

<WA1/>
<AW1/>
2026

Applicazioni Web I

Web Applications I

Introduction to the course

Fulvio Corno, Luigi De Russis

Francesca Russo, Luca Scibetta, Alberto Monge Roffarello

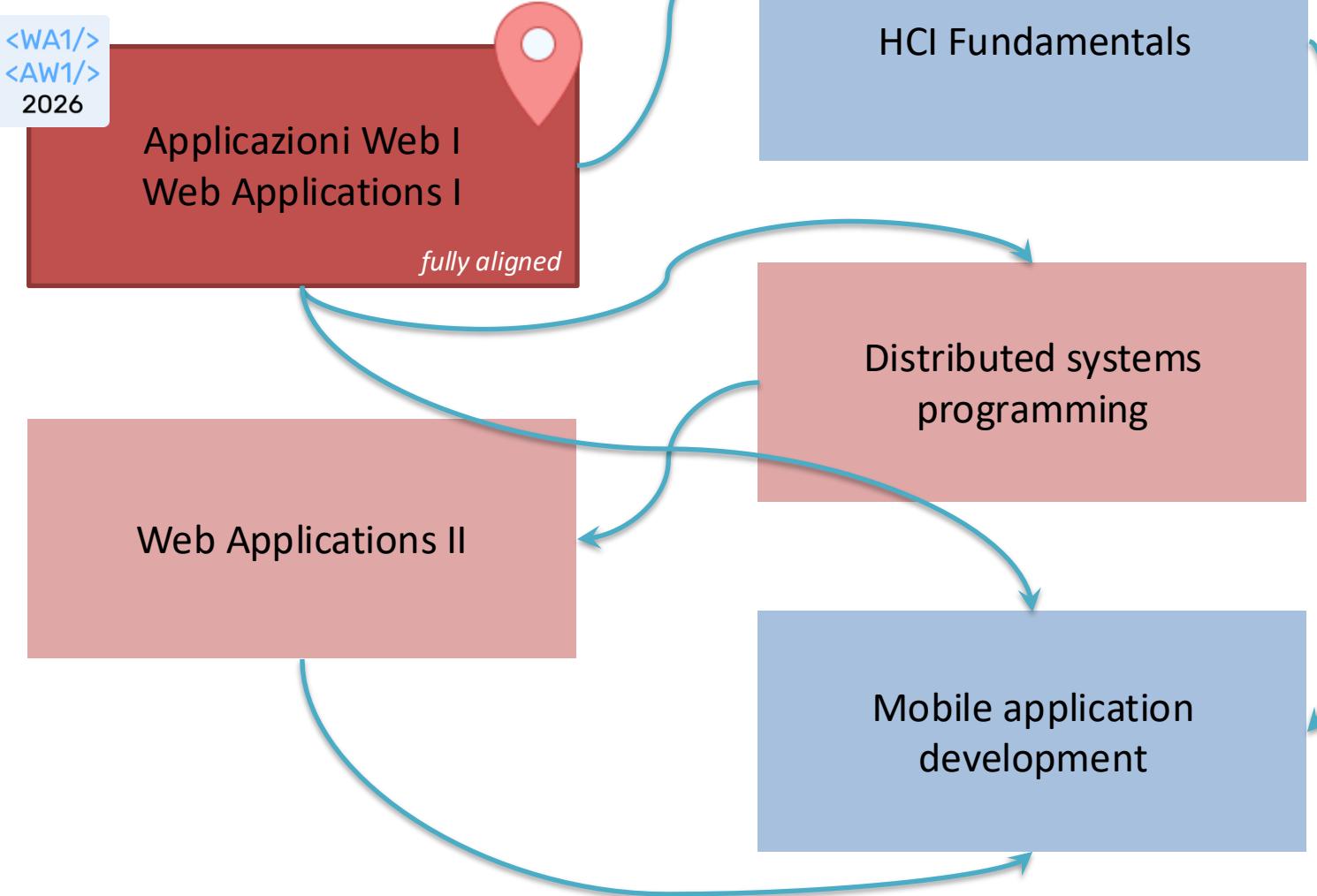


Goal

- Understanding web architectures
- Understanding and mastering web application design and development
- Gaining in-depth knowledge of the JavaScript language and ecosystem
- Becoming familiar with one of the most popular JavaScript frameworks (React)
- ...with special focus on the front-end

The Bigger Picture

- Web architecture
- JavaScript
- Browsers
- **Front-End** programming
- **Back-end** programming
- Scalability
- Large-scale

