bash cheat sheet

# The most useful command

## man <command>

Manual

Opens the manual page for the specified command

Similar to help(function) in Python

Press q to exit

# Navigating directory structures

## pwd

Print working directory

Tells you which folder your terminal is currently in

## cd <path>

Change directory

Changes the folder your terminal is currently in to the one specified by <path>

### cd ..

Move to the parent directory

### cd ~

Move to the home directory

### cd -

Move to the last directory you were in

## ls <path>

List

Lists the files in the specified directory (<path>)

If <path> is omitted, then lists the files in the current directory

### ls .

List files in the current directory

### ls ..

List files in the parent directory

### ls ~

List files in the home directory

### ls -1

Only one file per line

### ls -a

Do not omit hidden files (those that start with a '.')

## tree

Tree

Similar to ls

Prints the directory structure as a tree

# Manipulating files and folders

## mkdir <new dir name>

Make directory

Creates a new directory with the specified name/path

## mv <target> <new destination>

Move

Move the target (file or directory) to its new destination

Can be used to simply rename a file

## cp <target> <new destination>

Copy

Copy the target file to a new destination

### cp -r <target dir> <new destination>

The -r (recursive) flag is required to copy directories

## ln <target> <new destination>

Link

Create a “hard link” at the new destination for the target file

Cannot be used on directories

### ln -s <target> <new destination>

The -s (symbolic) flag creates a symbolic link

Can be used on directories

## rm <target>

Remove

Permanently deletes the target file

This action cannot be undone

### rm -r <target dir>

The -r (recursive) flag is required to remove directories

# Other useful commands

## clear

Clear

Clears the terminal so that it looks fresh

Equivalent to ^+L

## history

history

Prints a list of previously used commands

## echo "<text>"

echo

Prints the specified text

Double-quotes allow variables to be replaced with their values

Single-quotes will be interpreted as raw literal strings

## cat <file>

Concatenate

Prints the contents of <file>

## touch <file>

Touch

Changes the modify date of <file> to the moment the command was executed

Creates empty <file> if it does not already exist

## wc <file>

Word count

Prints the number of lines, words, bytes, and file name (by default) of <file>

### wc -l

Prints the number of lines

### wc -w

Prints the number of words

### wc -c

Prints the number of bytes

### wc -m

Prints the number of characters

## less <file>

less

Interactive text-viewer

User can only view text; no modification allowed

Press q to exit

### Commands **within** the viewer

#### -S

(uppercase)

Doesn’t wrap long lines

Can also be used as a flag when opening less

#### -s

(lowercase)

Wraps long lines

Can also be used as a flag when opening less

#### -N

Display line numbers

Can also be used as a flag when opening less

#### -n

Hide line numbers

Can also be used as a flag when opening less

#### /

Search within a file

Regular expressions are allowed

Press return to begin search

Press n to go to the next match

Press N to go to the previous match

Press p to go to the first match

#### g

Go to the beginning of the file

#### G

Go to the end of the file

#### q

Exit

## grep "<pattern" <file>

General regular expression (file pattern searcher)

Prints all the lines where something matches a given pattern in the file

### grep -E

Accepts extended regular expressions

### grep -P

Accepts Perl regular expressions

### grep -c

Prints the number of matching lines instead

### grep -o

Prints only the matching content

## sed -E "s/<find>/<replace>/g" <file>

Stream editor

In this class, we will always use the -E (extended regular expressions) flag

In this class, we will always use the substitute (s) mode

In this class, we will always perform global (g) substitutions.

# Redirection

## <cmd> | <cmd>

Pipe

Accepts the output of the command on the left and uses it as input to the command on the right

## <cmd> > <file>

Greater than

Accepts the output of the command on the left and writes it to the file on the right. Creates a new file if the specified one does not exist.

Overwrites anything in the file if it already exists.

Similar to open("filename", 'w') in python

## <cmd> >> <file>

Double greater than

Accepts the output of the command on the left and appends it to the file on the right. Creates a new file if the specified one does not exist.

Adds the new content to the bottom of the file if it already exists.

Similar to open("filename", 'a') in python

## <file> < <cmd>

Less than

Accepts the output of the command on the right and writes it to the file on the left. Creates a new file if the specified one does not exist.

Overwrites anything in the file if it already exists.

Similar to open("filename", 'w') in python

# Keyboard shortcuts

## ^+A

Move the cursor to the beginning of the terminal prompt (line)

Does not work when using screen

## ^+E

Move the cursor to the end of the terminal prompt (line)

## ^+C

Kill the currently running process

## ^+L

Clear the screen

Equivalent to clear

## ^+Z

Move the current process to the background

Process can be retrieved with the command fg (foreground)

## ^+R

Reverse history search

Allows the user to search previously executed commands

## ^+D

Ends the current session

Equivalent to exit

## ^+W

Word backspace.

Deletes the word to the left of the cursor