

<WA1/>
<AW1/>
2024

Applicazioni Web I Web Applications I

Introduction to the course

Fulvio Corno, **Luigi De Russis**

Luca Mannella, Luca Pezzolla, Juan Pablo Saenz

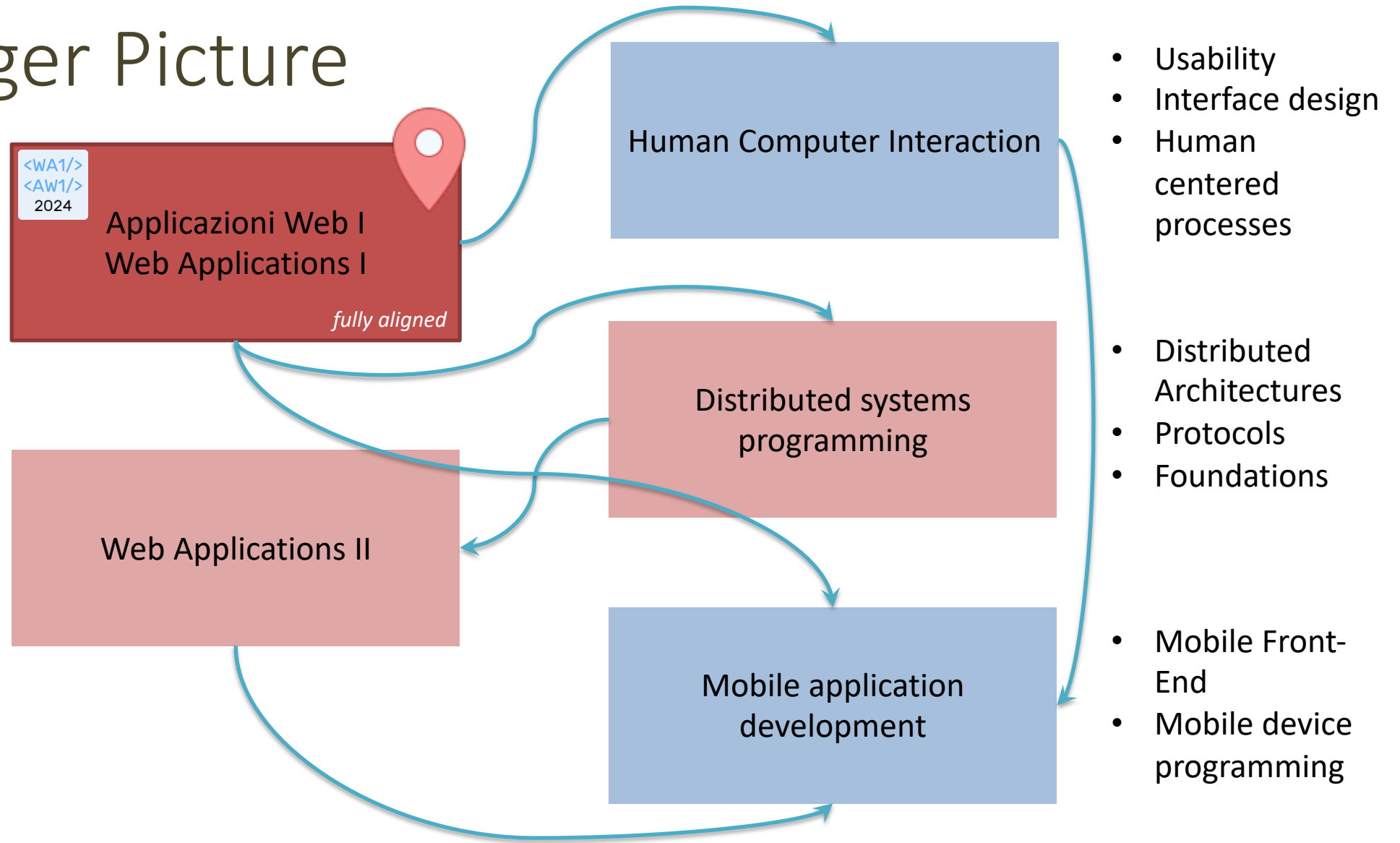


Goal

- 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



What We Will Learn

JavaScript as a language

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

The JavaScript logo, consisting of the letters "JS" in a bold, black, sans-serif font on a yellow square background.