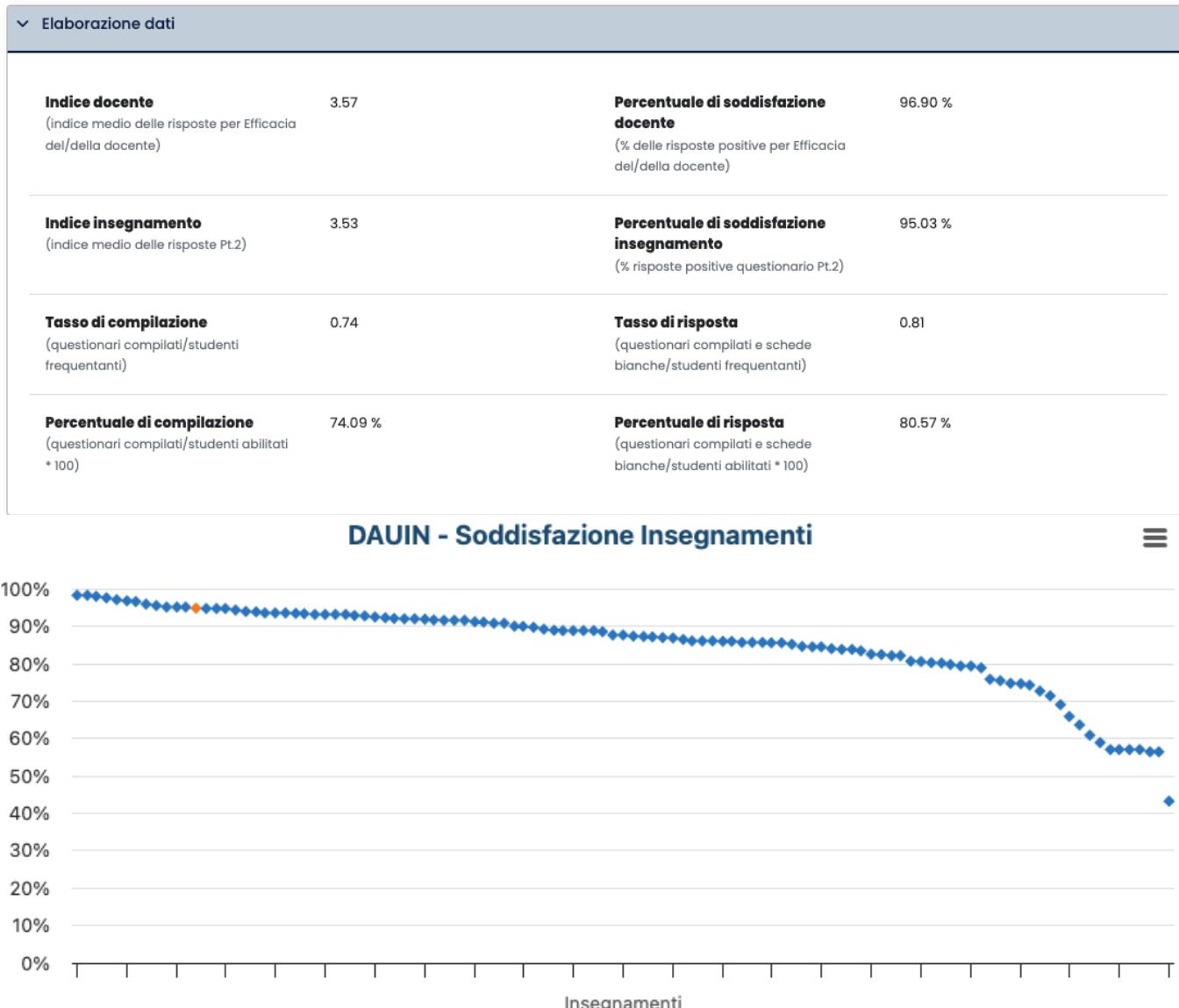


- Usability
- Interface design
- Human centered processes
- Distributed Architectures
- Protocols
- Foundations
- Mobile Front-End
- Mobile device programming



