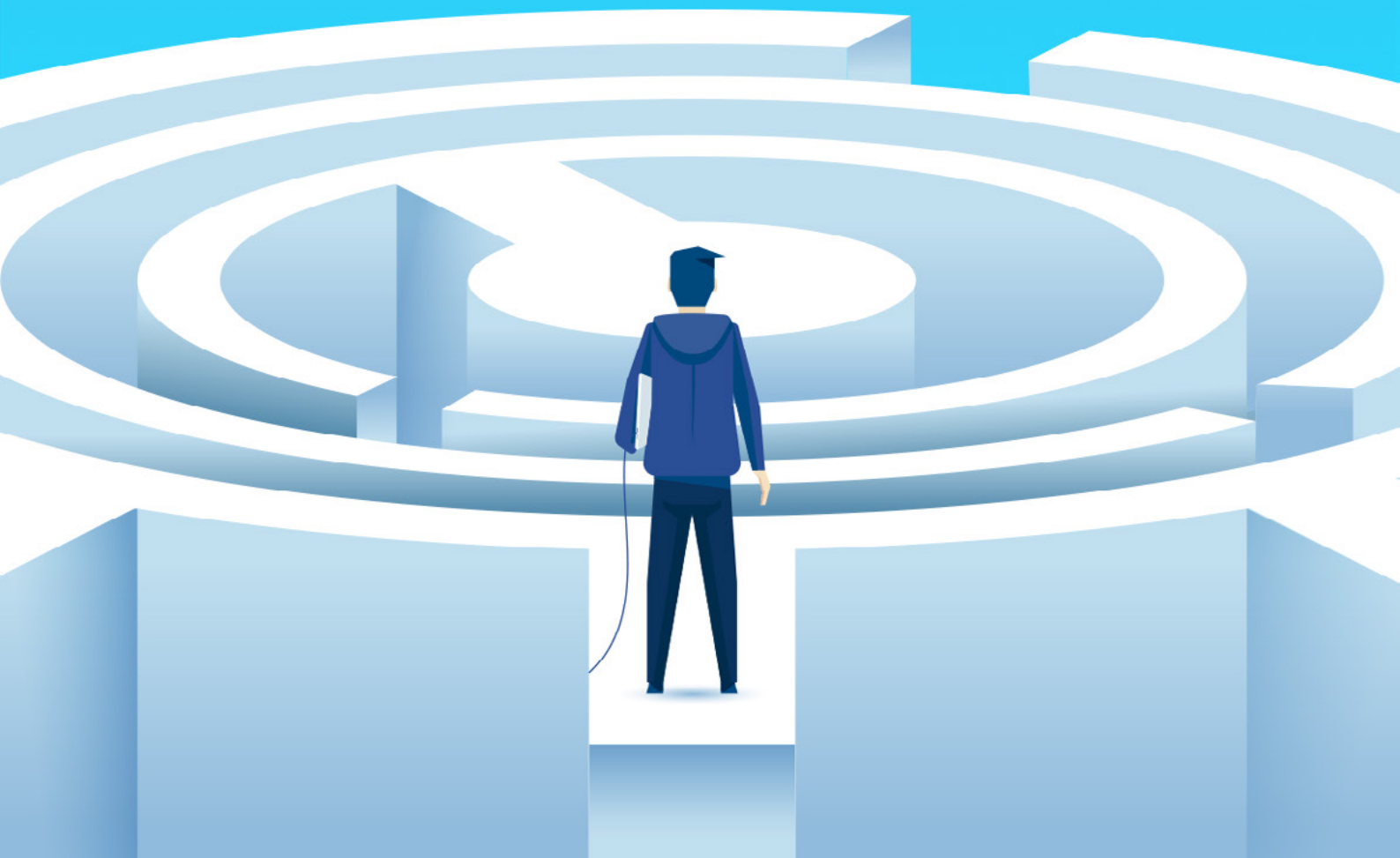


# Coding 101

learn **programming** the right way



# INTRODUCTION

---

Before you begin learning how to code, it's first useful to discuss why you would want to learn web development. There are lots of good reasons why you should learn how to code. **Here are a few of them:**

- Coding is a skill that is in **high demand** in the technology industry. Jobs as a coder can allow you to work wherever you want. All you need is a computer and the internet!
- Coding is a **necessary** skill to be involved in one of the most cutting-edge job fields currently available.
- Many of the most exciting opportunities in emerging industries today are in the **cryptocurrency** and **AI space**. Coding is crucial for both.
- Coding is both challenging and worthwhile. You'll be amazed at what you're able to create after just a small time spent learning to code.
- The skills you learn will be relevant in the future. The tech industry is only going to **grow!**
- You don't need a degree to learn how to code. You can get started **without** any college/university debt!

