Command Line Basics

Foundations Course

Introduction

Feeling scared of the command line? You're not alone. We have this image of developers staring intently at a black screen with white or green text flashing across as they wildly enter incomprehensible commands to hack into the corporate mainframe (no doubt while guzzling soda and wiping neon orange Cheetos dust off their keyboard).

That black screen or window is the command line interface (CLI), where you're able to enter commands that your computer will run for you. While there's no need for you to reenact the scene above, working with the command line is a critical skill for you to learn as a developer. The command line is like our base of operations, from which we can launch other programs and interact with them. It has a syntax of its own to learn, but since you'll be entering the same commands dozens of times, you'll quickly pick up the commands you need most.

In this introductory lesson to the command line, you'll learn how to navigate around your computer and how to manipulate files and directories (also known as folders) directly from the comfort of the command line. You'll soon see that this isn't as difficult as you may think. The commands you will learn in this lesson are very straightforward. So don't let the prospect of using the command line for the first time intimidate you.

Test Drive Your Terminal

Open a terminal on your computer.

- **Linux**: Open the programs menu and search for "Terminal". You can also open the terminal by pressing `CTRL + ALT + T` on your keyboard.

- **macOS**: Open your Applications > Utilities folder and find "Terminal". You can also use Spotlight search to open Terminal. Press `CMD + SPACE` to open Spotlight, and search for "Terminal". Press `RETURN` to open it.

Before we do anything, take a look at the following text: `$ whoami` This is a terminal command because it begins with a `$`. The `$` is saying "Hey! Enter what follows in your terminal." This means that we must exclude the `$` when entering any command. In the example above, we would only enter `whoami` in our terminal. This is a common indicator so make sure that you aren't entering `$` before a command. Now that you are aware of what `$` does, take your terminal for a test run! Make sure your terminal is open, type the command mentioned above, and press enter on your keyboard.

It returns your username. Cool!

Why learn this now?

You will be making heavy use of the command line throughout this curriculum, and the upcoming installations project will need you to install many different software programs using the command line. Additionally, you will primarily be using Git within the command line (more on this later). As part of the bigger picture, you may well be using the command line on a daily basis in your career as a software developer, making it an indispensable skill in your toolset.

Lesson Overview

This section contains a general overview of topics that you will learn in this lesson.

- Describe what the command line is.
- Open the command line on your computer.
- Use the command line to navigate directories and display directory contents.
- Use the command line to create a new directory and a new file.
- Use the command line to rename or destroy a directory and a file.
- Use the command line to open a file or folder in a program.

Assignment

Note: Many of these resources assume you're using a Mac or Linux environment. If you did our previous installation lesson, you should already have Linux installed in dual-boot or a virtual machine. Or, you might be using MacOS. If you don't have MacOS, or any version of Linux installed, please return to the operating system installation guide.

2. Visit The Unix Shell course designed by the Software Carpentry Foundation. There you will find a full complement of lessons on using the CLI, but for now just focus on completing the following lessons:
    - Setup (Download the exercise files, but skip the Software Install step, you already have everything you need)
    - Introducing the Shell
    - Navigating Files and Directories
    - Working With Files and Directories

- Pipes and Filters

3. With your newly discovered CLI super powers, practice creating a folder and a few files using the `mkdir`, `touch`, and `cd` commands introduced in the previous step. As an example, a basic website might have a main `index.html` file, a CSS stylesheet file called `style.css`, and a folder for `images`. Think about how you could create these files with the commands and put it into practice!

4. Let's practice creating files and directories and deleting them! You'll need to enter the commands for the steps below in your terminal. If you can't recall how to open a terminal, scroll up for a reminder.

    2. Create a new directory in your home directory with the name `test`.
    3. Navigate to the `test` directory.
    4. Create a new file called `test.txt`. *Hint: use the `touch` or `echo` command.*
    5. Open your newly created file in VSCode and make some changes, save the file, and close it.
    6. Navigate back out of the `test` directory.
    7. Delete the `test` directory.

    That's it–you're done with practice! If you commit to doing most things from the command line from here on out, these commands will become second nature to you. Moving and copying files is much more efficiently done through the command line, even if it feels like more of a hassle at this point.

Use the Command Line Like a Pro

There's something important that you need to know about programmers. Programmers are lazy. Like, really lazy. When forced to do something over and over again, the odds are good that they'll figure out a way to automate it instead. The good news is that you get to take advantage of the many shortcuts they've created along the way. It's time to learn how to use the command line like a pro (which is to say, in a really lazy way).