You are here

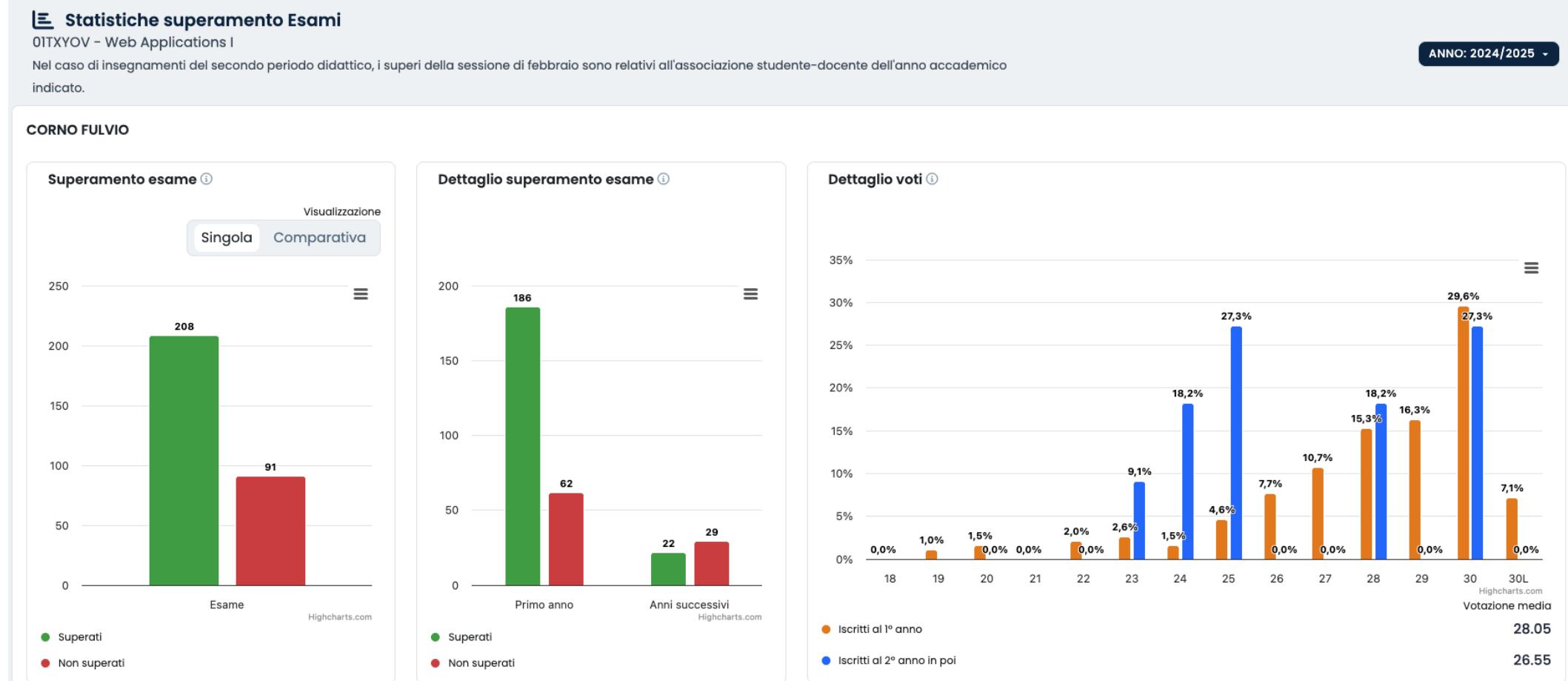
2025 End-of-course Questionnaire



- Relevant critiques*:
 - Labs, in group and not mandatory, don't work
 - Not enough time to reason about each topic
 - Exam discussion is challenging

* considered in planning this year's course

Exam statistics (2024/25)



https://didattica.polito.it/pls/static/esami/statistiche/?p_cod_ins=01TXYOV&p_a_acc=2025

What We Will Learn

JavaScript as a language

- ECMAScript ES6
- Language constructs
- In-depth semantics
- Functional, Asynchronous, Modular, ...



The browser ecosystem

- HTML, CSS, page structure
- DOM
- Events, Properties, Handlers, APIs



Single Page Applications

- Server-side (bare minimum) with node
- API development
- Backend storage
- Sessions and Authentication



React framework

- Components, Properties, State
- JSX
- Hooks
- Router



Weeks and Calendar... At a Glance!

1. Intro to JS: basics, objects, functions
2. Intro to JS: async programming, callbacks, DB interaction + Intro to Web
3. Server-side with Express
4. HTML, CSS, Bootstrap
5. Intro to React + DOM
6. React: props and state
7. React: context, life cycle, forms
8. React router
9. Fetch and client-server interaction (in React)

Course Organization

- Classes
 - 3 h/week
 - Lectures + Exercises (*mixed*)
- Laboratories
 - 1.5 h/week
 - 2 Lab groups
 - From the 2nd week
- **Exception:** first week
 - Class instead of Lab

| | MO | TU | WE | TH | FR |
|-------|----|----|----|----|----|
| 08:30 | | | | | R2 |
| 10:00 | | 2P | | | R2 |
| 11:30 | | 2P | | | |
| 13:00 | | | | | |
| 14:30 | | | | | |
| 16:00 | | | | | |
| 17:30 | | | | | |

Classes

- **In person**, in rooms with power outlets at the desks
 - bring your own computer, if possible, to follow the examples/exercises
- **Video-recorded** and made available soon after each class
 - *not* streamed live
- Once during the course, we will give you some materials to read/watch instead of a lecture (i.e., **readings**)
 - published well *in advance*

Laboratories

- From the 2nd week
- In rooms with power outlets at the desks
- Text online, some days in advance
- Exercises to be done during Lab hours
- Solution will be posted on GitHub
 - around 1 week after the end of each lab

Laboratories

- You will build a simple project during the labs
 - Step by step, following the course topics
- 2 slots:
 - 08:30-10:00: Last name A-L
 - 10:00-11:30: Last name M-Z

Exam project simulation

- Last two weeks of the course
 - All classes (all together), all lab hours (in the two slots)
- We will design and *start* implementing an exam-like project
 - Interactively, reasoning and discussing together

Learning Material

- Course website – <https://elite.polito.it/wa1>
 - Slides (in English)
 - Full schedule
 - Links and supplementary material
- Video lectures (screencasts)
 - YouTube - <<https://www.youtube.com/playlist?list=PLqRTLlwsxDL8pBPQOiq0ObMJU6rvYiT1r>>
 - Portale della Didattica / Polito Students App
- GitHub - <https://github.com/polito-webapp1>
 - Examples, exercises, labs, exams, ...