Above all, coding is about problem solving. It's not so much about memorising all these functions that different lines of code create (although that's incredibly useful). You can look those functions up if you forget how to do something specific. Your skills as a **problem solver**, which are the most important skills regarding coding, will develop with the more time you spend training them.

# TABLE OF CONTENTS

<b>1. Learn the Basic Coding Languages</b>	<b>4</b>
HTML	4
CSS	5
Javascript	5
<b>2. Building Your First Website</b>	<b>7</b>



# 1 LEARN THE BASIC CODING LANGUAGES

First things first - you have to begin by learning the **basic coding languages**.

## 1. 1. HTML

If you've never done any web development whatsoever, you're going to want to start with HTML.

HTML stands for **Hyper Text Markup Language**. Put simply, it's used to make basic web pages. It's not technically a programming language at all. However, if you're completely new to coding, it's still nothing like any language you'll be familiar with.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>My First Heading</h1>
6.
7. <p> My first paragraph. </p>
8.
9. </body>
10. </html>
```

Whenever you see images or text on the web, they will have been put there primarily using HTML. HTML allows you to place text, images, videos, songs, and other content onto a webpage using what are called "**tags**".

There isn't a whole lot you can do with HTML but what you can do appears all over the internet. It's therefore really important to get a good grasp of HTML before you start out on your career or hobby as a web developer.

## 1. 2. CSS

The next logical step to learning how to develop websites is to learn CSS. This still isn't really a programming language though! However, it does introduce a load more variety into your websites and is another crucial step you need to take before you start programming in more interesting languages!

CSS makes the stuff you made using HTML **look good**. You can do all kinds of neat tricks with it such as adding colour, padding (borders), margins, and changing fonts.



A good grasp of **CSS** and **HTML** can make a half-decent looking website. However, if you really want to be a developer, you're going to have to learn some real programming languages too!

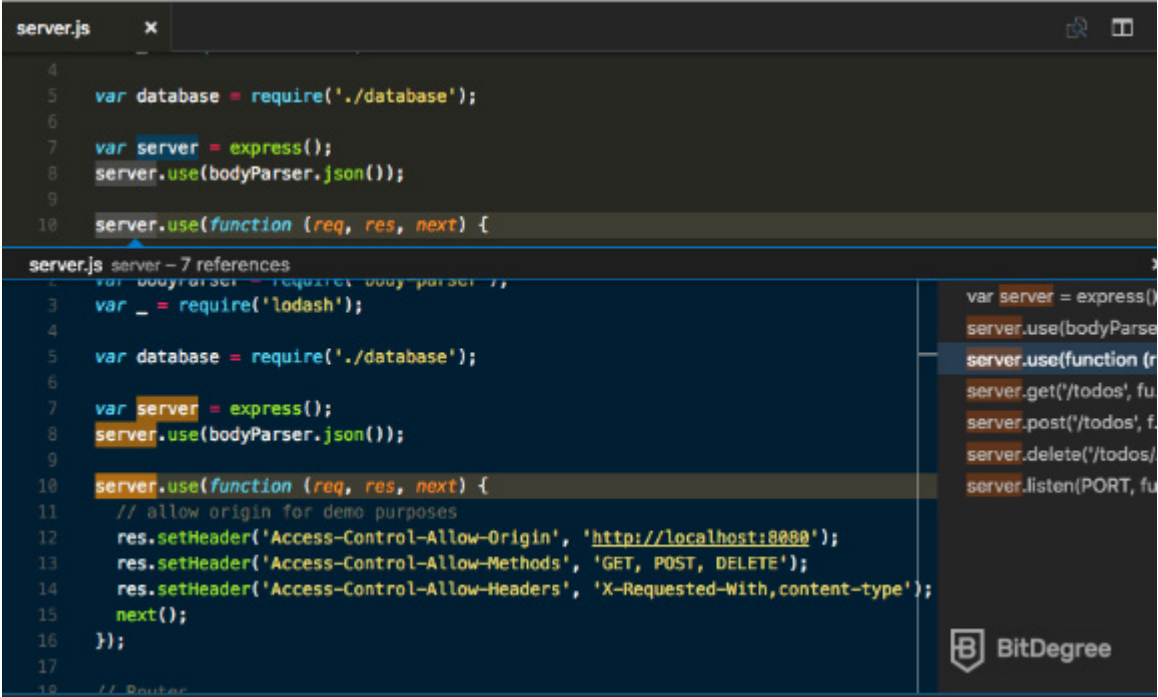
## 1. 3. Javascript

Let's not sugar coat it. If you want to learn how to start coding to a high level, you're going to need a good knowledge of JavaScript.

When learning how to code, JavaScript is a must. It's everywhere. These days, you'll be hard-pressed to find a company who doesn't want some kind of online presence. JavaScript will be used to build most of their applications and websites.

Being such a widely used programming language, job opportunities are great for someone who's skilled at programming in JavaScript. That said, in a recent survey of developers by StackOverflow, it was found that a massive 62.5% of respondents said that they used it.

This means that finding a JavaScript job is going to be hugely competitive too. It will be worth it though. PayScale estimate that the average salary of a JavaScript web developer is almost \$60,000. There's lots of money to be made if you're good!