The browser ecosystem

- HTML, CSS, page structure
- DOM
- JavaScript in the browser
- 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
3. Intro to Web and server-side with Express
4. HTML, CSS, Bootstrap
5. DOM and JS in the browser
6. Intro to React
7. React: props and state
8. React: context, life cycle, forms
9. React router
10. Fetch and client-server interaction (in React)
11. Authentication

Course Organization

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

	MO	TU	WE	TH	FR
08:30		2P		R2	
10:00		2P		R2	
11:30					
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
- A few times during the course, we will give you some materials to read/watch instead of a lecture
 - published well *in advance*

Laboratories

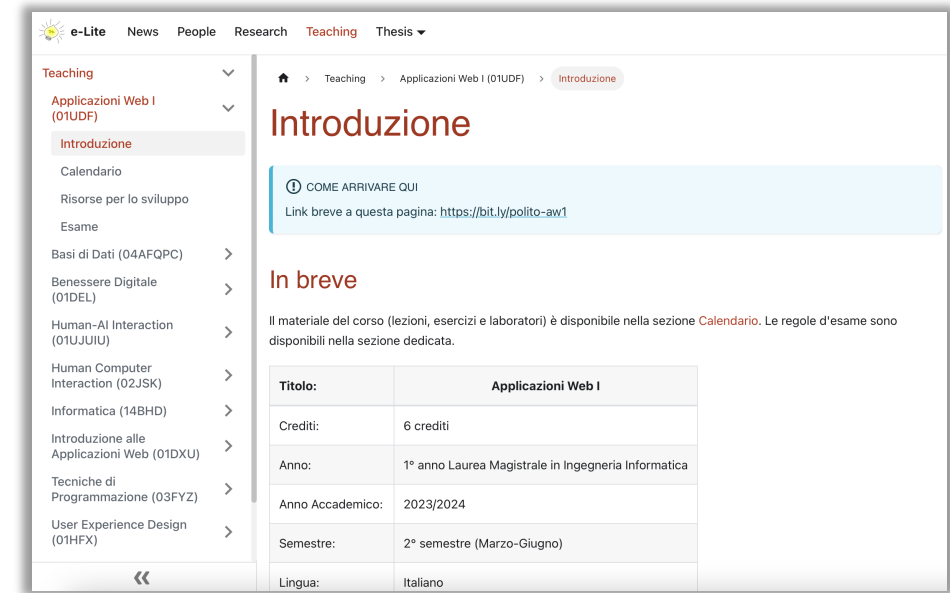
- Starting 14/03/2024
- In rooms with power outlets at the desks
- Text online, some days in advance
- Exercises to be done during Lab hours
- Solution will be posted on GitHub
 - around 1 week after the end of each lab

Laboratories

- You will build a simple project during the labs
 - Step by step, following the course topics
- Some labs will last one week, others will span multiple weeks
- 2 slots:
 - Thursday, 08:30-10:00, room R2, surnames: [A-L](#)
 - Thursday, 10:00-11:30, room R2, surnames: [M-Z](#)