The screenshot shows the e-Lite course website for "Web Applications I (01TXY)". The top navigation bar includes links for e-Lite, News, People, Research, Teaching, Thesis, and a gear icon. The main content area displays course details: "Official website of the course Web Applications I (code 01TXY) offered to students in the 1st year of the M.S. in Computer Engineering. Academic Year: 2024/2025". Below this, there are sections for "Information" (General information, Course schedule), "Development Resources" (Development Resources (tools, documentation)), and "Exam" (Exam). A sidebar lists other courses: "Applicazioni Web I (01UDF)", "Basi di Dati (04AFOPC)", "Benessere Digitale (01DEL)", "Designing for Mindful Human-Computer Interaction (01GOFIU)", "Human Computer Interaction (02JSK)", "Informatica (14BHD)", "Introduzione alle Applicazioni Web (01DXU)", "User Experience Design (01HFIX)", and "Web Applications I (01TXY)". The "Web Applications I (01TXY)" section is expanded, showing "Information", "Schedule", "Development Resources", and "Exam". Navigation buttons include "Previous", "Next", and "Information".



Communications



- We will use **Telegram** for the main communications
 - among students, with teachers, etc.
- Announcements, official information, and Q&A
- Feel free to contact the teachers for feedback and questions
 - questions of general interest must be posted in the group, so that everybody can see the answer
- Link to the Telegram group (**mandatory**):
<https://t.me/politowebapp2026>

Exam: Two Parts

1. Project development (up to 26 points)

- Individual, using GitHub
- Starting from shared requirements
- 20 days of time

2. Oral discussion (up to 4 points)

- individual and mandatory
- “live” correction and discussion of the submitted project
- when: the official exam day (or starting from that day)

Exam: Score and Process

- Project development + oral discussion: up to 30 points
- Evaluate the functionality, code quality and organization of the submitted project, as well as the student's understanding of their design choices and motivations
- Up to 2 extra points for students whose projects demonstrate a high quality and for the richness and precision of the answers during the discussion

Note: If it emerges that the student does not have mastery of the written code, the exam will be immediately canceled, without a numerical evaluation.

Full exam rules in the course website (under “Exams”)

Project Development

What

- Develop a web application using
 - React + JavaScript
 - Node + Express
 - SQLite
- According to a functional specification
 - published 20 days before each official exam date

How

- Individually (i.e., not in group)
- Using GitHub Classroom
 - commit + push your project
 - GitHub commits and usage are checked
- Teacher's Evaluation
 - running the application on the teacher's laptop
 - examining the code

Oral Discussion

Goals

- To ensure that each student developed the web application **by themselves**
- To evaluate how much the student can explain the **exact** behaviour of the code

Evaluation Criteria

- Theoretical and practical knowledge of the project design
- Theoretical and practical knowledge of the project code base
- Readiness and clarity in the replies

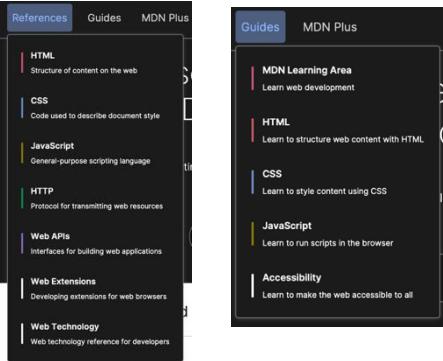
Using External Resources and AI

- Getting help during the course and the exam project is normal: from documentation, Stack Overflow, peers, or AI tools.
- In the course, as in *professional practice*, what matters is your mastery of what you design and implement.
- **You are fully responsible for everything you create and submit**, and you must be able to:
 - Explain all parts of your implementation.
 - Justify your design choices.
 - Acknowledge and own its limitations.
- Submitting code you don't fully understand is unacceptable, regardless of its source.

Resources (fundamentals)

The screenshot shows the MDN homepage with a dark background. At the top, it says "Resources for Developers, by Developers". Below this, there's a section titled "Featured Articles" with four cards: "Introducing the CSS Cascade" (CSS), "<dialog>: The Dialog element" (HTML), "Asynchronous JavaScript" (JavaScript), and "Canvas tutorial" (Web APIs). The "JavaScript" card has a small error in the text: "In this module, we take a look at asynchronous JavaScript, why it is important, and how it can be used to effectively handle potential...".

Mozilla Developer Network
(MDN)
<https://developer.mozilla.org/>

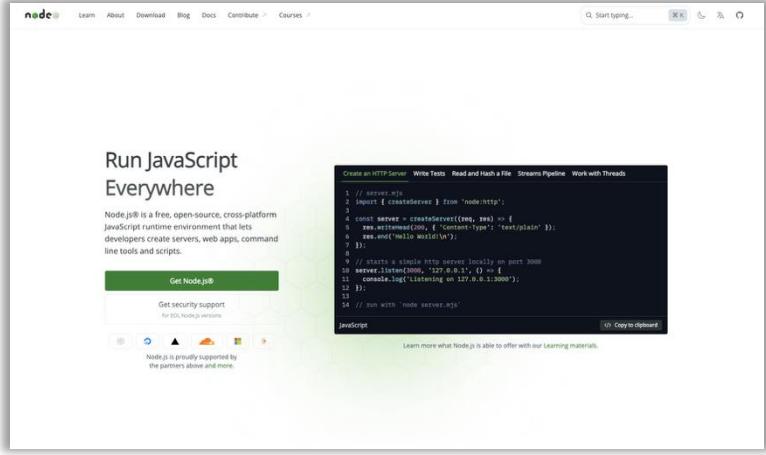


The screenshot shows the React library homepage. It features a large blue atom icon and the word "React". Below this, it says "The library for web and native user interfaces". There are two buttons: "Learn React" and "API Reference". The main content area has a large heading "Create user interfaces from components" and a subtext "React lets you build user interfaces out of individual pieces called components."

