

<WA/>

2025

# Web Applications

## Introduction to the course

Enrico Masala

Antonio Servetti



# 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
- Some attention on basic security aspects in web applications
  - *More on this in other courses in “Cybersecurity Engineering” course of study*

# 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



# 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; API design
4. HTML, CSS, Bootstrap
5. JS: modules and other topics, + JS in the browser
6. Intro to React
7. React: props and state
8. React: context, life cycle, forms
9. React router
10. Data fetching and client-server interaction (in React)
11. Authentication

# Course Organization

- Classes
  - 6 (or 3) h/week
  - Lectures + Exercises (*mixed*)
- Laboratories (room 9i and 10i)
  - 1.5 or 1.5+1.5 h/week (using class slots)
  - 3 Lab groups (see later for the split)
  - Starting 2<sup>nd</sup> week
- **Detailed schedule week-by-week**
  - <https://github.com/polito-WA-2025/.github/blob/main/profile/SCHEDULE.md>

	MO	TU	WE	TH	FR
08:30					
10:00					
11:30					
13:00	29B			1P	
14:30	29B			1P	
16:00		10i			9i
17:30					9i

# Classes

- In person, (mostly) 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 before a lecture
  - relatively *short* and published *in advance*

# Laboratories

- Starting March 4, 2025
- In rooms with power outlets at the desks
  - No computers are available in the room, bring your own
- 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

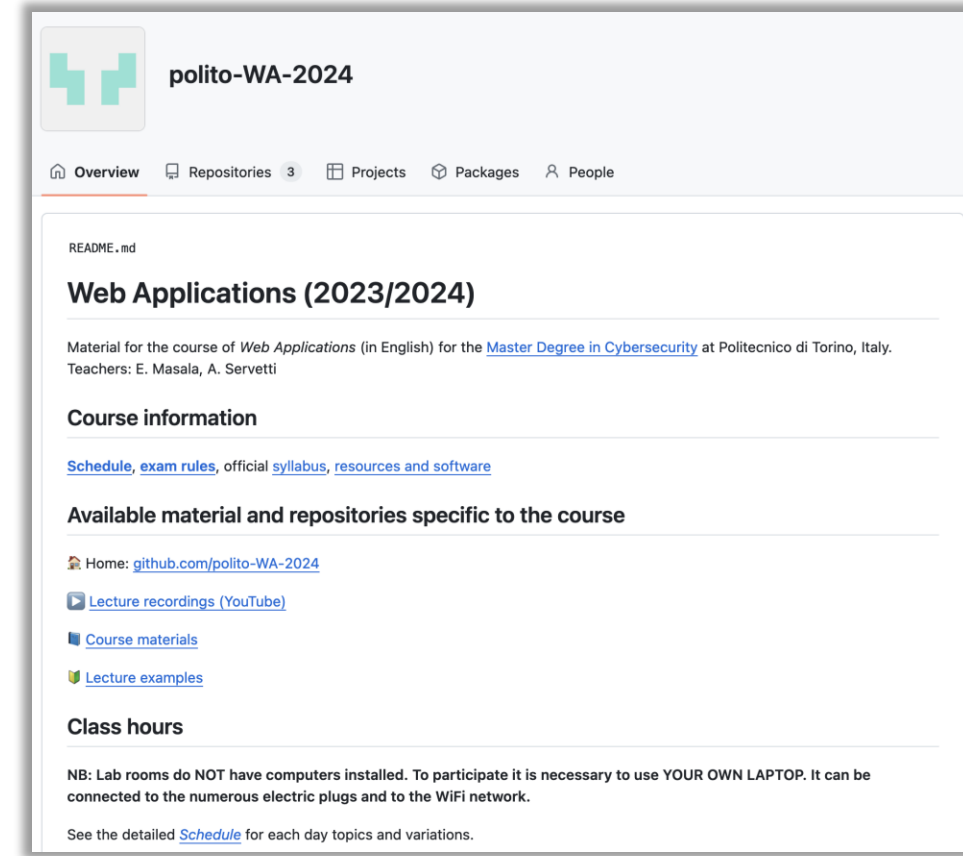
# Laboratories

- 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
- Three slots, divided by last name:
  - AA-FE
  - FF-MZ
  - NA-ZZ