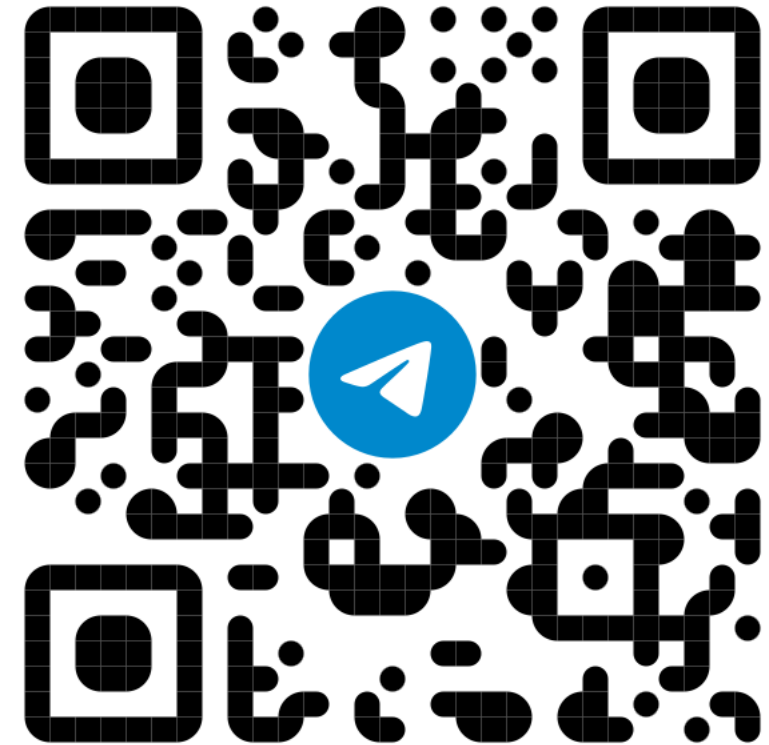
Learning Material

- Course website – <https://bit.ly/polito-aw1>
 - Slides (in English)
 - Full schedule
 - Links and supplementary material
- Video lectures (screencasts)
 - YouTube - https://www.youtube.com/playlist?list=PLs7DWGc_wmwTz6XD62wCYGRaVXMztRpwd
 - Portale della Didattica
- GitHub - <https://github.com/polito-webapp1>
 - Examples, exercises, labs, exams, ...



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:
<https://t.me/+du8qADsfwMgxOTI0>
 - “mandatory” participation in the group
- Emails can be an **alternative** for slower, more articulated, and private individual communications



Students' Hours

- **Why?**

- An opportunity to discuss issues or needs
- To get clarifications or ask questions on the course
- To discuss academic or professional goals (e.g., thesis, what to do next)
- To know more about specific topics
- ...

- **When?**

- **Wednesday 16:00-17:00** in my office; send a message at least one day before
- On request, in person (at my office) or remotely (via Zoom)

Exam: Two Parts

1. **Project development** (up to 26 points)

- Individual
- Starting from shared requirements
- 20 days of time – submit by 23:59 on the day before the official exam date

2. **Oral discussion** (up to 4 points)

- individual and mandatory – in-person, only
- “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
- 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
 - strict submission procedure
- Teacher's Evaluation
 - running the application on the teacher's laptop
 - examining the code

Project Evaluation Criteria

- Correct and complete implementation of the specification
 - Functional requirements
 - Non-functional requirements
- Soundness of design choices and implementation choices
- Consistency with standards and good practices
- Quality and clarity of the code

Oral Discussion

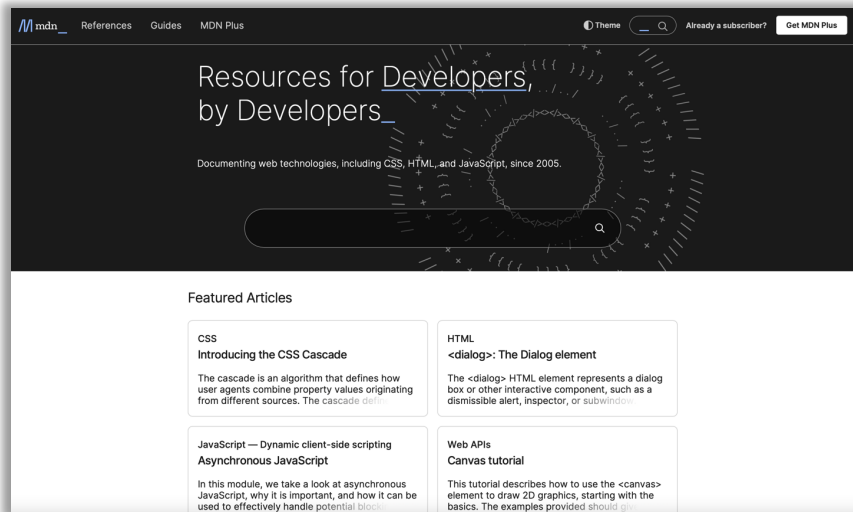
Goals

- 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
- To discuss design and technical choices

