

**Command line Module**

I really don’t know how long this will take so buckle up!

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
| 1 | What is command line? | 1 |
| 2 | * Machine Learning   + Applications   + Processes | 2 |
| 2-5 |
| 5-7 |
| 3 | * Data Science   + Processes   + Data Science vs ML | 8 |
| 8-9 |
| 9 |

Module on command line and Linux

**Basic Commands**

1. **alias**

$ alias pd="pwd"

The alias command allows you to create keyboard shortcuts, or aliases, for commonly used commands.

1. **cd**

$ cd Desktop/

cd takes a directory name as an argument, and switches into that directory.

$ cd jan/memory

To navigate directly to a directory, use cd with the directory’s path as an argument. Here, cd jan/memory/ command navigates directly to the **jan/memory** directory.

**cd ..**

$ cd ..

To move up one directory, use cd ... Here, cd .. navigates up from **jan/memory/** to **jan/**.

1. **cp**

$ cp ada\_lovelace.txt historical/

cp copies files or directories. Here, we copy the file **ada\_lovelace.txt** and place it in the **historical/** directory

1. **env**

$ env

The env command stands for “environment”, and returns a list of the environment variables for the current user.

**env | grep VARIABLE**

$ env | grep PATH

env | grep PATH is a command that displays the value of a single environment variable.

1. **export**

export USER="Jane Doe"

export makes the variable to be available to all child sessions initiated from the session you are in. This is a way to make the variable persist across programs.

1. **grep**

$ grep "Mount" mountains.txt

grep stands for “global regular expression print”. It searches files for lines that match a pattern and returns the results. It is case sensitive.

**grep -i**

$ grep -i "Mount" mountains.txt

grep -i enables the command to be case insensitive.

**grep -R**

$ grep -R Arctic /home/ccuser/workspace/geography

grep -R searches all files in a directory and outputs filenames and lines containing matched results. -R stands for “recursive”.

**grep -Rl**

$ grep -Rl Arctic /home/ccuser/workspace/geography

grep -Rl searches all files in a directory and outputs only filenames with matched results. -R stands for “recursive” and l stands for “files with matches”.

1. **HOME**

$ echo $HOME

The HOME variable is an environment variable that displays the path of the home directory.

**ls**

$ ls

2014 2015 hardware.txt

ls lists all files and directories in the working directory

**ls -a**

$ ls -a

. .. .preferences action drama comedy genres.xt

ls -a lists all contents in the working directory, including hidden files and directories

**ls -l**

$ ls -l

drwxr-xr-x 5 cc eng 4096 Jun 24 16:51 action

drwxr-xr-x 4 cc eng 4096 Jun 24 16:51 comedy

drwxr-xr-x 6 cc eng 4096 Jun 24 16:51 drama

-rw-r--r-- 1 cc eng 0 Jun 24 16:51 genres.txt

ls -l lists all contents of a directory in long format. [Here’s what each column means](https://www.codecademy.com/courses/learn-the-command-line/lessons/command-line-manipulation/exercises/ls-l).

**ls -t**

ls -t orders files and directories by the time they were last modified.

1. **mkdir**

$ mkdir media

mkdir takes in a directory name as an argument, and then creates a new directory in the current working directory. Here we used mkdir to create a new directory named **media/**.

1. **mv**

$ mv superman.txt superhero/

To move a file into a directory, use mv with the source file as the first argument and the destination directory as the second argument. Here we move superman.txt into superhero/.

1. **nano**

$ nano hello.txt

*nano* is a command line text editor. It works just like a desktop text editor like TextEdit or Notepad, except that it is accessible from the command line and only accepts keyboard input.

1. **PATH**

$ echo $PATH

/home/ccuser/.gem/ruby/2.0.0/bin:/usr/local/sbin:/usr/local/bin:/usr/bin:/usr/sbin:/sbin:/bin

PATH is an environment variable that stores a list of directories separated by a colon. Each directory contains scripts for the command line to execute. PATH lists which directories contain scripts.

1. **pwd**

$ pwd

/home/ccuser/workspace/blog

pwd prints the name of the working directory

1. **rm**

$ rm waterboy.txt

rm deletes files. Here we remove the file waterboy.txt from the file system.

**rm -r**

$ rm -r comedy

rm -r deletes a directory and all of its child directories.

1. **sed**

$ sed 's/snow/rain/' forests.txt

sed stands for “stream editor”. It accepts standard input and modifies it based on an *expression*, before displaying it as output data.

In the expression 's/snow/rain/':

* s: stands for “substitution”.
* snow: the search string, the text to find.
* rain: the replacement string, the text to add in place.

1. **sort**

$ sort lakes.txt

sort takes a filename or standard input and orders each line alphabetically, printing it to standard output.

1. **source**

source ~/.bash\_profile

source activates the changes in **~/.bash\_profile** for the current session. Instead of closing the terminal and needing to start a new session, source makes the changes available right away in the session we are in.

1. **touch**

$ touch data.txt

touch creates a new file inside the working directory. It takes in a file name as an argument, and then creates a new empty file in the current working directory. Here we used touch to create a new file named keyboard.txt inside the 2014/dec/ directory.

If the file exists, touch is used to update the modification time of the file

1. **uniq**

$ uniq lakes.txt

uniq, short for “unique”, takes a filename or standard input and prints out every line, removing any exact duplicates.