# Learning Material

- Everything is on GitHub:  
<https://github.com/polito-WA-2025>
- Course website
  - Slides
  - Full schedule
  - Links and supplementary material
  - Examples, exercises, labs, exams, ...
- Video lectures (screencasts)
  - YouTube - <https://www.youtube.com/playlist?list=PLuZyhAOPm9pNtxSPd9qRvwBEYoDXjgeYQ>
  - Portale della Didattica (download only)



# Communications



- We will use **Telegram** for the main communications
  - Among students, with teachers, etc.
  - Announcements and official information, and Q&A (using “topics” in Telegram)
- Feel free to contact the teachers for feedback and questions
  - **questions** of general interest (including exam) must be posted in the group, so that everybody can see the answer. NB: Do not exchange suggestions to solve the exam.
- Link to the Telegram group: <https://t.me/+nOA1i4v9FUkyMWE0>
  - Any nickname is ok, but tell who you are for personal issues (especially in DM)
- Emails can be an **alternative** for slower, more articulated, and private individual communications

# About the Exam: Project

- **Individual** project development
  - 20 days of time for development
- Develop a web application according to the given functional specification, using the approach/technologies seen during the course
  - React + JavaScript, Node.js + Express, SQLite
  - Different technologies/approaches will NOT be accepted and lead to exam failure, without testing the project. If in doubt, ask the teacher in advance!
- Assignment published 20 days before each official exam date
  - Different for each exam date, submission deadline before the exam date  
(via *GitHub Classroom* – more details in specific instructions on course website)

# After Project Submission

- Similarity checks will be run: excessively similar solutions will lead to exam failure without oral discussion
- The teacher will automatically load all solutions on a Linux server, so that they are ready to be tested during the oral discussion
  - **STRICT** conformance to submission instruction is extremely important!!
  - Check carefully library install and imports, filename upper/lowercase, ports, etc.
- A schedule for oral exams is prepared and published
- *Complete and detailed exam rules and submission instructions in the course website (under "Exams")*




# Oral Discussion

- Test of the application in student's presence, running it on a Linux server
- Assessment criteria:
  - Evaluation of the application architecture, responses to user actions (UI update, API called, content of API requests and responses, etc.)
  - Evaluation of the code (both client and server): programming patterns, security management and checks, code clarity, code uniformity and coherence, etc.
  - Evaluation of the student's theoretical and practical knowledge of the project design, code base, readiness and clarity in the replies: fundamental to ensure that each student developed the web application by her/himself
- NB: It is NOT a presentation of the project given by the student