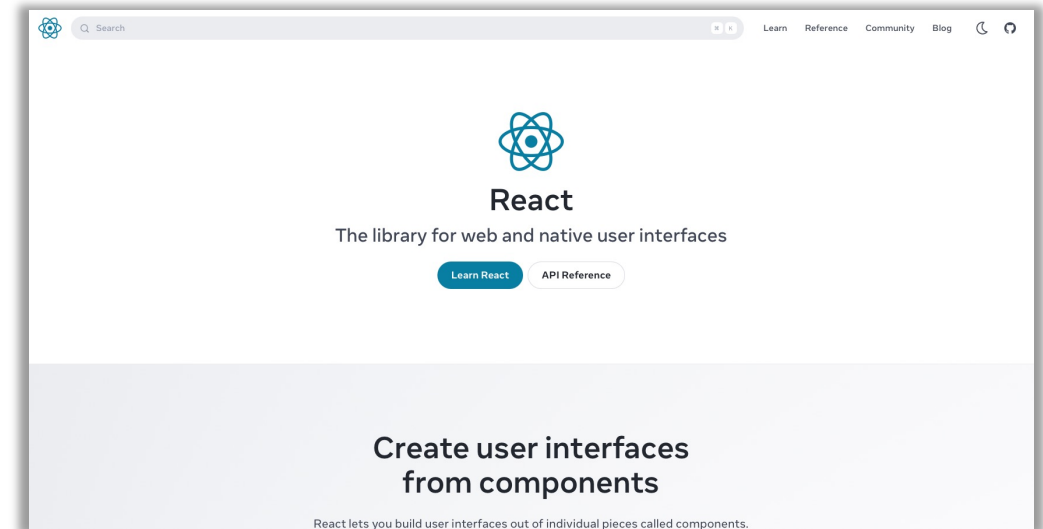
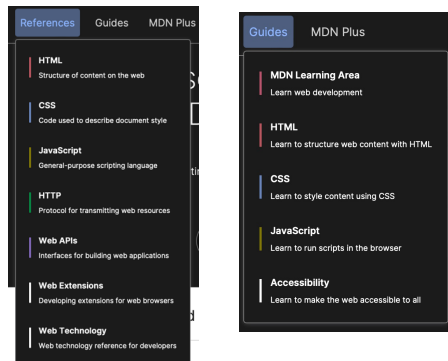
Evaluation Criteria

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

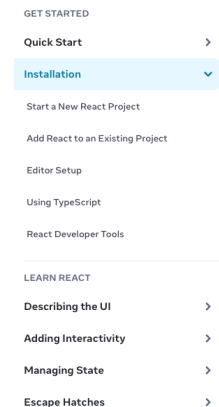
Resources (fundamentals)



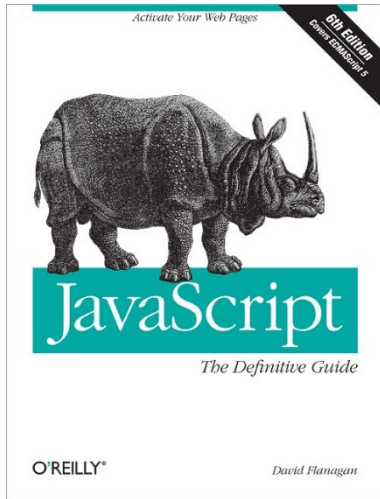
Mozilla Developer Network
(MDN)
<https://developer.mozilla.org/>



