<WA1/>
<AW1/>
2025

Applicazioni Web I
Web Applications I

Introduction to the course

Fulvio Corno, **Luigi De Russis**

Francesca Russo, **Luca Scibetta**

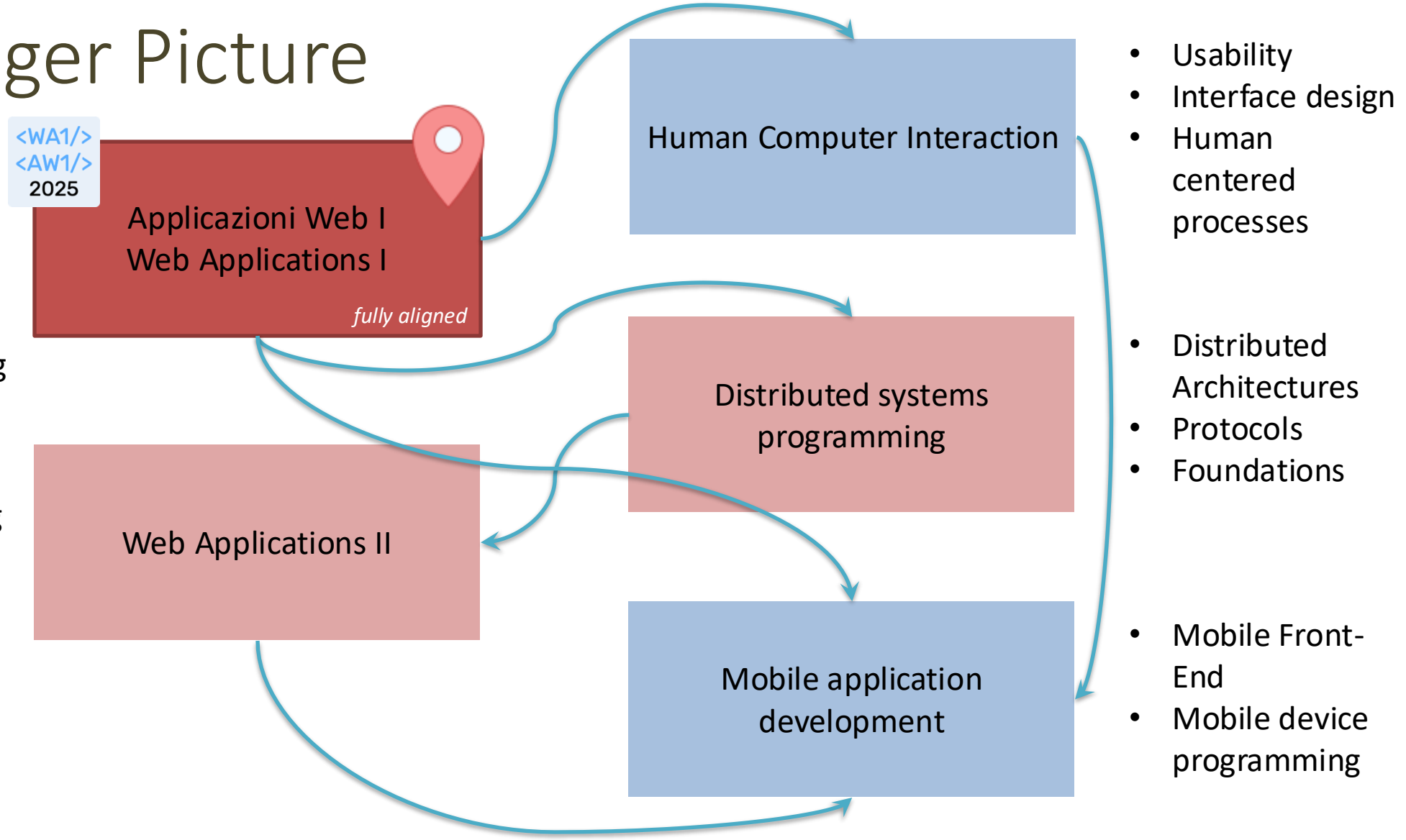