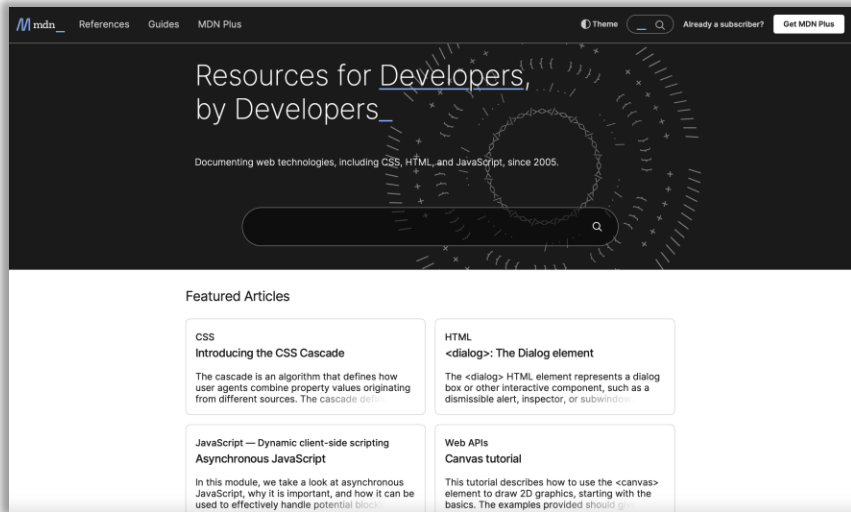
# Exam Caveats

- Many students have experience in web development. In this case:
  - DO NOT use technologies not explained in the course; if in doubt, ask the teacher.
  - In any case, full knowledge of all technologies used in the project is expected and required
- The objective of the exam is not only to create an application that behaves as specified in the exam specification
- The exam is especially about how to correctly implement features, giving **proper consideration** to the various aspects involved in the design
  - API design, what should stay on the client, on the server, which information is transmitted, how security is handled, how code checks for authorizations, etc.
- Also, being able to explain code behavior is a fundamental requirement: if the student cannot explain, the exam is failed (regardless of the app quality)

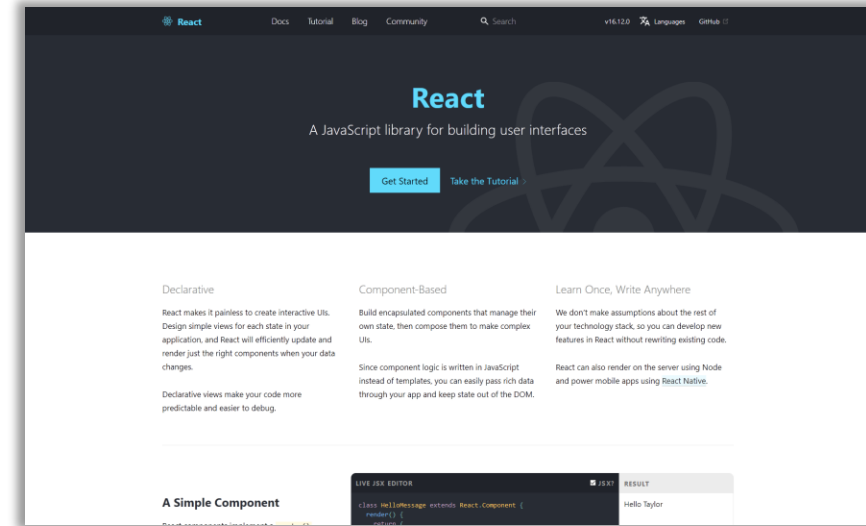
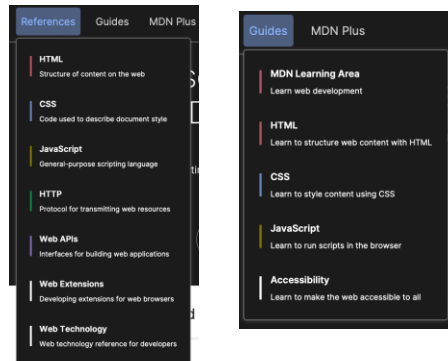
# AI and Code Development

- Many automatic tools can generate code nowadays
  - **ChatGPT, Copilot**, etc.
- Typically good for small pieces of code, can be confusing in larger projects
- If you use them, be careful (a lot)! 
-  You **MUST UNDERSTAND** the code they propose
-  You **MUST BE ABLE TO EXPLAIN** in detail **ALL** the code you included in your project submission, regardless of the source, not to fail the exam
  - Code coherency, uniformity and style will be evaluated at the exam (dumb cut & paste not recommended)

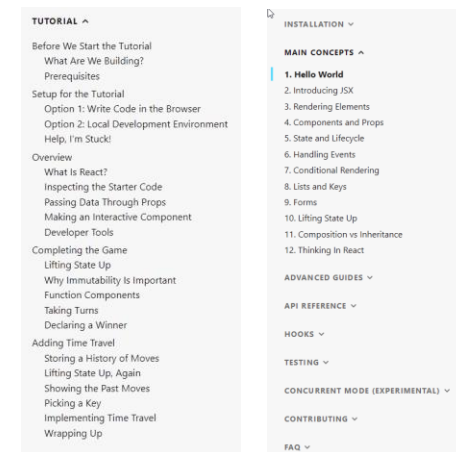
# Resources (fundamentals)