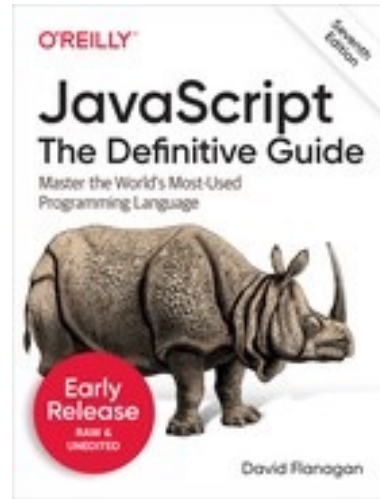
React
<https://react.dev>



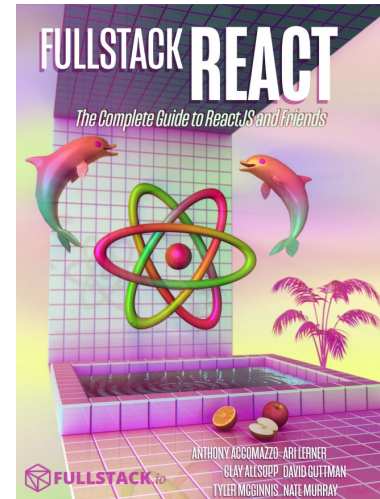
Resources (books)



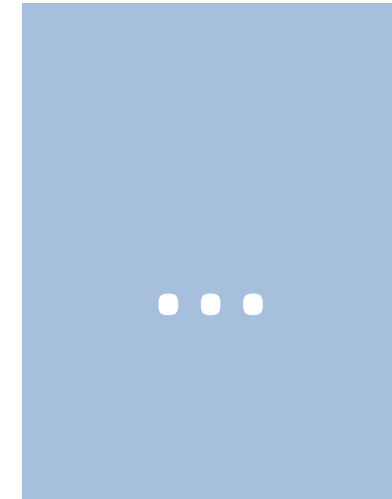
JavaScript: The Definitive Guide,
6th Edition
By David Flanagan
ISBN 978-0596805524
Release Date: May 2011
(not very updated...)



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

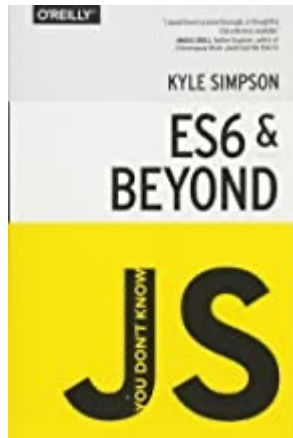
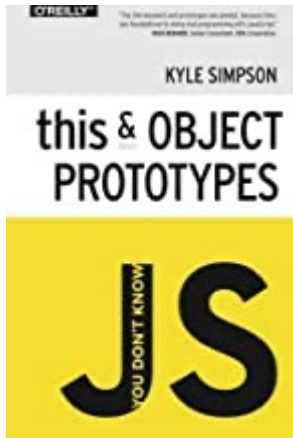
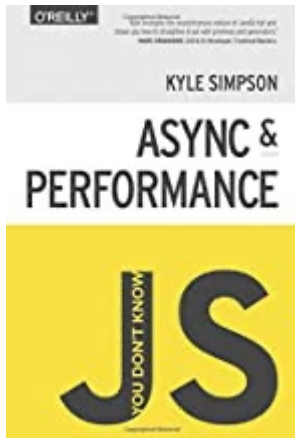
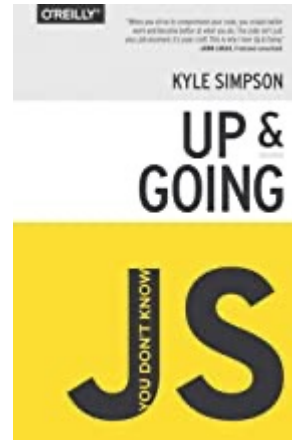
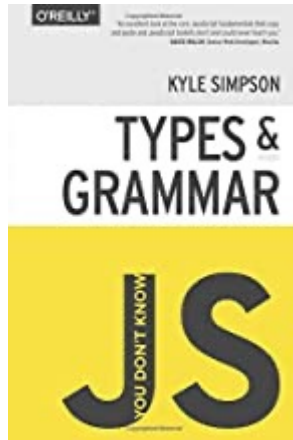
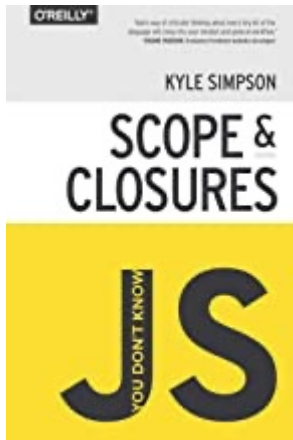


