# NGS 2017 Workshop Cheatsheet

# 1. Basic Linux Commands

## 1.1. Browse the directory structure

| | |
|---|---|
| `pwd` | tell you where you are |
| `ls` | list the content of the current directory |
| `ls <directory>` | list the content of a directory |
| `cd <directory>` | go to the specified directory |
| `cd ~ (or cd)` | go to your home directory |
| `cd ..` | go to the parent directory |
| `tree <directory>` | list the content of a directory in a tree-like format |
| `mkdir <directory>` | create the specified directory |

## 1.2. View the content of a file

| | |
|---|---|
| `less, more` | view text with paging |
| `head` | print first lines of a file |
| `tail` | print last lines of a file |
| `cat` | print the content of a file to the screen |
| `zcat` | print the content of a `gzip` compressed file to the screen |

## 1.3. File manipulations

| | |
|---|---|
| `rm <file>` | remove `file` |
| `cp <file1> <file2>` | copy `file1` to `file2` |
| `mv <file1> <file2>` | rename `file1` to `file2` |

## 1.4. Some other useful commands

| | |
|---|---|
| `find <folder>/ -type f` | recursively find all files in a specific folder |
| `find . -name '<pattern>'` | recursively find anything whose name contains <pattern> in the current folder (Single quotes must be used in order to avoid wildcard expansion by the shell) |
| `grep <pattern>` | show lines of text containing a given pattern |

| | |
|---|---|
| `grep -v <pattern>` | show lines of text not containing a given pattern |
| `sort` | sort lines of text files |
| `wc` | count words, lines and characters |
| `> (output redirection)` | allow to redirect the output to a file |
| `| (pipe)` | allow to send the output from one program to another |
| `cut` | extract selected portion of each line from one or more files |
| `echo` | input a line of text and display it on standard output |

## 1.5. AWK programming

**AWK** - UNIX shell programming language. A fast and stable tool for processing text files.

| | |
|---|---|
| `awk '/www/ { print $0 }' <file>` | search for the pattern www in each line of the file |
| `awk '$3=="www"' <file>` | search for the exact match of www in the third column of the file |
| `awk 'length($0) > 80' <file>` | print every line in the file that is longer than 80 characters |
| `awk 'NR % 2 == 0' <file>` | print even-numbered lines of the file |

### 1.5.1. Some built-in variables

| | |
|---|---|
| `NR` | Number of records |
| `NF` | Number of fields |
| `FS` | Field separator character |
| `OFS` | Output field separator character |

See    www.grymoire.com/Unix/Awk.html    and    www.tutorialspoint.com/awk/ awk_basic_examples.htm for more information

# 2. Writing and editing files

## 2.1. GNU nano

GNU nano is a text editor for Unix-like operating systems using a command line interface. The following sections describe the commands used for opening, editing and saving files using nano.

### 2.1.1. File Control

| | |
|---|---|
| `nano main.nf` | Open or create the file `main.nf` |
| `kbd:[Ctr+o]` `kbd:[Enter]` | Save changes |
| `kbd:[Ctrl+x]` | Quit |

### 2.1.2. Navigating through file contents

| | |
|---|---|
| `kbd:[Ctrl+a]` | Move to the beginning of the current line |
| `kbd:[Ctrl+e]` | Move to the end of the current line |
| `kbd:[Ctrl+v]` | Move down one page |
| `kbd:[Ctrl+y]` | Move up one page |
| `kbd:[Ctrl+w]` `kbd:[Ctrl+y]` | Go to the beginning of the file |
| `kbd:[Ctrl+w]` `kbd:[Ctrl+v]` | Go to the end of the file |

### 2.1.3. Copy and Paste

| | |
|---|---|
| `kbd:[Ctrl+c]` `(Mac kbd:[Cmd+c])` | Copy the current selection to the clipboard |
| `kbd:[Ctrl+v]` `(Mac kbd:[Cmd+c])` | Paste the contents from the clipboard at the current cursor position |
| `kbd:[Ctrl+k]` | Cut from the current cursor position to the end of the current line |
| `kbd:[Ctrl+d]` | Delete the character at the current cursor position |

### 2.1.4. Search and Replace

| | |
|---|---|
| `kbd:[Ctrl+w]` | Search for a target string |