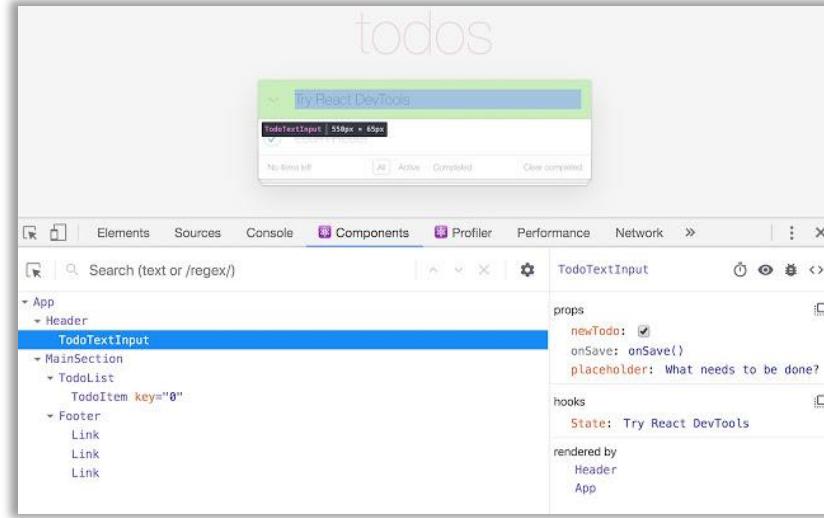
React
<https://react.dev>

- GET STARTED
 - Quick Start >
 - Installation >
- Start a New React Project
- Add React to an Existing Project
- Editor Setup
- Using TypeScript
- React Developer Tools
- LEARN REACT
 - Describing the UI >
 - Adding Interactivity >
 - Managing State >
 - Escape Hatches >

Tools

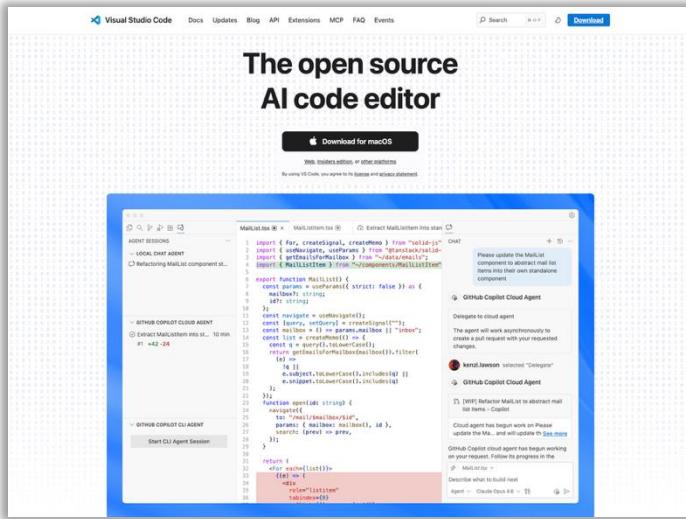


Node.js runtime
Version 24.x LTS
<https://nodejs.org/en/>



React Developer Tools
Extension for [Chrome](#) and [Firefox](#)