Goal

- 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



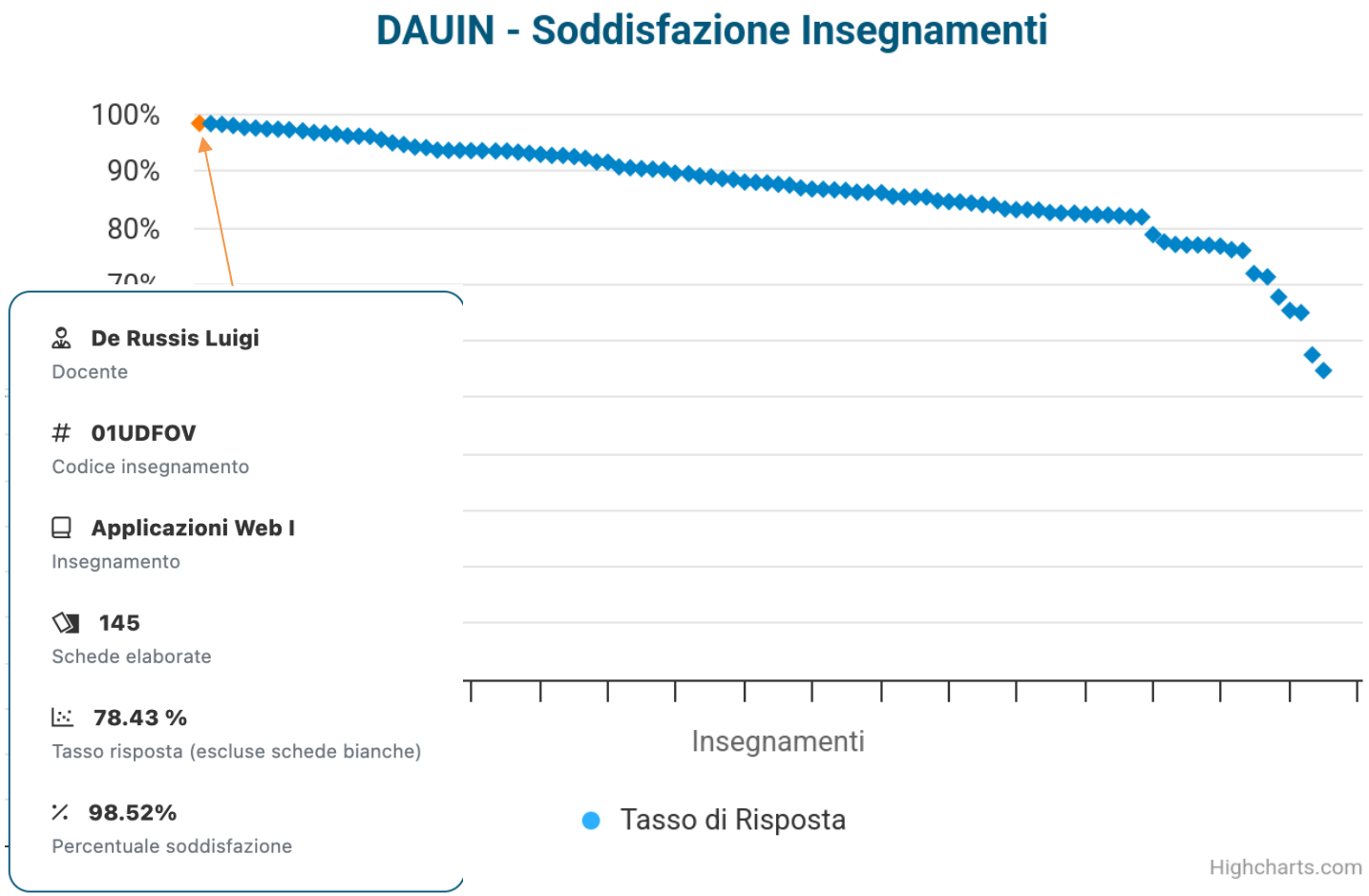
- Usability
- Interface design
- Human centered processes

