

# What Is React? And Why Would We Use It?

React is a JavaScript library for  
building user interfaces

React makes building **complex**,  
**interactive** and **reactive** user  
interfaces **simpler**

# What is React.js?

React.js



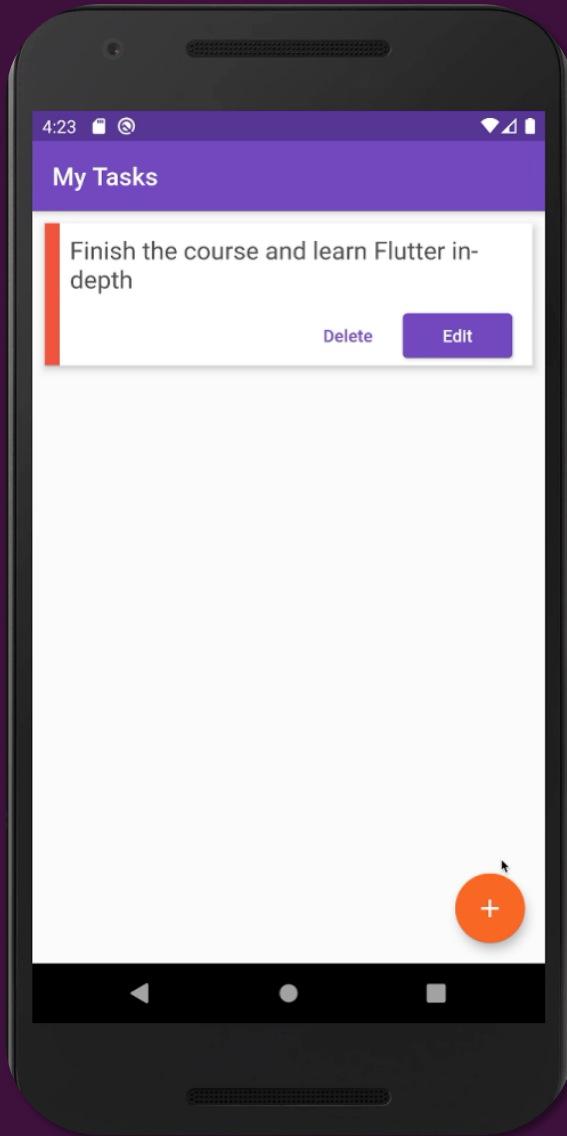
A client-side JavaScript  
library



All about building modern,  
reactive user interfaces for  
the web



Declarative, component-  
focused approach

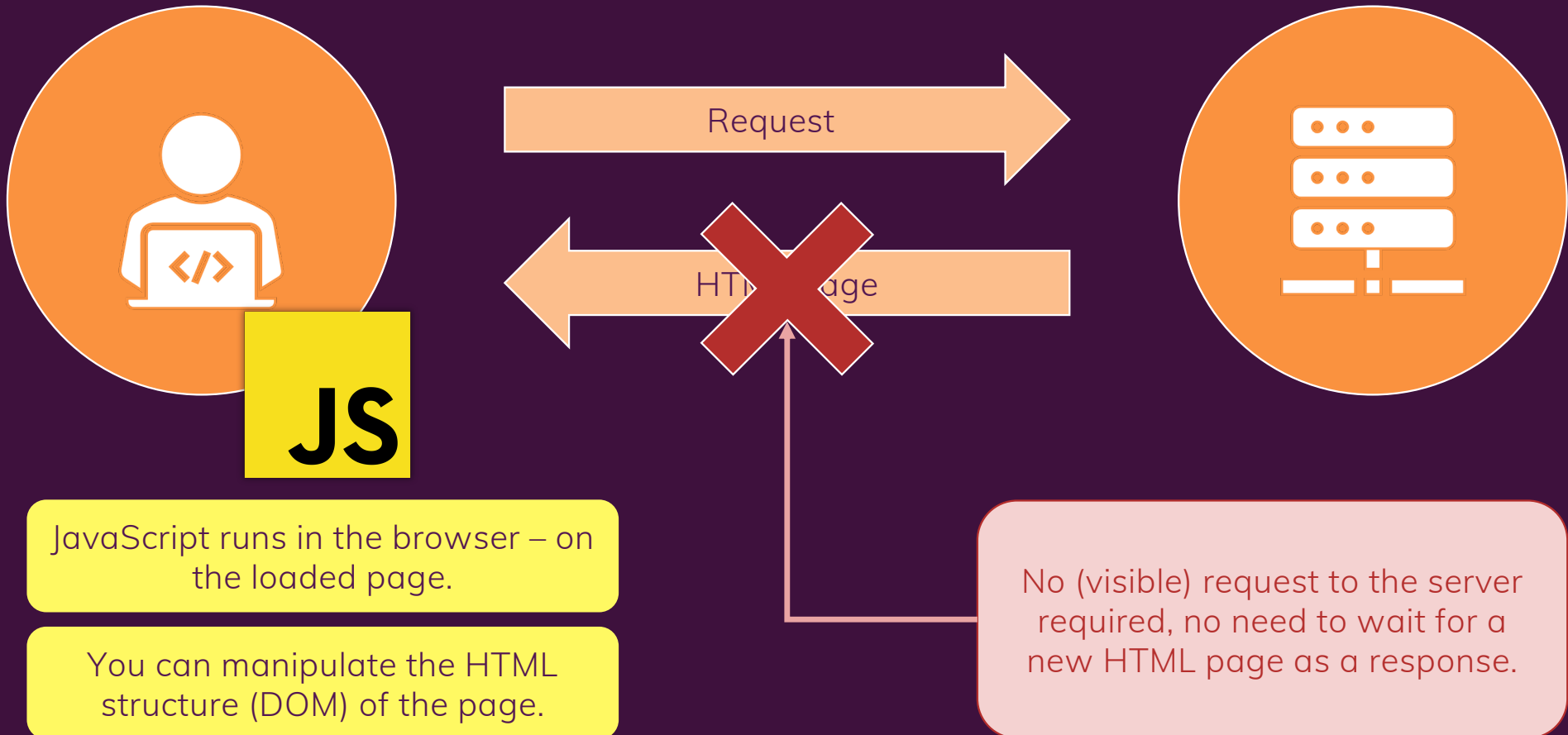


Mobile apps and desktop apps **feel** very “**reactive**”: Things happen instantly, you **don’t wait** for new pages to load or actions to start.

Traditionally, in web apps, you click a link and wait for a new page to load. You click a button and wait for some action to complete.

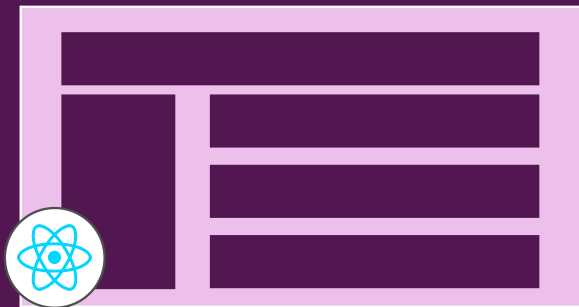


## JavaScript To The Rescue!



# Building Single-Page-Applications (SPAs)

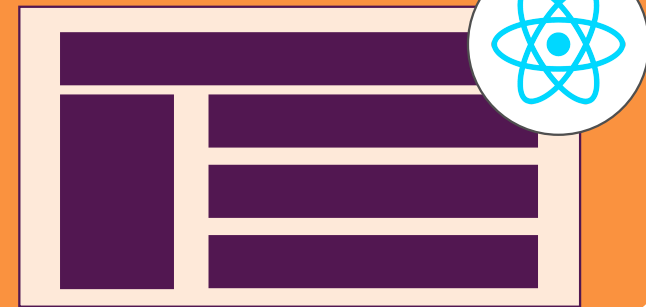
React can be used to **control parts** of HTML pages or entire pages.



“**Widget**” approach on a multi-page-application.  
(Some) pages are still **rendered on and served by a backend server**.

In this course!

React can also be used to **control the entire frontend** of a web application



“Single-Page-Application” (**SPA**) approach. Server **only sends one HTML page**, thereafter, React takes over and controls the UI.

**HTML, CSS & JavaScript** are about  
building user interfaces **as well**



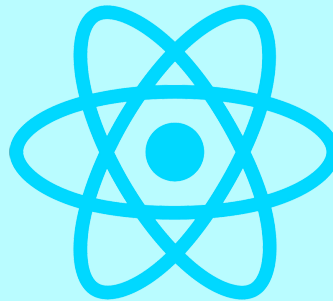
## React.js Alternatives

Angular



Complete component-based UI framework, packed with features. Uses TypeScript. Can be overkill for smaller projects.

React.js



Lean and focused component-based UI library. Certain features (e.g. routing) are added via community packages.

