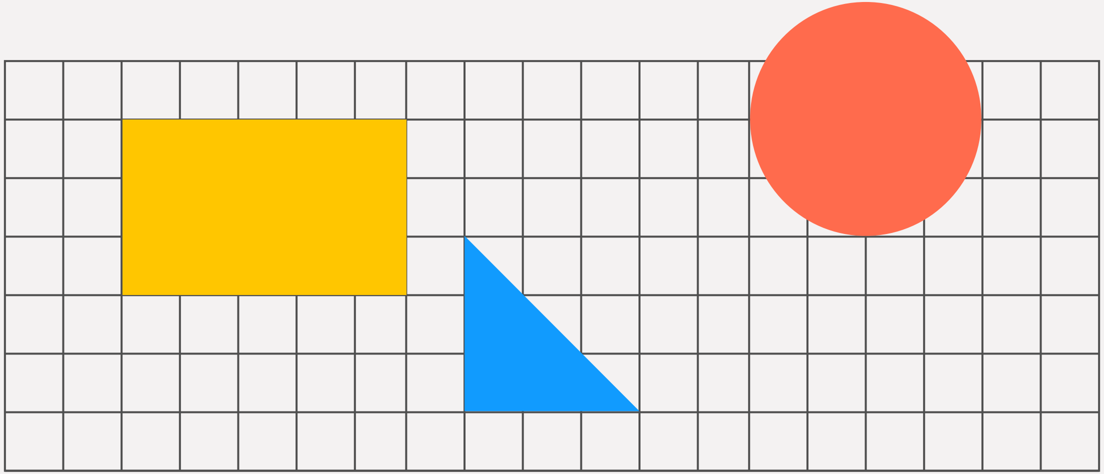
UIs That React (Literally)

Ali Abrishami



In This Lecture You Will...

a

Understand the problem with classic FE dev.

b

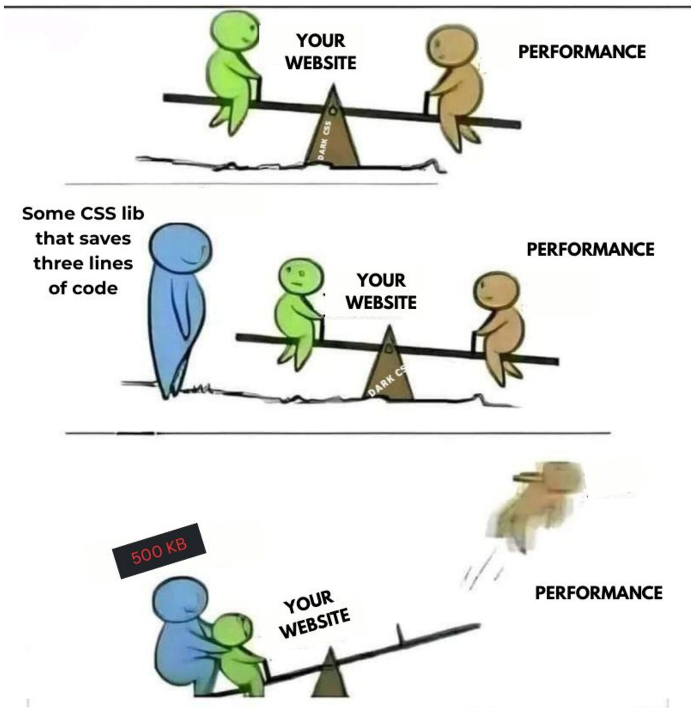
Learn what React is

c

Understand the dynamicity in React

Classic FE: The Challenges

- Tangled **DOM manipulation** with vanilla JS/jQuery
- Inconsistent and hard-to-manage state
- Poor code reuse and maintainability
- Performance bottlenecks from manual updates
- Difficult collaboration and scaling