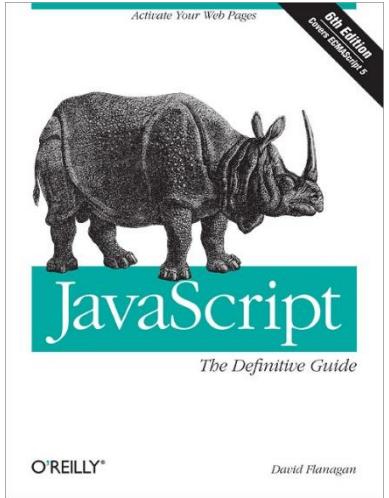
Programming Environment



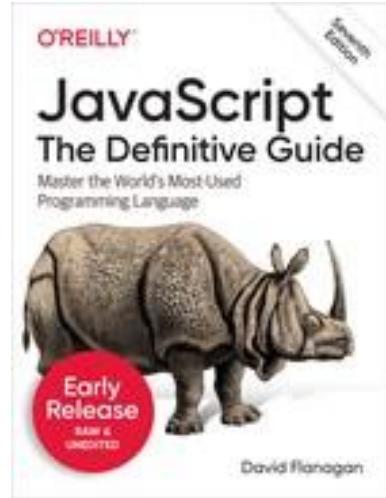
Visual Studio Code

<https://code.visualstudio.com/>

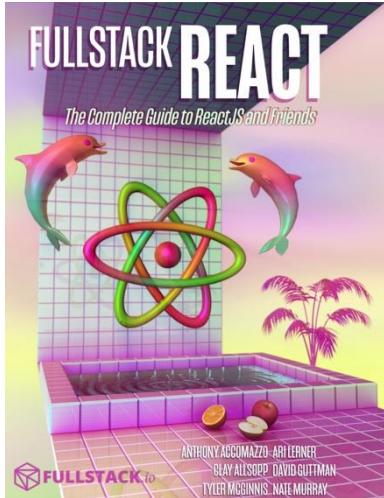
Resources (books)



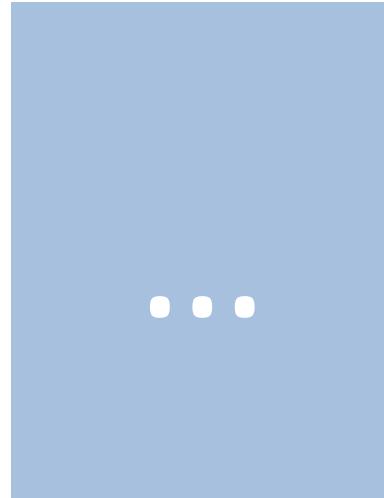
JavaScript: The Definitive Guide,
6th Edition
By David Flanagan
ISBN 978-0596805524
Release Date: May 2011
(not very updated...)



JavaScript: The Definitive Guide,
7th Edition
By David Flanagan
ISBN 978-1491952023
Release Date: July 2020

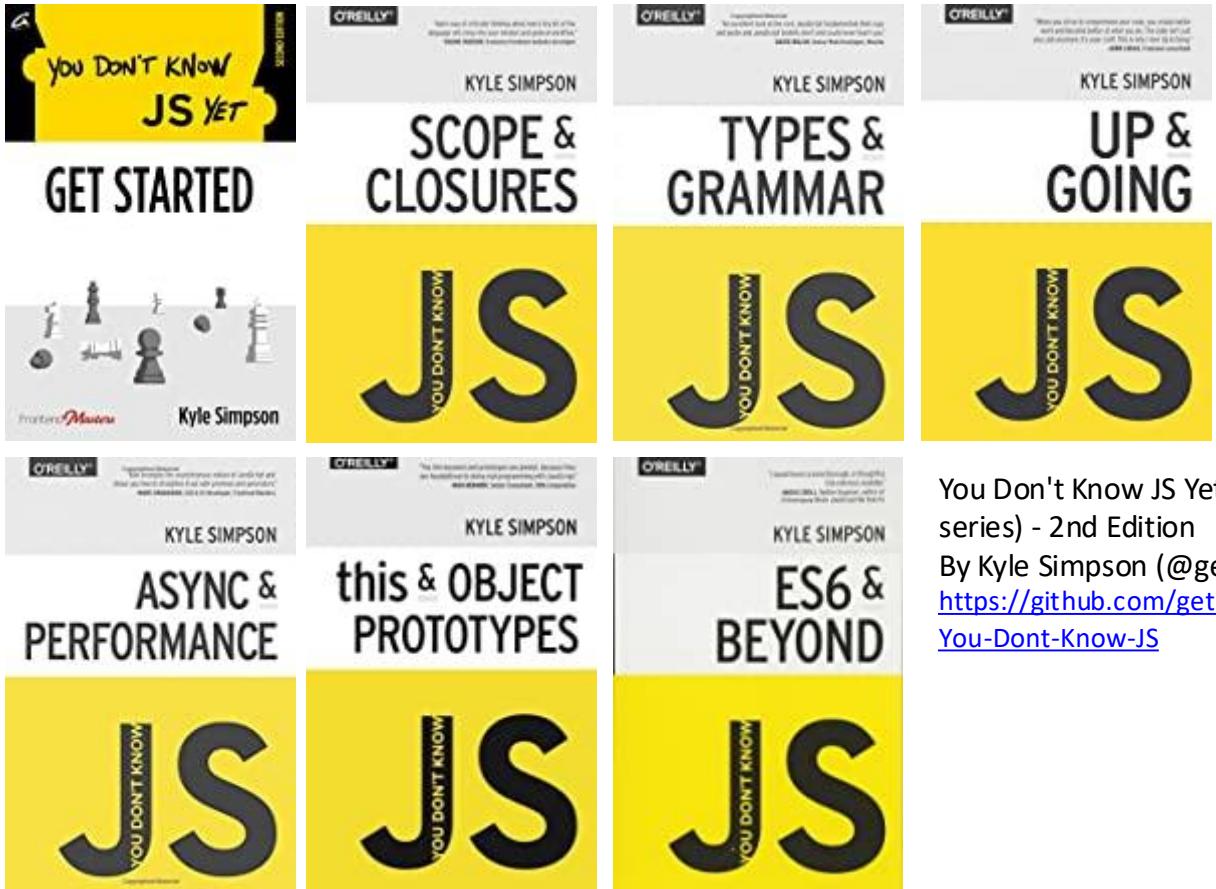


Fullstack React
By Anthony Accomazzo, Nate Murray, Ari Lerner, Clay Allsopp, David Guttman, and Tyler McGinnis
<https://www.newline.co/fullstack-react>
Release: r40 (January 2020)

