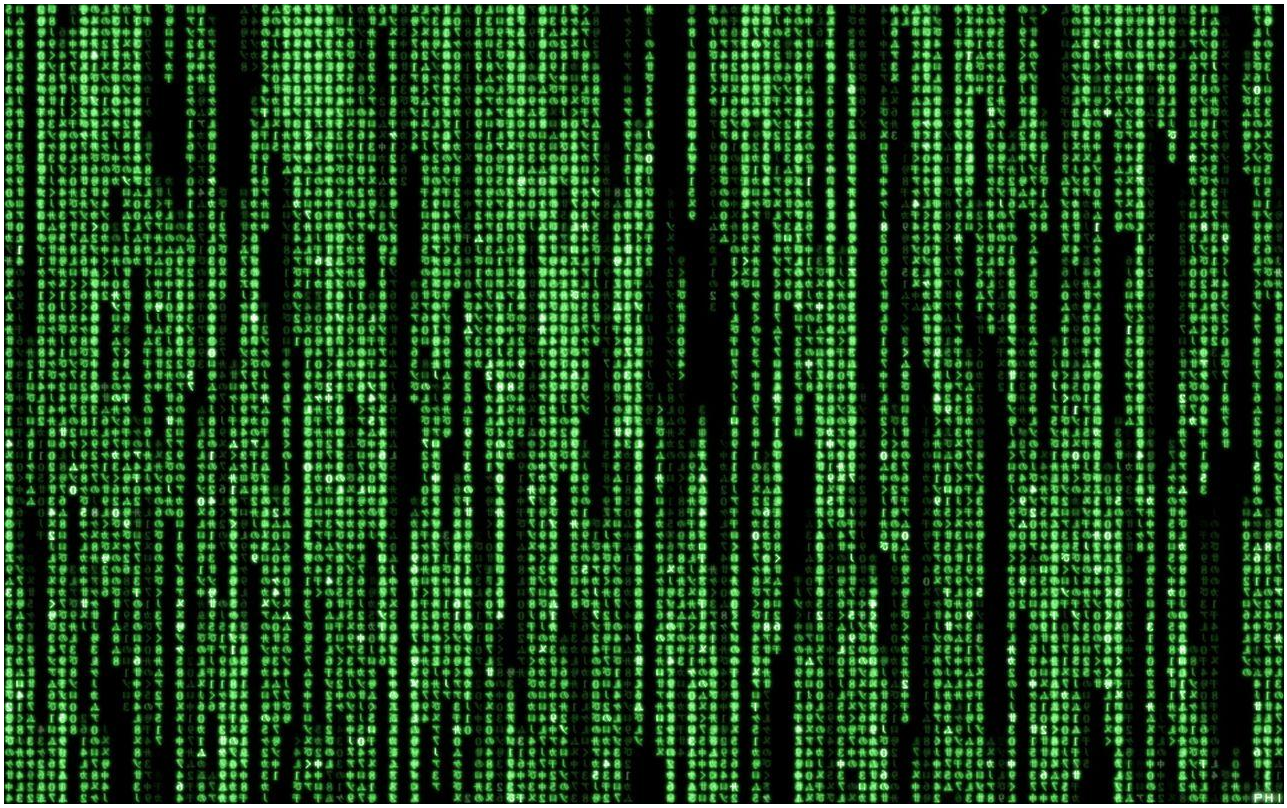
Week 02 - Introducing the CLI

Louis Botha, `louis.botha@tuni.fi`



The command line (CLI) is a tool for talking to your operating system (e.g., macOS, Windows, Linux, etc.) using text instead of by moving around a mouse and clicking on things. Just as you can use your mouse to open folders, move and rename files, and launch programs, so too can you use the terminal to ask your operating system to do the same things. There are *lots* of names that float around that basically mean the same thing, including terminal, shell, command line, and bash.

Why Should I Learn the Command Line?

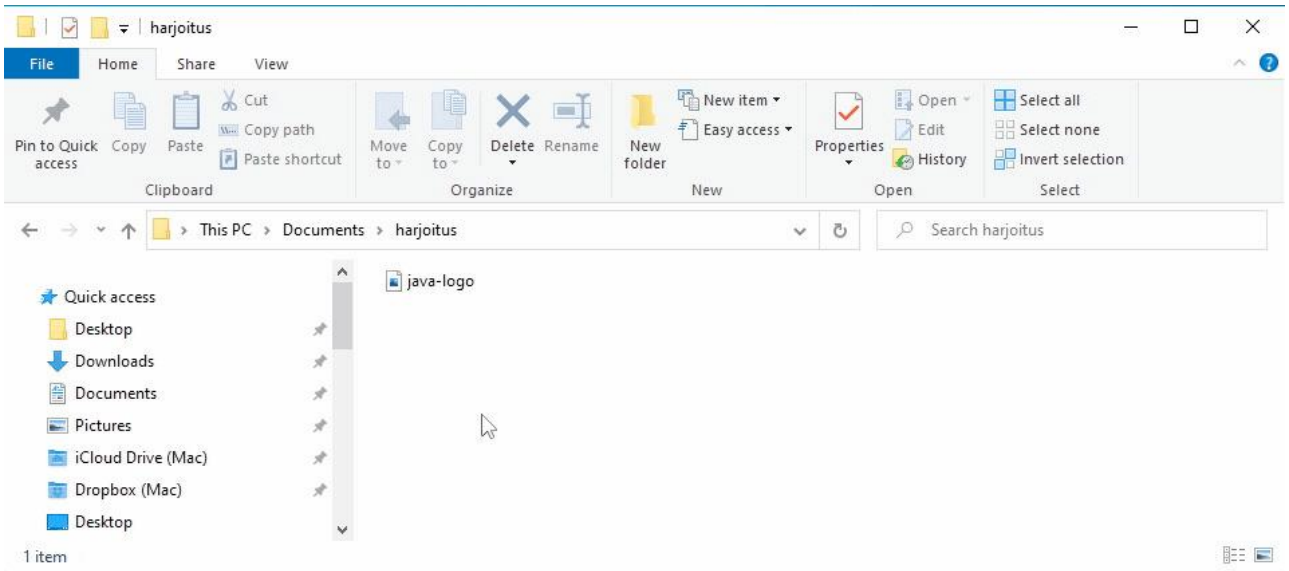
There are three main reasons to learn to use the command line.

1. **More tools are available through the command line than through graphical user interface.** It turns out that it takes a lot of work to make a program that has

pretty icons and windows that pop up, so lots of people make tools but don't bother to make pretty graphical interfaces. This is especially true for free and open-source software. So, by learning to use the command line, you gain access to *lots* of powerful tools that would otherwise be unavailable. For example, you basically *have* to use the command line to:
    - install and manage packages in many programming languages like Python and JavaScript
    - manage Git repositories
2. **You won't always have a graphical interface.** If your work ever requires you to use a remote server with, say, more computing power, those systems probably won't offer a graphical user interface.
3. **Sometimes you want to do something OVER AND OVER.** Renaming a file by clicking on it, deleting the name, typing in a new name, and hitting return is great if you only need to rename one file. But what if you want to rename hundreds? The command line gives you the ability to write little scripts to do this kind of work for you.

