

<WA1/>

<AW1/>

2023

# Applicazioni Web I

# Web Applications I

## Introduction to the course

Fulvio Corno, Luigi De Russis, Enrico Masala

Luca Mannella, Juan Pablo Saenz, Antonio Servetti

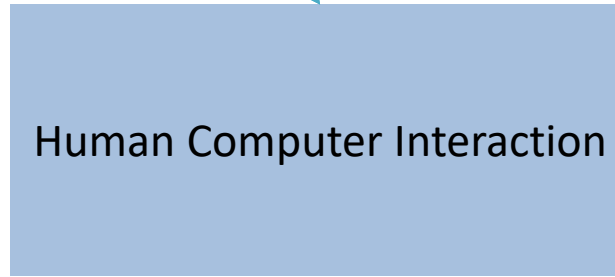


# Goal

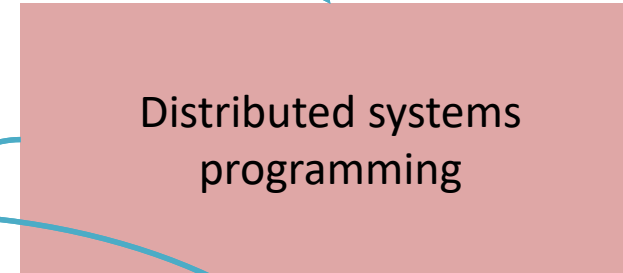
- Understanding web architectures
- Understanding and mastering web application design and development
- Gaining in-depth knowledge of the JavaScript language and ecosystem
- Becoming familiar with one of the most popular JavaScript frameworks (React)
- ...with special focus on the front-end

# The Bigger Picture

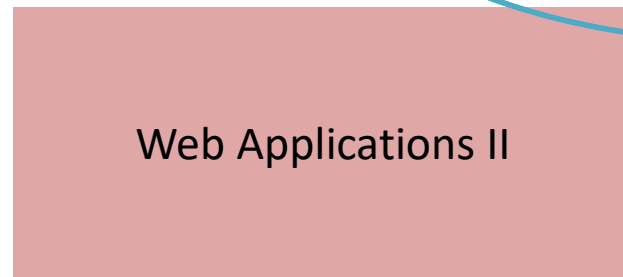
- Web architecture
- JavaScript
- Browsers
- **Front-End programming**
- **Back-end programming**
- Scalability
- Large-scale



- Usability
- Interface design
- Human centered processes



- Distributed Architectures
- Protocols
- Foundations



- Mobile Front-End
- Mobile device programming



You are here

# What We Will Learn

## JavaScript as a language

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



JS

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

# Course Organization

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

	MO	TU	WE	TH	FR
08:30				8I	
10:00		4P		8I	
11:30		4P			
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 before a lecture
  - relatively *short* and published *in advance*

