



# Conda and the Terminal

The command line/terminal is a text-based program used to powerfully interact with the computer's systems and processes. Here you can perform operations like running python scripts and managing software or hardware, without the use of the operating system's graphical user interface (GUI).

This notebook will cover how to

- Navigate the file system
- Use the Miniconda Python package manager
- Write and run a python program

## Navigation

Navigating around the computer without the graphical file explorer can be challenging at first if dragging/dropping and clicking on things is what you're used to.

The below examples list six common Windows and equivalent Unix commands.

1. dir/ls: Lists each file in the directory you are currently in.
2. pwd: Displays the current working directory.
3. cd: Move into the directory given and make this the working directory.
4. mkdir: Create a directory with the given name and path.
5. echo/touch: Create a file with the given name and path.
6. move/mv: Move a file to the given destination directory path.

## Windows

```
dir
pwd
cd <new_location>
```

```
mkdir <new_dir>
echo > <new_file>
move <file> <location>
```

## Mac/Linux

```
ls
pwd
cd <new_location>
mkdir <new_dir>
touch <new_file>
mv <file> <location>
```

## Miniconda

Python is not lightweight and often your projects will require large libraries.

It is not necessary or good practice to install libraries directly onto your base Python installation and Anaconda/Miniconda is software that allows you to create and manage 'virtual environments' that contain only the libraries you really need.

### Installation

1. Navigate to <https://docs.conda.io/en/latest/miniconda.html>
2. Windows: Download and run the relevant installer making sure to check the 'Add to PATH' option.  
OSX: Download the installer, navigate to its location in terminal and run the following command -make sure to accept the option to add the installation to the path.

```
bash Miniconda3-latest-MacOSX-x86_64.sh
```

## Commands

As with the default commands included in your OS, conda provides us with many of their own. A comprehensive list of these can be found here -

<https://docs.conda.io/projects/conda/en/latest/commands.html#conda-general-commands>

Some important ones are as follows, these are the same in both DOS and Unix based systems.

```
conda --version
conda env list
conda create -n <env_name> python=<py_version> <required module/s>
conda remove -n <env_name> --all
conda activate <env_name>
conda deactivate <env_name>
conda install <required_module/s>
conda remove <installed_modules>
```

## Hello World

Using the tools above, we can now write and run some Python.

Programming is giving a computer instructions to perform. Python is an **interpreted** language which means every instruction is evaluated and executed in real-time, whereas **compiled** languages evaluate all the instructions ahead of time and create a binary file which then gets executed later.

## The Interpreter

1. Create a new directory for this session using the command line and navigate into it.
2. Type 'python' to run the conda python interpreter. Here is where we execute python instructions directly.

```
C:\Users\user1>python
Python 3.7.7 (default, Apr 15 2020, 05:09:04) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## From File

1. Create a new file called 'temp.py'. The .py extension lets the computer and interpreter know what kind of file this is and how to run it.
2. Write an instruction or two you tried in the interpreter ie

```
print(3+4)
print("Hello World")
```

3. Save this file and run it in the command line typing 'python temp.py', ie calling the interpreter and pointing at your instructions in file format. Is there any difference between these python execution methods?