The ***browser sends additional request*** to the server
  - ➢ Download the CSS and JavaScript resources
- Render the page
- JavaScript execution
- Re-rendering and updating
- Final display

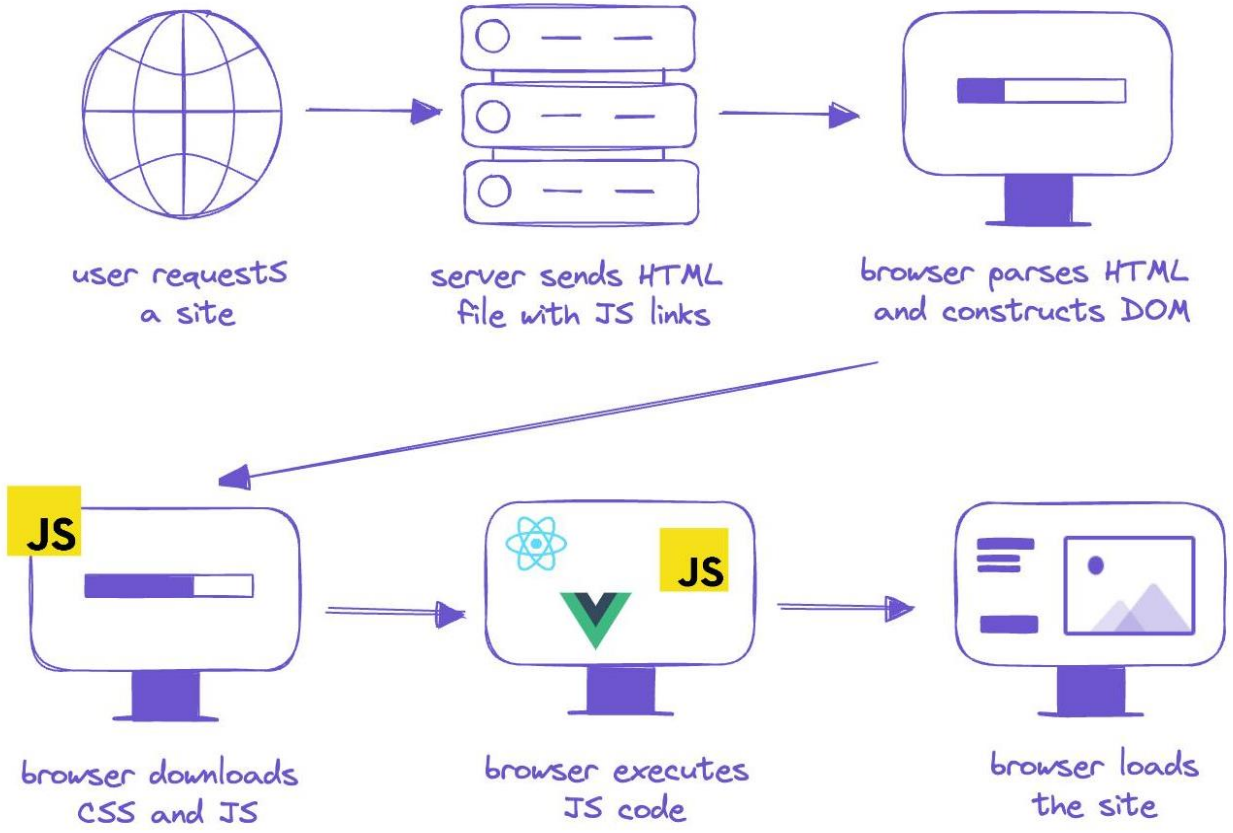

user requests a site
server sends HTML file with JS links
browser parses HTML and constructs DOM
JS
browser downloads CSS and JS
JS
browser executes JS code
browser loads the site

Image from [Prismic]

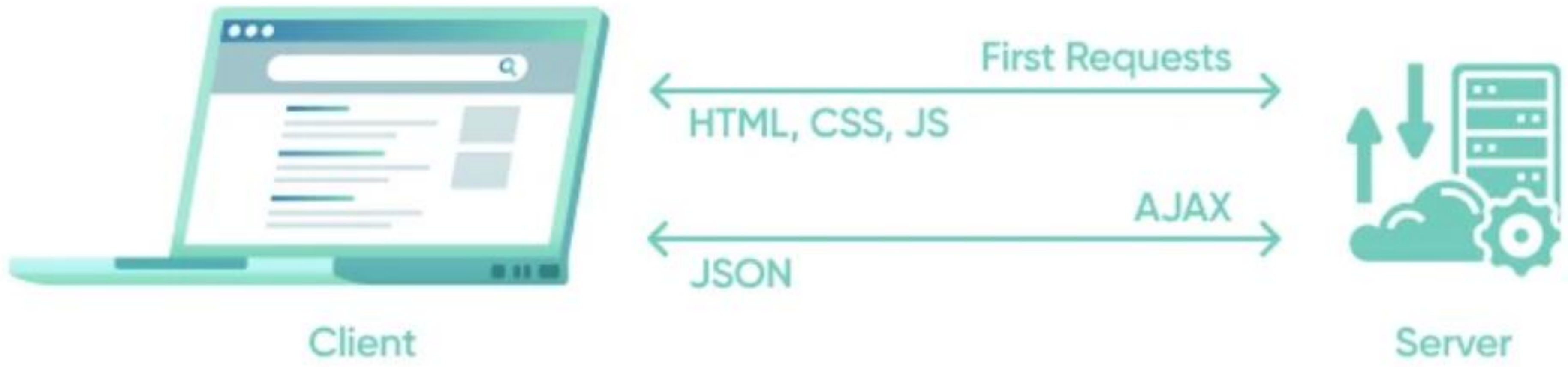


Image from [Soft Kraft]

Single-page Application (SPA)

- A web app that interacts with the user
  - ➢ By ***dynamically rewriting*** the current page
  - ➢ Rather than loading entire new pages from the server
  - ➢ The page is updated in real-time
  - ➢ Is designed to provide a ***smoother, more responsive user experience***

Server-Side Rendering(SSR)

- A user visits a URL via the browser
- The server receives a request for the page
  - ➢ This action sends a request to the server
- Fetching data
  - ➢ The server retrieves the data from DB, external APIs, or other relevant sources.
  - ➢ This data will be used to populate the HTML template
- Template rendering:
  - ➢ The server generates the HTML markup
  - ➢ using a templating engine or a framework
  - ➢ That combines the fetched data with the predefined structure of the page
- Sending the rendered page
- Rehydration
  - ➢ The server also sends the JavaScript required to handle client-side interactivity and dynamic behavior
- Repeat the rendering process