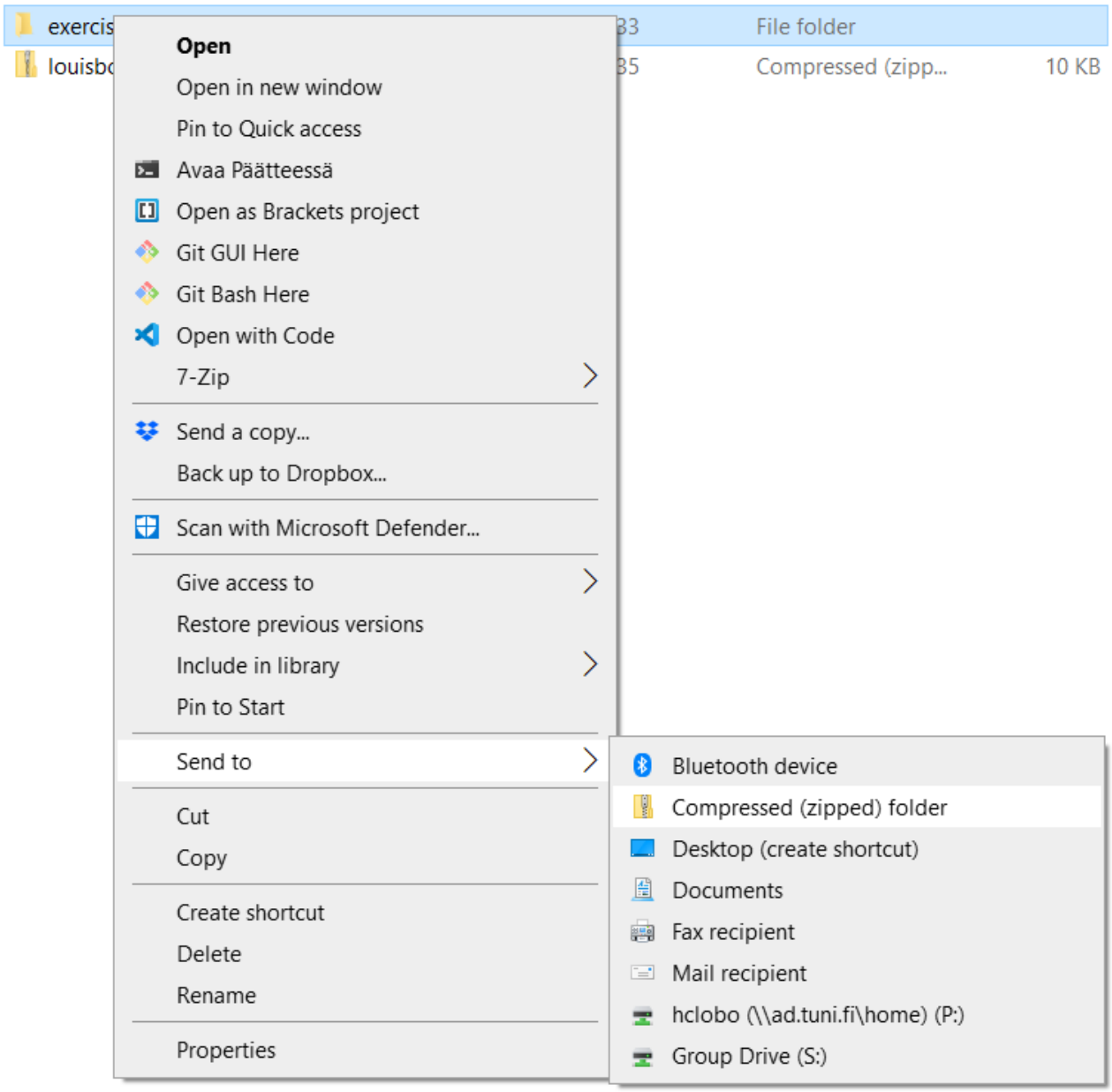
1. Operating system

- Create a directory on the Desktop or any place you choose (in `c:\users\<username>\Documents` for example) called `exercise`.
- Open the Notepad text editor.
    - macOS: TextEdit
- In the document write the names of 3 different programming languages (:google_sign:: common programming languages). Save the file to the `exercise` directory you created in the step above and give it your full name as the filename, e.g `clint-eastwood.txt`.
- Use google image search to find logos for the programming languages you wrote in the text document, eg. "python language logo".
- Browse to the page where the image is found. Right click on the image and use "save image as" to save it to the `exercise` directory you created.
- Rename the image files to have simpler filenames, eg `python.png`, `java.png`
- Set the Windows File Explorer to show file extensions (n/a to macOS)



File Explorer > View > File name extensions

- Create a ZIP package of all the files in the `exercise` folder. You will find the option when you Right-Click on the folder and under the send to.



- Give your name as the name of the zip package, eg. `clinteastwood.zip`

Exercise - Part 1

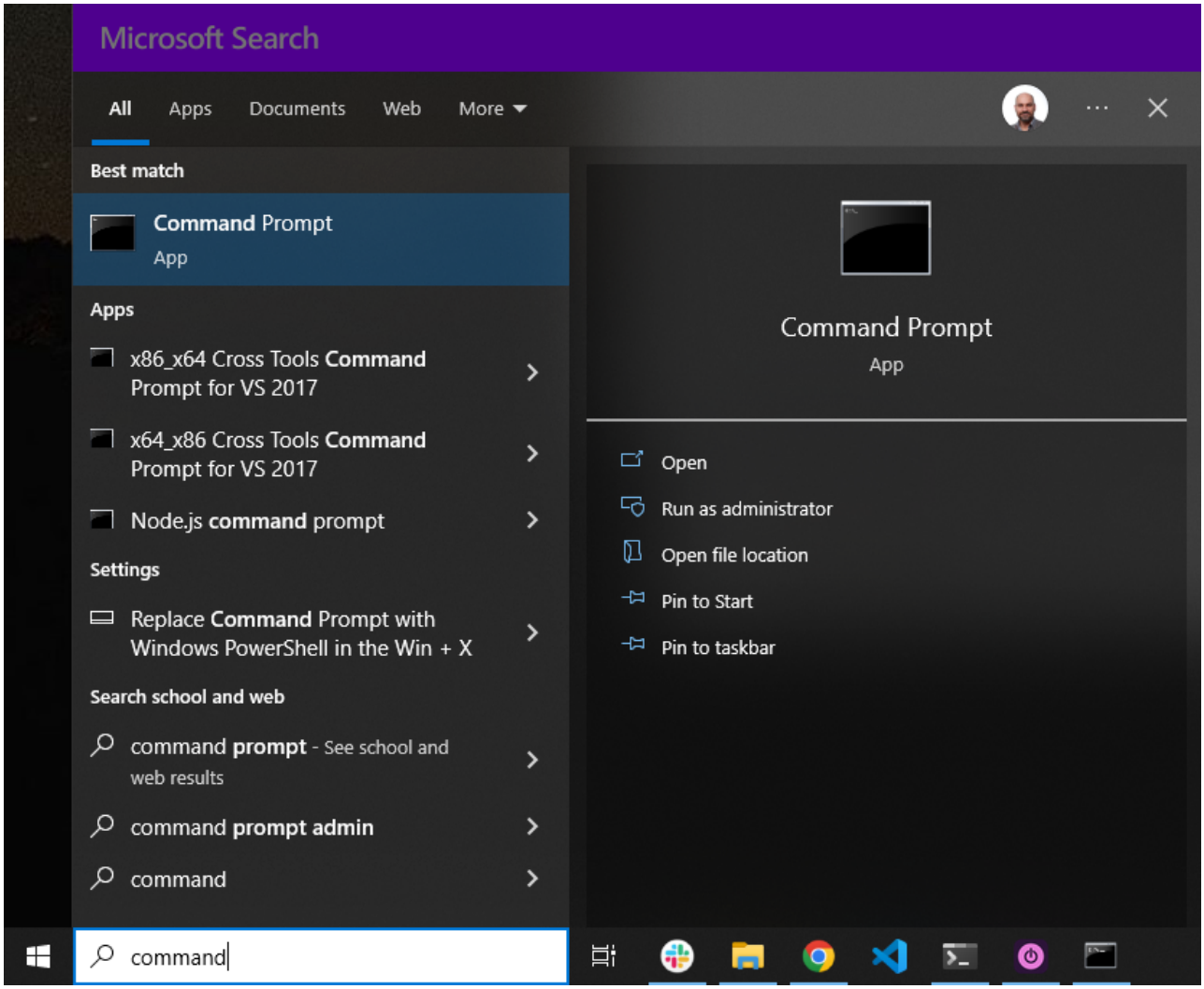
Upload and post the zip package you created to ::slack: in the #general channel in the 🧵 Introducing the CLI - Part 1 thread.