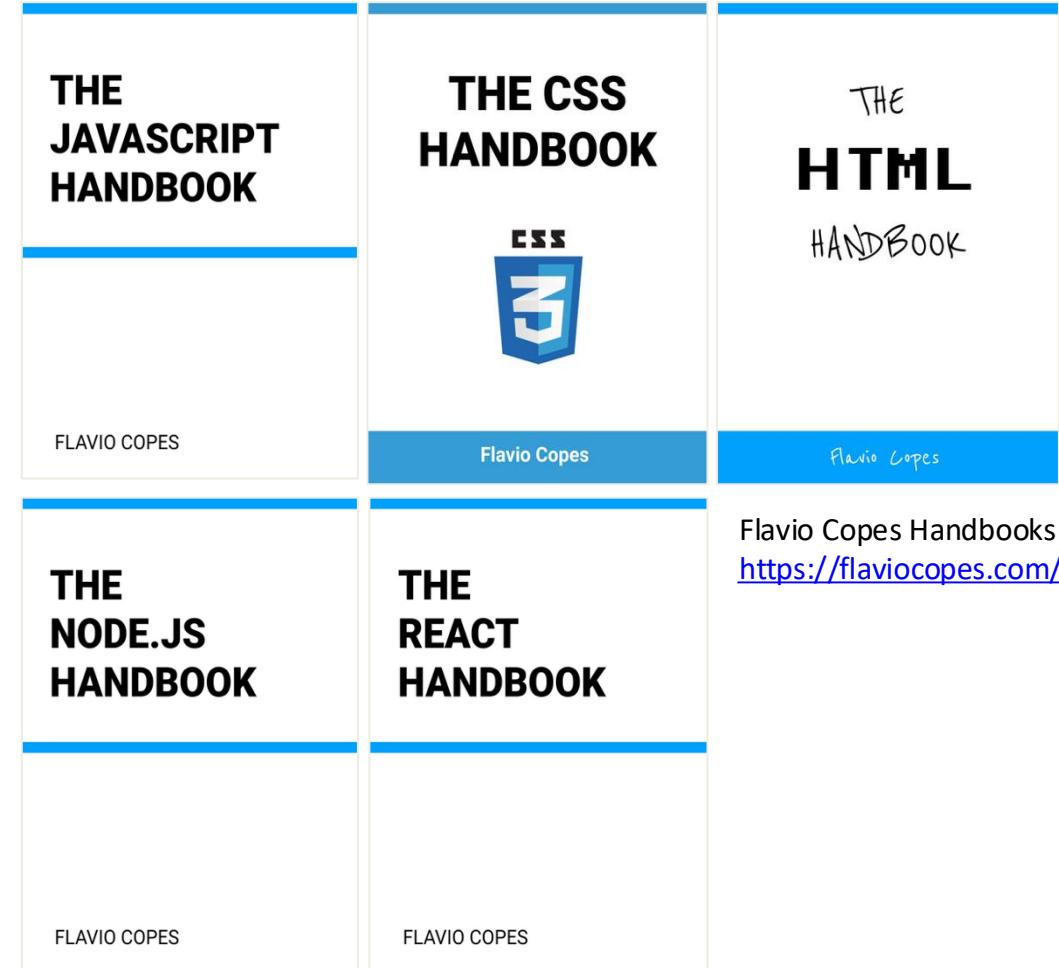


... and many others

Resources (on-line books)

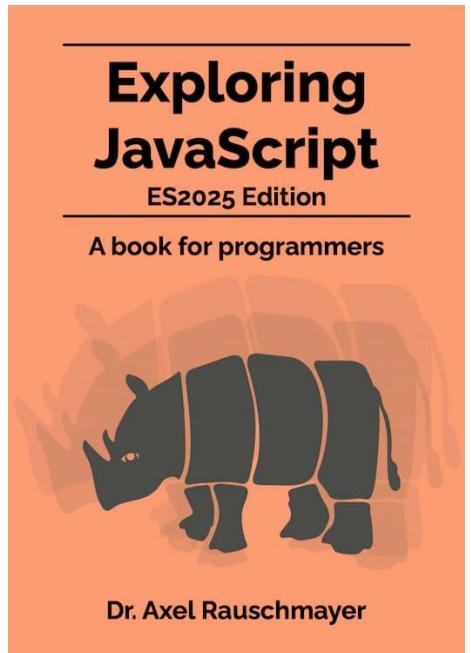


You Don't Know JS Yet (book series) - 2nd Edition
By Kyle Simpson (@getify)
[https://github.com/getify/
You-Dont-Know-JS](https://github.com/getify/You-Dont-Know-JS)