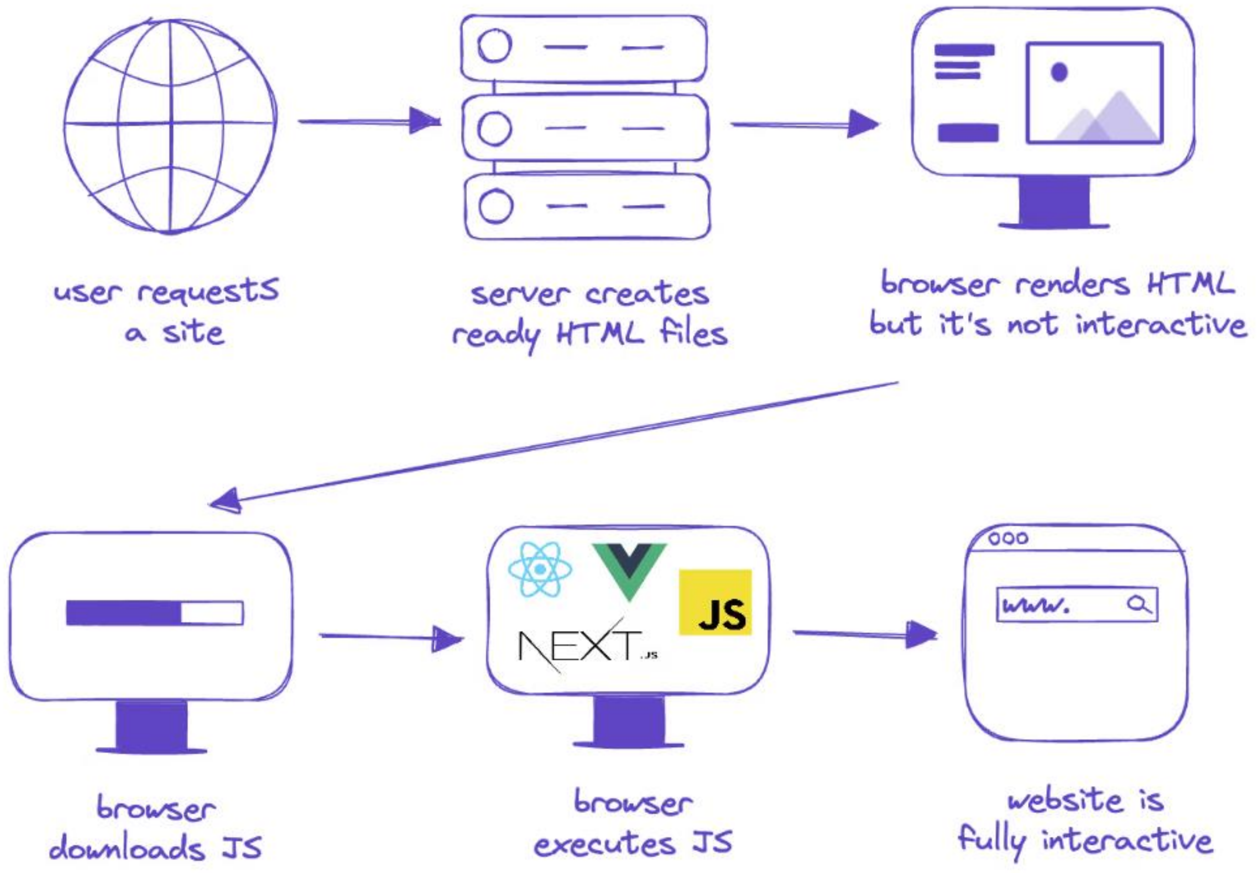

user requests a site
server creates ready HTML files
browser renders HTML but it's not interactive
browser downloads JS
NEXT.JS
JS
browser executes JS
www.
website is fully interactive

Image from [Prismic]

