Day I - Part I - Introduction to command line

December 2, 2020

1 Introduction to command line

Identify and open your command line interface. First, it is important to note that this will now differ (but not substantially) depending on whether or not you're on a Windows machine:

- Windows: we will choose to use **Anaconda Prompt** (which was installed on your machine when you installed Anaconda).
- nix (another way to describe Mac OS and/or Linux machines): we will use the system **terminal**.

2 Finding your computer's name.

Let us first let's find out the name of your computer by running:

Windows

```
$ echo %USERNAME%
```

nix

```
$ whoami
```

3 Finding your current location

Now let's find out which directory (folder) we are currently in:

Windows

```
$ cd
```

This stands for "current directory".

nix

```
$ pwd
```

This stands for "present working directory"

Type the command in and press enter. It should list where you are currently located in your command line interface.

4 Seeing what is in your current location

To view the contents of the current directory:

Windows

```
$ dir
```

This stands for "directory".

nix

```
$ ls
```

This stands for "list"

Type the command in and press enter. You should see a list of the various files and directory in your current directory. Open your current directory in a graphical user interface and compare.

5 Moving to another location

On both Windows and nix if you want to enter a directory that is in your current diractory type:

```
$ cd <directory>
```

Try moving to your Desktop. It should be something like:

```
$ cd Desktop
```

6 Creating a directory

To create a directory:

```
$ mkdir <directory_name>
```

Experiment with creating a directory for this workshop:

```
$ mkdir rsd-workshop
```

If your directory structure looked like this:

```
|--- home/
|--- Desktop/
     |--- research
     |--- photos
```

It will now look something like:

```
|--- home/
|--- Desktop/
     |--- research
     |--- photos
     |--- rsd-workshop
```

As an exercise move into the directory we just created:

```
$ cd rsd-workshop
```

and create two further directories:

```
|--- rsd-workshop
     |--- src
     |--- test
```

If you now wanted to go back to the "parent" directory:

```
$ cd ..
```

Where `..` is short hand for a previous directory.

Experiment with these, in combination with the command to find your current location as well as the command to list the contents of your directory.

7 Creating a file

To create a directory:

Windows

```
$ echo <file_name>
```

nix

```
$ touch <file_name>
```

Experiment with creating a file named `addition.py` in the directory `rsd-workshop`.

Windows

```
$ echo addition.py
```

nix

```
$ touch addition.py
```

If you type `ls` you will see that the file has been created.

8 Copying files

To copy a file:

Windows

```
$ copy <file> <new_file_directory_and_name>
```

nix

```
$ cp <file> <new_file_directory_and_name>
```

Experiment with copying any file.

9 Moving/renaming files

To move a file:

Windows

```
$ move <file> <new_file_directory_and_name>
```

nix

```
$ mv <file> <new_file_directory_and_name>
```

Experiment with moving any file. Note that if you want to rename a file you can do this by passing the new name in the same directory. WARNING When using the command line interface you will not be prompted for confirmation if move/mv were to overwrite another file. Be careful.

10 Deleting files

To delete a file:

Windows

```
$ del <file>
```

nix

```
$ rm <file>
```

11 Copying and removing directories

To copy a directory:

Windows

```
$ xcopy <dir> <target>
```

nix

```
$ cp -r <dir> <target>
```

To remove a directory:

Windows

```
$ rmdir /s <dir>
```

nix

```
$ rm -r <dir>
```