---

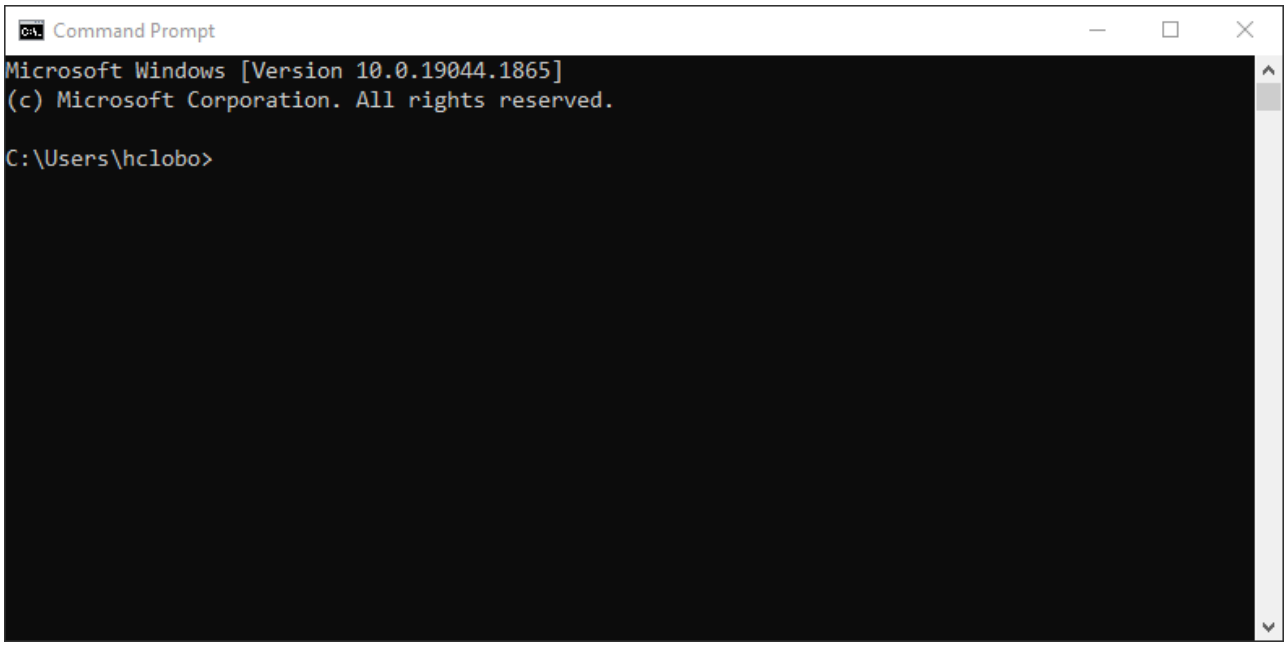
2. Getting to the Command Line

Windows OS

- Open the Command Prompt directly.

