Learning Journey Stage 2 🗺️

Congratulations on making it to the next stage of your learning journey with School of Code! 🥳

The following tasks are designed to build on what you learned in Stage 1 (through creating your Avatar character and Scratch game), to get you set up for coding success and to build a strong foundation in web development. Completing all of the activities in this guide will give you a really good grounding in some essentials and - if you're invited to join a bootcamp - will mean that you're up to speed with where our curriculum starts. You may be familiar with some of the material - even so, do take the time to go through the resources and refresh your understanding.
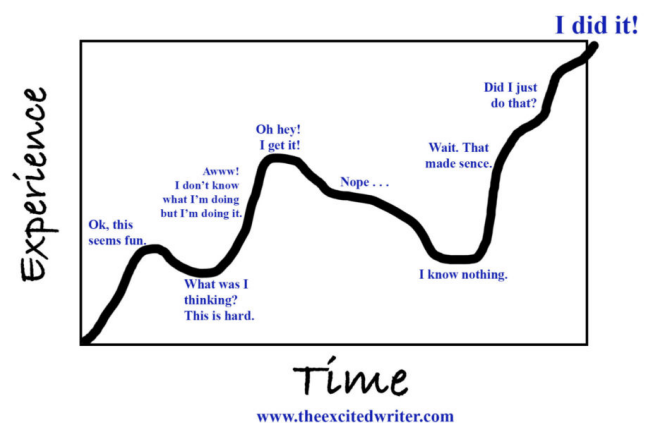
By the end of this material, you'll be able to:

- Set up your development environment
- Set up your file system and understand naming conventions
- Understand how information flows over the internet
- Use HTML to build a static webpage and CSS to style it
- Start navigating through your computer using commands in the command line
- Know the basics of Git and GibHub for version control

- Create a tribute page - your first project!

We've given you some rough time indicators to help you plan your time. Remember, these are only *approximate estimates*. Everybody comes to their School of Code learning journey with a unique combination of previous experience, existing knowledge and exposure to tech. Everybody learns at a different rate (and often at different rates for different topics). The key is to work steadily at your own pace, and remember that learning (although sometimes intense) is amazing! If you keep at it, it'll be worth it! 🎯

The Learning Curve

Experience

Time

I did it!

Did I just do that?

Oh hey! I get it!

Wait. That made sence.

Awww! I don't know what I'm doing but I'm doing it.

Nope . . .

Ok, this seems fun.

What was I thinking? This is hard.

I know nothing.

www.theexcitedwriter.com

The tasks you need to complete are listed below. They are numbered on the left with links to any relevant resources under each one. The text on the right is there to guide and help you through. If you get really stuck, remember - Google is your friend 🤝.

There is a quick google form to complete at the end of each task - make sure to fill each of these in so we can see how you're progressing with your School of Code Learning Journey Stage 2. 👉**YOU WILL NEED YOUR APPLICATION ID TO COMPLETE THESE FORMS** (you can find it in the email giving you the link to this document)👈

Good luck - and enjoy! 😊

Learning Journey Stage 2 Tasks

Task 1: Setting up your development environment

Approx learning time 30 mins +

✅**1.1 Learn about IDEs**
   What is an IDE article
✅**1.2 Download VS Code**
   Link to download VS Code
   Getting started video
   Guide to installing VS Code
✅**1.3 Practice shortcuts**
   Mac shortcuts
   Windows shortcuts
   Searchable reference

✅ **Task 1: Feedback**

Your IDE (integrated development environment) is where you write your code. Just think about Microsoft Word for writing text - it's not just a notepad, there's a lot of added benefits to using it because it is designed to be a great environment for creating documents. That's what an IDE is for writing code.

At School of Code we use Visual Studio Code (usually just referred to as VS Code). VS Code is infinitely customisable with extensions, and it has an integrated terminal, live share, and other handy capabilities that make our coding life easier. Watch the video to get started with the basics and spend some time just familiarising yourself with the program and practice the keyboard shortcuts.