Fullstack React
By Anthony Accomazzo, Nate
Murray, Ari Lerner, Clay
Allsopp, David Guttman, and
Tyler McGinnis
<https://www.newline.co/fullstack-react>
Release: r40 (January 2020)

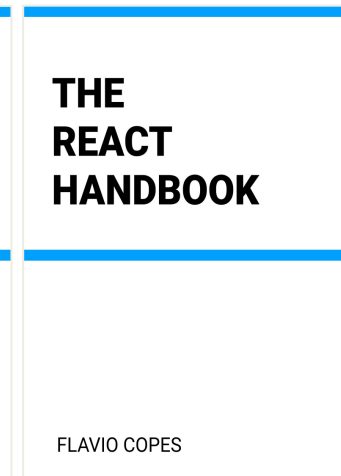
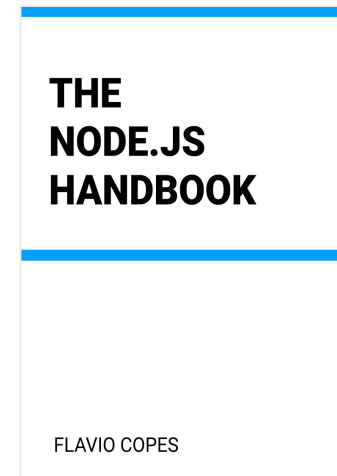
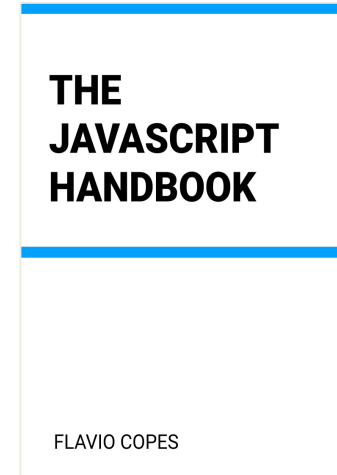


... and many others

Resources (on-line books)

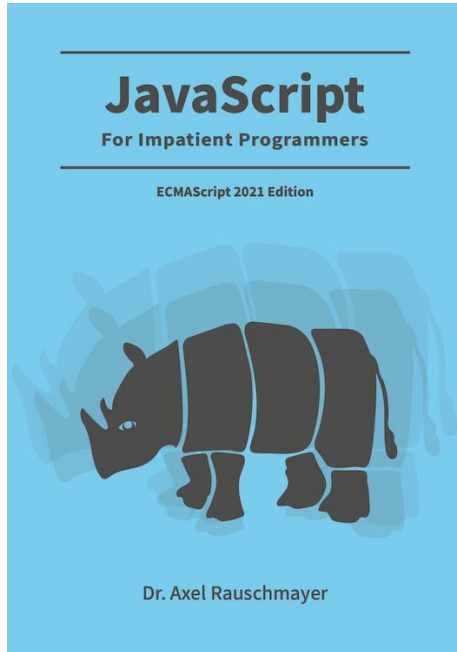


You Don't Know JS Yet (book series) - 2nd Edition
By Kyle Simpson (@getify)
<https://github.com/getify/You-Dont-Know-JS>