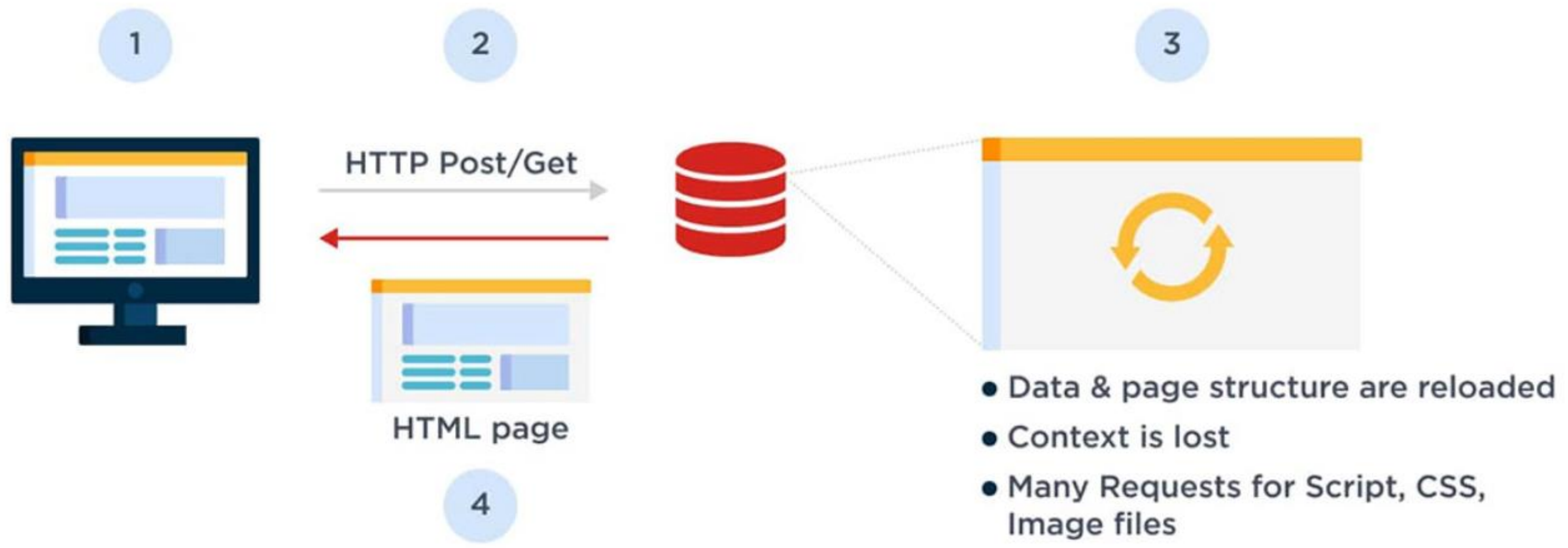
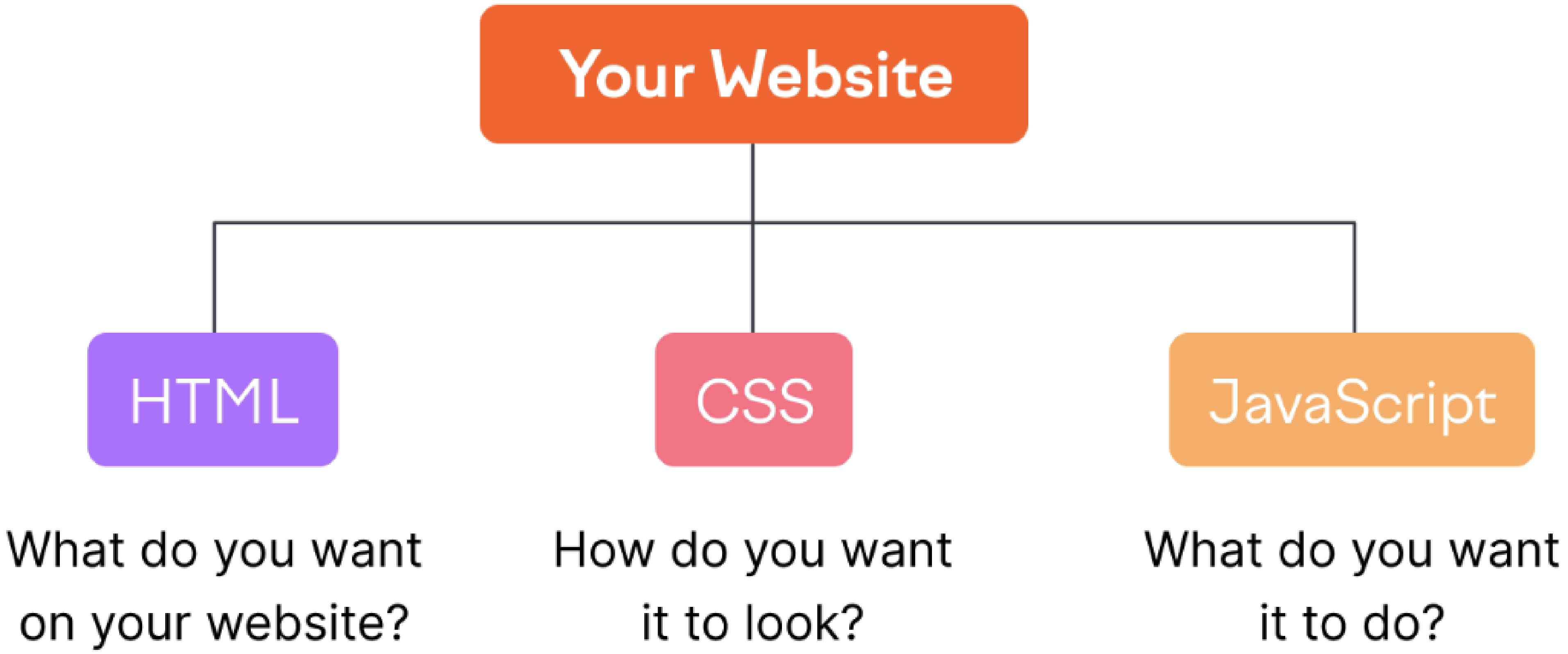

1
2
3
4
HTTP Post/Get
HTML page
● Data & page structure are reloaded
● Context is lost
● Many Requests for Script, CSS, Image files

Image from [Blog]

Multi-page Application (MPA)

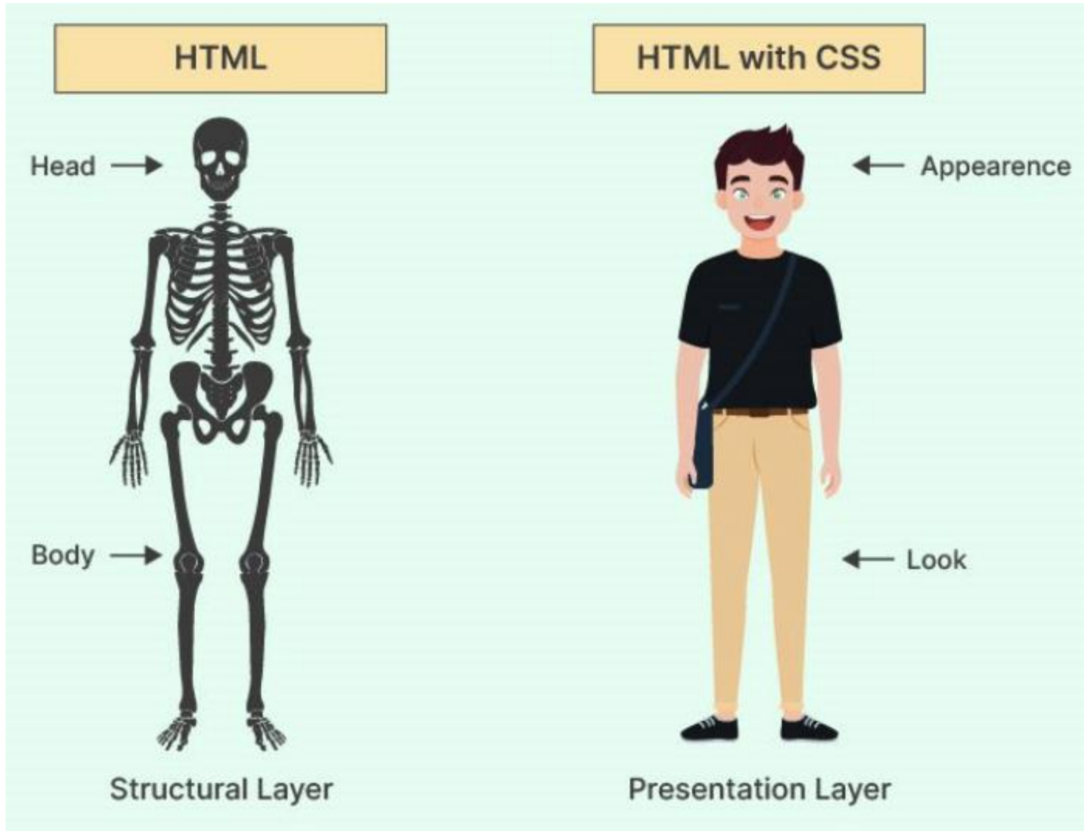
- Web app
  - ➢ Consisting of a large number of pages
  - ➢ Completely refreshed every time
  - ➢ When data changes on them
- Any data changes/transfer to the server
  - ➢ Leads to a new page displayed in the browser

The Core Foundations of Front-End Development: HTML, CSS, and JavaScript


Your Website
HTML
CSS
JavaScript
What do you want on your website?
How do you want it to look?
What do you want it to do?