- Distributed Architectures
- Protocols
- Foundations

- Mobile Front-End
- Mobile device programming

You are here

2024 End-of-course Questionnaire



- Relevant critiques*:
  - Labs should be more structured and supportive
  - Need more time for in-class exercises
  - Not enough time to reason about each topic

** considered in planning this year's course*

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)
10. 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		R4b			
10:00		R4b			
11:30					
13:00					
14:30	3P				
16:00	3P				
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 (i.e., **readings**)
  - published well *in advance*

Laboratories

- Starting 2nd week
- In rooms with power outlets at the desks

- Build a simple project extracted from recent **Exam assignments**
- In groups of 2-5 students
- **Weekly step-by-step** design and implementation, following the course topics
- Text online, some days in advance

- Support, discussion, help, during the lab sessions
- Attendance to the labs is **highly recommended**

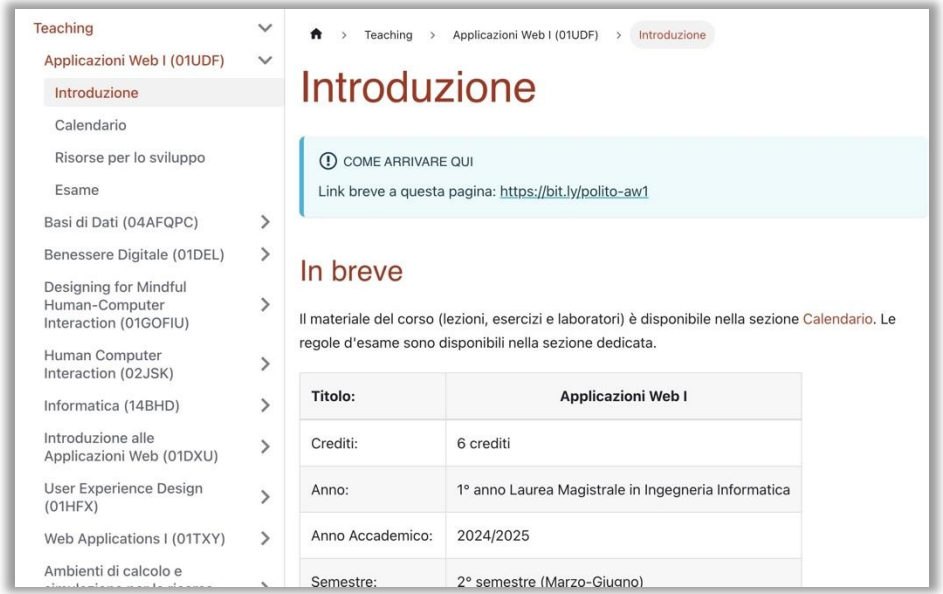
Groups and Laboratories

- 2 slots:
  - Tuesday, 08:30-10:00, room R4b, surnames: A-L
  - Tuesday, 10:00-11:30, room R4b, surnames: M-Z

- Groups to be created within the two slots above
  - People from one slot *can* join a group in another slot, if needed

- Group composition via GitHub Classroom by the end of **March 3, 2025**:
  - https://classroom.github.com/a/BIwZinRv

Learning Material



- Course website – https://bit.ly/polito-aw1
  - Slides (in English)
  - Full schedule
  - Links and supplementary material
- Video lectures (screencasts)
  - YouTube - https://youtube.com/playlist?list=PLs7DWGc_wmwQoZtq-xf4mO4sw9VuZkTd0
  - 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/+K8wNMKS-Jww3MDM8
- Emails can be an **alternative** for slower, more articulated, and private individual communications

Office Hours

- **Why?**
  - An opportunity to discuss issues or needs
  - To clarify information or ask questions on the course
  - To discuss academic or career goals (e.g., thesis, what to do next year)
  - To know more about certain topics
  - …
- **When?**
  - **Tuesday 15:00-16:00, Thursday 11:30-12:30, Friday 14:00-15:00** in my office; book a 30-min slot at: https://calendly.com/luigi-derussis/office-hours
  - Also: on request, in person or remotely