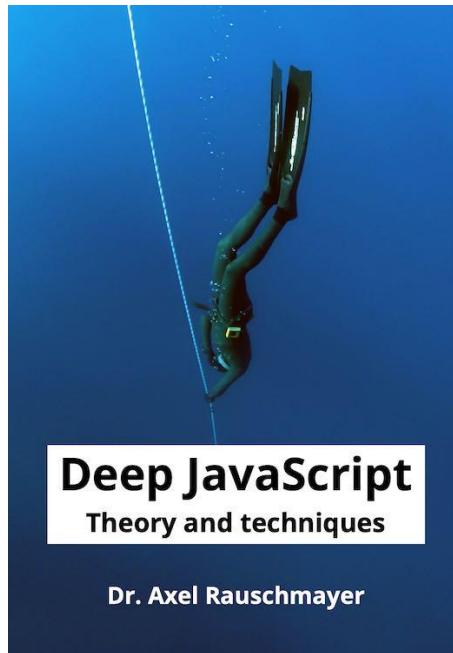
Flavio Copes Handbooks
<https://flaviocopes.com/>

Resources (on-line books)

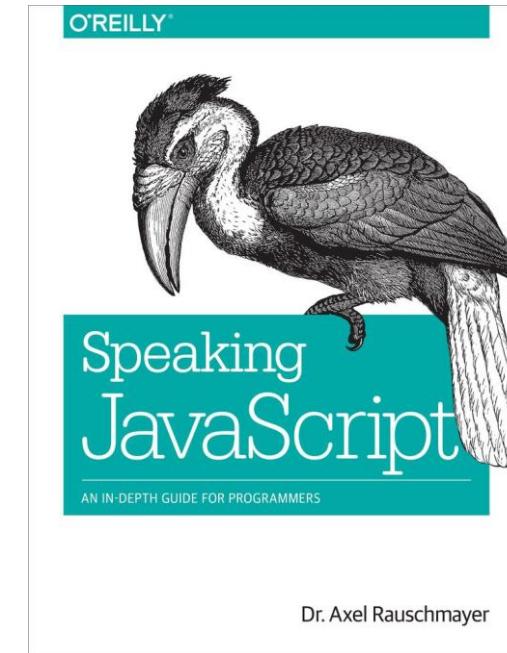


<https://exploringjs.com/impatient-js/index.html>

+

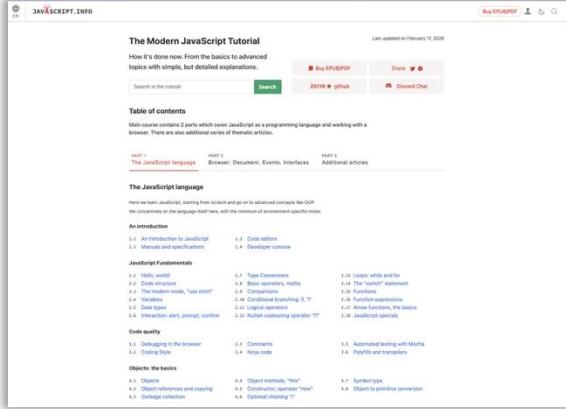


<https://exploringjs.com/deep-js/index.html>

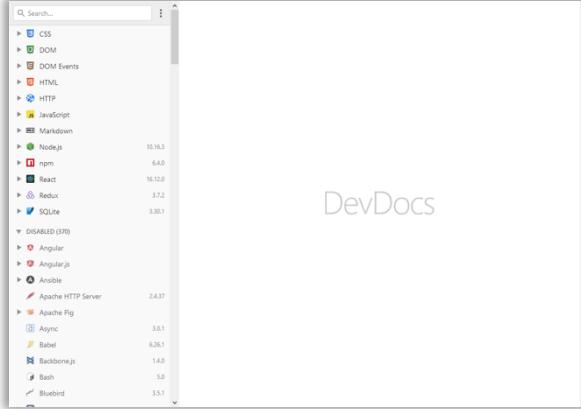


<https://www.oreilly.com/library/view/speaking-javascript/9781449365028/>

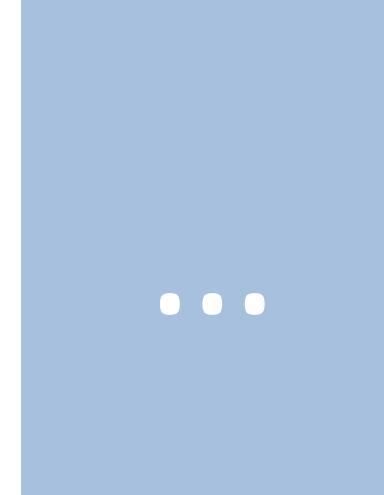
More Resources...



The Modern JavaScript Tutorial
<https://javascript.info/>



DevDocs: API Documentation
Browser
<https://devdocs.io/>



... and many others



License

- These slides are distributed under a Creative Commons license “**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**”
- **You are free to:**
 - **Share** — copy and redistribute the material in any medium or format
 - **Adapt** — remix, transform, and build upon the material
 - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
 - **Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **NonCommercial** — You may not use the material for [commercial purposes](#).
 - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.
 - **No additional restrictions** — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>