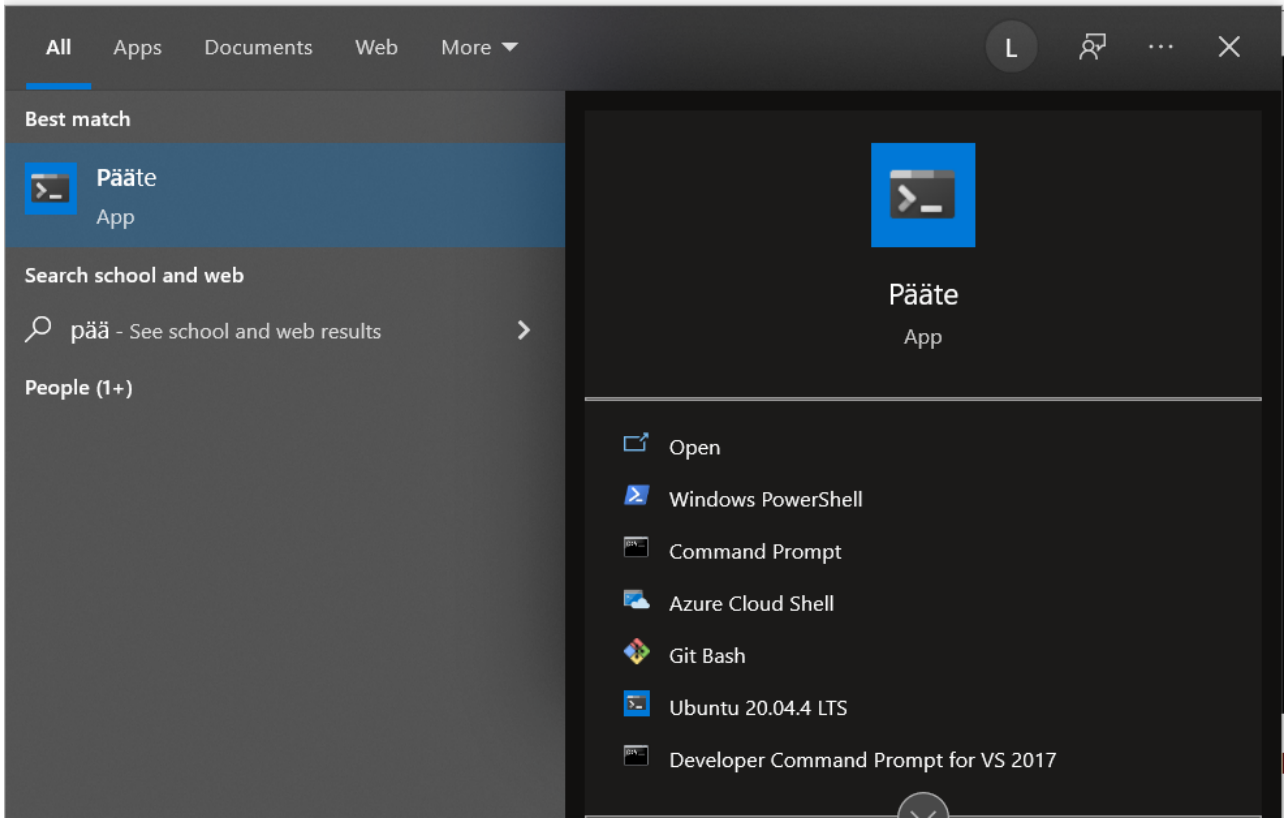


User-uploaded image: image.png

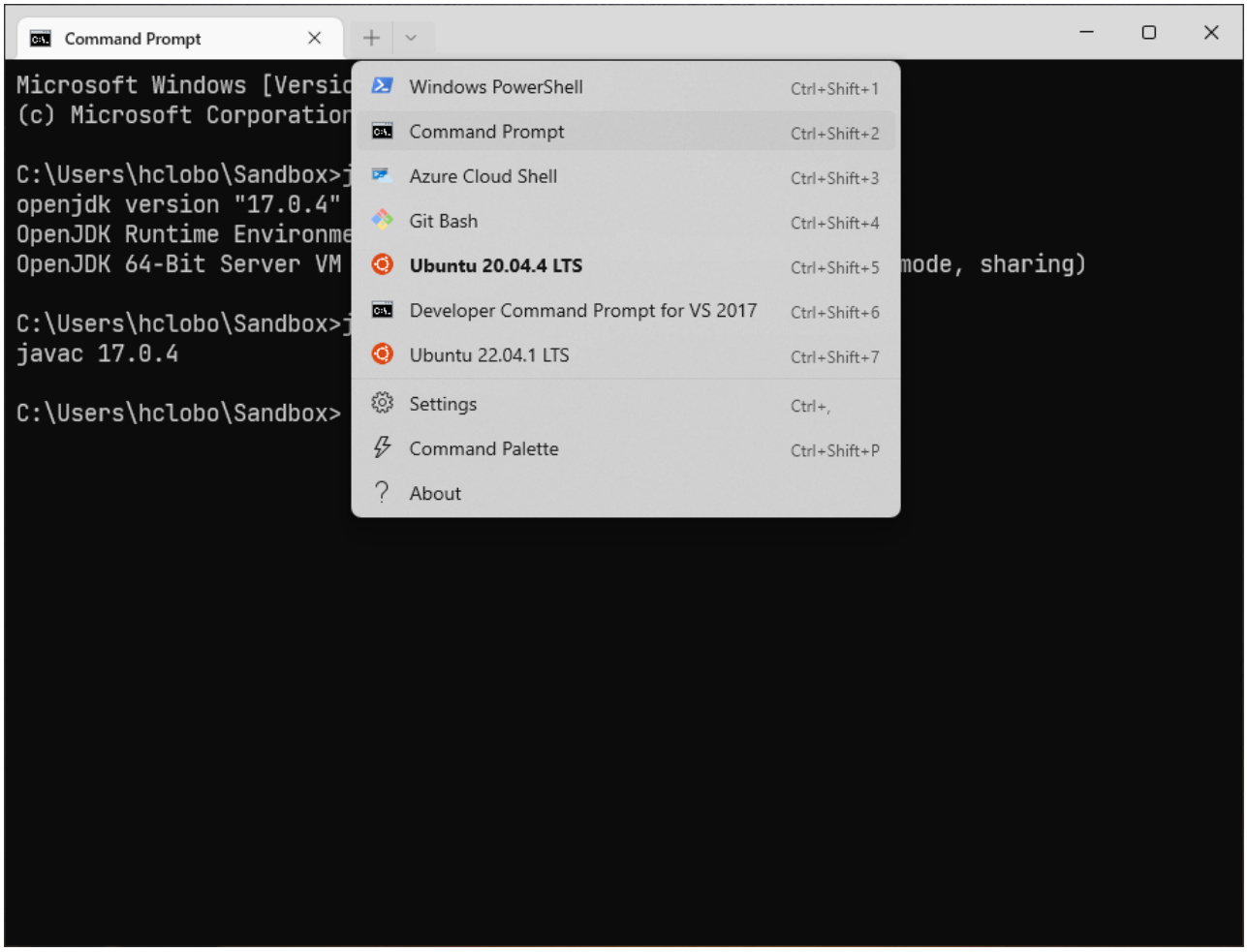


User-uploaded image: image.png

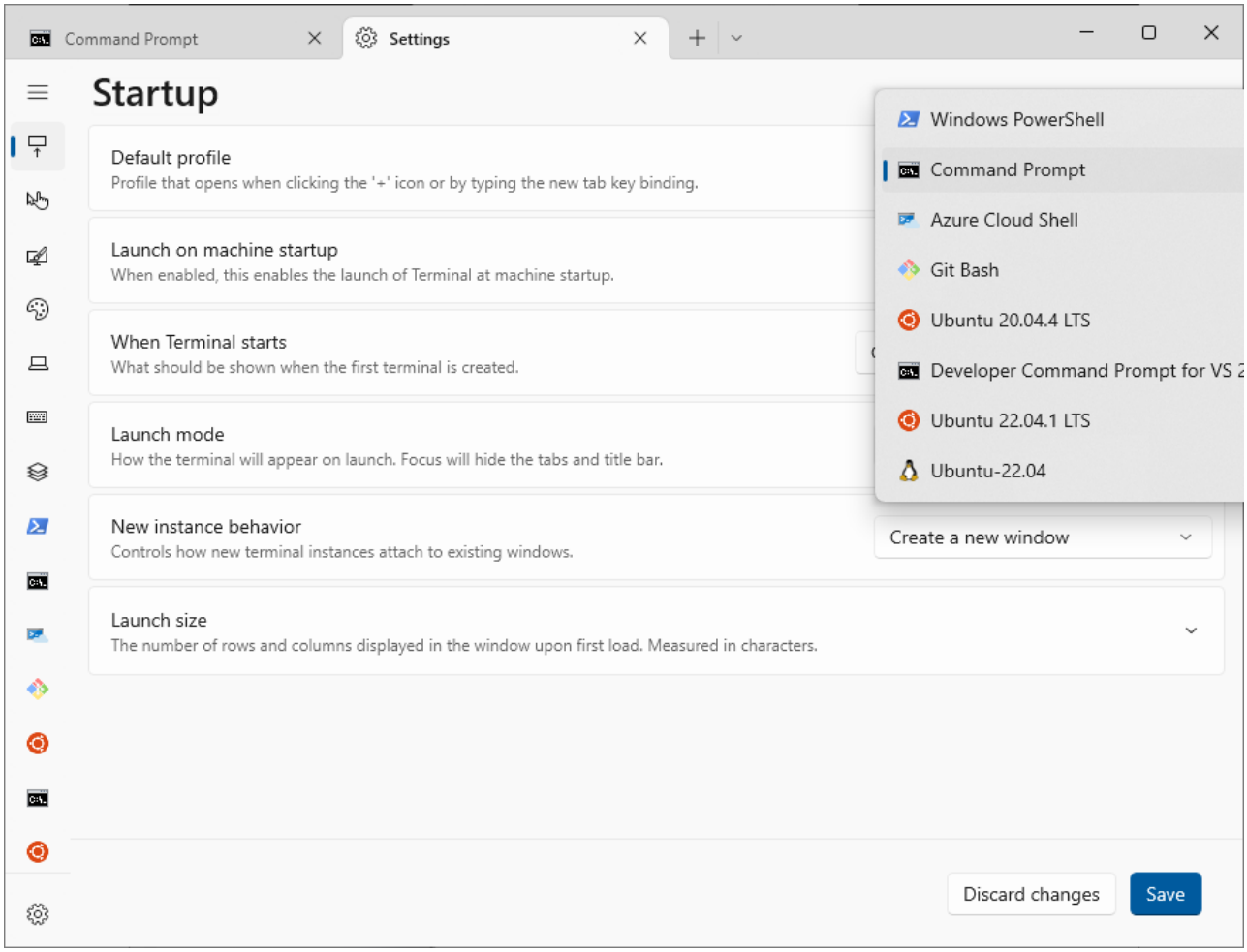
- Open Command Prompt from the Windows Terminal (Recommended way)
  - If you have a Finnish computer it will most likely be called **Pääte**



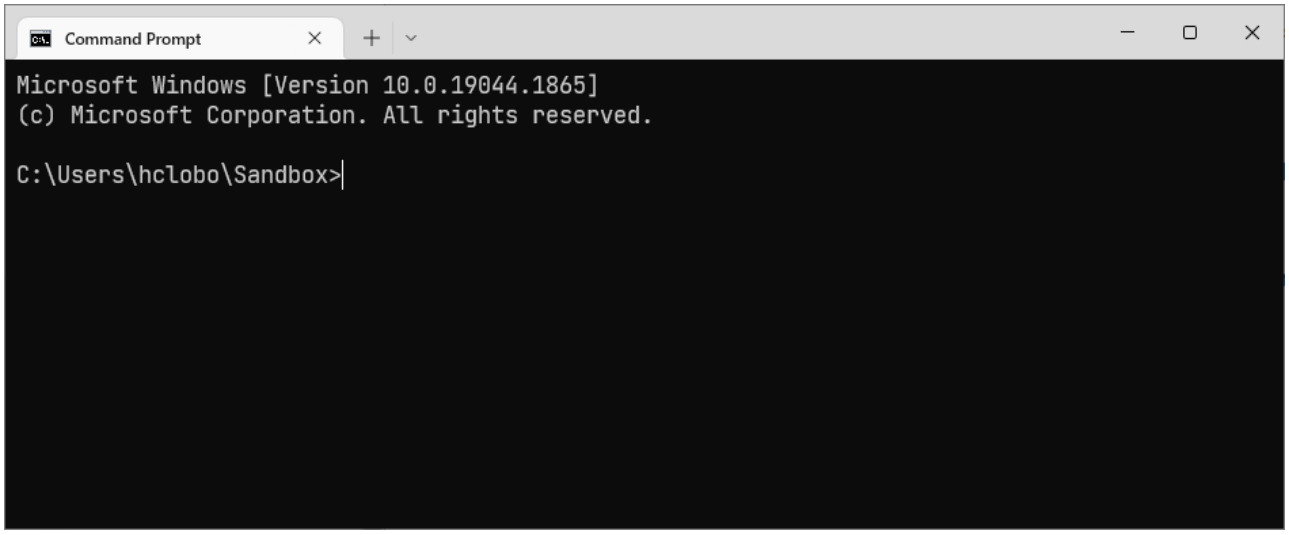
The default might be to open the Windows PowerShell. Press the little down arrow to get a selection of available command line shells available. Select **Command Prompt** you have installed.



Under settings there is an option to select the default profile. I am using the Command Prompt as default.



The terminal should look something like this.



User-uploaded image: image.png

macOS

- Terminal

- iTerm2

3. Using the Command Line

- **Where am I?**
  You can see in which directory the command line prompt is with:

```
#GitBash and Linux
pwd
/home/louis

#Windows
cd
C:\Users\hclobo

echo %cd%
```

```
C:\Users\hclobo

#macOS
pwd
/users/louis
```

Note: By default most terminals will open in your HOME directory.