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

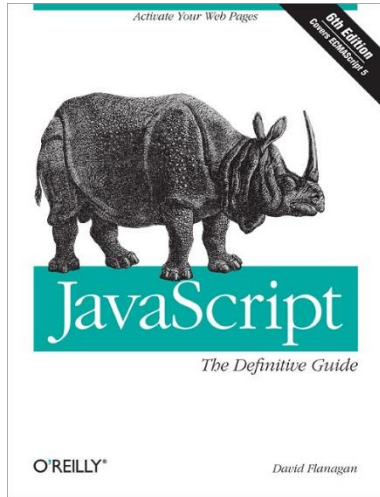


React Library  
<https://reactjs.org/>

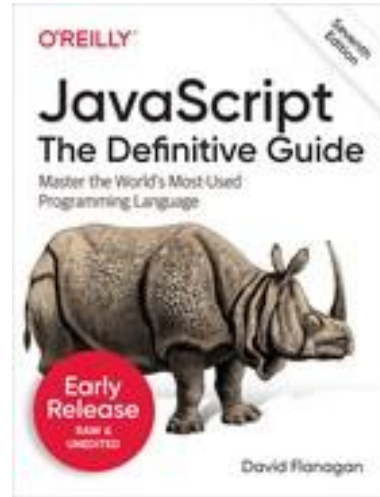




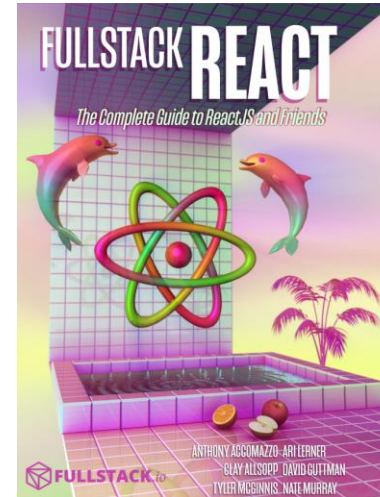
# Resources (books)



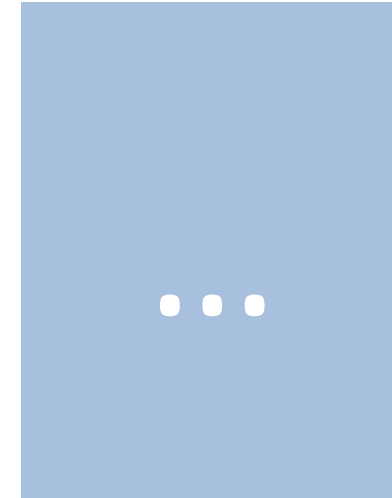
JavaScript: The Definitive Guide,  
6th Edition  
By David Flanagan  
ISBN 978-0596805524  
*Release Date: May 2011*  
(not very updated...)



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

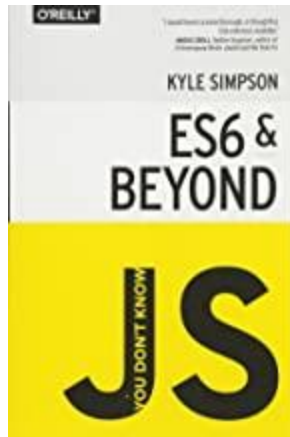
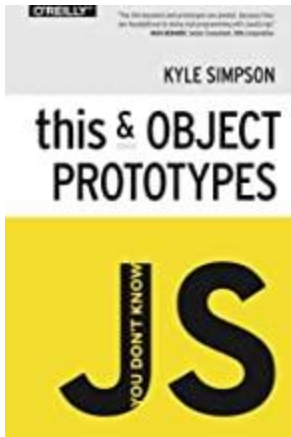
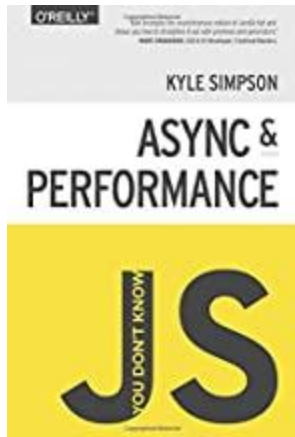
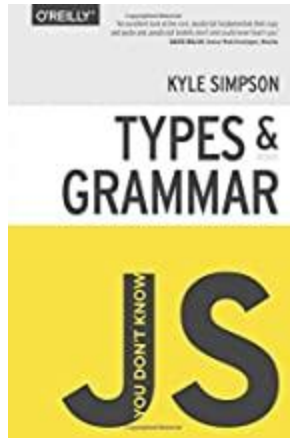
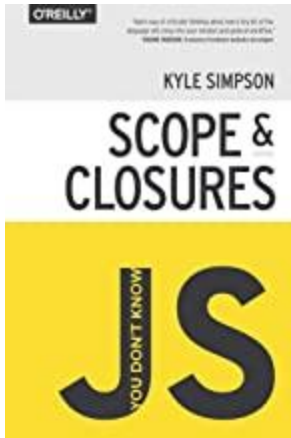