```
server.js x
4
5 var database = require('./database');
6
7 var server = express();
8 server.use(bodyParser.json());
9
10 server.use(function (req, res, next) {
11
12
13
14
15
16
17
18 // Router

server.js server - 7 references
3 var bodyParser = require('body-parser'),
4 var _ = require('lodash');
5 var database = require('./database');
6
7 var server = express();
8 server.use(bodyParser.json());
9
10 server.use(function (req, res, next) {
11 // allow origin for demo purposes
12 res.setHeader('Access-Control-Allow-Origin', 'http://localhost:8080');
13 res.setHeader('Access-Control-Allow-Methods', 'GET, POST, DELETE');
14 res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With, content-type');
15 next();
16 });
17
18 // Router

var server = express()
server.use(bodyParser
server.use(function (r
server.get('/todos', fu
server.post('/todos', f
server.delete('/todos/
server.listen(PORT, fu
```

JavaScript adds interactive features to your list of web developer tools. These might be games or responses to text being entered – that kind of thing.

You can start out by making small adjustments to your basic HTML web pages. These can include **welcome messages** and **image changes**. However, you can progress to making full three-dimensional games if you get good enough at coding in JavaScript!



## 2 BUILDING YOUR FIRST WEBSITE

The next step in learning how to code is to actually **try and build** your own website. There are loads of good things you can do to continue advancing your learning at this point though. **Some of them include:**

### 1. Learn more about debugging programs and sites using the Chrome developer tools.

The screenshot shows the official Chrome DevTools website. At the top, there's a blue navigation bar with links to HOME, CHROME DEVTOOLS, LIGHTHOUSE, PUPPETEER, WORKBOX, and CHROME USER EXPERIENCE REPORT. The main content area is titled "Chrome DevTools" and features a video player showing a demonstration of the tools. To the left, there's a sidebar with a list of topics including "View and Change CSS", "UI References and Overviews", and "Remote-Debug Android Devices". To the right, there's another sidebar with a "Contents" section listing "DevTools for Beginners", "Get started", and "Discover DevTools".

### 2. Find and attend meetups about coding and using JavaScript in your local area.

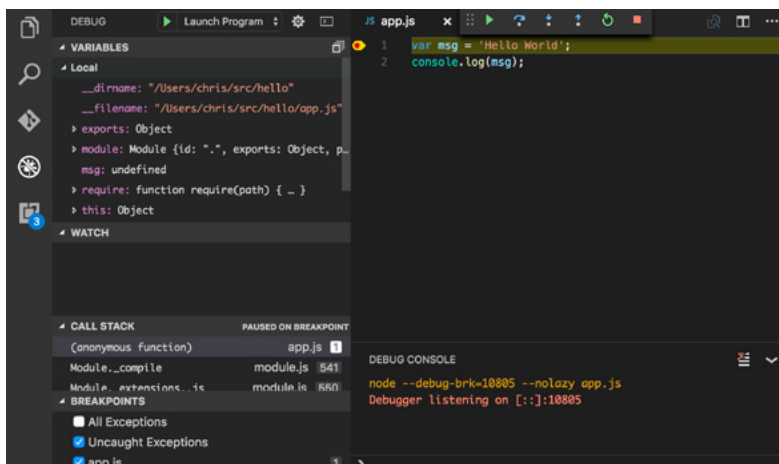
### 3. Learn about synchronous and asynchronous JavaScript.

### 4. Learn about event loops – this will open your eyes to a whole load more great stuff you can do with JavaScript.

4. Learn about event loops – this will open your eyes to a whole load more great stuff you can do with JavaScript.

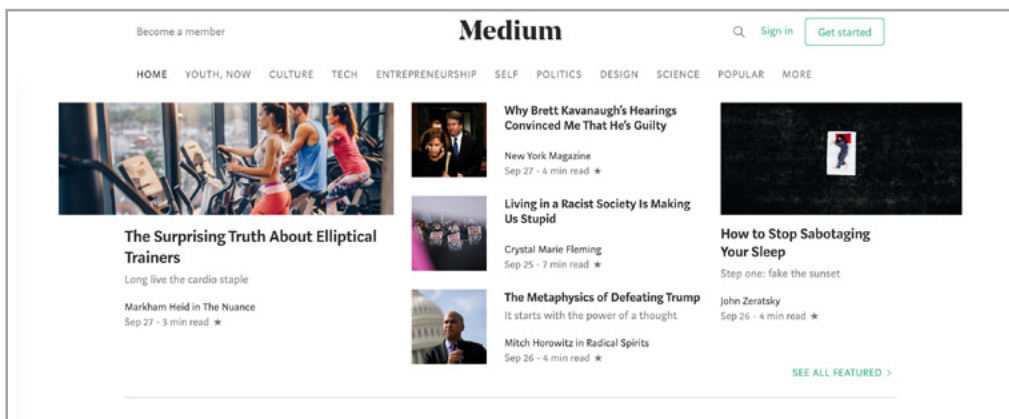


5. Download and study node.js and npm. These tools are great for all aspiring web developers.



6. Learn about React.js.