# Laboratories

- Starting 09/03/2023
- 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

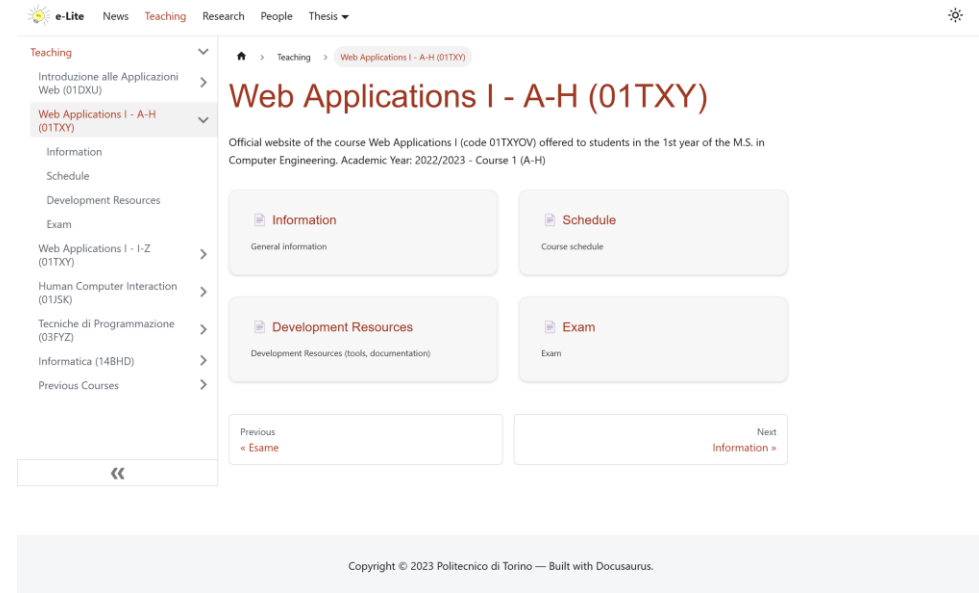


# Laboratories

- 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:
  - AA-DE
  - DI-ZZ

# Learning Material

- Course website – <https://bit.ly/polito-wa1-ah>
  - Slides (in English)
  - Full schedule
  - Links and supplementary material
- Video lectures (screencasts)
  - YouTube - <https://youtube.com/playlist?list=PLqRTLLwsxDL8WgeiSZVJzjEr1f9aHy2gz>
  - Portale della Didattica
- GitHub - <https://github.com/polito-WA1-AW1-2023>
  - Examples, exercises, labs, exams, ...



The screenshot shows the course website for "Web Applications I - A-H (01TXY)". The page features a navigation menu on the left with options like "Information", "Schedule", "Development Resources", and "Exam". The main content area includes sections for "Information" (General information), "Schedule" (Course schedule), "Development Resources" (Development Resources (tools, documentation)), and "Exam" (Exam). There are also "Previous" and "Next" buttons at the bottom. The footer indicates "Copyright © 2023 Politecnico di Torino — Built with Docusaurus."



# 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/+7rIW\\_ZT\\_2ANIZGZk](https://t.me/+7rIW_ZT_2ANIZGZk)
- Emails can be an **alternative** for slower, more articulated, and private individual communications

# About the Exam

## 1. Project development

- Individual
- up to 26 points
- 20 days of time

## 2. Oral discussion (on the project)

- individual and mandatory
- “live” correction of the submitted project and discussion
- up to 6 points
- when: the official exam day (or starting from that day)

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 a clean recent Linux distro (with node)
  - 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)

Resources for Developers, by Developers

Documenting web technologies, including CSS, HTML, and JavaScript, since 2005.

Featured Articles

- CSS**  
Introducing the CSS Cascade  
The cascade is an algorithm that defines how user agents combine property values originating from different sources. The cascade defines...
- HTML**  
<dialog>: The Dialog element  
The <dialog> HTML element represents a dialog box or other interactive component, such as a dismissible alert, inspector, or subwindow.
- JavaScript** — Dynamic client-side scripting  
Asynchronous JavaScript  
In this module, we take a look at asynchronous JavaScript, why it is important, and how it can be used to effectively handle potential block...
- Web APIs**  
Canvas tutorial  
This tutorial describes how to use the <canvas> element to draw 2D graphics, starting with the basics. The examples provided should g...

React

A JavaScript library for building user interfaces

Get Started Take the Tutorial

**Declarative**  
React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.  
Declarative views make your code more predictable and easier to debug.