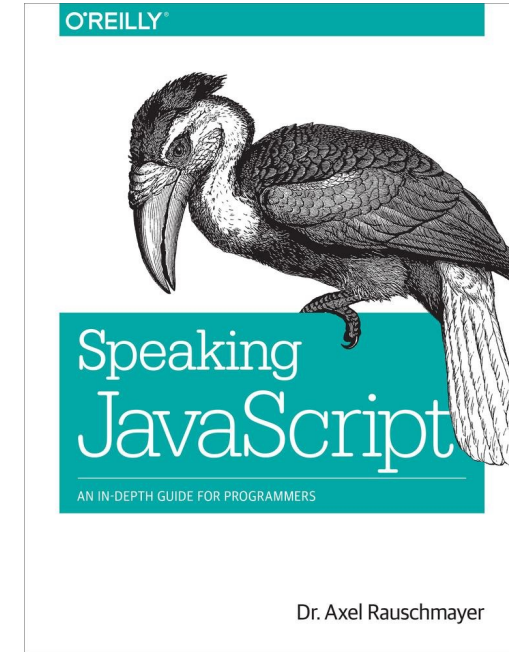
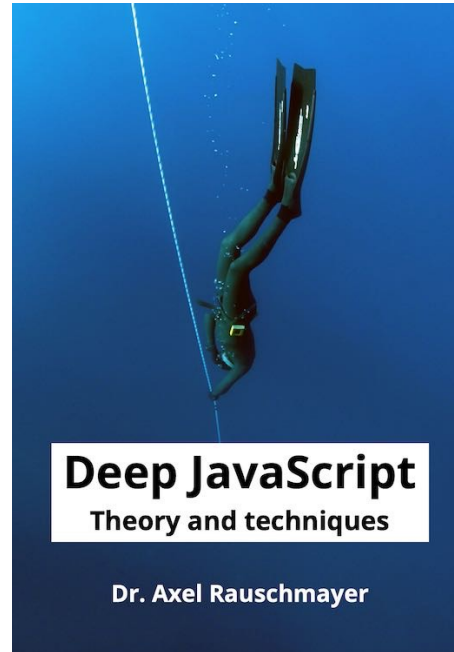


Flavio Copes Handbooks
<https://flaviocopes.com/>

Resources (on-line books)



+

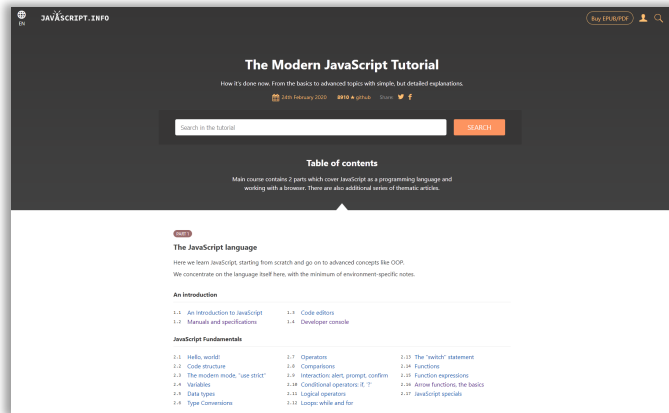


<https://exploringjs.com/impatient-js/index.html>

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

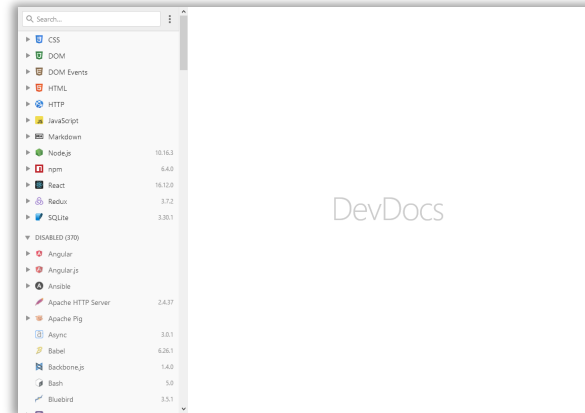
<http://speakingjs.com/>

More resources...