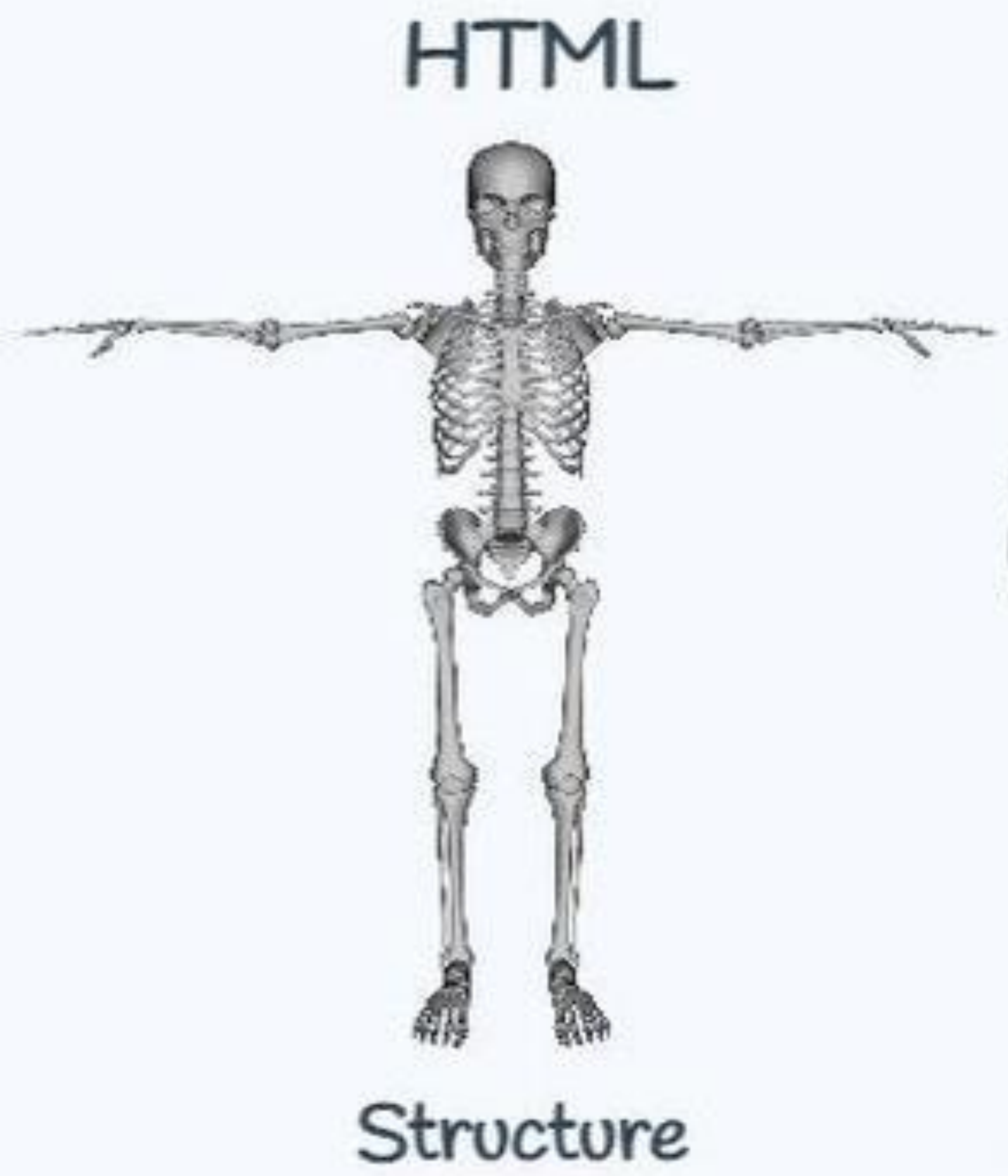
Fundamentals of Web Development

Image from [Semrush]



	HTML	CSS	JS
Language	**HTML**	**CSS**	**Javascript**
Purpose	Structure, Objects, Things	Looks, Style	Actions
Syntax	`<p> <h1>  `	P {color: red;}	var x = 5;
Grammar	nouns	adjectives	verbs
Building	Walls, structure	Paint, curtains	Electrical, Plumbing, AC