**Component-Based**  
Build encapsulated components that manage their own state, then compose them to make complex UIs.  
Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

**Learn Once, Write Anywhere**  
We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.  
React can also render on the server using Node and power mobile apps using React Native.

```
class HelloMessage extends React.Component {
  render() {
    return 'Hello Taylor';
  }
}
```

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

References Guides MDN Plus

- HTML: Structure of content on the web
- CSS: Code used to describe document style
- JavaScript: General-purpose scripting language
- HTTP: Protocol for transmitting web resources
- Web APIs: Interfaces for building web applications
- Web Extensions: Developing extensions for web browsers
- Web Technology: Web technology reference for developers

Guides MDN Plus

- MDN Learning Area: Learn web development
- HTML: Learn to structure web content with HTML
- CSS: Learn to style content using CSS
- JavaScript: Learn to run scripts in the browser
- Accessibility: Learn to make the web accessible to all

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

TUTORIAL

- Before We Start the Tutorial
- What Are We Building?
- Prerequisites
- Setup for the Tutorial
- Option 1: Write Code in the Browser
- Option 2: Local Development Environment
- Help, I'm Stuck!
- Overview
- What is React?
- Inspecting the Starter Code
- Passing Data Through Props
- Making an Interactive Component
- Developer Tools
- Completing the Game
- Lifting State Up
- Why Immutability Is Important
- Function Components
- Taking Turns
- Declaring a Winner
- Adding Time Travel
- Storing a History of Moves
- Lifting State Up, Again
- Showing the Past Moves
- Picking a Key
- Implementing Time Travel
- Wrapping Up

INSTALLATION

MAIN CONCEPTS

- 1. Hello World
- 2. Introducing JSX
- 3. Rendering Elements
- 4. Components and Props
- 5. State and Lifecycle
- 6. Handling Events
- 7. Conditional Rendering
- 8. Lists and Keys
- 9. Forms
- 10. Lifting State Up
- 11. Composition vs Inheritance
- 12. Thinking in React

ADVANCED GUIDES

API REFERENCE

HOOKS

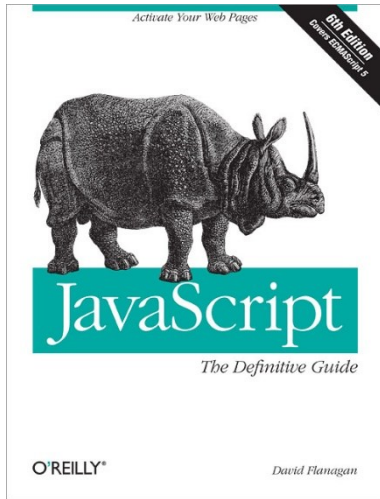
TESTING

CONCURRENT MODE (EXPERIMENTAL)

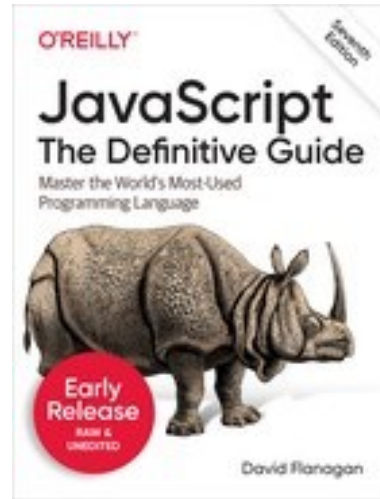
CONTRIBUTING

FAQ

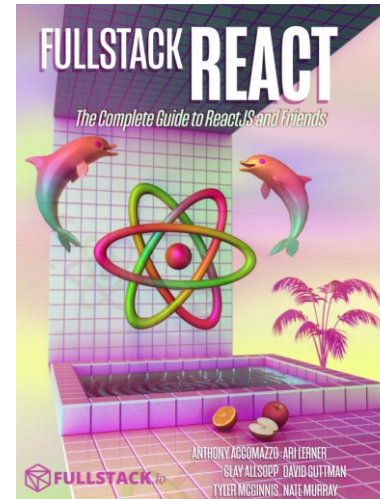
# Resources (books)



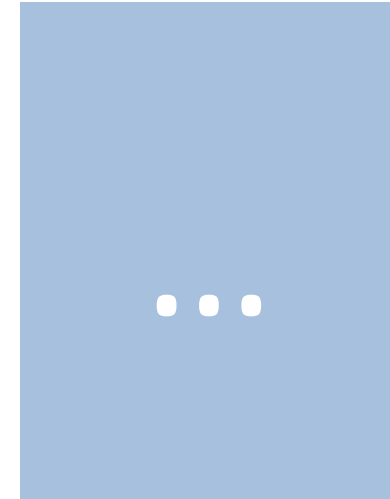
JavaScript: The Definitive Guide,  
6th Edition  
By David Flanagan  
ISBN 978-0596805524  
*Release Date: May 2011*  
(not very updated...)



