Command Line Lecture

Class	01_visual
Created	@Jan 7, 2021 1:31 PM
Materials	
Reviewed	□
Type	

The Command Line

1) Start of the course

What is your most important tool over the next 12 weeks?

- Python
- Anaconda
- Terminal
- StackOverflow
- git
- slack
- Our mind
- Computer

2) How can we navigate our Computer?

- Terminal / Command Line Interface
- Graphical User Interface (around 1980) - GUI

3) Why do we care about the Command Line Interface?

- Automation of tasks is a lot easier with a CLI

- It is not always possible to use a GUI, eg. on Ubuntu; more generally when we communicate with some device that does not provide graphical output - eg. Cloud computing
- More efficient error handling
- A GUI is an additional piece of software that might be missing
- Often a job requirement
- You will need it for tomorrow - git

4) Shell

- Interface between the User and the Operating system
- Most commonly used shell is `bash (Bourne Again Shell)` . Bash is the default shell for Linux.
- Bash was the default shell on Mac until some time in 2019
- Newer Macs by default use `zsh` . Zsh is different shell, it is very similar to bash
- Bash is a fully fletched programming language

5) Create a spiced projects folder

- `pwd` - print working directory: prints the path to the current working directory
- `cd` - change directory
- `ls` - list files (and directories) in current directory
- `mkdir <name-of-directory>` - make directory
- `clear` - clears the terminal window from all previous commands and output; `ctrl + l` does the same job

6) Bash Commands seen

- `cd <name-of-directoy>` - change directory to ...
- `cd ..` - go one directory up (same as `cd -`)
- `cd` - go back to the home directory
- `ls` - list files
- `ls -a` - show hidden files
- `ls -l` - list files with additional information
- `mkdir <name-of-directory>` - create a directory
- `pwd` - print working directory
- ⬆ - last command
- **Tab key** - auto-complete
- `source <name-of-program>` - execute a bash program
- `less` - display content of a file
- `nano <name-of-file>` - open a file in the nano text editor
- `man <command>` - displays the manual of a specific command
- `cp <file> <path>` - copies a file to a different path
- `*` - all files in a directory
- `rm <filename>` - remove a file. Be careful, this removal is permanent!
- `rm -r <directory-name>` - remove a directory and all files inside
- `python <filename.py>` - run a python file from the command line
- `clear`- clear the screen