Your Pre-Journey Into Tech

Your Pre-Journey Into Tech

Introduction

Welcome to your Pre-Journey into tech. We're sure you're really excited to have embarked on your journey into (or back into) tech. There's a little while between now and the start of the course and we'd like you to use this time to complete some preparatory work.

We do expect everyone to have completed this work by the time the course starts so we're all starting with some prior knowledge under our belts, so it's really important that you keep in touch with us and let us know if you're struggling for whatever reason so we can support you. We really hope you enjoy it, and that it gives you a taste of what's to come.

The aims of the Pre-Journey are:

- To get your laptops ready to be used as software development machines
- To understand what code is
- To gain basic JavaScript knowledge
- To practice solving problems with JavaScript
- To understand what the Command Line is and practice using it
- To practice working with a Unix-based Operating System
- To sign up with a few important services such as Github
- To learn what Git and Github are and learn to use them
- To develop your study skills
- To begin to build a growth mindset
- To refresh your HTML and CSS skills from the Pre-Interview stage (if you have completed a Pre-Interview with us)
- To get you involved in our community and the wider local tech community

Don't worry - everything we cover in the Pre-Journey will be revisited in the main course so don't feel like you have to master everything. The main aim is familiarity and giving you a flavour of what's to come.

How to follow the Pre-Journey

We have organised the content of the Pre-Journey into logical order, so please start at the beginning and

work your way through each section. Some sections will take longer than others, especially the practical sections.

The Pre-Journey (like the course itself) is not a box-checking exercise. Don't rush through just to get to the end - it's more important that you understand everything you read and do.

Getting help

Feel free to email our coaches at any time if you need help! We also encourage you to chat to your peers on Slack - you'll all be working on the same material and probably getting stuck in the same places, so why not help one another?

Our coaches:

james.heggs@techreturners.com
harriet.ryder@techreturners.com

Contents

1.Setting up your laptop

For the course, and from section 4 onwards of the Pre-Journey Into Tech, you will need to have access to a laptop running a **Unix-based operating system**. You can read a little bit more about the history of the Unix operating system **here**. Mac laptops run an operating system called MacOS which is derived from the Unix operating system, so this is fine, however if you have a Windows operating system we ask you to install an additional operating system called Ubuntu, which is Unix-based.

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

Installing Ubuntu on a Windows machine

Ubuntu can be installed alongside the Windows OS so that you have a choice of operating systems to use. To do this, you will first need to create a bootable disk for the Ubuntu operating system. There are various guides for how to do this, such as **this one**. You will then need to boot your laptop from this disk, and select the option to install Ubuntu alongside your current operating system.

If you would like assistance with this process please let us know so we can book some time for you to come in and see us or provide remote help. It would be a good idea to get this sorted **as soon as possible**, so it doesn't impede you from moving on with the Pre-Journey.

Preparing your Ubuntu laptop

Once you have the Ubuntu operating system running, 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 both the computer can interpret them. You've already written code - HTML and CSS - which are coding languages designed to instruct the computer how to display a web page. HTML deals with telling the computer **what** content to display on a webpage and CSS deals with instructing it what the content should **look like**.

Computers, at the end of the day, only understand 1s and 0s though. So in fact, when we write HTML and CSS, 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.