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)



GET STARTED



ASYNC & PERFORMANCE



SCOPE & CLOSURES



this & OBJECT PROTOTYPES



TYPES & GRAMMAR



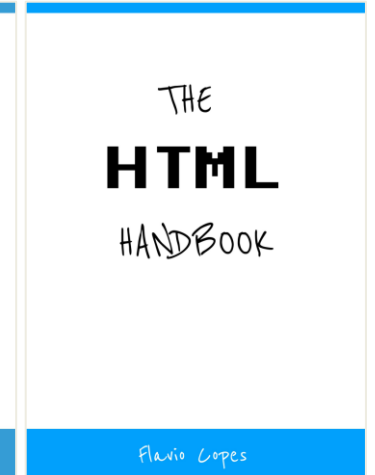
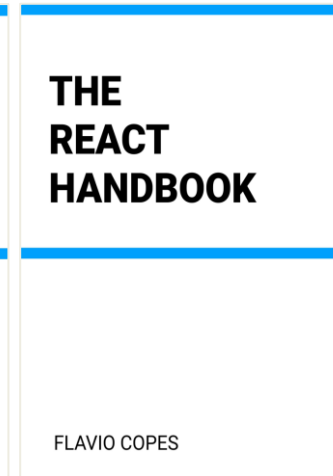
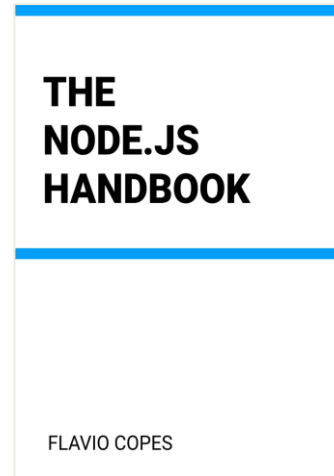
ES6 & BEYOND



UP & GOING

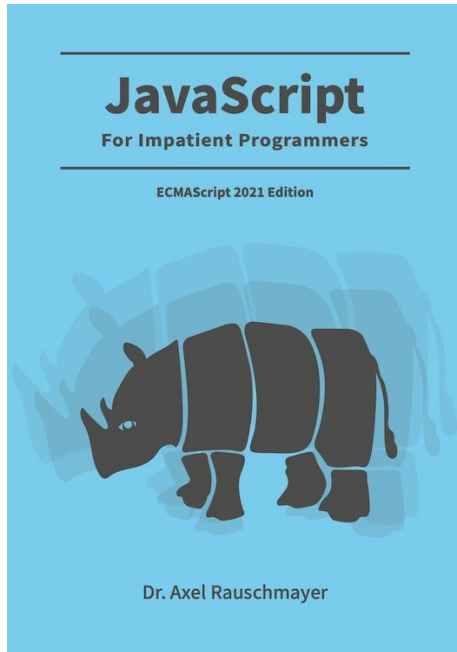


You Don't Know JS Yet (book series) - 2nd Edition  
By Kyle Simpson (@getify)  
<https://github.com/getify/You-Dont-Know-JS>