Exam: Two Parts

1. **Project development** (up to 26 points)
   - Individual
   - 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
- 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
- Teacher's Evaluation
  - running the application on the teacher’s laptop
  - examining the code

Oral Discussion

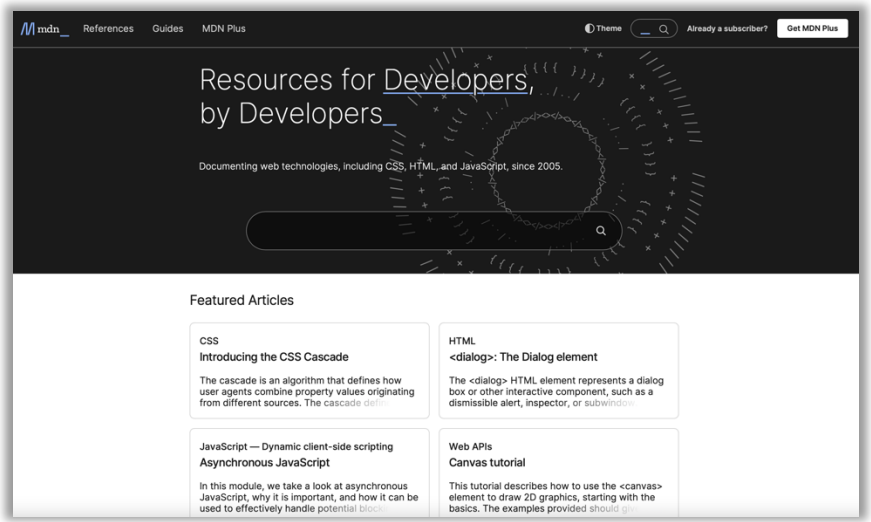
Goals

- 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

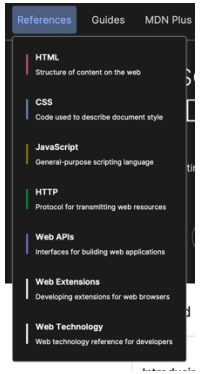
Evaluation Criteria

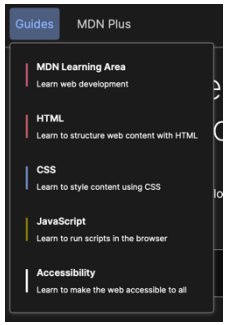
- Theoretical and practical knowledge of the project design
- Theoretical and practical knowledge of the project code base
- Readiness and clarity in the replies

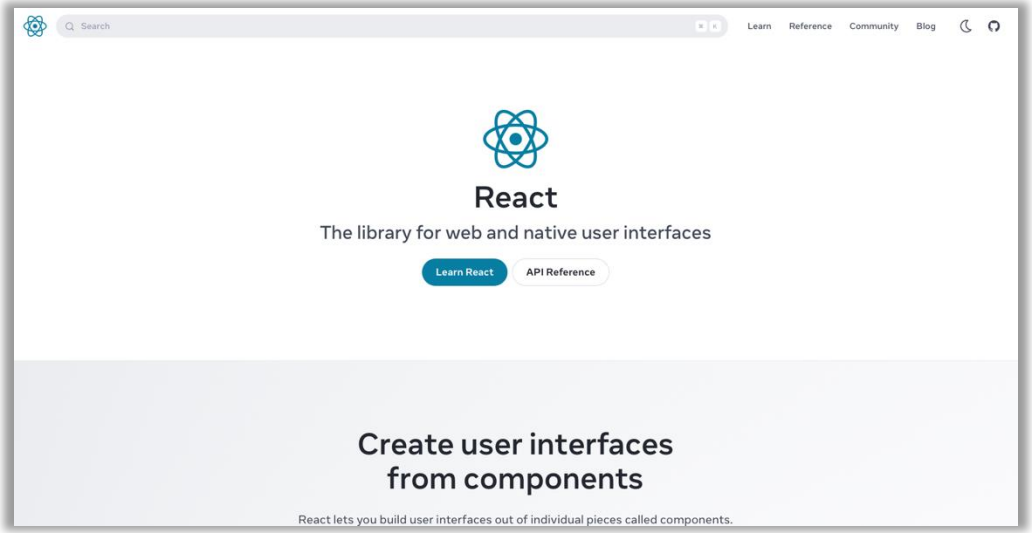
Resources (fundamentals)



