

BASH Cheat Sheet

Adopted from <https://phoenixnap.com/kb/bash-commands> and <https://www.educative.io/blog/bash-shell-command-cheat-sheet>

ls

lists all files and directories in the current directory; can provide information on file sizes, permissions, timestamps, etc.

```
$ ls [options] [file|dir]
```

Example:

```
$ ls -lrth
```

lists files and directories in reverse chronological order and displays information on file size.

```
$ ls -a
```

lists all files, including hidden ones

cd

changes the current directory to the specified directory.

```
cd [directory]
```

Example:

```
$ cd ..
```

switches to a directory above

```
$ cd
```

equivalent to `cd ~`, switches to home directory

pwd

prints the path of the current working directory; useful to get the path to the directory for further use in a bash script.

```
$ pwd [options]
```

rm

removes files or directories.

Important: `rm` deletes files without prompting for confirmation! deleted files are not recoverable. best used with `rm -i`.

```
$ rm [options] [file|dir path]
```

Example:

```
$ rm -r
```

deletes a directory and everything inside it; removes recursively directories and their contents, including subdirectories and files within them.

mkdir

creates a new, empty directory.

```
$ mkdir [options] [dir]
```

cp

copy files and directories between locations.

```
$ cp [options] [source] [destination]
```

Example:

```
$ cp -r timetables/ timetables-backup
```

copies the specified directory and its entire contents, including subdirectories and their contents into a new location.

mv

moves directories and files; can also be used to rename files or directories.

```
$ mv [options] [source] [destination]
```

find

searches for files or a specific string of characters in a directory hierarchy.

```
$ find [location] [expression] [options]
```

- `[location]` is the directory where the search will begin, if not specified, the search starts from the current directory.
- `[expression]` is used to combine multiple search criteria using logical operators (`-and`, `-or`, `-not`).
- `[options]` can be: `-name XXX` search by name; `-type f/d/l`, where `f` is files, `d` is directories, `l` is links; ...

grep

searches inside files for lines that match a given regular expression.

```
$ grep [options] [search pattern] [file]
```

- `[search pattern]` is the text or regular expression you're searching for

- `[file]` are the files in which you want to search for the pattern.
 - `[options]` can be `-i` to ignore case-sensitivity; `-r` for recursive, ...
-

head

previewa the beginning section of a text file or input stream.

```
$ head [options] [file...]
```

Example:

```
$ head -100 script.sh
```

shows the first 100 lines of the file.

tail

displays the last [default = 10] lines of a file,

```
$ tail [options] [file...]
```

```
$ tail -100 script.sh
```

shows last 100 lines in the file

more

displays one screenfull of file contents at a time. use space bar to more to next screenfull

```
$ more [file]
```

history

lists recently executed commands within a terminal session in chronological order.

```
history [options]
```

alternatively, `arrow up` allows to navigate the previous commands within the command line.

echo

displays text or variables in the terminal as the standard output; often used in a script.

```
echo [options] [string]
```

| pipe

allows users to connect the standard output of one command to the standard input of another, thus enabling the flow of data between commands.

```
$ [command1] | [command 2]
```

> >> < redirect operator

a mechanism that allows users to control the input and output of commands.

- `>` redirects the standard output of a command to a file, overwriting its content if the file already exists.
- `>>` appends the standard output to the specified file.
- `<` redirects the standard input of a command from a file.

```
$ date > time.txt
```

writes current time and date into a file

scp

secure copy. Like a copy command, allows copying a file or directory between two machines

```
$ scp [options] [file] [destination]
```

Example:

```
$ scp -r mycodes/ s123456@eddie.ecdf.ed.ac.uk:~/project1/.
```

copies directory `mycodes` from the local machine to remote machine, and places into a directory `project1` located in a home directory.

ssh

secure shell, connects to a remote machine.

```
$ ssh [options] [user]@[server]
```

Example:

```
$ ssh -Y s123456@eddie.ecdf.ed.ac.uk
```

logs in a user `s123456` to Eddie HPC, allowing for visualisation of graphics on a remote machine.

qsub

submits a gridengine job on a supercomputer.

Example:

```
$ qsub myjob.sh
```

qstat

checks for the status of submitted gridengine jobs to a supercomputer for a given user.

Example:

```
$ qstat -u s123456
```