# DSCI 510: Principles of Programming for Data Science
Spring 2023 – Lab 1

Instructor: Prof. Jose Luis Ambite
Course Producers: Sanjana Parakh, Samuel Vara, Swarnita Venkatraman, Gursimar Kaur

**Introduction to Shell and basic shell commands**

The shell is the command interpreter for linux systems. It is the command line interface that interacts with the users in the terminal. Shell commands are instructions to the system, prompting it to do some action.

*For the rest of the semester, when given shell commands to type, note that when you see a "$" that is to indicate the command prompt from either the Mac/Linux terminal or Windows "command prompt" shell. You do not type the dollar sign, only the commands after it*

Let's explore some basic shell commands. Open the terminal window if you're using a Mac system or open the command prompt if you're using a windows system.

1. **$ man [command]**

   The man command provides information or help about the command. To exit the manual, press 'q'

2. **$ more [filename]**

   The more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large. You can specify different options to change the way the file is displayed.

   Eg : more -p sample.txt : The '-p' option clears the screen and then displays the text.

3. **$ pwd**

   The pwd command prints the path of the current working directory.

4. **$ ls**

   ls prints the names of the files and directories in the current directory. It also has lots of other options.

4.1. **$ ls -l** : It lists the contents of the current directory with extra details, such as user permissions and the time/date when the item was last modified.

4.2. **$ ls -t** : It sorts the files in the directory by descending time modified.

4.3. **$ ls -l | more** : In case a directory has many items we pipe the results of 'ls-l' into 'more'. The more command takes the output of the 'ls -l' as input and the final output of ls-l is displayed one screen at a time.

Note : The '|' is a pipe character. A pipe is a form of redirection that is used to send the output of one command to another command for further processing.

5. **$ cd [path]**

The cd command is used to change the current working directory to the required folder. You could use the relative path or the absolute path.

**. (a single period)** means the current directory

**.. (two periods)** means the parent directory

**~** means your home directory

Eg : **$ cd ..** would change the directory to the parent directory of the current directory

6. **$ mkdir [name]**

This creates a new directory in the present working directory

7. **$ mv [source] [destination]**

You could also use the command line commands to move or copy files. The mv command works in two ways, you could use it to move a file to a different location or to move directories to different locations.

8. **$ cp [source] [destination]**

The cp command works in a similar way to the mv command. The cp command only copies a file from its source to the destination.

9. **$ rm [file/folder name]**

This command deletes the file.

To delete a directory use the command **rm -r [path]**

10. **$ chmod [options] [permissions] [file name]**

A set of flags are associated with each file that determine who can access the file and how they can access the file. These flags are called permissions or modes. ‘Chmod’ stands for change mode; it can change permissions for a file or directory.

File permissions are usually nine characters long, they represent the settings for the three sets of permissions. The first three characters represent the permissions for the user who owns the file, the middle three characters show the permissions for members of the file group, and the last three characters show the permissions for anyone not part of the group.

Each set of permission has three characters. The characters indicate the presence or absence of the permission. If the character is a dash, then that represents that the permission is not granted. The characters ‘r’, ‘w’ or ‘x’ mean that the permission has been granted.

The letters represent :

**‘r’ ( read permission ) : the file can be opened, and its contents can be viewed.**

**‘w’ ( write permission ) : the file can be edited, modified or deleted**

**‘x’ (execute permission ) : if the file is a script or program it can be run**

While using the chmod command, we need tell who are we setting the permissions for (users (u), groups (g), others (o), or all (a)), what changes are we making ( ‘+’ : adding permission, ‘-’ : removing permissions, ‘=’: set permissions exactly as specified and remove others), and which permissions are we setting (‘r’, ‘w’, ‘x’)

Examples :

- chmod +r *.py : make all files ending in .py readable by everybody
- chmod og-wx *.py : disallow users in the same “group” and “others” to write (modify) or execute files ending in .py

**11. $ touch [filename]**

This command is used to create an empty file [filename] or if the file already exists it will update the access and modification time of [filename]

**12. $ clear**

This command is used to clear the terminal

This is just an introduction to the basics of a UNIX Terminal, there’s a lot more that Terminal can do. Learn and try it out yourself!