- **How do I see this place in the Operating System File Explorer?**

```
#GitBash
explorer.exe .

#Windows
explorer.exe .

#macOS
open .
```

- **What's in this directory?**

```
#GitBash and Linux / macOs
ls
Courses  Downloads  Playground  Sandbox  Temp  moss

#Windows
dir
17.05.2022  10.54    <DIR>          OneDrive - TUNI.fi
11.04.2022  08.35    <DIR>          Sandbox
10.01.2022  13.25    <DIR>          Saved Games
25.02.2022  14.10    <DIR>          Searches
18.05.2022  12.52    <DIR>          Videos
               5 File(s)         10 658 bytes
              24 Dir(s)  384 399 884 288 bytes free
```

- **How do I change the directory I am in?**
  cd <directory> <ENTER>

```
#GitBash and Linux / macOS
cd Downloads
/home/louis/Downloads
~/Downloads >

#Windows
C:\Users\hclobo>cd Documents
C:\Users\hclobo\Documents>
```

> *Most modern shells have completion enabled. This means that you can use TAB to see suggestions. This is extremely helpful for traversing through the directories.*

For example:

```
# I have a directory that has 2 directories Downloads and Docu
ments
cd D <Press TAB>
# Shows that there is 2 possibilities
Documents/ Downloads/
cd Doc <Press TAB>
cd Documents/ >
```

- **How do I go back from a directory?**

```
#GitBash and Linux / macOS
cd ..

#Windows
cd ..
```

Note: You can go back multiple directories with `cd ../../../`, each ../ represents a directory.
Note: In some shells you can go back to your HOME directory with just `cd`

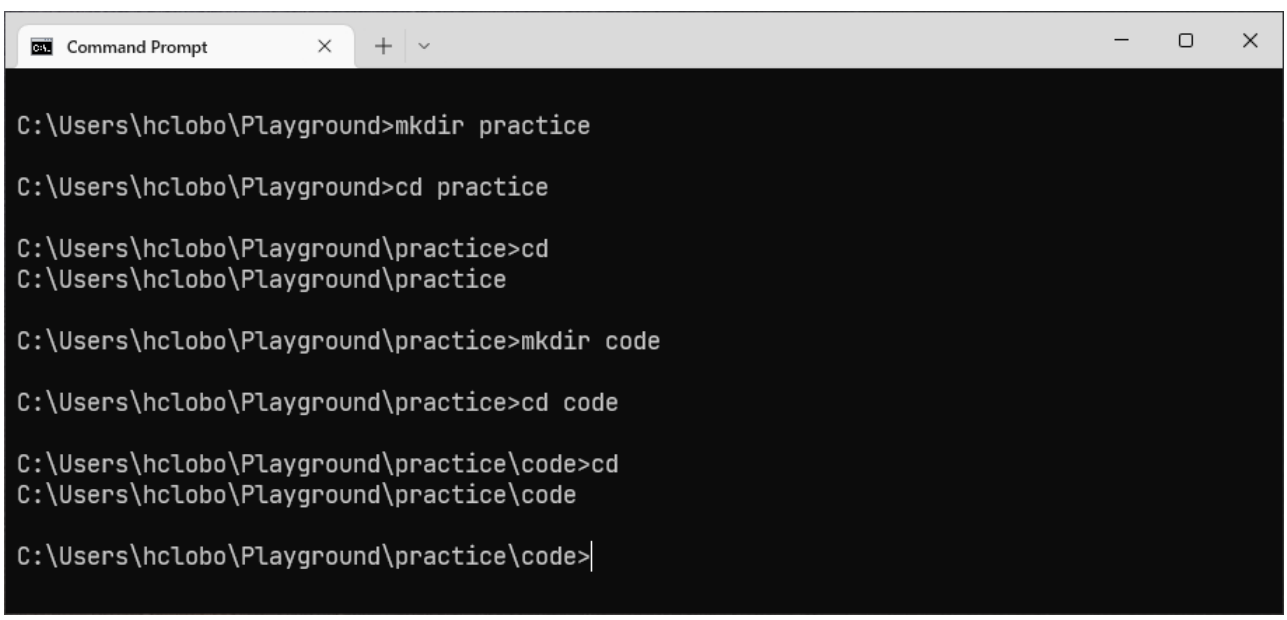
- **How do I make a directory?** mkdir <name>

```
#GitBash and Linux / macOS
mkdir exercise


#Windows
mkdir exercise
```

Let's Practice

1. Create a directory called `practice`.
2. Change into the `practice` directory.
3. Display where you are.
4. Create a directory called `code`.
5. Change into the `code` directory.

Windows Command Prompt



GitBash

```
MINGW64:/c/Users/hclobo/Pl

hclobo@WKS-98407-LT MINGW64 ~/Playground
$ mkdir practice

hclobo@WKS-98407-LT MINGW64 ~/Playground
$ cd practice

hclobo@WKS-98407-LT MINGW64 ~/Playground/practice
$ pwd
/c/Users/hclobo/Playground/practice

hclobo@WKS-98407-LT MINGW64 ~/Playground/practice
$ mkdir code

hclobo@WKS-98407-LT MINGW64 ~/Playground/practice
$ cd code

hclobo@WKS-98407-LT MINGW64 ~/Playground/practice/code
$ pwd
/c/Users/hclobo/Playground/practice/code

hclobo@WKS-98407-LT MINGW64 ~/Playground/practice/code
$
```

- **How do I quickly create a file?**
  touch <filename.extension> <ENTER>
  type nul > <filename.extension> <ENTER>

```
#WSL and Linux / macOS
touch Code.txt
ls
Code.txt


#Windows
type nul > Code.txt
dir
18.05.2022  14.47                    0 Code.txt
```

```
               1 File(s)              0 bytes
               2 Dir(s)  383 943 254 016 bytes free
```

- **How do I open the file for editing?**
  Visual Studio Code should be installed. We will open the file with VSCode :vscode:.

```
#WSL and Linux / macOS
code Code.txt

#Windows
code Code.txt
```

> *:attention: Mac users might need to install VSCode to the path first.*

Copy the following piece of code into the file that is opened in VSCode :vscode:.

```
class Code {
    public static void main(String[] args) {
    }
}
```

Save (CTRL + S | CMD + S) and close VSCode :vscode:.

- **How do I see what is in a file?**

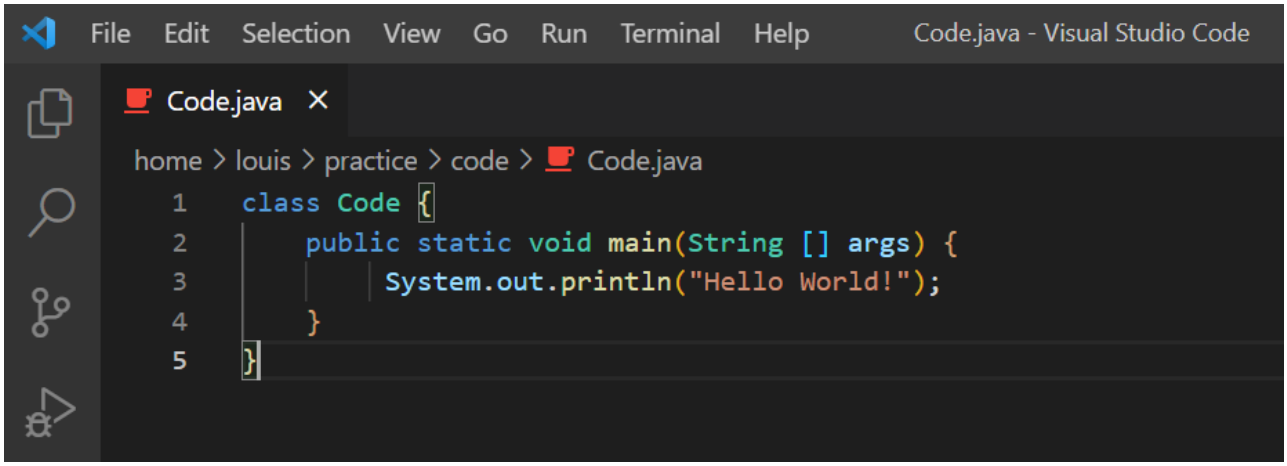
```
# WSL and Linux / macOS
cat Code.txt

# Windows
type Code.txt
```

- **How do I rename a file?**

```
# WSL and Linux / macOS
mv Code.txt Code.java

# Windows
```

```
rename Code.txt Code.java
```

Now open the Code.java file with VSCode :vscode:. Notice that the .java file extension is recognized by VSCode :vscode: and that there is now syntax highlighting.