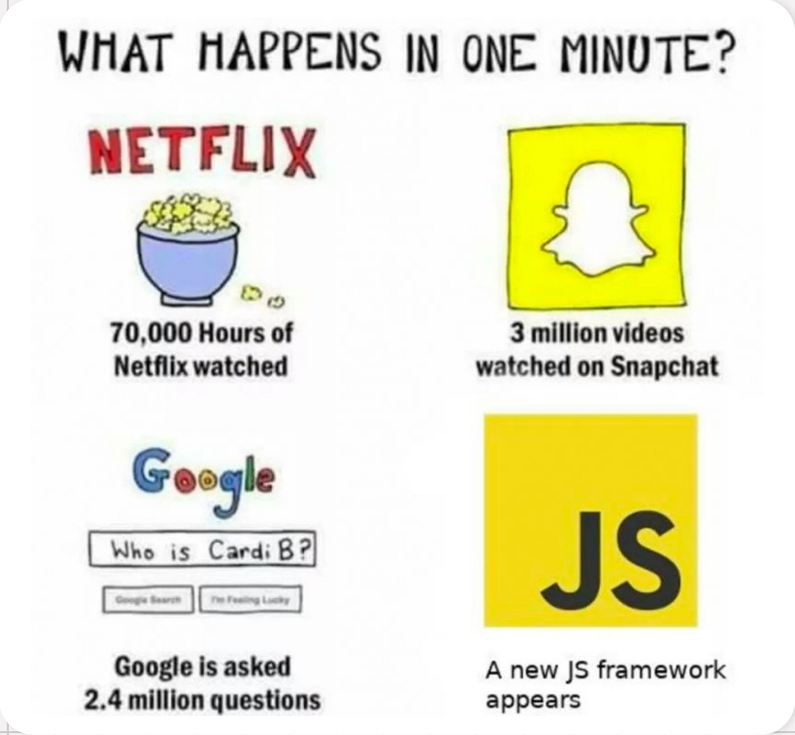


jQuery Rises!

- Made DOM manipulation simpler and faster
- Abstracted away **cross-browser issues**
- Popularized easy AJAX requests
- Brought a cleaner API vs. raw JavaScript
- But still left state and structure problems unsolved

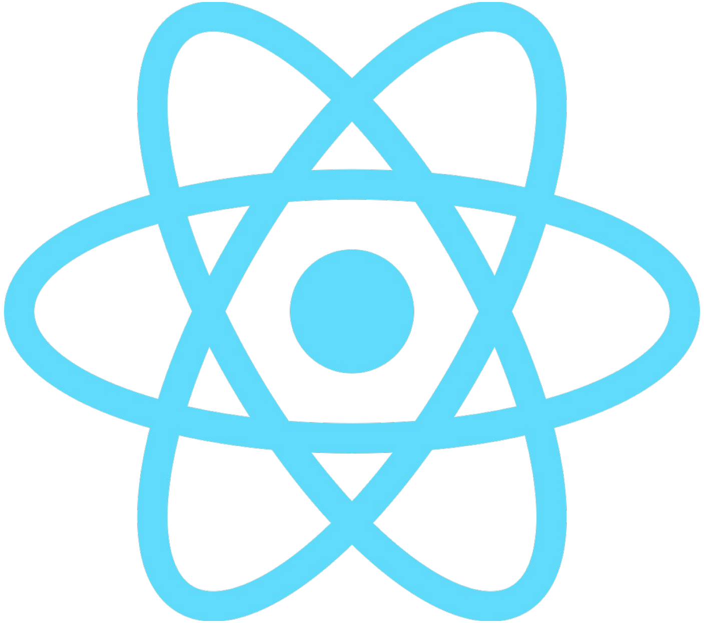
Enter The Frameworks

- Introduced structure and conventions for building UIs
- Improved **state management** and **data flow**
- Encouraged **reusable** components and **modular** code
- Boosted developer productivity and **collaboration**
- Set the stage for modern libraries like React, Vue, Angular


WHAT HAPPENS IN ONE MINUTE?
NETFLIX
70,000 Hours of Netflix watched
3 million videos watched on Snapchat
Google
Who is Cardi B?
Google Search
I'm Feeling Lucky
Google is asked 2.4 million questions
JS
A new JS framework appears

What is React?