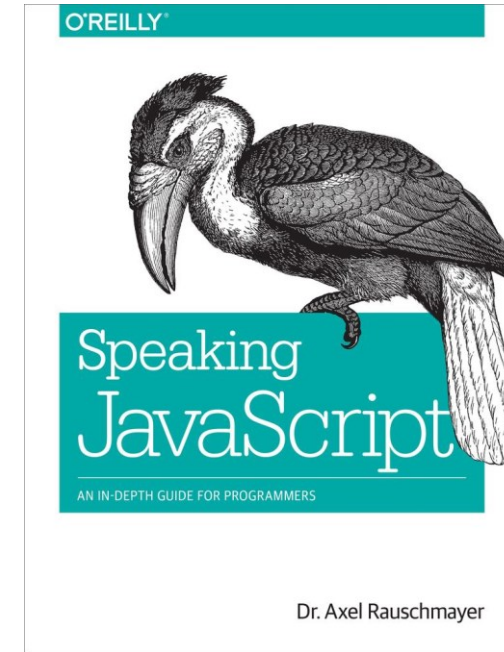
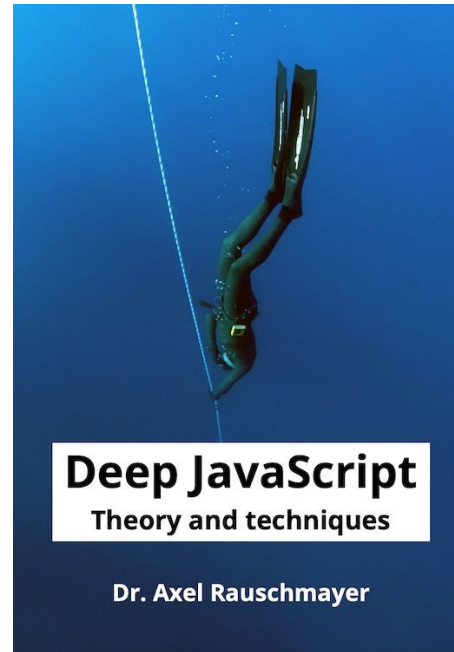


Flavio Copes Handbooks  
<https://flaviocopes.com/>

# Resources (on-line books)



+



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

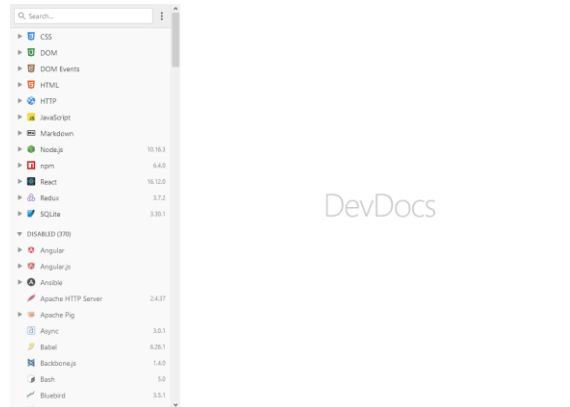
<https://exploringjs.com/deep-js/index.html>

<http://speakingjs.com/>

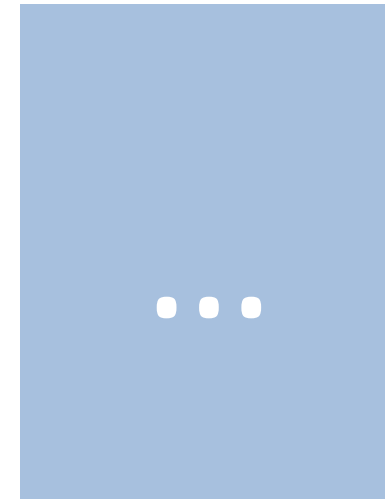
# More resources...



