C2C3 - Session 0

Introduction, setup, tools

Goals:

- Understanding the aim of the course
- Installing the tools needed
- Tips and tricks for an effective course

FAQ

Who is this course for?

Welcome to Couch 2 Coder 3! In the 3rd iteration of this course, you'll learn how to build dynamic, responsive web applications from the ground up. We will cover both front-end and back-end development, introducing you to essential tools and frameworks like HTML, CSS, Node.js, Express, and data storage fundamentals. Whether you're a beginner or looking to expand your skill set, this course will provide you with a solid foundation to create complete, functional web applications using JavaScript.

But I have not attended the previous two Couch 2 Coder courses! Also I have never coded before in my life...

Not a problem - even though the course will cover advanced concepts, everything will be explained from the ground up. The content also does not have any strict connection to the previous iterations. You may, however, encounter familiar concepts from both courses, like programming fundamentals, or basic HTML and CSS. Speaking of which...

I took part in both previous Couch 2 Coder sessions - Is there anything new for me worth joining for?

Absolutely! While you might find concepts that are repeated from the previous course (HTML fundamentals, CSS, programming basics like variables and data structures), the end goal this time is different. While the first course focused on basic data analytics using Python, the second was mainly centered around building a website using HTML and CSS, this time

around we will expand our horizons and build a full stack website (meaning both the user facing and the business logic will be covered), this time using JavaScript. The course will offer you a new programming language, expanded toolset and a broader overview of how we actually can build a web application that is not just static text and images, but has dynamic data.

What equipment do I need to participate?

The course is aiming to be as inclusive as possible - there is absolutely *no* need for any high-end laptops to attend the course. There are some limitations, however:

- You must use a laptop or desktop PC with either Mac, Windows, or a flavour of Linux installed on it.
- You must have basic familiarity with concepts when it comes using computers (creating files, folders, installing apps, and if you'd like to keep up and code along, a reasonable typing speed could be beneficial - but don't worry, no need to be a virtuoso piano player to attend!)
- You must either be the user with admin power on your device, or at least have immediate access to the admin if needed for installing/running apps
- You need internet access, and have ideally Google Chrome installed as your browser for the course (we will build our website on Google Chrome. Should work on other browsers too, but for consistency's sake, I recommend Chrome!)

And that's pretty much it! Unfortunately the course currently cannot be properly done on a Chromebook or tablets, we insist on participants using either a Mac, or a Windows/Linux based PC.

I'm worried I'm going to fall behind on my own...

Not to worry! The course will be pre-recorded, and it's not only optional, but actually recommended to pause, rewind, and rethink the covered topics. No need to rush, the point is to do the course at your own pace! You'll also be added to a group of like minded individuals who will work their way through the course, and you'll have a chance to connect and collaborate if you'd like to!

Will I be able to host the website and share it with friends and family?

Unfortunately, due to the limited timeframe, we will not have time to cover the hosting of the website itself. This means that you can showcase your app on your machine, but cannot share it online, as that topic could cover a week long course by itself. Having said that, there's no telling how much you can research and learn by yourself if you're driven enough -

the content is out there!

Required apps

As mentioned above, we'd prefer if participants downloaded Chrome and used it as their default browser for the duration of the course.

Our chosen programming language is going to be JavaScript - the reason for that is twofold. First, it's one of the most popular programming language currently, while also being relatively beginner friendly and easy to get started with. Second, since we are building a full stack app, we will need to code on the client/user facing side as well, and the only language available in any browser/web app is going to be JavaScript.

Installing JavaScript is easy - just go to the following website and follow either the terminal installation or the direct download steps:

Download NodeJS here

If you get stuck, refer to the video for detailed steps and troubleshooting!

You'll notice that we install an app (more like runtime) called NodeJS, not JavaScript - this is basically a tool for us that gives us the ability to run JavaScript code on our machine. If you want to view and edit a spreadsheet, you will need the appropriate app installed - and if you want to code with JavaScript, we need NodeJS!

Make sure the installation was successful by opening up the shell/terminal of your device, and typing in `node -v` to confirm that we have JavaScript running locally. If you get stuck anywhere, watch back the intro video - we will cover how to get out of seemingly sticky situations multiple times!

