

14 - Final Words and Where to Go Next?

What to do next with our newly acquired skills?

WE'LL COVER THE FOLLOWING



- 14 - Final Words and Where to Go Next?
- Congratulations!
- Where to Go Next?

14 - Final Words and Where to Go Next?

#

Congratulations!

Having completed the course, you should now be familiar with the basics of the JavaScript language and with programming in general, as the concepts that we learned are widely applicable to different programming languages.

Think of learning what past tense is, in a natural, spoken, language like in English. How you construct a past tense sentence might structurally differ in between English and Japanese, but the fact that you have a conceptual understanding of what past tense is, would allow you to transfer your knowledge from one language to next in a much easier manner once you figure out the mapping in between them.

In this course, we learned about the fundamental structures in programming languages such as variables, operators, loops, conditionals, functions, objects and arrays using JavaScript. All the programming languages that I personally had to interface so far had an implementation of these concepts in one way or another. The syntax for them might differ but the workings are pretty similar. Generally speaking, the knowledge you acquire in one programming language is highly transferrable to other languages.

Alongside with JavaScript, we also learned about the p5.js library along the way. This means that if you wanted to utilize your newly found programming knowledge, you could continue to do so by using p5.js with great comfort.

The thing that is currently missing from our knowledge is to deploy our work on the web, thereby sharing it with rest of the world. Since this has been a course focused on learning how to program, I didn't want to occupy us with secondary, operational, concerns like this. But this is something that I will eventually explore in future lessons as it is something that you might want to learn how to do.

You might find yourself constrained a bit at this point especially if you don't want to keep sharpening your newly acquired skills in p5.js. Where should one go next from here?

Where to Go Next?

As mentioned at the beginning of the course, learning JavaScript will open a world of opportunities in front of you as there are a lot of domains that you can apply your knowledge. Where to go next, surely depends on where you would like to grow your skill-set in.

If you want to move beyond p5.js and create user-facing web applications, you will surely need to learn the basics of **HTML** and **CSS** to build graphical interfaces for your projects. After that, learning some **DOM API** would be useful to be able to hook up the interface elements on the page to execute JavaScript commands. If you wanted to build slightly more complicated interfaces, then **jQuery** might help as it makes working with JavaScript slightly easier when working with web pages. If you are looking to build more advanced interfaces where there are lots of interactions and state dependencies in between page elements, then a front-end framework such as **AngularJS** or a library like **React** can be very useful. I would suggest not to rush to learn these tools even though they are all the hype nowadays. Starting out with pure JavaScript and then moving to jQuery to build slightly more challenging interfaces will help you appreciate the problems that these more complicated tools are built to solve.

Maybe you might want to do some server side programming. Then **Node.js** is

Maybe you might want to do some server-side programming. Then **Node.js** is the tool that you should pick up. Server-side programming might involve any back-end calculations that doesn't need to happen on the client-side, on the browser, and might involve building scripts that interface with the operating system of the host computer to do things like: creating files and folders, deleting them, etc. A **Node.js** script doesn't necessarily need to be in service of a web application backend. The prime advantage of **Node.js** is that it provides you with an environment on your machine, rather than on the browser, to run JavaScript in. Node.js sets the JavaScript free from the constraints of the browser environment. For example, **Electron**. It is a **Node.js** based framework that allows you to build native desktop applications using web technologies. You can use **Electron** to build a web browser using JavaScript rather than to build something on the browser using JavaScript!

Automating web pages, programmatically connecting to the internet and scraping online data might be your thing. For that, there is **Phantom.js** and **Casper.js**. If you wanted to do advanced data visualizations, then you should check out the **D3** library. Maybe you want to get into robotics, or program IOT devices using JavaScript. How about giving **Johnny-Five** a try then? You can build native mobile apps (**React Native**), HTML5 games (**Phaser**), databases (**mongoDB**) 3D visuals and animations (**Three.js**) or even cognitive applications that makes use of Artificial Intelligence (**IBM Watson**) all by using JavaScript.

When learning a new programming language it is best to have a plan: a vision for something that you would like to build. Ideally, the minimum viable project that you can lift off the ground without encumbering yourself with too much work or too many technicalities. Getting to that point and then building the next thing. Always getting better, always learning something new. Welcome to the world of programming. I can't wait to see what you will come up with.