Fundamentals of Web Development



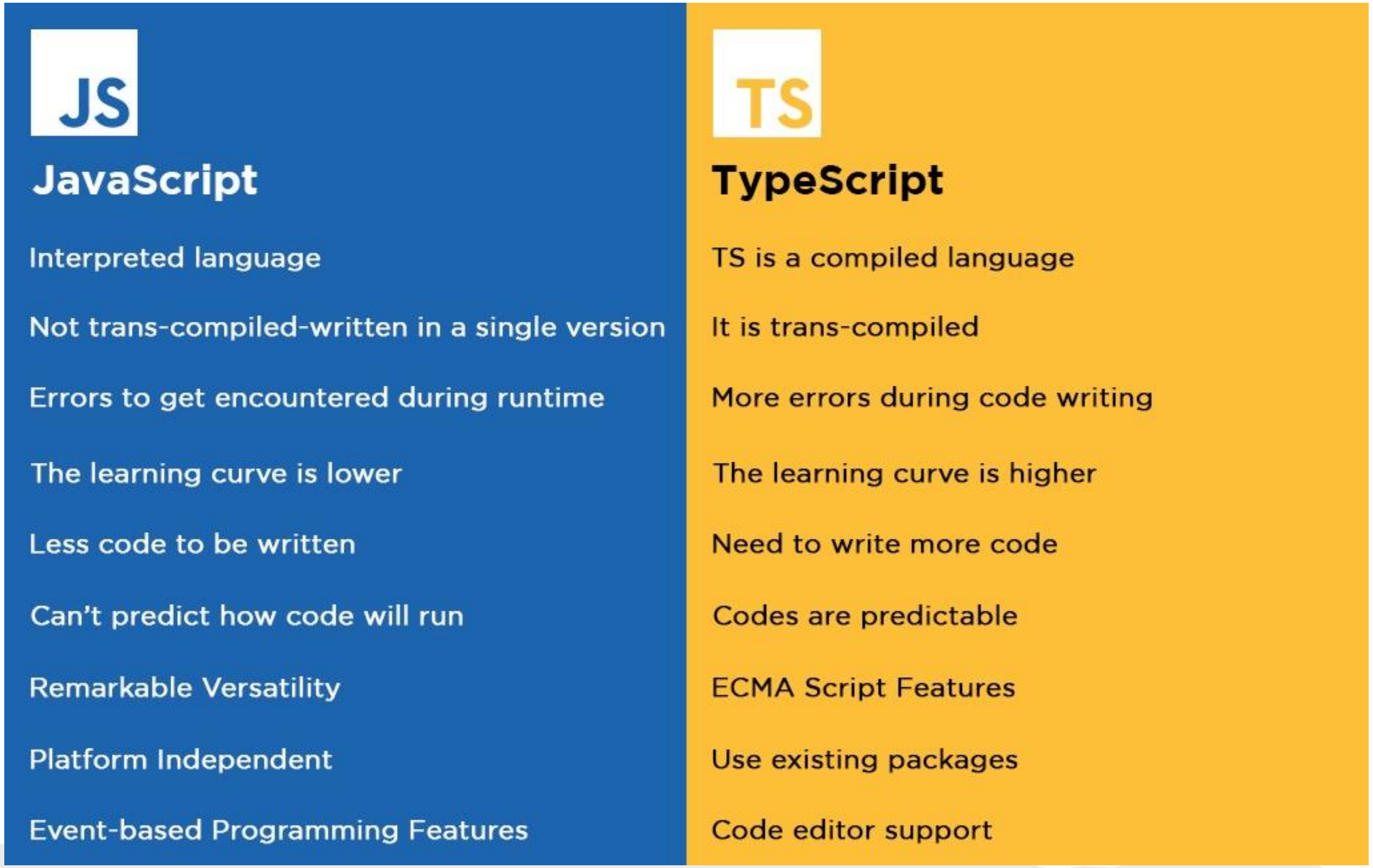
CSS
Presentation

JavaScript
Behavior

Image from [Pinterest]

What is JavaScript?

- A high-level, interpreted programming language
- Widely used for web development
- Client-side scripting language, meaning it runs in the user's web browser
- Dynamically-typed language, allowing variables to hold values of different types
- JavaScript frameworks and libraries like React, Angular, Vue.js have greatly enhanced its capabilities for building complex web applications



JavaScript	TypeScript
Interpreted language	TS is a compiled language
Not trans-compiled-written in a single version	It is trans-compiled
Errors to get encountered during runtime	More errors during code writing
The learning curve is lower	The learning curve is higher
Less code to be written	Need to write more code
Can't predict how code will run	Codes are predictable
Remarkable Versatility	ECMA Script Features
Platform Independent	Use existing packages
Event-based Programming Features	Code editor support

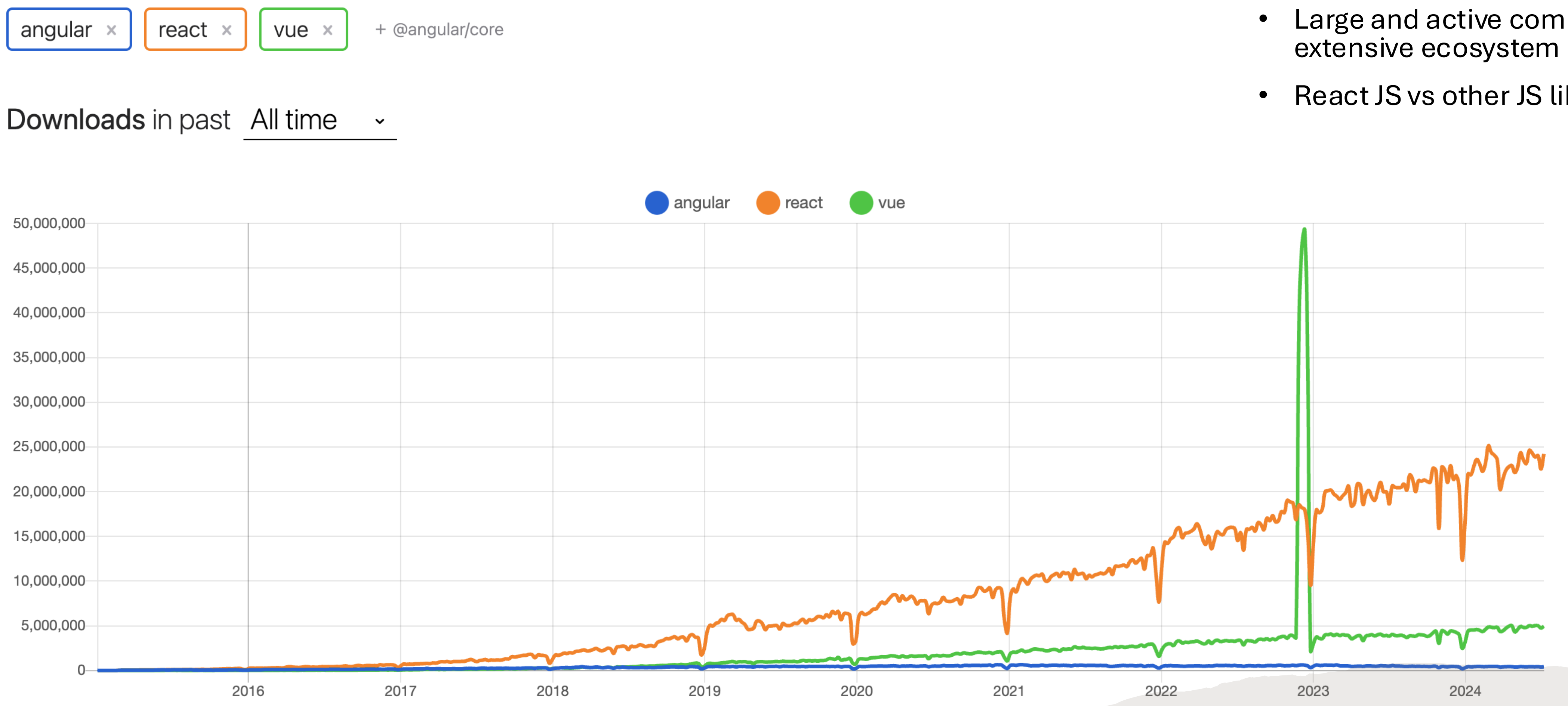
Image from [Ace Infoway]