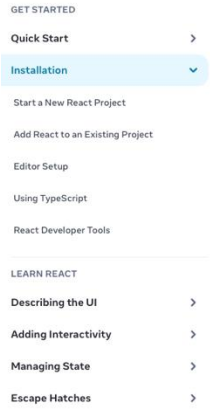
Mozilla Developer Network (**MDN**)
https://developer.mozilla.org/



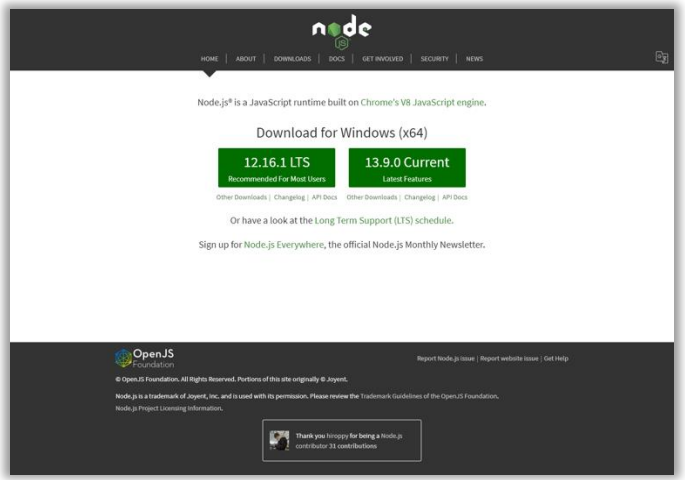




React
https://react.dev



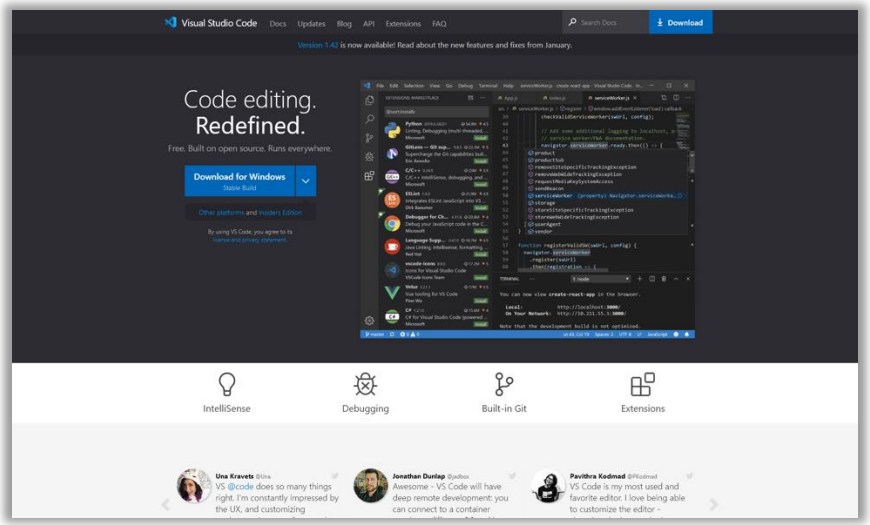
Tools



Node.js runtime
Version 22.x LTS
https://nodejs.org/en/

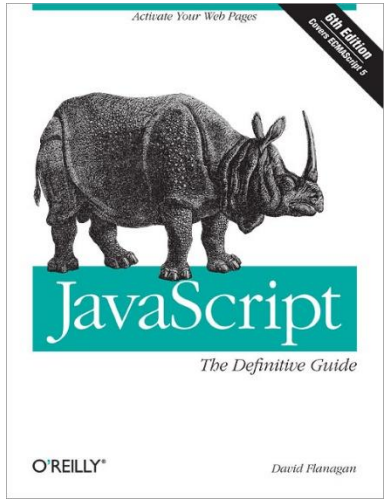
React Developer Tools
Extension for Chrome and Firefox