The Modern JavaScript Tutorial

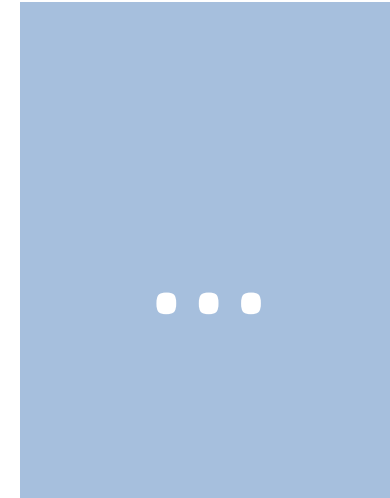
<https://javascript.info/>



DevDocs: API Documentation

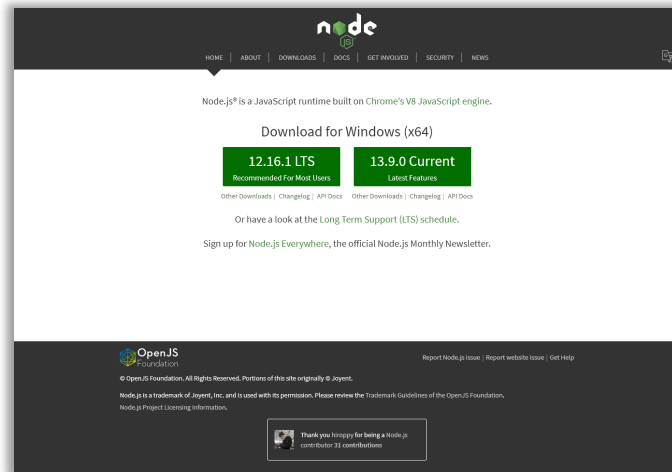
Browser

<https://devdocs.io/>

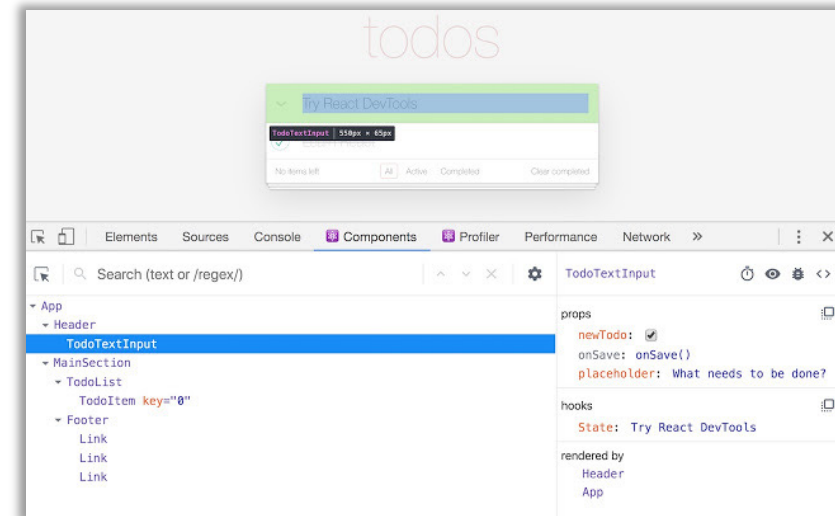


... and many others

Tools

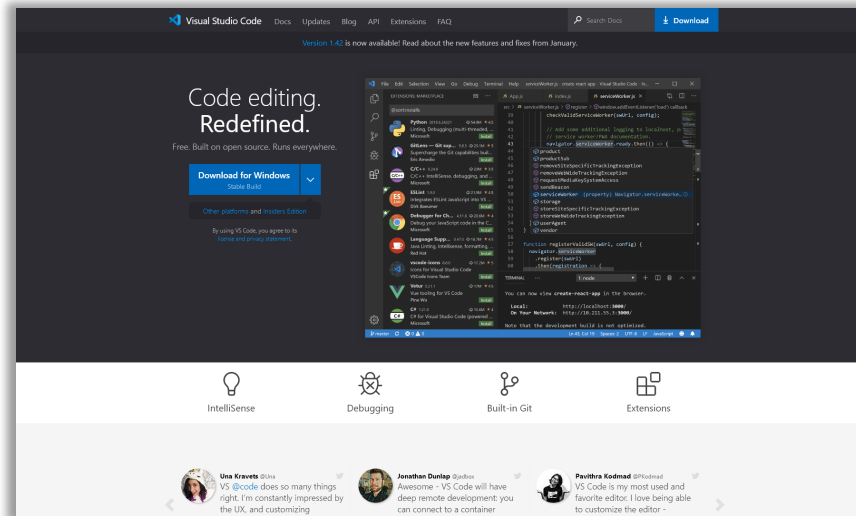


Node.js runtime
Version 20.x LTS
<https://nodejs.org/en/>



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

Programming Environment



Visual Studio Code

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

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/>