Fundamentals of Web Development

React and Node.js: A Foundation for Modern Web Development

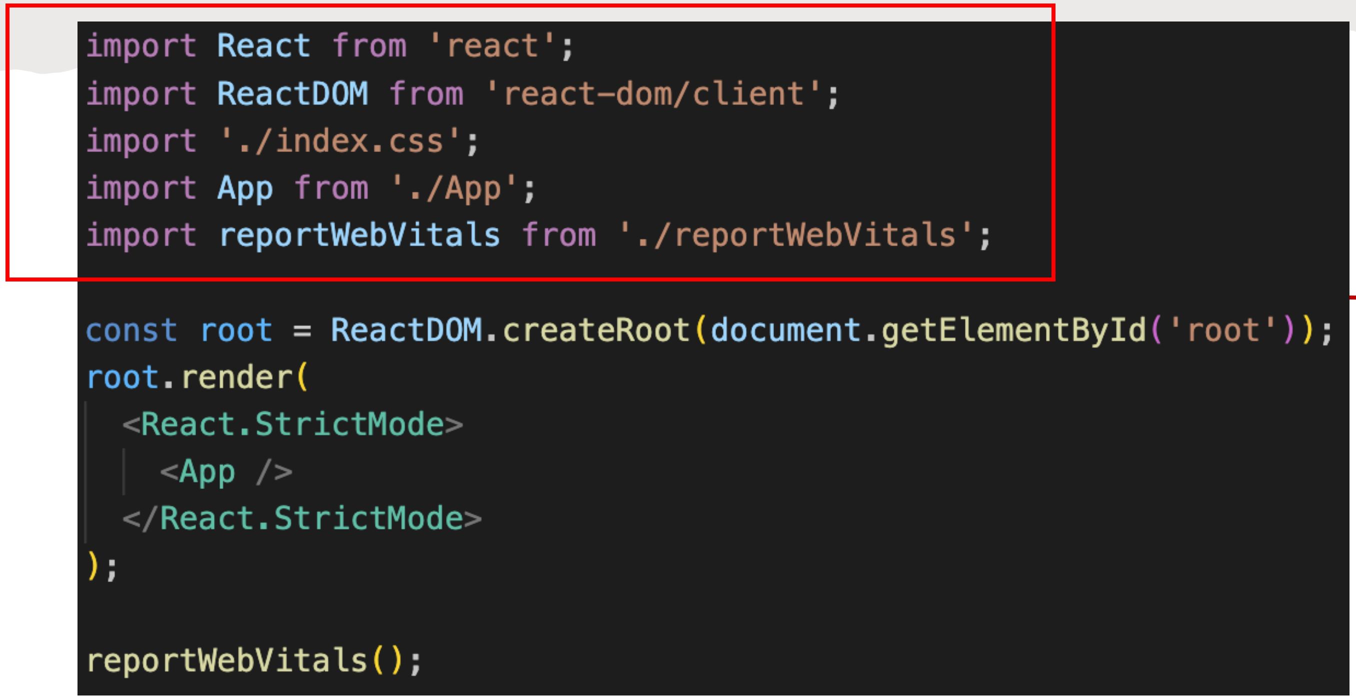
What is React?

- The best JS UI library for creating and maintaining views
- Reusable and composable components
- Large and active community with extensive ecosystem
- React JS vs other JS library



What is React?

- A JavaScript library for building user interfaces



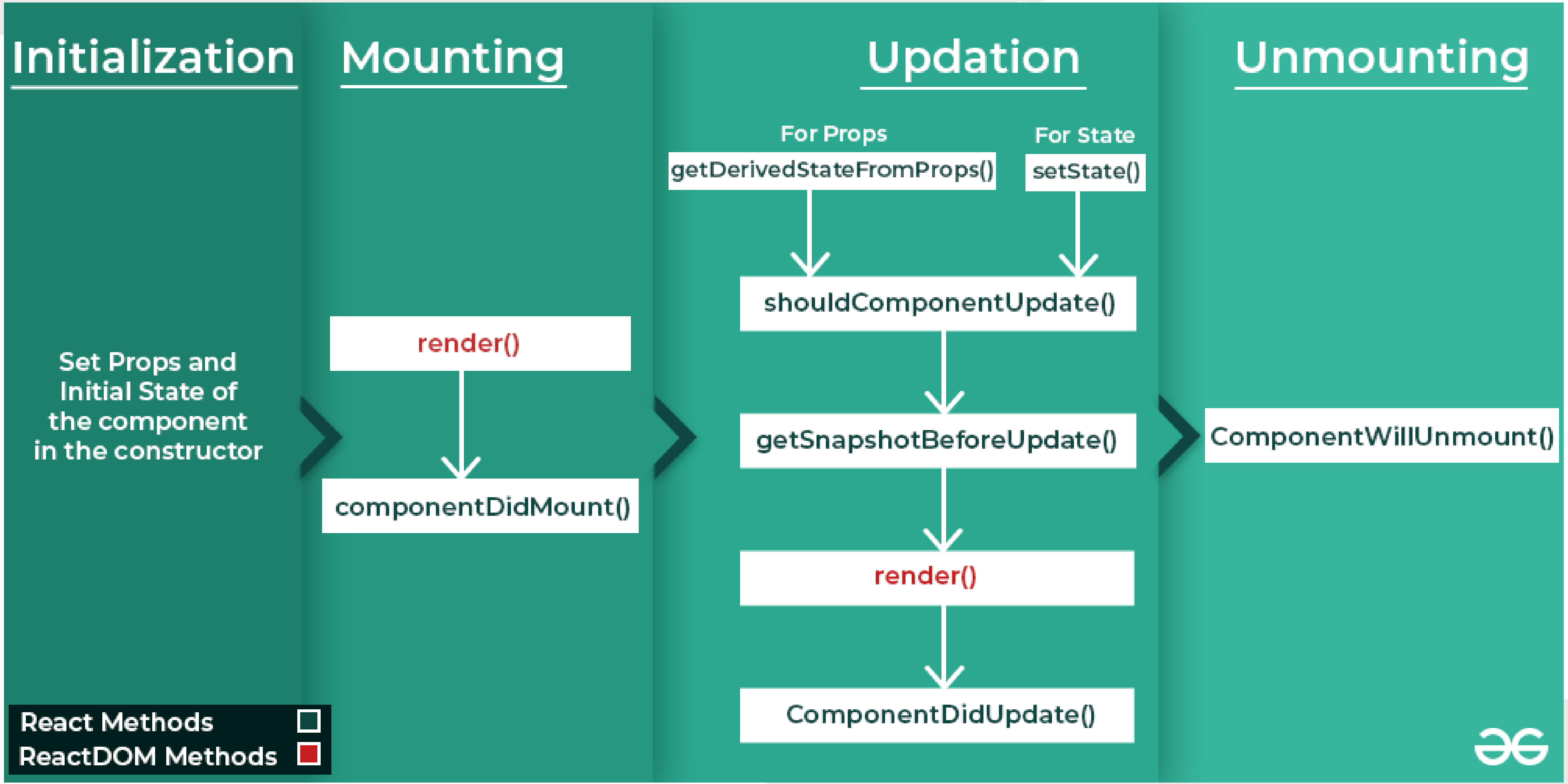
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

