Step into Tech

Introduction

Hello and welcome to the BBC Step into Tech. We're sure you're really excited to have embarked on your journey into (or back into) tech. There's a short amount of time between now and the start of the course and we'd like you to use this time to complete some preparatory work and get a sense of what's to come over the next few months.

Getting help

Feel free to email our coaches at any time if you need help!

Our coaches:

james.heggs@techreturners.com

rebecca.taylor@techreturners.com

Contents

1.Setting up your laptop

For the course, you will need to have access to a laptop running a **Unix-based operating system** or a **Windows Operating System with bash shell command line tool installed**. You can read a little bit more about the history of the Unix operating system **here**.

The **bash shell command line tool**, available on Windows 10, allows you to use a unix-like environment from within a Windows PC.

NOTE: If your laptop runs an earlier version of Windows, please get in touch with James.

Why Unix?

- Most developers work with a unix-based operating system, so learning how to use it is another important part of learning software development

- You will likely be given a machine running a unix-based operating system in your role as a software developer
- So we are all using a consistent operating system when working in class and on projects

Preparing a Windows laptop

If you have a Windows Operating system, please follow **these instructions** for installing the bash shell command line tool.

If step 12 does not work, open the Windows Store from the Start menu instead of via the command line, and search for Ubuntu on Windows and install it. Then continue following the instructions.

Then, follow **this guide** to set up various tools and programs that we will be using throughout the course.

Preparing an Ubuntu laptop

If you are using an Ubuntu operating system, please follow **this guide** to set up various tools and programs that we will be using throughout the course.

Preparing your Mac laptop

If you have a Mac laptop, please follow **this guide** to set up various tools and programs that we will be using throughout the course.

2. What is Code?

When we talk about code, we're talking about writing instructions to the computer in such a way that the computer can interpret them. There are various languages which can be used to instruct the computer in various different ways - HTML can instruct the computer what content to display on a webpage, CSS can instruct the computer how this content should look, and languages like JavaScript, Python, Ruby and Java offer us a way to instruct the computer to perform logic.

Computers, at the end of the day, only understand 1s and 0s though. So in fact, when we write HTML, CSS, JavaScript, Python, Java etc., that code that we write has to be compiled into binary code (1s and 0s) before the computer actually understands it. So why we don't simply write 1s and 0s in the first place?

The answer is that this would be hard for us to do, and hard for other developers to read and understand! So when we write code, we need to remember we're not just writing code for the computer, but also for ourselves and each other.

And that's what code is! Instructions that make sense both to the computer, and to us.

If you'd like to read a bit more around code, including the history of writing code and computer software, we can really recommend:

📚 **What is Code? By Paul Ford** (an excellent article that appeared in Bloomberg magazine in 2015)

📚 **CODE by Charles Petzold**

Exercise

Before moving on, try to answer the following questions:

1. How many coding/programming languages have you heard of?
2. What is the difference between software and hardware?
3. Think of what you did yesterday. Can you list all the things/processes you interacted with that involved software or hardware in some way?
4. What is the difference between “front end” and “back end” programming?
5. How has computer programming changed over the last 50 years?
6. What are some of the most exciting advances in computing happening today?

3. Personal and Study skills for success

Growth Mindset

We firmly believe that the key to success in developing your technology skills is deeply rooted in your mindset.

Please watch the following talk and reflect on the ideas Dweck presents:

🎥 **Carol Dweck on the concept of Growth Mindset**

You might also be interested in her book:

📚 **Mindset by Carol Dweck**

Note taking

As you've probably already found with any HTML/CSS you've learnt, there are a *lot* of things to remember when it comes to programming and software development. Some of these things are **conceptual**, such as "What is a database?" or "How does information travel across the internet?" or "What does the cloud mean?". However, there is also a lot of **procedural** and information to remember such as "How do I open a file in VS Code?" or "What are the steps to get this application running?". Many of the commands you followed when you were setting up your laptop were procedural, i.e. Do this thing, then do the next thing, then do this third thing. Other information is **factual** such as "What does the `<head>` section of an HTML page do?" and "What are all the CSS selectors?".

When it comes to remembering procedural and factual information, don't expect to remember everything! There's far too much - so it's a great idea to begin writing everything down.