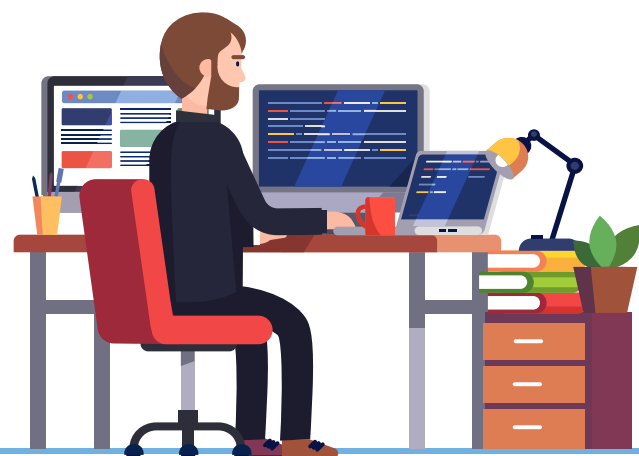
7. Read as many coding articles you can on sites like Medium. There are loads of different levels, but your understanding of industry jargon should be pretty good by now!





After learning these basic coding languages and concepts, you'll be starting to get a feel of what you're good at and what you like to create using code. Below are some of the next areas you **should** consider studying:

- Learn how to use **Git** and **GitHub**. After familiarising yourself with these tools, you can sign up for a GitHub account and start making commits. Employers often favour active GitHub users.
- **SQL** – if you love the heavy **database** stuff, SQL is a great programming language to learn. Since practically every application uses databases in some way, SQL programming knowledge is a great skill to have.
- **Solidity** – the programming language of the **Ethereum blockchain** is a particularly good one to learn. It's used to implement smart contracts and Ethereum developers with such skills are in high demand right now. It's only a matter of time before someone develops a killer **dApp application** on the Ethereum network using Solidity – it could be you!
- **jQuery** – jQuery is like a tool box for JavaScript developers. It's basically been developed to speed up typing JavaScript code. Many tasks are grouped together to mean they can be added with a single line of code.
- **Swift** – Swift is the programming language of **iOS applications**. Since Apple products are so consistently popular, being a skilled Swift coder will certainly land a great job. What's more, iOS applications are generally a lot more profitable than those that appear on Google Play or other such application stores.





# FINAL WORDS

So, that's my short ebook on how to start coding. I hope you discovered exactly how to get started in this exciting field.

However, this is just the beginning of your education. The thing with technology is that it doesn't stay the same for long. This is particularly true with web development. New frameworks and libraries are **constantly** being released for JavaScript meaning there is always something new to learn. Then there are loads of different programming languages. Realistically, your efforts to learn coding **are only just beginning!**

This guide should have given you a solid understanding of how everything in computer science fits together though. All the extra things you learn going forward will slot into place much more easily with such a base of knowledge.

Above all, remember that programming and coding is all about solving problems. You've been given a great set of tools to start solving problems that arise when you're building applications and websites. New tools will come along that will make things easier, but your foundation is there already.