reportWebVitals();
```

Let's explore the details further in the seminar. ☺

- Efficient and declarative way to build complex UIs
- Dynamically generate web pages
- Developed and maintained by Facebook

React Lifecycle

- Is defined as the series of methods that are invoked in different stages of the component's existence
- Four stages: initialization, mounting, updating, unmounting



What is Node.js?

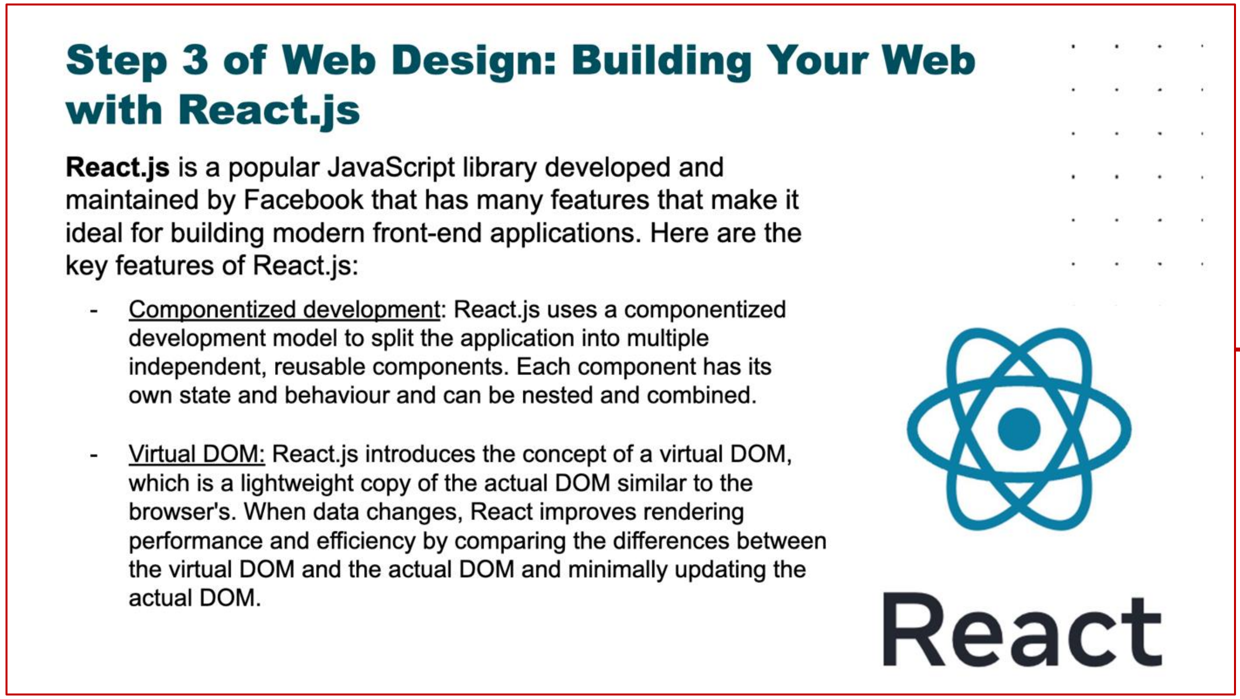
- A JavaScript runtime environment
- Allows running JavaScript code outside a web browser
- Enables server-side JavaScript execution, allowing developers to build scalable and high-performance web applications
- Has a vast ecosystem of packages and modules available through the npm (Node Package Manager) registry
- Commonly used for building web servers, APIs, real-time applications, command-line tools, and more

Image from [WikiPedia]

However, why do we need Node.js to use React?

Feature	React	Node.js
Purpose	Frontend library for building user interfaces	Backend runtime environment for server-side development and more
Usage	Used for creating interactive, component-based UI elements in web applications	Primarily used for server-side scripting and handling asynchronous requests
Language	JavaScript	JavaScript (for server-side applications)
Architecture	Component-based (UI components)	Event-driven, non-blocking I/O model
Main Focus	Building efficient, dynamic user interfaces	Building scalable network applications
Ecosystem	Focused on the View layer of the MVC model	Full runtime environment with various modules
Installation	Installed via 'npx' or 'npm'	Node.js needs to be installed to run 'npm' (Node Package Manager), which is required for installing and managing React