We recommend getting used to using a notetaking application which has support for code syntax. For example, *Bear* for Mac, *Evernote* for Windows and *BoostNote* for Ubuntu all allow you to write snippets of code directly in your notes. If you write your code snippets between sets of 3 backticks, it will format your code in a really nice way, specific to certain languages:

For example:

~~~
```js
let num = 42;
if (num > 10) {
  console.log("That's a lot of dogs!");
}
```
~~~

Is displayed as:

```js
let num = 42;
if (num > 10) {
  console.log("That's a lot of dogs!");
}
```

Googling

As well as being comfortable with not being able to remember everything, you'll need to be comfortable with googling. Professional developers do it all the time - it's absolutely not "cheating" and is actually an essential part of the job. There is far too much information to remember off the tops of our heads, and computers have a habit of finding an immesaurable number of errors to surprise us with! Many of them we'll never have seen before, so we head over to Google to see what the internet's got to say!

StackOverflow will often be the first result that comes up when searching for programming-related problems. Sometimes it's not the easiest site to understand, but it does have a vast collection of relevant and useful troubleshooting advice.

Mozilla Developer Network (MDN) is an extremely thorough and reliable source of information, although it can also be hard to understand for beginners.

W3 Schools is also a popular site for information about HTML, CSS and JavaScript, and is a little more accessible for beginners.

Get used to googling, and using these resources when you need to!

In the weeks and months that follow, you're going to be learning a lot - some of it will come easily and some of it will be challenging, and push you outside of your comfort zone. Please do try and remember the lessons learned in this section!

Without further ado, let's get back to learning some tech!

4. Introduction to JavaScript

As you probably know, we teach full stack JavaScript. We think JavaScript is a great language to learn as it's really popular nowadays, coveted by employers, can be used in many types of software development, and has a large and active community and great resources around it. However, remember that the things you learn to do with JavaScript can be done with any programming language!

JavaScript is a great programming language to learn as it's really popular nowadays, and it can also be used both on the frontend and the backend. This means we can use it to program the interactivity on the frontend of web applications, and also use it on servers to handle logic such as fetching and updating data from a database.

To get a feeling for the JavaScript language, lease work through **sections 1-8** of the below course:

💻 An Introduction to JavaScript

This will likely take you some time - but that's okay. And don't worry if you don't finish it, or it doesn't seem to be sinking in - we start from the very basics in Immersion Week so you won't be missing out. It's just a bonus to be a bit familiar with the syntax before we start!

5. Problem Solving with JavaScript

If you do manage to get through the CodeWars JavaScript sections, you might want to practice solving problems with JavaScript using a site called **CodeWars**. CodeWars is an app that offers many bitesized problems for you to solve with any programming language of your choice.

First, watch this video introducing you to CodeWars if you haven't used the site before:

🎥 Getting started with Codewars

Then, work through the following problems. We've listed a few suggestions below but feel free to explore others too!

🎥 Solving Problems on Codewars

🎥 Using JSBin to practice

Problems to solve:

https://www.codewars.com/kata/return-the-day

https://www.codewars.com/kata/5265326f5fda8eb1160004c8

https://www.codewars.com/kata/56bc28ad5bdaeb48760009b0 https://www.codewars.com/kata/5545f109004975ea66000086

https://www.codewars.com/kata/542ebbdb494db239f8000046

https://www.codewars.com/kata/5ad0d8356165e63c140014d4

https://www.codewars.com/kata/58d248c7012397a81800005c

https://www.codewars.com/kata/57f780909f7e8e3183000078

https://www.codewars.com/kata/l1-set-alarm

https://www.codewars.com/kata/57eae20f5500ad98e50002c5

6. Additional Reading/Resources

If you fancy some extra reading or resources, here's a list of things we like.

👂 **The Code Newbies Podcast**

💻 **JavaScript for Cats** (extra JavaScript practice - and cats!)

💻 **FreeCodeCamp** - check out the sections on “Algorithm Scripting” as part of the JavaScript section for more challenges a bit like Codewars. FreeCodeCamp also has an active forum.

📚 **Hello World by Hannah Fry**

📚 **Algorithms to Live By: The Computer Science of Human Decisions by Brian Christian and Tom Griffiths**

💻 **BaseCS** - computer science concepts explained simply

💻 **Dev.to** - a great community of developers, with many interesting blogs and articles

💻 **The Tech Returners blog** - read about other people's experience on the course and keep up to date with our latest news!

📚 **You Don't Know JS** - book series available for free online, the Up and Going book is great at this stage