Fullstack React  
By Anthony Accomazzo, Nate  
Murray, Ari Lerner, Clay  
Allsopp, David Guttman, and  
Tyler McGinnis  
<https://www.newline.co/fullstack-react>  
*Release: r40 (January 2020)*

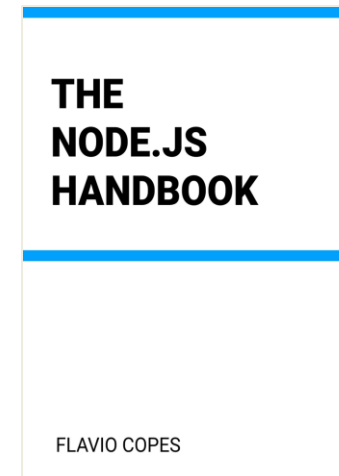
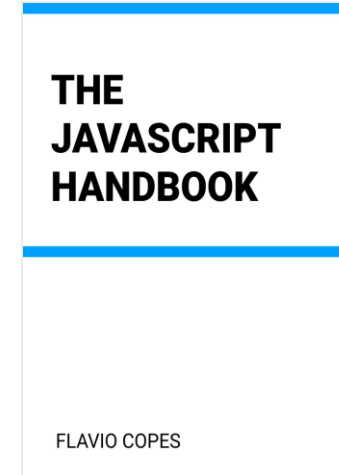


... and many others

# Resources (on-line books)

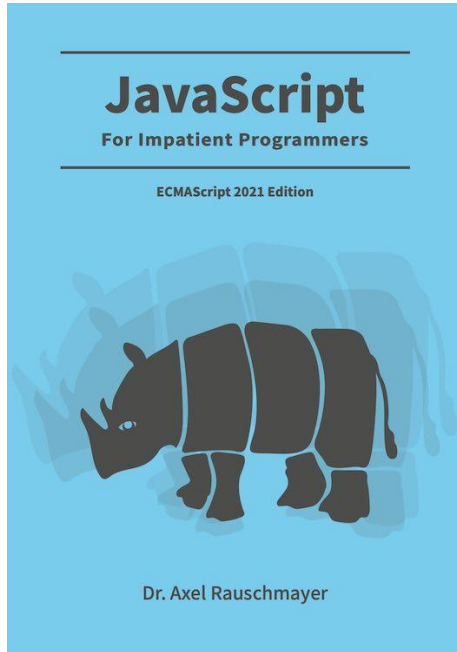


You Don't Know JS Yet (book series) - 2nd Edition  
By Kyle Simpson (@getify)  
<https://github.com/getify/You-Dont-Know-JS>

