

The terminal

As a person new to programming, using the terminal interface of a computer may be something you'll do for the first time in this lesson.

We often access computer programs through their Graphical User Interface or GUI. This is simply the visual component of a computer program. For example, word processing programs, such as Microsoft Word, generally have a GUI that contains a white area for us to type in, a cursor to show where we are typing, and a variety of buttons, menus, and options to edit our work. This is a GUI. Another example is an email inbox that displays each email's details, allows us to open an email by clicking on its subject line, and offers buttons to reply and format text.

However, when we are developing we will often use our devices' terminal interface. The terminal is a text-based interface that allows a user to interact with the computer by entering commands on a command line. You have probably navigated through the folders and files on your device using a GUI tool such as Finder on a Mac or Explore on Windows. In this lesson, we'll explore how we can create, update, delete and navigate our folders and files using the terminal interface. We'll also see how using the terminal gives us access to other commands that we do not have when using the GUI tools our computers provide.

Opening the terminal

On Mac/Linux open 'terminal'.

If you are using Windows 10 , you are required to use Bash on Ubuntu on Windows. The instructions for installing Bash on Ubuntu can be found on the following [link](#).

From this point onwards, we are only going to refer `Bash on Ubuntu on Windows` as `Windows` . We expect you to understand that all commands should be run in `bash` instead of normal `cmd` application.

If you are using windows 8 or below, please install a Ubuntu Virtual Machine (VM) on Windows instead.

You will now see a new window to the inside of your computer. It might look a bit intimidating at the beginning, but it is a lot simpler to navigate then using the GUI once you get used to it.

Let's create a directory (folder)

When using a GUI, the term 'folder' is used to describe a storage space to organize files. On the command line, the term 'directory' is used instead. 'Folder' and 'directory' are essentially the

same thing.

You are currently in your home directory (the highest level of the directory). To double check, you can type `cd ~` and then press enter. You should now see a `~` indicating that you are in your home directory.

First, type `ls`. Does anything happen? Not really does it? This is because the command `ls` lists all the files in the current directory, however we haven't added anything there yet.

Next, type `mkdir code`. The `mkdir` command creates a new directory with the name 'code'. If you now type `ls`, you should see a new 'code' directory.

Afterwards type `cd code`. You should see `~` being replaced with the name 'code'. `cd` stands for 'change directory' which allows us to enter or change directories.

Let's add a file

Let's do this:

Type `touch index.html` and then type `ls` again. You should see a file with the name 'index.html'. If you want you can use your GUI to navigate to your code folder and double check that the file really exists.

Cheat Sheet

These are some of the basic commands that will help you throughout this course.

```
cd [path] # Change directory
cd ~ # Home directory, e.g. 'cd ~/folder/'
ls # List Files in directory
ls -a # Listing incl. hidden files
touch [file] # Creates file
code [file] # Opens the file in Visual Studio Code
code . # Opens the whole directory in Visual Studio Code
cp [file] [dir] # Copy a file to a directory
rm [file] # Removes file, be careful with this command!!
rm -r [directory] # Removes a whole directory with content. Again be extra careful here
sudo [command] # Run command with the security privileges of the
exit # Exit
clear # Clear screen
```

Special for Windows

On Windows, the drives are mounted in a special location `/mnt/`. For example, C Drive is mounted on `/mnt/c` and D drive is mounted on `/mnt/d`. You can access your windows files in C Drive by using the following commands:

```
# Change Directory to C Drive
cd /mnt/c

# List all the files.
ls -al
```

Optional: if you don't want to access your C-Drive every time by the path `/mnt/c` . You can run the following commands to create a symbolic link in your home directory.

```
cd ~
ln -s /mnt/c
cd ~/c
# Now in the path of /mnt/c
```

From now on , if you are trying to install any command line based applications like `nodejs` , `git` or `ngrok` , you should always use ubuntu to install the software.

On the other hand, if you are installing graphical applications (such as `Visual Studio Code` or `Slack`), you should use windows to install the software.

Where do I place my code?

For windows, you are not supposed to use windows application to modify the files in ubuntu bash. Therefore , we strongly advise you to keep your code in a directory in the folder `/mnt/c/` such that you can access the files from both windows application and ubuntu bash shell.

Extra reading about the command line

You can read more in [here](#) why it is not advised to use windows application to modify the files in ubuntu bash.