First, you might have already noticed that copying and pasting inside the command line doesn't work the way that you'd expect. When you're inside the command line, you'll need to use `Ctrl+Shift+C` (Mac: `Cmd+C`) to copy and `Ctrl+Shift+V` (Mac: `Cmd+V`) to paste. For example, to copy and paste commands from your browser into the command line, you'll highlight the command text and use `Ctrl+C` as usual and then paste it in your terminal using `Ctrl+Shift+V`. Test it out!

Second, you need to learn about tab completion. Seriously, this tip will save you so much time and frustration. Let's say that you're in the command line and that you need to move into a folder that's far away, something like `~/Documents/Odin-Project/foundations/javascript/calculator/`. That's a long command to type out, and everything needs to be exactly right in order for it to work. Nope, we're *way* too lazy for that! Basically, by hitting `Tab`, the command line will automatically complete commands that you've started typing once there's only one option. For example, it's pretty common to have a `Documents` folder and a `Downloads` folder in the home directory. If you've typed `cd D` and then press `Tab`, the command line will let you know that it's not sure which one you want by showing you the different options that match what you've typed so far: `bash $ cd D Documents/ Downloads/ $ cd D` But once you've typed in a little bit more, it will complete the name for you, making it possible to write out the full file path above by typing as little as `cd Doc[tab]O[tab]f[tab]j[tab]cal[tab]` (depending on what other folders exist

on your computer). Test it out, and get comfortable with how this works. You're gonna love it.

Third, there's a really handy shortcut for opening everything within a project directory: `.` Once you've installed a text editor, you can use this shortcut to open up an entire project and all its files in one go. This shortcut is also commonly used with Git (later on it's covered in detail) with commands like `git add .` to add all the files inside of a directory into Git's staging area. For example, if you have VS Code installed, you can `cd` into the project directory and then type `code .` (with the period). It will launch VS Code and open up the project folder in the sidebar. See the next section of this lesson for a more detailed example.

A Note on typing passwords: When using a command in the terminal that requires you to enter your password for authentication (such as `sudo`), the characters should not be visible to you as you type them. While you might think the terminal isn't responding, don't worry! This is a security feature to protect confidential information, like how password fields on websites use asterisks or dots. By not displaying the characters you write, the Terminal keeps your password secure.

Opening files in VSCode from the Command Line

- **Linux**: You can open VSCode from the command line by typing `code`, and you can open folders or files by adding the name of the location after it: `code my_awesome_project/`.

- **macOS**: Some setup is required. After installing VSCode, launch it any way you're comfortable with. Once it's running, open the Command Palette with `CMD + Shift + P`. In the little dialog that appears, type `shell command`. One of the choices that appears will be `Shell Command: Install 'code'`

`command in PATH`. Select that option, and restart the terminal if you have it open.

Additional Resources

This section contains helpful links to related content. It isn't required, so consider it supplemental.

- The Art of Command Line is a complete beginner's pro-maker. It serves as an open-source repository. This also has a lot of pro tips!
- The online book, Learn Enough Command Line to Be Dangerous is a great resource for mastering the command line. Chapter 1 is free and provides a good introduction to command line tools. The rest of the book is not free and goes into more depth than you really need at this point, but feel free to buy and read the rest of the book if you like.
- ExplainShell.com is a great resource if you want to deconstruct a particularly strange shell command or learn how Bash works through guess-and-check.
- Unix/Linux Command Cheat Sheet contains a list of important commands that you can refer to regularly as you become familiar with using Linux. You can print it out so you can have a physical copy with you when you're not at your computer.
- Command Line Flashcards by flashcards.github.io.
- Video Series from LearnLinuxTv contains 24 videos explaining the basics of the command line. Videos are brief enough for beginners but, at the same time, detailed enough to get you started and light your inner curiosity.

Knowledge Check

This section contains questions for you to check your understanding of this lesson on your own. If you're having trouble answering a question, click it and review the material it links to.

- What is the command line?
- How do you open the command line on your computer?
- How can you navigate to a particular directory?
- Where will `cd` on its own navigate you to?
- Where will `cd ..` navigate you to?
- How do you display the name of the directory you are currently in?
- How do you display the contents of the directory you are currently in?
- How do you create a new directory?
- How do you create a new file?
- How do you destroy a directory or file?
- How do you rename a directory or file?

View Course

Login to track progress

Next Lesson

Improve this lesson on GitHub

Have a question?

Chat with our friendly Odin community in our Discord chatrooms!

Open Discord