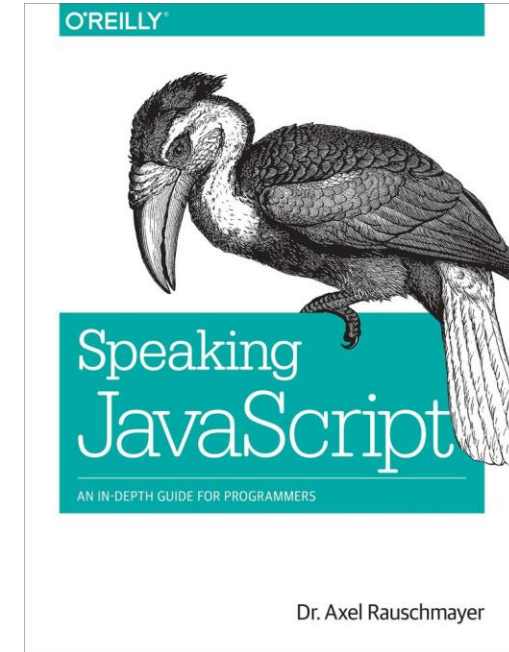
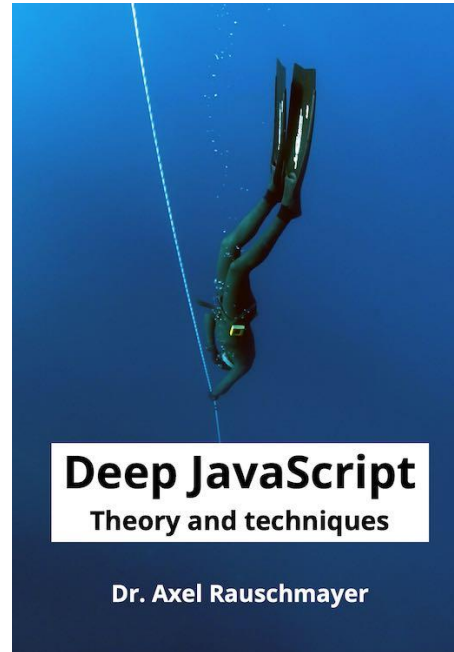


Flavio Copes Handbooks  
<https://flaviocopes.com/>

# Resources (on-line books)



+

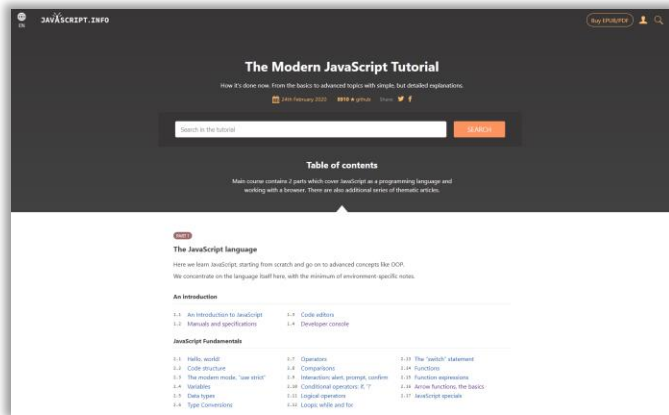


<https://exploringjs.com/impatient-js/index.html>

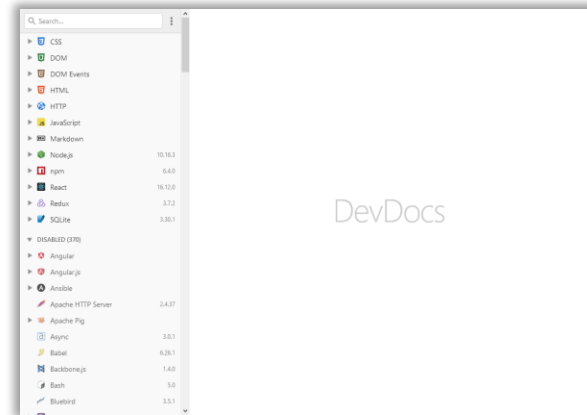
<https://exploringjs.com/deep-js/index.html>

<http://speakingjs.com/>

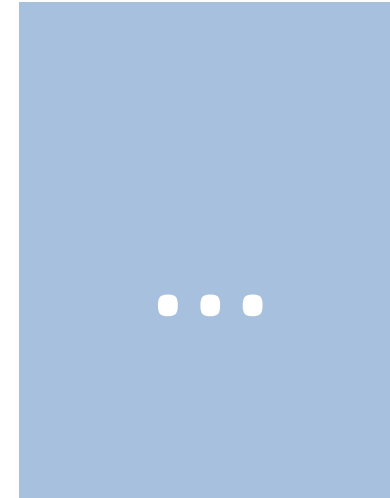
# More resources...



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/>

