Summary and Perspectives

This lesson summarizes what we have covered in this whole course.

This course covers a lot of ground. Here are some key takeaways.

- JavaScript is a *programming language* created in 1995 for adding interactivity to web pages. Still the language of the web, it has also pervaded many other environments: servers, mobile devices, connected objects, etc.

- JavaScript has been standardized under the name **ECMAScript** and is continuously improved. Its latest major version is **ES2015**, also known as ES6.

- JavaScript is *dynamically typed*: the type of a variable is deduced from the value stored in it.

- JavaScript is a *multi-paradigm* language: you can write programs using an imperative, object-oriented or functional programming style.

- JavaScript's object-oriented model is based on *prototypes*.

- When run into a browser, JavaScript can use the **DOM** (Document Object Model) to access and update dynamically the structure of a web page.

- JavaScript can send **AJAX** calls (asynchronous HTTP requests) to exchange data with web servers or use web APIs, enabling the creation of *web applications*.

- The **Node.js** platform, created in 2009 and built on Chrome's V8 engine, brought JavaScript outside the browser.

- Node emphasizes modularity. Its `npm` package registry is the largest ecosystem of open source libraries in the world.

The Road Ahead

This course is designed to be the first step of a JavaScript-centered learning path. I hope it inspired you to go further. If so, you are only at the beginning of a long and exciting journey!

If you feel overwhelmed for now, take heart: Rome wasn't built in a day. Grasping the concepts of programming takes time and practice. Don't get discouraged and be sure to follow the guidelines provided in the next few paragraphs.

> All the resources listed here are freely available online, although some of them require registration or are also sold in another format. They were picked for their relevance, without any affiliation.

1. Keep on Practicing

This is by far the most important advice. Nothing will sharpen your skills and make you more confident as a JavaScript developer than practicing your craft on a regular basis. In particular, I urge you to start building a *personal project* right now. This is the best solution to stay focused and create something meaningful while having fun along the way. You may already have a project idea in the back of your mind. If not, here are a few things you could build:

- An interactive quiz application.

- A client for your favorite social network.

- A contact or TODO list manager.

- A simple game, like the famous Connect 4

Still undecided or looking for more? Take Wes Bos' [JavaScript30](#) coding challenge and build 30 things in 30 days. Yes you can!

2. Get a Deeper Understanding of the Language

You hopefully learnt a lot already, but JavaScript is a surprisingly deep language that still has a lot in store for you. Some prominent resources to strengthen your knowledge are:

- Kyle Simpson's [You Don't Know JS](#), a book series diving deep into the core mechanisms of the language.
- Axel Rauschmayer's authoritative [blog](#) and [books](#).
- Ilya Kantor's [javascript.info](#), an online tutorial with a lot of interesting content.
- The JavaScript section of the [Mozilla Developer Network](#), full of reference material, guides and tutorials.

Lastly, you might want to check out my own little [curated collection](#) of JavaScript-related links.

3. Get Social

JavaScript's popularity is peaking. Why not becoming a part of its huge developer community?

Thanks to code sharing platforms like [GitHub](#), you can easily look for examples of code, see how other developers work and even get involved in a project. That's the beauty of open source! When facing a problem, knowing how to find the best answers is a critical skill. [Stack Overflow](#) is your best bet for asking questions and sharing knowledge with tons or other developers.

Lastly, you could join a learning community in order to share experiences (good or bad) and build relationships. There are JavaScript user groups in many cities of the world, look for one in your neighborhood. Should you prefer socializing online, Quincy Larson's [freeCodeCamp](#) features a broad

curriculum including many projects and a helpful, beginner-friendly

community. Whichever road you may take, I wish you a great journey into the wonderful world of JavaScript!