Vue.js



Complete component-based UI framework, includes most core features. A bit less popular than React & Angular.

# About This Course & Course Outline

Theory / Small  
Demos & Examples



More Realistic  
(Bigger) Example  
Projects



Challenges &  
Exercises

Components & Building  
UIs

Working with Events &  
Data: "props" and "state"

Styling React Apps &  
Components

Introduction into "React  
Hooks"

**Basics & Foundation**  
*(Introducing Key Features)*

Side Effects, "Refs" & More  
React Hooks

React's Context API &  
Redux

Forms, Http Requests &  
"Custom Hooks"

Routing, Deployment,  
NextJS & More

**Advanced Concepts**  
*(Building for Production)*

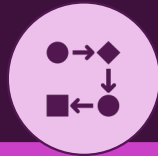
JavaScript Refresher

ReactJS Summary

React Hooks Summary

**Summaries & Refreshers**  
*(Optimizing your Time)*

## Taking This Course: Two Options



### Standard Approach *(Recommended)*

Start with lecture 1 in section 1 and go through the course step by step

Skip JavaScript refresher module if you don't need it

Use React summary module at the end to summarize what you learned or to refresh knowledge in the future



### Summary Approach *(If you're in a hurry)*

Skip forward to the React summary module

Optionally also take JavaScript refresher module if you need it

Go through the entire course after going through the summary module and / or if you got more time in the future

## How To Get The Most Out Of The Course



**Watch the Videos**  
*(choose your pace)*



**Code Along & Practice**  
*(also without me telling you)*



**Debug Errors & Explore Solutions**  
*(also use code attachments)*



**Help Each Other & Learn Together**  
*(Discord, Q&A Board)*