**FULL STACK TRAINING**

**The Languages and Technologies You’ll Learn, And Why They’re Awesome**

Fundamentals, Front End, Backend, Full Stack...

Hi Guillain,

With coding being around for so many decades, it means that lots and lots of programming languages are available. How do you know which ones you should learn?

And what about when you're ready to deploy your web application to the cloud? Do you learn Google Cloud, Microsoft Azure, or Amazon Web Services?

At Nucamp, we teach the languages that help students get a job in the tech sector.

And when it comes to cloud platforms, ***we teach all three.***

**Expand your cloud experience**

Deploy projects to all three of the top cloud providers: Google Cloud, Microsoft Azure, and Amazon Web Services



Here are some of the languages and technologies we'll cover in The Complete Software Engineering Bootcamp Path:

* **HTML & CSS**: True building block languages, HTML & CSS control how any website looks and feels. To create a unique, beautiful online experience, you need to know these languages fluently.
* **JavaScript**: This is the language that lets you build dynamic features. When you see features on a website that are animated or refresh without you doing anything, that’s JavaScript at work.
* **Python**: The back end language that will allow you to connect to SQL databases, build powerful apps and run Data Science projects as well. Python has replaced Java as preferred back end language.
* **Bootstrap**: Create responsive mobile experiences with Bootstrap, a major repository for HTML, CSS and JavaScript code. Knowing how to use Bootstrap means you won’t have to create code from scratch. Instead, you can leverage freely-available blocks of code to help get your projects up and running.
* **React**: As a JavaScript library that helps you build functional user interfaces, React lets developers create web and mobile experiences that are dynamic without having to reload content. In essence, React lets you create digital experiences that people will actually want to use because they are easy to use.
* **React Native**: React Native is React’s mobile cousin, letting you do everything React does but on mobile devices. With React Native, you can package a website into a mobile app letting you reach people on Apple’s App Store or Google Play.
* **NodeJS**: NodeJS lets web developers do something called server-side scripting, which means writing code that lets a website or app behave according to a user’s behavior. This can include everything from performing database operations and user authentication to application logic.
* **MongoDB**: Behind the scenes of any website or app is data, lots and lots of data. MongoDB lets you actually store that data and store it in a way that’s flexible and accessible so that the websites and apps you create run quickly and smoothly.

With a curriculum focused on both Python, JavaScript and the additional libraries that support them, you can be assured Nucamp’s bootcamps teach what’s in demand in today’s job market!

### **Exercise: Using the Bash CLI**

NOTE: Scroll down below the video for written instructions.

Click here if you wish to open this video in its own window.

### **What You'll Do**

* You will learn how to use basic bash CLI commands.

### **Instructions**

* Open Git Bash (Windows) or macOS Terminal.
* Git Bash users only: Enter the following command to ensure that you are in the home directory/folder, using the **cd**command, which stands for **change directory**, and the **~** character, which is a shortcut to the home folder:

cd ~

* Enter the following command, which stands for **print working directory**, to verify that you are in the home folder (which should end with /Users/<your account name>):

pwd

* Next, use the **cd** command again to move into the Desktop folder:

cd Desktop

* + **NOTE FOR WINDOWS USERS:** If your Windows account is linked to **OneDrive**, your Desktop folder will have been moved into the OneDrive folder. In this case, use this command: **cd OneDrive/Desktop**
* Use it once again to move into the NucampFolder:

cd NucampFolder

* Use the command **ls**, which is short for **list**, to view the folder contents:

ls

* Next, make a new folder named **testdir** with the **mkdir**command:

mkdir testdir

* Move into this new folder:

cd testdir

* Enter the following two commands to create two new empty files. Make sure you place a dot in front of the name hiddenfile:

touch testfile

touch .hiddenfile

* List the folder contents:

ls

* You should see the testfile, but not the .hiddenfile. This is because files with a dot/period/fullstop as the first character in their names are hidden by default.

#### **Flags**

* Most CLI commands have a set of flags that can be used to pass options when executing the command.
* The **ls** command has a flag of **-a**. The **a**stands for all. Try it:

ls -a

* Now you should be able to see the hidden file.
* The ls command also has a flag of **-l**. The **l**stands for long. Try this as well:

ls -l

* This shows a detailed view of the folder contents, including file permissions, file size, owner, date modified, etc.
* You can combine flags like this:

ls -al

* The order of the letters does not matter. This would be equivalent:

ls -la

#### **Paths**

* Enter the following command to move up to the parent directory of **testdir**, which is **NucampFolder** :

cd ..

* The two dots, **..**, refer to the parent directory.
* Enter the following command from anywhere in the filesystem to return to the home folder:

cd ~

#### **Relative vs. absolute paths**

* There are two ways to move around the filesystem: using relative or absolute paths.
* A relative path is relative to the present working directory. It will only work from that directory.
* An absolute path will work from anywhere.
* To move from the home folder to the NucampFolder using a relative path, enter:

cd Desktop/NucampFolder

* + **NOTE FOR WINDOWS USERS WITH ONEDRIVE:**You will need to use **cd OneDrive/Desktop/NucampFolder**here.
* Then go back to the home folder:

cd ~

* Now enter the following command to use an absolute path, substituting <your username> with your actual username:
  + Windows:

cd c:/Users/<your username>/Desktop/NucampFolder

* + - (or cd c:/Users/<your username/OneDrive/Desktop/NucampFolder if using OneDrive)
  + macOS:

cd /Users/<your username>/Desktop/NucampFolder

* Try using the up arrow to bring back the command "cd Desktop/NucampFolder". You will need to use it two or more times.
* Once you see it, press enter. This will produce an error, demonstrating that the relative path only works from the home folder.
* Use the up arrow again multiple times, until you see the command to cd to the NucampFolder via its absolute path. Press enter.
* This will work, and you won't see an error message. (Though since you should already be in the NucampFolder, you won't actually go anywhere).
* There are other ways to recall previously entered commands.
* One way is the reverse search. Type**ctrl+r** to use it, then type in **mk**. This should bring back the **mkdir testdir** command.
* If you press enter, you will see an error message because there is already a **testdir** in the NucampFolder directory.
* Another way is this command:

history

* This command will show a list of recently used commands.
* Finally, you can enter the command **exit** to close the bash session. (It's fine to just close the window as well.)

### **Additional Resources**

* [Bash Commands Guide](https://medium.com/@duruldalkanat/bash-commands-guide-129c81cbfe87)
* [An A-Z Index of the Linux command line: bash + utilities](https://ss64.com/bash/) (Note: most but not all will work in Git Bash)