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

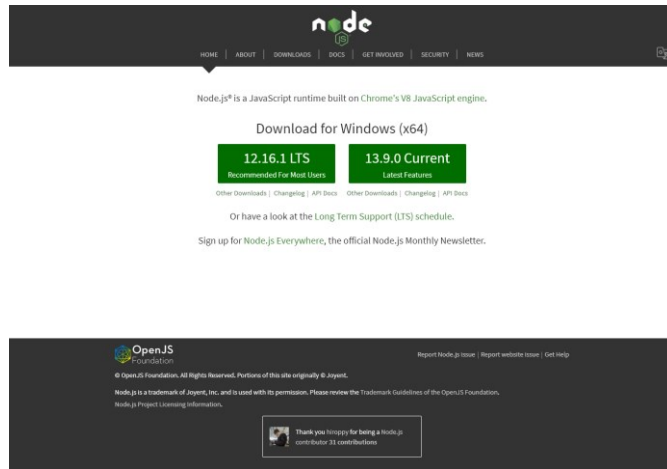


DevDocs: API Documentation  
Browser  
<https://devdocs.io/>



... and many others

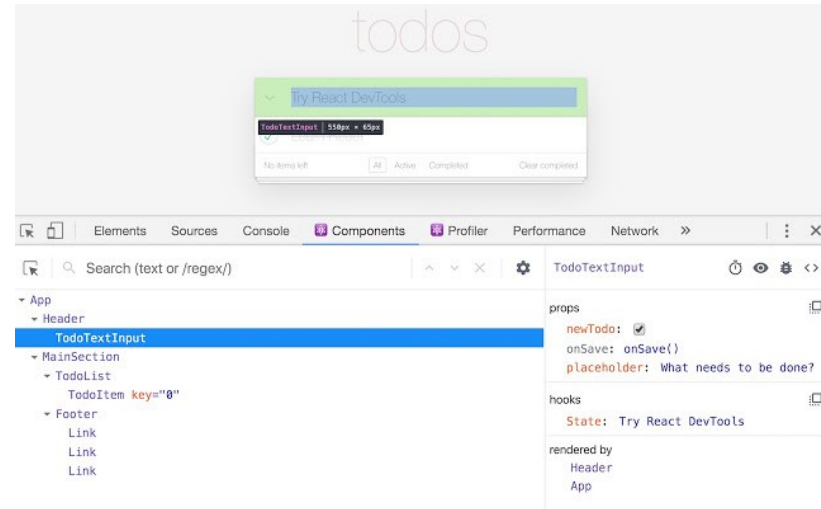
# Tools



Node.js runtime  
Version 18.14 LTS

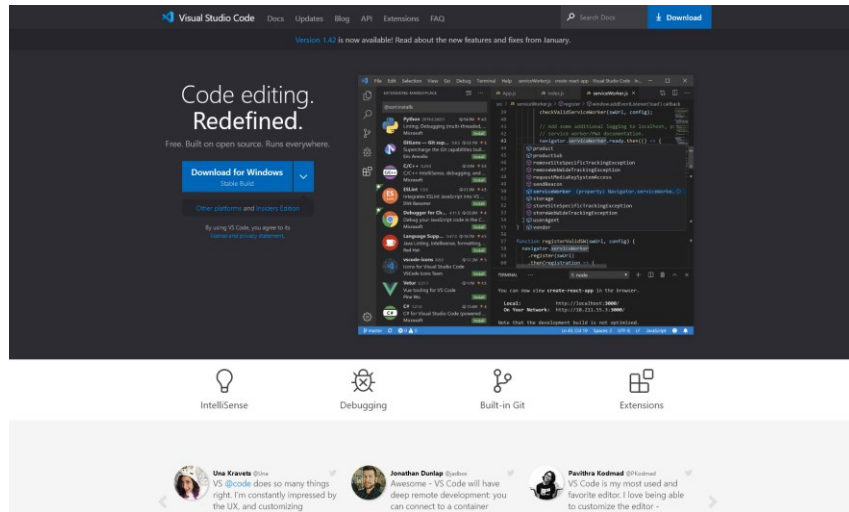
<https://nodejs.org/en/>

Install on Linux using the instructions on  
<https://github.com/nodesource/distributions>



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/>