Close VSCode :vscode:.

- **How do I copy a file?**
  💡 On the command line it is always FROM → TARGET

```
# WSL and Linux / macOS
cp Code.java Copy.java
'Code.java' -> 'Copy.java'

ls
Code.java  Copy.java

# Windows
copy Code.java Copy.java
        1 file(s) copied

dir
```

```
18.05.2022  14.45                 0 Code.java
18.05.2022  14.45                 0 Copy.java
               2 File(s)              0 bytes
               3 Dir(s)  383 931 199 488 bytes free
```

- **How do I install a command line tool?**
  Let's install a tool called **tree**. This is WSL and macOS only. Tree is available on Windows by default.

```
# WSL and Linux
sudo apt install tree


[sudo] password for louis:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 1s (64.5 kB/s)
Selecting previously unselected package tree.
(Reading database ... 32549 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

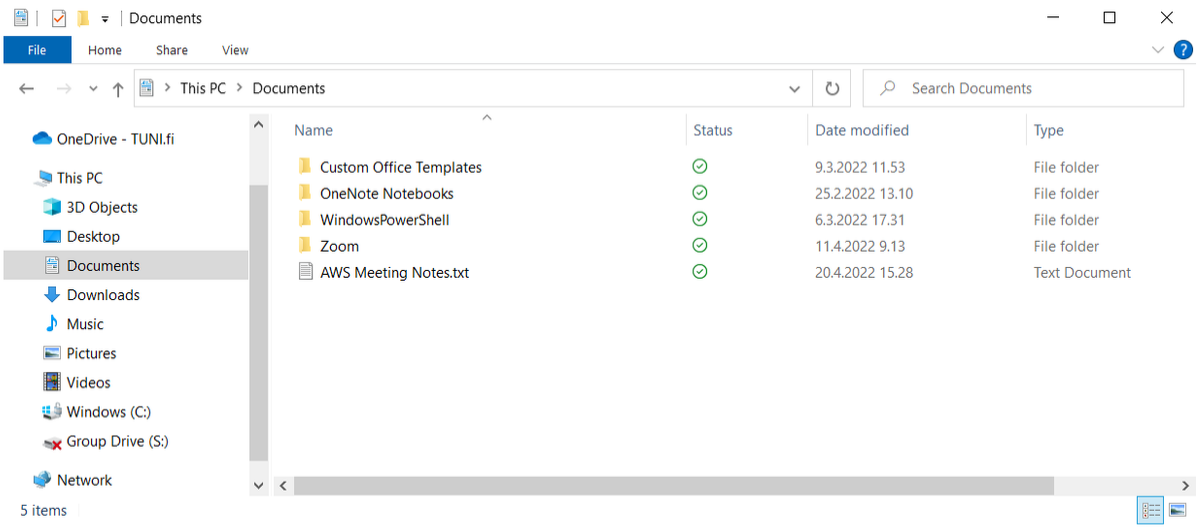
```
tree
.
└── code
    └── Code.java
1 directory, 1 file

# macOS
brew install tree
...
tree
.
└── code
    └── Code.java
1 directory, 1 file

# GitBash - Need to call the Windows tree command
$ cmd //c tree //f
Folder PATH listing for volume Windows
Volume serial number is 9E10-A7E3
C:.
└───code
        Code.java

// Create an alias in bash
$ alias tree='cmd //c tree //f'
$ tree
Folder PATH listing for volume Windows
Volume serial number is 9E10-A7E3
C:.
```