Programming Environment

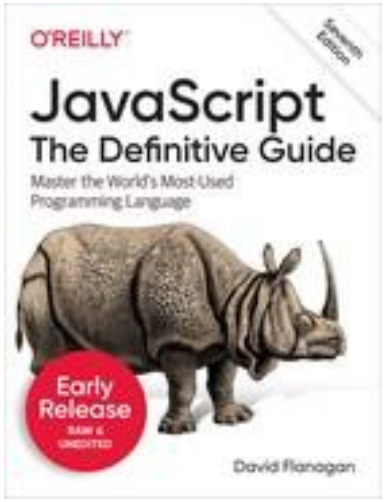


Visual Studio Code
https://code.visualstudio.com/

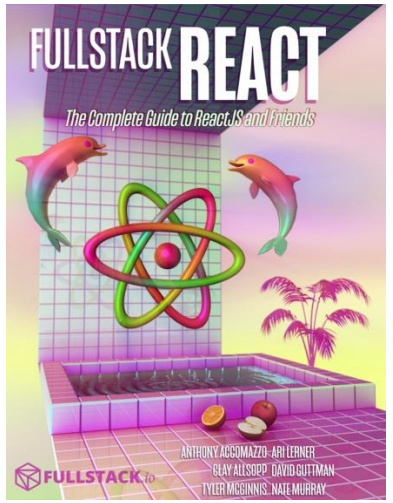
Resources (books)



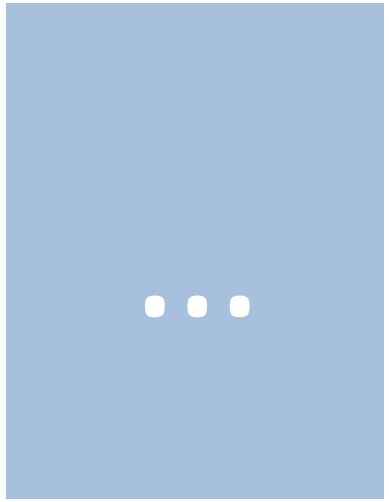
JavaScript: The Definitive Guide, 6th Edition
By David Flanagan
ISBN 978-0596805524
Release Date: May 2011
(not very updated…)



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

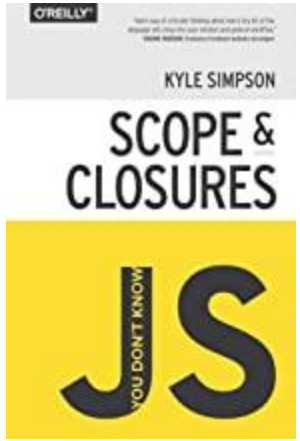


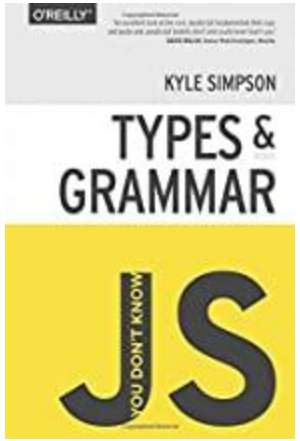
Fullstack React
By Anthony Accomazzo, Nate Murray, Ari Lerner, Clay Allsopp, David Guttman, and Tyler McGinnis
https://www.newline.co/fullstack-react
Release: r40 (January 2020)