And finally, we will need a text editor - if you already have a favourite one, feel free to use it, but for anyone just beginning their coding journey, I recommend downloading VS Code from the following link:

Download VS Code here

Any further optional extensions will be discussed in our video guide.

Effective text editing

Any time you get started with a task, you aim to work on it using the most appropriate tools. We could use pen and paper, draw up tables and use a calculator to calculate values from spreadsheets - or we can use a tool designed for this job, Microsoft Excel/Google Sheets.

For our full stack development journey, we could utilise a couple of tools - Atom is free and open source, Sublime is powerful but costs money. VSCode however seems to be the text editor of choice for most developers for a lot of programming languages, and especially for CSS/HTML and JavaScript. Originally developed by Microsoft, it's now an open source tool that has tons of customization options and extensions.

We will cover some of the shortcuts and extensions now.

It may seem trivial but it can make a huge impact to your productivity as a developer.

You probably realised early on when you were working on your thesis or any assignment that clicking `File` -> `Save` with the mouse is time consuming, especially if you do it dilligently every couple of paragraphs, but clicking `ctrl` + `s` is no time at all once you get into the habit. Learn the shortcuts, and you'll save yourself a lot of time!

VSCode

VSCode is:

- multi-platform (MacOS, Windows, Linux)
- popular (widely used for web development)
- free and open source
- extensible (we can add functionality via plugins called 'extensions')

It also comes with a lot of really cool shortcuts and other tools we can use.

Command Palette

If you only ever memorise one keyboard shortcut, make it `cmd` + `shift` + `p` or `ctrl` + `shift` + `p` (if that doesn't work on Windows, try `F1`). This opens up the Command Palette, a list of every function VSCode can do, either built-in or through packages. You can type in what you want to do as text (e.g. "Copy") and it will search for the appropriate command. If a keyboard shortcut exists for the command, it will also show up here.

Menus

VSCode's menu system is also pretty in-depth, with a lot (but not all!) of its functions sorted into relevant menus. The menus also let you know keyboard shortcuts when they are available.

Useful Shortcuts

Keypress	Action
`cmd` + `shift` + `p`	take us to a command palette where we will be given menu options without leaving our keyboard.
`cmd` + `s`	save changes in current tab - DO IT OFTEN!!!
`cmd` + `q`	quit
`cmd` + `w`	close one tab at a time
`cmd` + `option` + `arrow`	switch between tabs
`ctrl` + `<number>`	switch directly to specific tab
`cmd` + `f`	search in your current file (see below)
`cmd` + `shift` + `f`	search the entire project
`option` + `shift` + `down-arrow`	Copy current line onto next line
`cmd` + `p`	open the file finder
`cmd` + `backspace`	delete the line before your cursor
`fn` + `backspace`	delete the character after your cursor
`cmd` + `right/left`	moves your cursor to the end/beginning of the line

You can get a cheat sheet for these and more at https://code.visualstudio.com/shortcuts/keyboard-shortcuts-macos.pdf

Multiple Cursors

A very useful feature of VSCode is being able to use more than one cursor, enabling us to type the same text in more than one place. `cmd` + `option` + `arrow keys` will add extra cursors to your window, allowing you to edit multiple elements at once.

You can also go to the selection menu and choose `Switch to cmd+click for multi-cursor`. Now if you hold down `cmd` and click in the file it will add a cursor wherever you click. When you type it will appear wherever there are cursors

This can be incredibly useful for renaming variables and functions. We can also use `cmd` + `d` to select the word (or name) under the cursor, and use `cmd` + `d` again to select duplicates of it. If we then type, we will replace all selected instances of the text.

Settings and Themes

`cmd` + `,` allows you to access VSCode's user settings.

From here you can make all sorts of changes to set up your environment, both to the Core VSCode app, and to the Editor window itself. Most of these settings are fine, but one that we instructors find useful for putting code on the projector is:

`Code` -> `Preferences` -> `Settings`

Search for `scroll beyond last line` and check the box.

You can also change the theme (i.e. make it light or dark) `Code` -> `Preferences` -> `Color Theme`

VSCode's extensions are one of the main reasons VSCode is so popular. The app is not just a text editor, but a rich ecosystem of extensions which add functionality to the editor. If there's something you wish VSCode did, or just did better, there's probably an extension for that!