```
└──code
        Code.java
```

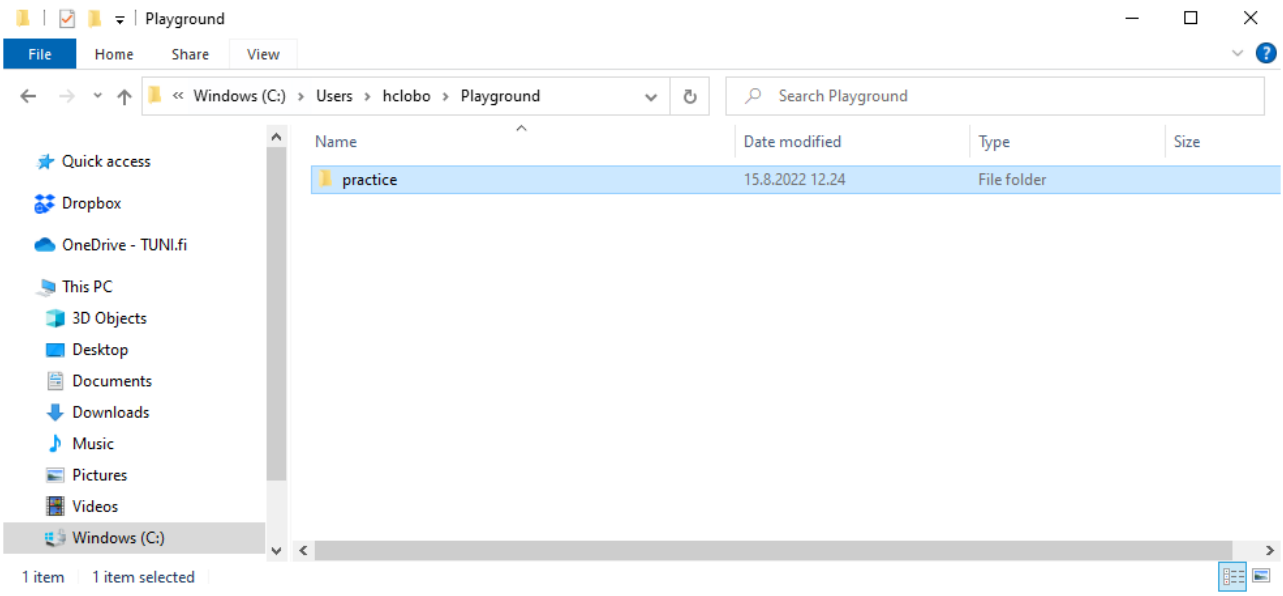
Close the terminal :terminal:.

- **How do I open a shell in a specific directory?**

On Windows, open the File Explorer

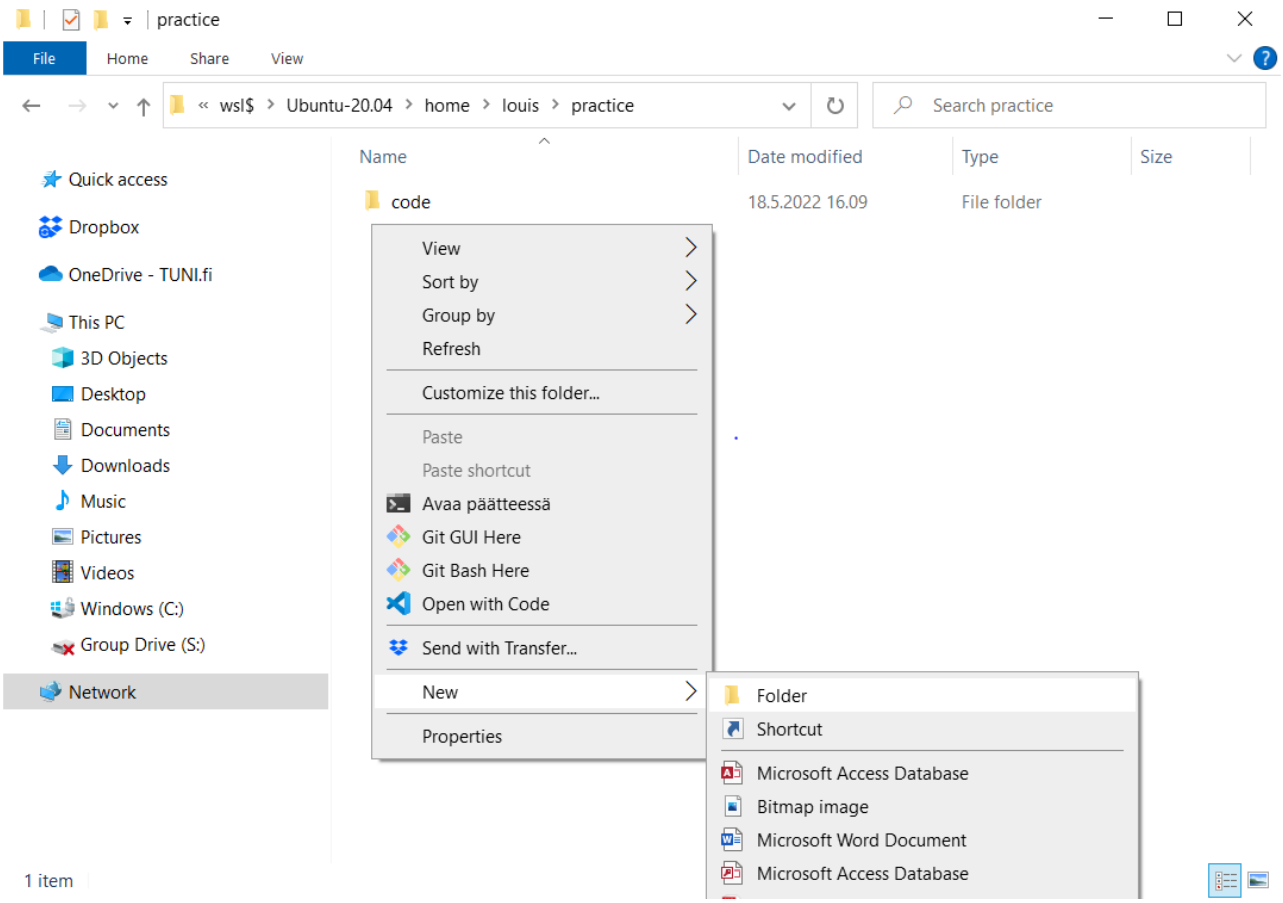


Find the `practice` directory

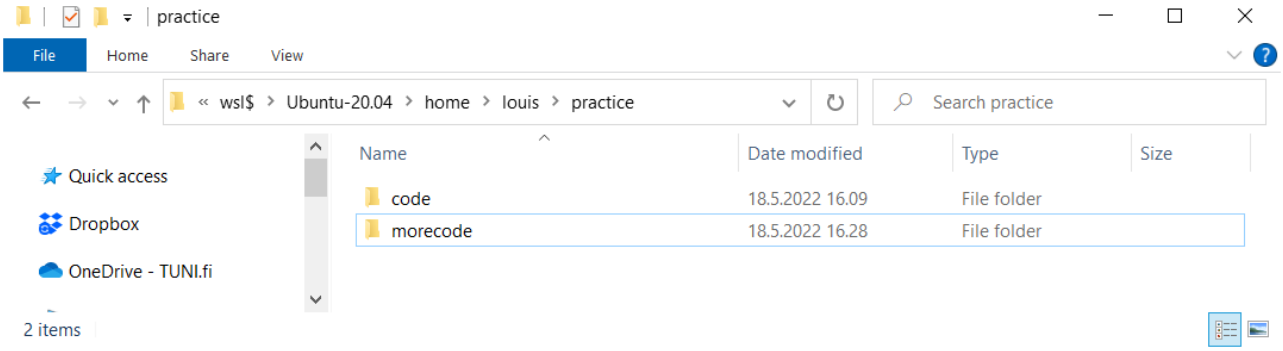


Double-Click on the `practice` directory to open it in the File Explorer.

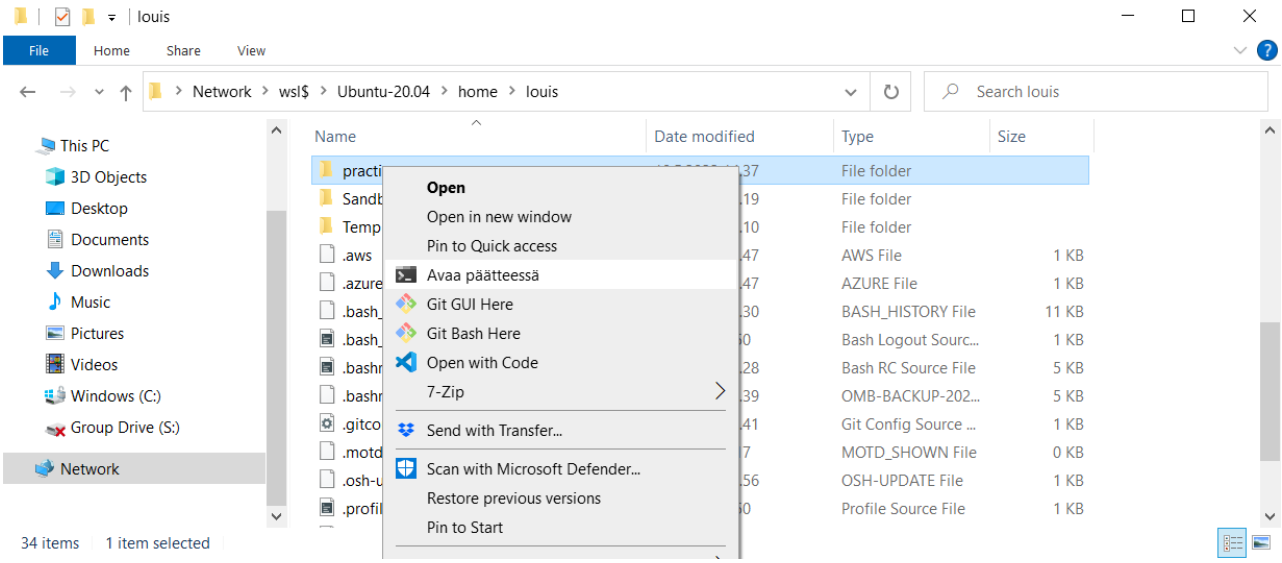
Create a new folder in the `practice` directory. Right-Click and select `New → Folder` and



Name the new directory `morecode`.

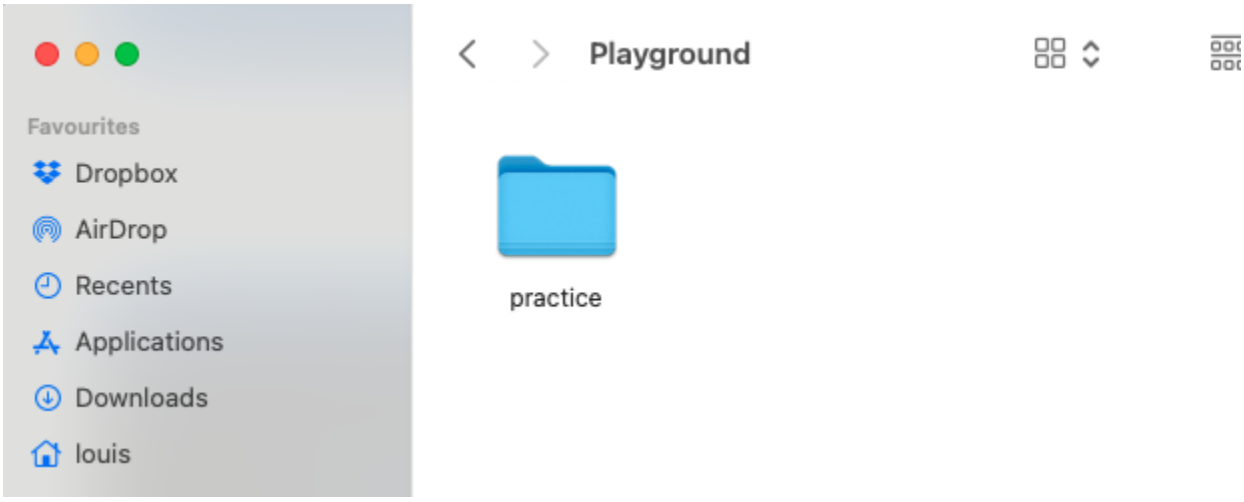


Right-Click on the `morecode` directory and select `"Open in Pääte" | "Open in Terminal | "Git Bash Here"`.



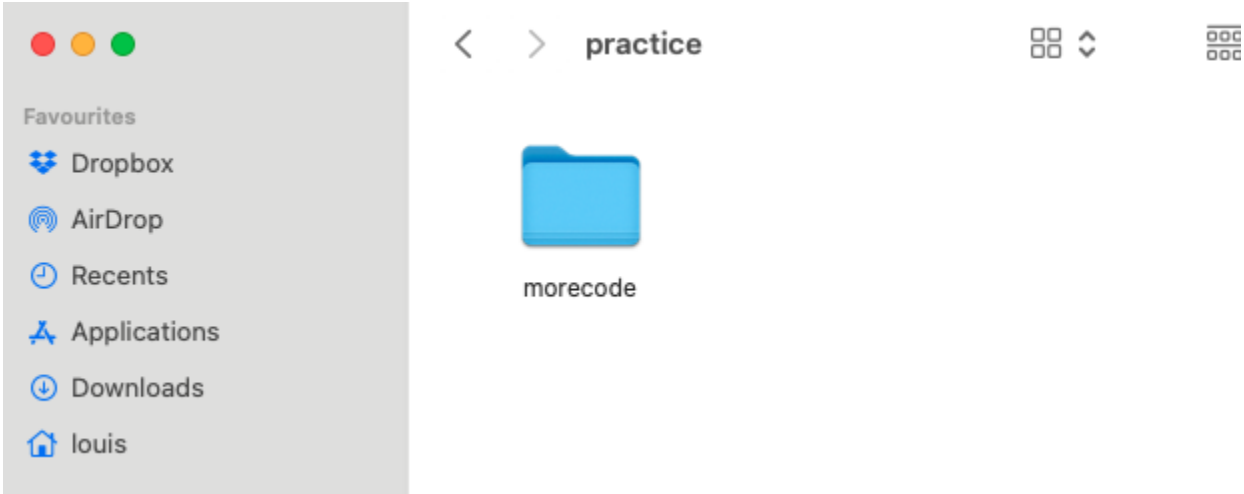
You will need to use the CLI commands to drill down to the directory.

On macOS, open Finder

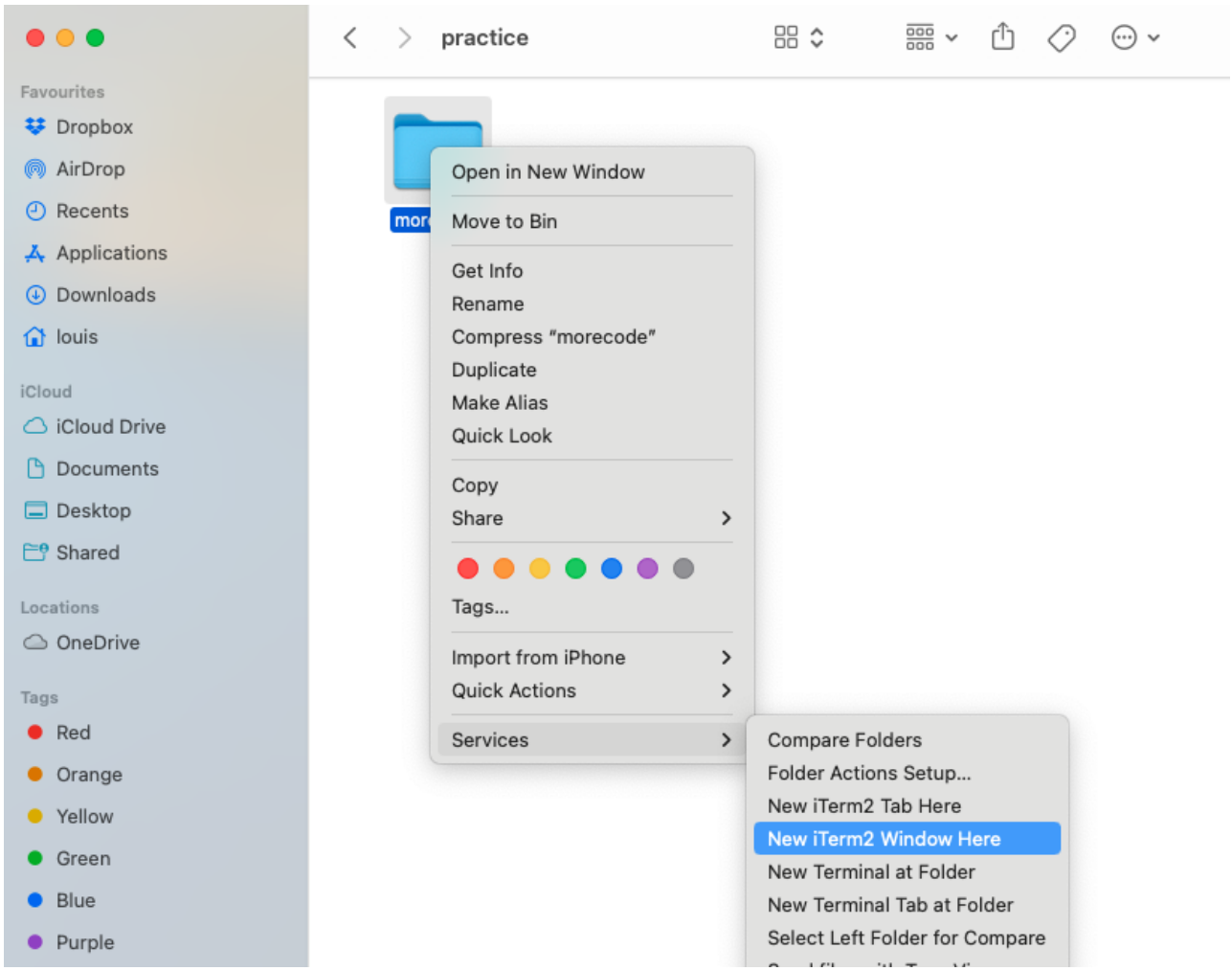


Double-Click on the `practice` directory to open it in the Finder.

Create a new folder in the `practice` directory. Right-Click and select `New → Folder` and



Right-Click on the `morecode` directory and select “New Terminal at Folder” | New iTerm2 Window Here”.



Terminal or iTerm will open in the selected directory.

