

Command lines essentials in UNIX

- For the purposes of working in any 'Big Data' real world situation, it is essential that you feel comfortable in the command line (Linux/Unix).
- This website will cover the very basics the command line (<http://linuxcommand.org/tlcl.php>)
- Unix is a multitasking, multiuser computer operating system. Most computers either run a Microsoft Windows operating system or some type of unix operating system(eg, Mac OS). The wonderful part of the unix operating system is that it is open source, which allows anyone to view the source code and make changes.

Common Commands

- In terms of navigation, we commonly will use:
- * ``pwd`` - prints your current location - ``print working directory``
- * ``cd`` - to ``change directory``
- * ``ls`` - will list all of the files in our current location - ``list``

- To create and remove files:
 - * ``touch`` - creates an empty file
 - * ``rm`` - to ``remove`` a file
- To create and remove directories:
 - * ``mkdir`` - to ``make directory``
 - * ``rmdir`` - to ``remove directory``
- To completely remove an entire directory we can run:
 - * ``rm -rf`` - this will ``remove``, ``recursively``, and ``force`` the removal of all items in a particular directory.

- Other common commands include:
- * ``cp`` - copy a file to new place - ``cp path\to\current\file path\to\new\file``
- * ``mv`` - move file to new place = ``mv path\to\current\file path\to\new\file``
- ****Note****: It is important that you know ``mv`` will overwrite a file if you name it the same as another file at the same path. You will obtain a warning depending on your distribution.

Some keyboard shortcuts

- <https://www.linux.com/learn/best-linux-keyboard-shortcuts>
- Note here that `^h` is the same as `ctrl-h` on a mac, and the equivalent on a Windows system (try `ctrl` or `alt`).
- Some of the commands I use most often include:
 - * `^u` - Erase input from the current location to beginning of the line.
 - * `^k` - Erases input from the current location to the end of the line.
 - * `^a` - Move to the beginning of the line.
 - * `^e` - Move to the end of the line.
 - * `^c` - Kill a running program
 - * `clear` - clear entire screen.

Some exercise

1. Open up your terminal and navigate to your root directory.
``cd`` or ``cd ~``
2. In your root directory, create a new directory called ``test_directory``. ``mkdir test_directory``
3. Navigate into ``test_directory``, and print the working directory to the screen. Is the pathname printed to the screen an absolute path or relative path?

```
```bash
 cd test_directory
 pwd ```
```

The path printed to the screen is an absolute path. You can tell because it is prefixed with a ``/``.

4. Create two files, calling them ``test_file1.txt`` and ``test_file2.txt``.

```
```bash
touch test_file1.txt test_file2.txt ```
```

- 5 Use ``ls`` to make sure that the files are in fact in the directory.

```
`ls`
```

You should see both of the files ``test_file1.txt`` and ``test_file2.txt`` present.

Some exercise

6. Copy ``test_file1.txt`` to a new file, calling it ``test_file3.txt``. How many files are there in your directory now?

```
`cp test_file1.txt test_file3.txt`
```

There should now be three files in your directory - ``test_file1.txt``, ``test_file2.txt``, and ``test_file3.txt``.

7. Rename ``test_file3.txt`` to ``test_file4.txt``. How many files are therein your directory now?

```
`mv test_file3.txt test_file4.txt`
```

There should still be only 3 files in your directory - ``test_file1.txt``, ``test_file2.txt``, and ``test_file4.txt``.

8. Within ``test_directory``, create a new directory called ``inner_directory``.

```
`mkdir inner_directory`
```

9. In one command, move all of the files from ``test_directory`` into the ``inner_directory``.

```
`mv test_file1.txt test_file2.txt test_file4.txt inner_directory`
```

10. In one command, delete ``inner_directory`` and all its contents.

```
`rm -r inner_directory`
```

11. Navigate back to your root directory, and delete ``test_directory`` using ``rmdir``. Why does ``rmdir`` work here (**Hint**: Think about what's left in ``test_directory``)?

```
```bash cd rmdir test_directory ```
```

# Resources for learning and practice python

- <https://learnpythonthehardway.org/book/>
- [Code Wars](<https://www.codewars.com/dashboard>)
- Leetcode

The more you do it, the better you will get!