

How to learn JavaScript in 10 not so easy steps

bit.ly/learnjs by @codylindley

Why isn't it simple?

Too much bad information

The browsers & DOM haven't exactly made it easy
but it's getting much better

It's only recently (last couple of years) JavaScript
has been taken seriously as a programming
language

Learn JavaScript, not:

BOM (Browser Object Model), DOM (Document

Object Model), CoffeeScript, Dart, Plug-ins,

Libraries, Frameworks, Design Patterns, Modules,

AMD, Node.js, Common JS, Application

Architecture, MVC, MVVC, UI Widgets, OOP,

Event Loop, Event Driven Programming

You want to learn
JavaScript, then learn
JavaScript

Wait, which JS are we
learning? ES3? ES4? ES5?
ES6?

Step 0. Build a base, a mental model

Read: How Computer Programming Works

Watch: Introduction to Programming and computer science (wth... you said learn JS not Python)

Hands on: [codecademy.com - Learn to code
JavaScript course](https://www.codecademy.com/learn/learn-javascript)

Step 1. The basics

Read: JavaScript: The Definitive Guide, 6th Edition
(chapters 1 to 8)

Read: Professional JavaScript for Web Developers, 3rd Edition (chapters 1 to 4)

Watch: Douglas Crockford: The JavaScript Programming Language

Watch: JS 101, JS 102, JS 103

Repeat step 1

Step 2. The not so basics

Read: JavaScript: The Definitive Guide, 6th Edition
(chapters 9)

Read: Professional JavaScript for Web Developers, 3rd Edition (chapters 5, 6, 7)

Watch: Crockford on JavaScript - Chapter 2: And Then There Was JavaScript & Level 7: ECMAScript 5: The New Parts

Repeat step 2

Step 3. Add style and wisdom

Read: [Principles of Writing Consistent, Idiomatic JavaScript](#)

Read: [Google JavaScript Style Guide](#)

Read: [Code Conventions for the JavaScript Programming Language](#)

Read: <http://dojotoolkit.org/community/styleGuide>

Read: [Felix's Node.js Style Guide](#)

Watch: [Crockford on JavaScript - Section 8: Programming Style and Your Brain](#)

Step 4. Learn what to un-learn

Read: [JavaScript: The Good Parts](#)

Watch: [JavaScript: The Good Parts \(video\)](#)

Step 5. Embrace first-class functions

Read: [Functional Programming](#)

Watch: [Crockford on JavaScript - Act III: Function the Ultimate](#)

Step 6. Make sure you get it, focus on the high notes

Read: [JavaScript Enlightenment](#)

Read: [JavaScript Garden](#)

Watch: [Learning to love JavaScript](#)

Interaction: <http://learnjs.info/>

Step 7. Put the correct gas in your programs

Read: [High Performance JavaScript - Build Faster Web Application Interfaces](#) (chapter 2, 4, 5)

Watch: [High Performance JavaScript](#)

Watch: [Speed Up Your JavaScript](#)

Step 8. Get intellectual with Dmitry Soshnikov

ECMA-262-3 in detail:

[Chapter 1. Execution Contexts](#)

[Chapter 2. Variable object](#)

[Chapter 3. This](#)

[Chapter 4. Scope chain](#)

[Chapter 5. Functions](#)

[Chapter 6. Closures](#)

[Chapter 7.1. OOP: The general theory](#)

[Chapter 7.2. OOP: ECMAScript implementation](#)

[Chapter 8. Evaluation strategy](#)

Notes:

[Note 1. ECMAScript. Bound functions.](#)

[Note 2. ECMAScript. Equality operators.](#)

[Note 4. Two words about "hoisting".](#)

Overview lectures:

[JavaScript. The Core](#)

ECMA-262-5 in detail:

[Chapter 0. Introduction.](#)

[Chapter 1. Properties and Property Descriptors.](#)

[Chapter 2. Strict Mode.](#)

[Chapter 3.1. Lexical environments: Common Theory.](#)

[Chapter 3.2. Lexical environments: ECMAScript implementation.](#)

Step 9. Get crazy, read the specs

Read: [Annotated ECMAScript 5.1](#)

Read: [ECMAScript 5.1 Language specification](#)

Step 10. Look up, look forward

Watch: Dave Herman: The Future of JavaScript

Watch: Brendan Eich: ECMA Harmony and the Future of JavaScript

Read: Harmony proposals

Parting advice

Mozilla currently has the most updated and best [JavaScript programming reference](#) for the language. Use it!

Educate yourself about browser/engine support for ECMAScript
[ECMAScript 5 compatibility table](#) & [ECMAScript Support Matrix](#)

Pick a library and study its code. I recommend [underscore.js](#) or
most anything you can find on [microjs.com](#)

Get an opinion about opinionated JavaScript by grokking [jslint](#) &
[jshint](#) validation