Take this opportunity during the Pre-Journey to get used to using a notetaking application which has support for code syntax. For example, *Bear* for Mac and *BoostNote* for Ubuntu both 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:

```
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. The Unix Operating System and command line

One of the most important things for you to begin to familiarise yourself with about your Unix-based operating system is the command line.

Firstly, please watch:

📹 **A brief introduction to the command line**

Then, please work through this course:

💻 **Learn the Command Line**

Before moving on, work through the following questions/instructions:

1. Make a list of all the commands you learnt so that you can easily refer back to them
2. What is a hidden file? How can you create a hidden file?
3. How can you delete a directory?
4. What is meant by the “root directory” and the “home directory”?
5. What is meant by the "prompt"?
6. What are some common GUI applications that you use on a regular basis?
7. What are some potential disadvantages of a GUI?
8. What does the `pwd` command stand for?
9. What are some alternate names for the "command line"?
10. How can you move your cursor to the beginning or the end of a line of text you've typed in the terminal?

5. 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, it’s one of the most important parts of the Pre-Journey.

The next stage of the Pre-Journey will have you practicing what you’ve learned to solve problems with JavaScript, at which point you might find you need to run through the course again, or at least some sections of it. This is very normal! You might also feel like you haven’t really “absorbed” the contents of the course - this is normal, too, and is why we ask you to practice what you’ve learned in the next section. It’ll take a while, but the more you practice the more you’ll begin to feel familiar and confident with how JavaScript works. But don’t expect it to be instant.

6. Problem Solving with JavaScript

For this section, we’d like you to put into practice some of your new JavaScript knowledge. We’ll use a

site called **CodeWars** which offers many bitesized problems for you to solve with any programming language of your choice.

First, please watch this video introducing you to CodeWars:

🎥 **Getting started with Codewars**

Then, work through the following problems which we have selected for you. This will take some time, and to get you started we've got a few tips for you first:

🎥 **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

https://www.codewars.com/kata/55d24f55d7dd296eb9000030

https://www.codewars.com/kata/reverse-list-order

https://www.codewars.com/kata/training-js-number-8-conditional-statement-switch

https://www.codewars.com/kata/abbreviate-a-two-word-name

https://www.codewars.com/kata/the-feast-of-many-beasts

https://www.codewars.com/kata/571f1eb77e8954a812000837

https://www.codewars.com/kata/sum-of-positive

https://www.codewars.com/kata/sum-mixed-array

https://www.codewars.com/kata/remove-exclamation-marks

https://www.codewars.com/kata/5b077ebdaf15be5c7f000077

As you will have seen, there are many more challenges available on Codewars so don't stop here - if you have any extra time between now and the course start date, the best thing you can do for yourself is keep working through challenges on Codewars to keep flexing that muscle memory for JavaScript.

You might find the below video helpful:

📹 **Solving further Codewars challenges**

7. HTML and CSS Recap

It's been a while since you first looked at HTML and CSS for your pre-interview challenge! And you've learnt so much since then! Let's take this opportunity to refresh what we learnt during that pre-interview work and brush up on any areas we're not quite sure about.

First, make sure you can answer the following questions:

1. What is HTML used for, and what is CSS used for?
2. How many different kinds of heading elements are there?
3. How many ways can you add styling to your page?
4. Which is the recommended best way of adding styling to your page? What are the advantages of this method over other methods?
5. What is the difference between a class and an ID in HTML?
6. What different CSS selectors do you know?
7. What is the head section of an HTML document for?
8. What version of HTML do we use nowadays and how do we declare the HTML version in our HTML page? 9.How do you create a link to another page?
9. How can you create a link to another page which opens in a new tab, rather than the same window?
10. How can you create rounded corners on an element?
11. What is meant by "semantic HTML"?
12. What element(s) does the * CSS selector select?
13. How would you change the font of some text using CSS?
14. How many different ways do you know of for specifying colours? Can you think of at least 3?

Next, go back to your original HTML/CSS project which you did for your interview. Is there anything you want to change about it or didn't have time to do? Can you extend it and make it better? Perhaps you wish you could start it all again and do it afresh - you can if you want!

Spend a bit of time re-familiarising yourself with HTML and CSS and improving (or re-doing) your original site. You might find you want to revisit the Codecademy **HTML** and **CSS** courses from the Pre-Interview stage or try this slightly different track: **Make a Website**.

When you're done, we'd like you to share your improvement with us by uploading your changes to Github. To learn how to do this, please watch the below video:

📹 **Uploading your updated websites to Github**

If you would like to actually get your website live (!!) then watch this video to learn how:

📹 Getting your website live on the web with Github Pages

8. Running Code on our Computers

So, what's the value of doing all these challenges on Codewars? How does this relate to writing actual software?

Well, being able to solve small problems with code is a huge stepping stone to being able to solve bigger problems with code. Having a familiarity with a programming language and knowing the kinds of things you can do with code are the most fundamental building blocks to being able to tackle larger projects with more complicated requirements.

Also, many (but by no means all) tech companies will give you a small coding challenge to complete as part of their interview process. Often the challenges are of a similar variety to the ones you find on Codewars, so getting plenty of practice with these kinds of challenges as early as possible is a great start.

But now it's time to move away from the environment of Codewars and look at how we can run JavaScript code on our own computers!

Learning:

Please watch the following videos:

📹 Running JavaScript code in the browser

📹 Running JavaScript code with NodeJS

9. Final JavaScript Exercise

Congratulations on making it so far!

For your final challenge we'd like you to create a short game, written in JavaScript, which can be played in the terminal.

The game will be **Rock, Paper, Scissors**. The rules of this game are that one person (the player) makes a guess of rock, paper or scissors. The computer should choose a random option from rock, paper or scissors and report on whether the user or the computer has won, or whether there has been a draw. The winner is calculated as follows:

- Rock wins over scissors
- Scissors win over paper
- Paper wins over rock

To get started, please watch the following videos:

📹 Setting up your project and linking to Github

📹 Capturing user input to the program

When you are done, please push this project up to Github if you haven't been keeping Github up to date, and let James or Harriet know you have completed the Pre-Journey!

10. 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

11. Getting involved in the tech community

If you're local to Manchester, there are lots of brilliant local meetups which you can attend to begin getting involved in the local tech community. Many of the below events also have regional groups in other cities.

CodeBar Manchester (also national):

A free coding workshop suitable for complete beginners through to more experienced developers who want to learn something new. Experienced mentors will help you work through tutorials, or your own projects. Note that CodeBar is for underrepresented groups only. Harriet usually attends these meetups if you'd like to meet some of our team!
https://codebar.io/manchester

CodeAndStuff:

A free workshop for women and non-binary people in Manchester who want to code together in a social

environment. Suitable for complete beginners through to more experienced developers.
https://codeandstuff-manchester.github.io/

FreeCodeCamp Manchester (also national):

A free, regular meetup for people learning to code. Often people choose to work through the FreeCodeCamp online curriculum although this is not a necessity! Open to anyone.
https://www.facebook.com/groups/free.code.camp.manchester/

CodeUp Manchester (also national):

A free monthly meetup for anyone learning to code. Mentorship is available for those who need some assistance, otherwise it's a great place for social coding and meeting other learners.
https://www.meetup.com/CodeUpManchester/

We also encourage you to be active on our **Slack channel** and begin chatting to your cohort members.

If you are not based in Manchester, have a look at what meetups and groups are available in your local area. Maybe someone on the Slack channel lives near you and would be interested in attending with you?