Rendering	Client-side rendering for a seamless user experience	Handles server-side processing, APIs, and database interactions
Why Needed for React?	Handles the frontend part of a web app	Needed to install and manage React packages via 'npm' and 'npx'

React VS Node.js



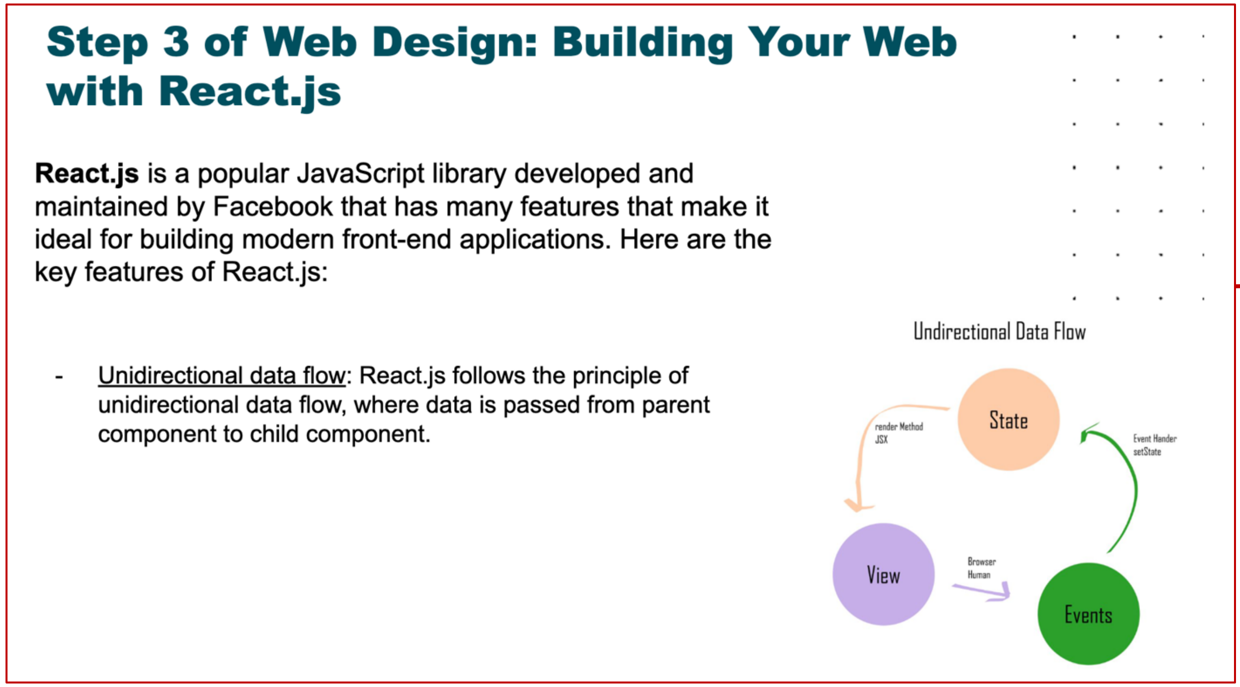
Step 3 of Web Design: Building Your Web with React.js

React.js is a popular JavaScript library developed and maintained by Facebook that has many features that make it ideal for building modern front-end applications. Here are the key features of React.js:

- <u>Componentized development</u>: React.js uses a componentized development model to split the application into multiple independent, reusable components. Each component has its own state and behaviour and can be nested and combined.

- <u>Virtual DOM:</u> React.js introduces the concept of a virtual DOM, which is a lightweight copy of the actual DOM similar to the browser's. When data changes, React improves rendering performance and efficiency by comparing the differences between the virtual DOM and the actual DOM and minimally updating the actual DOM.

React



Step 3 of Web Design: Building Your Web with React.js

React.js is a popular JavaScript library developed and maintained by Facebook that has many features that make it ideal for building modern front-end applications. Here are the key features of React.js:

- <u>Unidirectional data flow</u>: React.js follows the principle of unidirectional data flow, where data is passed from parent component to child component.

Does the left-hand side look familiar to you? 🤔

Take a Break and Get Ready to Dive into React in Lecture 02! ☺

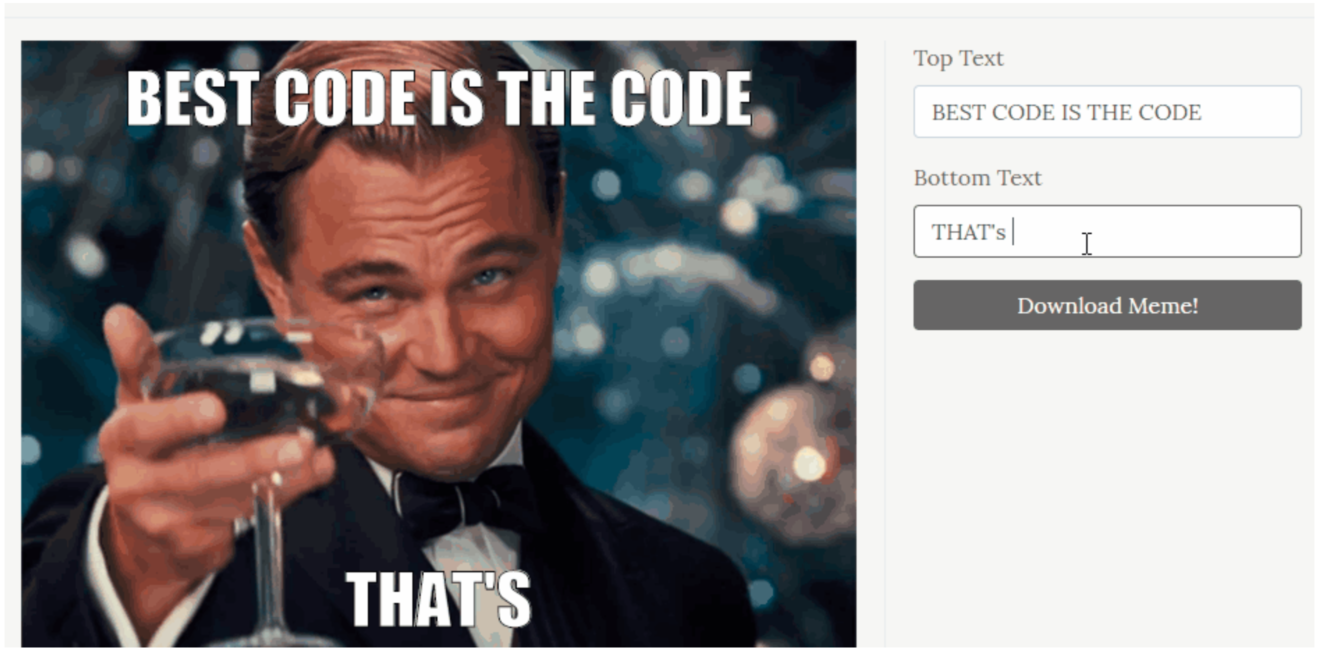


Image Source: [React for Beginners: Building a Meme Maker with React. (2023). *FreeCodeCamp*]