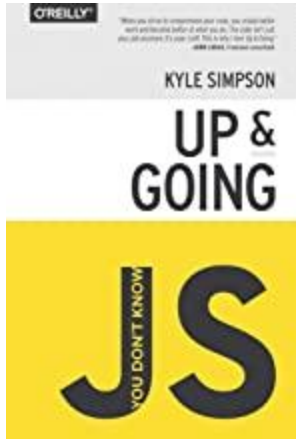


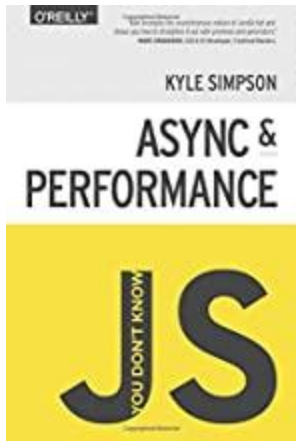
… and many others

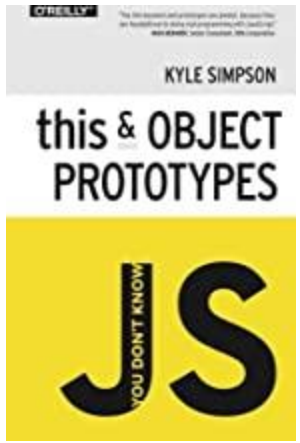
Resources (on-line books)

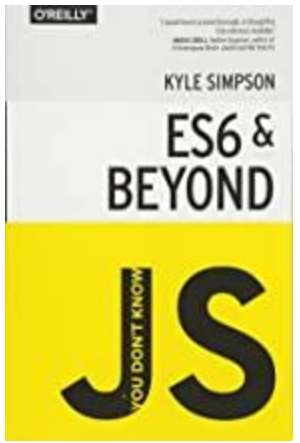




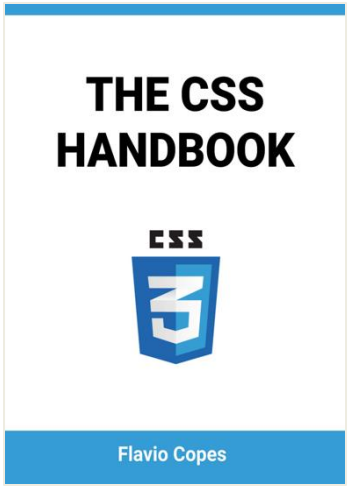


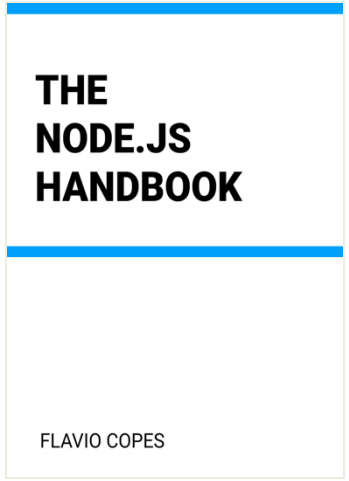






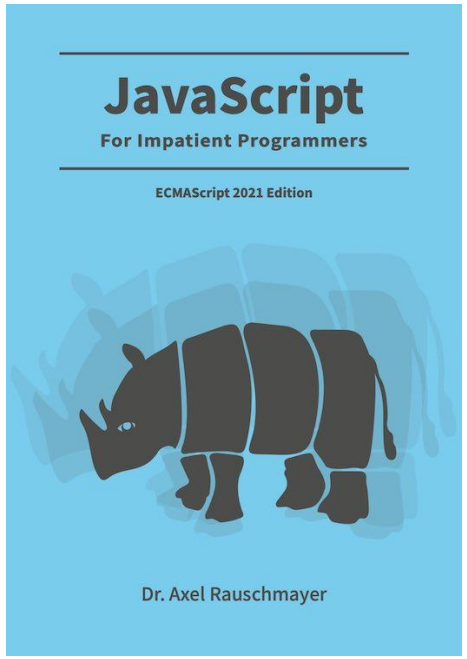
You Don't Know JS Yet (book series) - 2nd Edition
By Kyle Simpson (@getify)
https://github.com/getify/You-Dont-Know-JS