Task 2: Preparing your file system

Approx learning time 20 mins +

✅**2.1 Understand file naming**
   Naming conventions guide
✅**2.2 Create a projects folder**

You probably know how to create a file, and a folder... but believe us when we say it really will save you a lot of headaches to understand how, as a budding developer, we want you to start creating them.

It is essential for you to become more conscious of how you use and maintain your computer. Gone are the days where you mindlessly let your downloads folder fill up until your system tells you you have no space remaining on your hard drive, or 'conveniently' saving every file on your desktop for quick access until your background is completely obscured by

a mosaic of icons. You need to be in control of the folder structures where your code is saved, and adhere to naming conventions so that other developers understand how you have categorised your work.

In your documents folder create a directory for your School of Code learning journey - you will be using this later. Remember to use one of the naming conventions you have learned about in Task 2.1!

✅ **Task 2: [Feedback](#)**

Task 3: How the internet works

Approx learning time 30 mins +

✅**3.1 Watch the videos**
[Computer Networks](#)
[The Internet](#)
[The World Wide Web](#)

✅**3.2 Optional reading**
[Deeper Dive Document](#)

What even is the internet? Is it different to the world wide web? How does it actually work? What happens when you request a web page? These are things we might wonder about if we are going to start building things for the web. We've selected a few videos for you to watch from a series of YouTube videos - if you like the series, then feel free to watch more!

OPTIONAL: There is also a link to a document which provides a great overview and path to more resources.

✅ **Task 3: [Feedback](#)**

Task 4: HTML and CSS: the building blocks of the web

Approx learning time 4 hours +

✅**4.1 FCC tutorial #1**
[HTML and HTML5 tutorial](#)
✅4.2 FCC tutorial #2
[CSS tutorial](#)
☑ **4.3 Optional tasks**
[FCC HTML5 & CSS3 video](#)
[Marksheet tutorial](#)
[UI fundamentals](#)
[Semantic HTML article #1](#)
[Semantic HTML article #2](#)
[Semantic HTML article #3](#)

It's time to jump into HTML and CSS, which form the building blocks of the web. HTML stands for "hyper-text markup language", and CSS stands for "cascading style sheets". If a web page was a body, HTML would be its skeleton, the basic structure of the page. 🦴 💀 CSS would then be the skin, hair, clothes, jewellery, and anything else that falls under the design of the page, how it looks and how it's styled. 🩳 👗

FreeCodeCamp (FCC) is a fantastic resource that makes learning HTML and CSS approachable and fun. Work your way through the Responsive Web Design tutorials for basic HTML and CSS (NOTE: you don't need to complete the certification; the aim is to work through **the first two** sections of interactive activities so that you have a foundation of knowledge).

OPTIONAL: If you like to learn via video, you can watch FreeCodeCamp's video course on learning HTML5 and CSS3 from scratch. You could also complete the MarkSheet tutorial on HTML and CSS. There is also a link to an introduction to the fundamentals of designing user interfaces (UI) to help you learn basic design principles. You'll be bashing our websites that not only work well, but look good too!

It's always good to understand the tools and the right way to use them. Semantic HTML is a vital part of the modern web, and learning how to use the right HTML tag for the job is key (which HTML more meaning that just using div tags everywhere).

✅ **Task 4:** [Feedback]

Task 5: Taking command with the command line

Approx learning time 30 mins +

☑**5.1 Complete challenge**
[Line Commander Game]

☑ **5.2 Optional resource**
[Game help - text reverse]

Most users interact with their modern computers using the GUI (graphic user interface) built into their operating system. This means that they can look through their files, open programs, etc. by clicking on icons like folders. 📁 However, as a programmer, you'll start to use the command line, where you type prompts into a text terminal, to tell your computer what to do directly. ⌨️ It might be weird at first, but trust us - when you get used to it, it's much faster. Plus, you'll start to feel a little bit like a hacker in a movie! 😎