- A JavaScript library for building user interfaces
- Created by Facebook in 2013
- **Component-based:** build UIs from reusable pieces
- **Declarative:** describe what UI should look like, not how to update it
- Efficient rendering with the **Virtual DOM**



Framework vs. Library

- Library: You call it (**you** are in control)
- Framework: It calls you (**it** is in control)
- Libraries provide tools, frameworks provide structure.
- Frameworks **enforce** rules and conventions

CLASS ACTIVITY TIME: Is React a framework or a library?

React notable projects

- **Facebook & Instagram**: React's birthplace and showcase
- **Airbnb**: highly interactive booking platform
- **Netflix**: optimized UIs for performance and scalability
- **Uber**: dynamic dashboards and real-time apps

Top 30 Global Companies That Use React.JS
YAHOO!
salesforce
The New York Times
POSTMATES
BBC
asana
Uber

Source: Artoon Solutions

The React Ecosystem & Architectures

- **SPA** (Single-Page Apps): Classic React web apps
- **SSR** (Server-Side Rendering): Next.js for SEO & performance
- **SSG** (Static Site Generation): Astro, Gatsby for fast static sites
- **Mobile Apps**: React Native for cross-platform development

React vs. Vanilla JS

ReactJS	Vanilla JS
VDOM updates	manual DOM updates
Centralized with hooks	scattered state
Modular development	spaghetti code
Composable building blocks	Copy-paste!
Better scaling	Better starting point (fast start)

Component-based Development

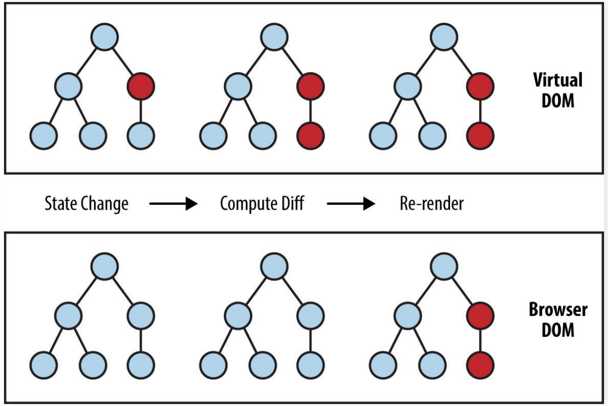
1. UIs built from small, independent pieces
2. Each one manages its own logic and UI
3. Components are reusable across the app
4. Easy to compose into larger features
5. Encourages clean, modular code
6. Simplifies maintenance and scaling

React Under the Hood

- Uses a **Virtual DOM** to optimize UI updates
- Employs Reconciliation to detect minimal changes
- Runs on the Fiber architecture for efficient rendering
- Applies a diffing algorithm to update only what’s needed
- Ensures smooth UI performance at scale

VDOM Updates

- React keeps a Virtual DOM copy of the UI in memory
- When state or props change, a new VDOM tree is created
- React diffs the new VDOM against the previous one
- Calculates the minimal set of changes (patches)
- Updates **only** the necessary DOM nodes, not the entire page
- Ensures fast and efficient rendering.



Source: Medium

React Starter Pack

- **Node.js**: runtime for running React tools & build systems
- **VS Code** or **WebStorm**: extensible code editor
- **Vite**: fast modern build tool for React projects
- **React Developer Tools**: browser extension for debugging components

React Starter Pack

- Install Node.js and a package manager (`npm` / `pnpm` / `yarn`)
- Create a project with Vite: `npm create vite@latest my-app`
- Choose React (or React + TypeScript) template
- Install dependencies: `npm i`
- Run! `npm run dev`