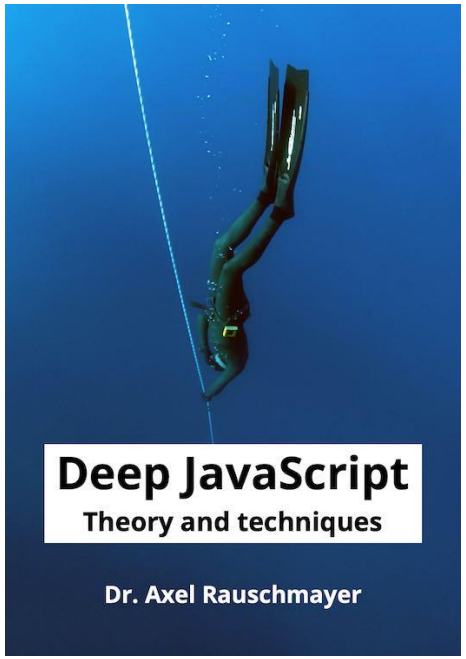
Flavio Copes Handbooks
https://flaviocopes.com/

Resources (on-line books)

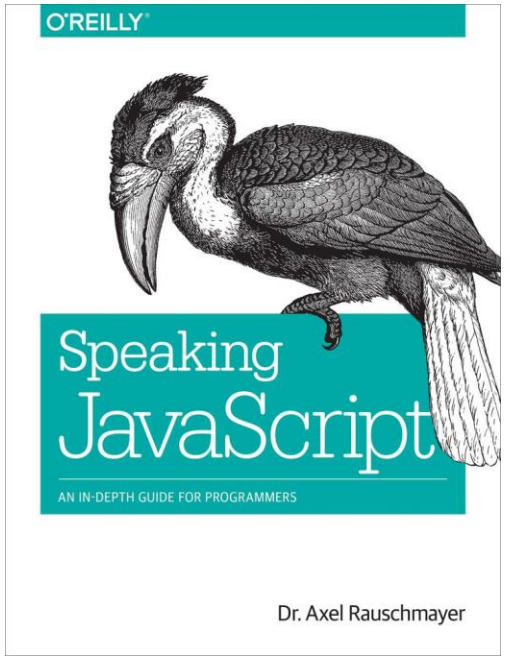


https://exploringjs.com/impatient-js/index.html

+

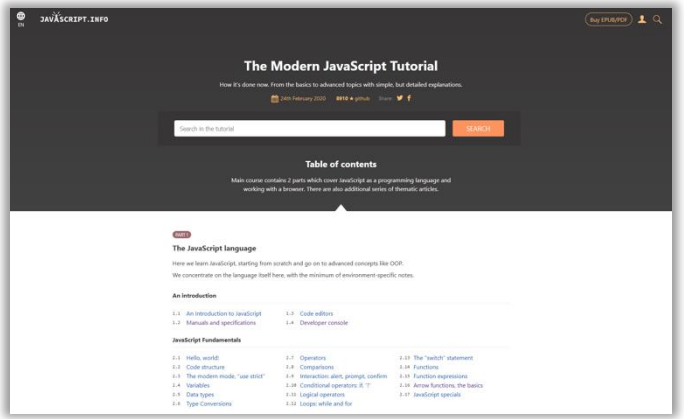


https://exploringjs.com/deep-js/index.html

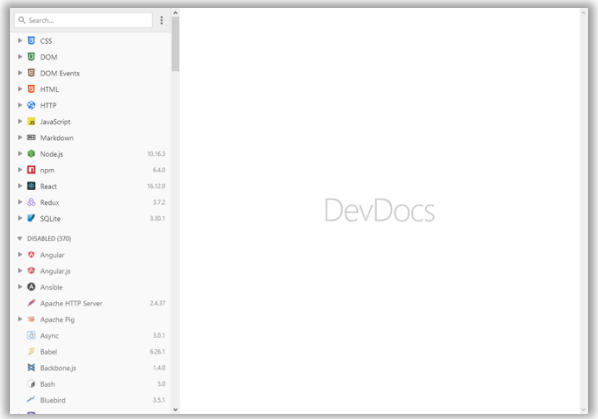


http://speakingjs.com/

More Resources…



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



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

… and many others

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/