Learning to navigate a computer using the command line is a skill that gives you lots of power. There are a few basic commands to learn first and more complex ones to learn later on. The important thing for you to take away is the pattern of using a command, options and input. Complete the game which should help you get to grips with the basics and also do some cool stuff! *NOTE - on some versions of Windows the rev command will not work. Don't worry - it's not an essential command to have installed! If you find the 'rev' command does not work on your computer, just use the text reverser in the optional resources to complete the challenge.*

✅ **Task 5:** [Feedback]

Task 6: Git and version control

Approx learning time 4 hrs +

✅ **6.1 Learn about version control & GitHub** [Git and GitHub video](#) ✅ **6.2 Install git** [Git first time setup](#) ✅ **6.3 Create GitHub account** [GitHub first time setup](#) ✅ **6.4 Learn the basics** [Using Git & GitHub tutorial](#) ✅ **6.5 Practice** [OhMyGit! Game](#) ✅ **6.6 Optional Resources** [Very basic Git basics](#) [Git cheat sheet](#) [Common issues/help](#) [Using Access Tokens & Git](#) ✅ **Task 6: [Feedback](#)**	The image of the lone coder sitting in a dark corner tapping away in solitude is a stereotype thanks to countless movies and TV shows. In reality, software engineering is a team sport. There are loads of benefits to multiple people working on the same codebase, but without a tool to manage how people can work on the same project simultaneously, the potential for getting wires crossed and overlapping each other's code could spell disaster... or at least be annoying. This is where a version control tool like git comes in to save the day. As it's industry standard, during the bootcamp we use git every day - it's a developer's best friend (once you get used to it). Hand in hand with Git, we also use GitHub. Watch the video to get an introduction to understanding Git and GitHub. Follow the instructions to ensure you've installed git correctly and have your credentials configured correctly. This will be a good chance to start practising interacting with your computer using the command line as well. Set up your GitHub account, and you're ready to start combining the two by working through the tutorial. Practice makes perfect, and what better way to practise than through a game, right? Work through the stages on OhMyGit game up to and including the section on "remotes".

Task 7: Putting it all together - your first project

Approx learning time - this is up to you! (We recommend you allow a minimum of 4 hours)

✅ **7.1 Complete FCC Tribute Page in CodePen** [Build a tribute page](#). ✅ **7.2 Get template repo** [Link to Template Repo](#) ✅ **7.3 Deploy your site** [GitHub pages](#)	By this point, you've done a load of learning, and you might feel like you have lots of different pieces of information swimming around in your head. It's time to solidify what you learned and give you the opportunity for some creativity as well by putting that knowledge into practice. It's time to complete your first project - a tribute page using HTML & CSS! 🥳 You can make your tribute page about anything you want - let your personality shine through!

This project is an element of FreeCodeCamp's Responsive Web Design Certification. Follow the link to the FreeCodeCamp (FCC) site and complete the instructions there - this task is just the tribute page only. *(It's up to you whether you want to submit your solution link to FreeCodeCamp)*

Once you've completed your tribute page in CodePen, we'd like you to solidify the other knowledge you've gained in the activities above - take your learning to the next level and get used to working with Git and in Github repositories.

Use the template repository in task 7.2 to create your own public repository and follow the readme instructions. You will use your code from CodePen.

Read through the instructions on deploying sites through GitHub Pages at 7.3 and then complete the final task in the ReadMe. Make a note of the URL of your brand new site!

✅ **Task 7.1:** [Feedback + submission of CodePen link](#)
✅ **Task 7.2:** [Feedback + submission of GitHub repo link](#)
✅ **Task 7.3:** [Feedback + submission of deployed Tribute Page URL](#)

That's it - you're done. Awesome work! 🥳

We hope you enjoyed Stage 2 of your learning journey with the School of Code!

Links to some useful sites below:

[W3 Schools](#)
[MDN Web docs](#)
[YouTube](#)
[Stack Overflow](#)