```
-zsh                                                    ⌥⌘1
Last login: Tue May 17 13:52:57 on ttys001
 ~/Playground/practice/morecode                          17:15:07
❯
```

Exercise - Part 2

1. Open the `morecode` directory you created in the CLI
2. Make a directory inside `morecode` called `fi`
3. Make a directory inside `fi` called `organisation`
4. Make a directory inside `organisation` called `project`

```
 ~  cd practice/morecode/
/home/louis/practice/morecode
 ~/practice/morecode  mkdir fi
mkdir: created directory 'fi'
 ~/practice/morecode  cd fi/
/home/louis/practice/morecode/fi
 ~/.../morecode/fi  mkdir organisation
mkdir: created directory 'organisation'
 ~/.../morecode/fi  cd organisation/
/home/louis/practice/morecode/fi/organisation
 ~/.../fi/organisation  mkdir project
mkdir: created directory 'project'
 ~/.../fi/organisation  cd project/
/home/louis/practice/morecode/fi/organisation/project
 ~/.../organisation/project
```

5. Copy Code.java from the `/practise/code/` directory to the `project` directory

```
✗  ~/.../organisation/project  cp ../../../../code/Code.java .
'../../../../code/Code.java' → './Code.java'
 ~/.../organisation/project  ls
Code.java
```

6. Navigate back to the `morecode` directory
7. Show the tree of the `morecode` directory

```
~/practice/morecode  tree
.
└── fi
    └── organisation
        └── project
            └── Code.java

3 directories, 1 file
```

8. Take a screenshot and post it to ::slack: in the #general channel in the 🧵 Introducing the CLI - Part 2 thread.

When you have completed both exercises, you can mark the Week 02 - 3 as done on the :google_sheets: sheet.

> *:attention: I will be using GitBash as my default command line during the course*