BASH CHEAT-SHEET

SHORTCUTS

```
CTRL+A  # move to beginning of line
CTRL+C  # halts the current command
CTRL+E  # moves to end of line
CTRL+K  # deletes (kill) forward to end of line
CTRL+L  # clears screen and redisplay the line
CTRL+N  # next line in command history
CTRL+P  # previous line in command history
exit    # logs out of current session
```

BASH BASICS

```
env                 # all environment variables
echo $SHELL         # the shell you're using
echo $BASH_VERSION  # bash version
bash                # use bash
whereis bash        # where bash is on your system
which bash          # which program is executed as 'bash' (default: /bin/bash)
clear               # clears content on window
```

DIRECTORY COMMANDS

```
mkdir <dirname>  # makes a new directory
cd               # changes to home
cd <dirname>     # changes directory
pwd              # current directory
```

CONDITIONAL STATEMENTS

```
#IF
if condition
then
  statements
[elif condition
  then statements...]
[else
  statements]
fi
```

```
#WHILE
while condition; do
  statements
done

until condition; do
  statements
done
```

```
#FOR LOOP
for x in {1..10}
do
  statements
Done

for name [in list]
do
  statements that can use $name
done

for (( initialisation ; ending condition ; update ))
do
  statements...
done
```

FILE COMMANDS

```
ls              # lists your files in current directory
ls <dir>        # lists of files in a specific directory
ls -l           # lists your files in 'long format' (e.g. file size, modified date)
ls -a           # lists all files, including hidden files

ln -s <filename> <link>       # creates symbolic link to file
cat <filename>                # prints file raw content (will not be interpreted)
more <filename>               # shows the first part of a file (move with space, q to quit)
head <filename>               # outputs the first lines of file (default: 10 lines)
tail <filename>               # outputs the last lines of file (default: 10 lines)

mv <filename1> <destination>  # moves a file to destination
cp <filename1> <destination>  # copies a file
rm <filename>                 # removes a file
diff <filename1> <filename2>  # compares files, and shows where they differ
wc <filename>                 # number of lines, words and characters in a file

chmod -options <filename>     # change the read, write, and execute permissions on file
gzip <filename>               # compresses files using gzip algorithm
gunzip <filename>             # uncompresses files compressed by gzip
gzcat <filename>              # look at gzipped file without gunzip it

grep <pattern> <filenames>    # looks for the string in the files
grep -r <pattern> <dir>       # search recursively for pattern in directory
```

Find more: https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh

BASH CHEAT-SHEET

FLOW CONTROLS

```
statement1 && statement2  # and operator
statement1 || statement2  # or operator
-a                        # and operator inside a test conditional expression
-o                        # or operator inside a test conditional expression
```

```
# STRINGS

str1 = str2         # str1 matches str2
str1 != str2        # str1 does not match str2
str1 < str2         # str1 is less than str2 (alphabetically)
str1 > str2         # str1 is greater than str2 (alphabetically)
-n str1             # str1 is not null (has length greater than 0)
-z str1             # str1 is null (has length 0)
```

```
# FILES

-a file             # file exists
-d file             # file exists and is a directory
-e file             # file exists; same -a
-f file             # file exists and is a regular file
-r file             # you have read permission
-s file             # file exists and is not empty
-N file             # file was modified since it was last read
-O file             # you own file
-G file             # file's group ID matches yours
file1 -nt file2     # file1 is newer than file2
file1 -ot file2     # file1 is older than file2
```

```
# NUMBERS

-lt                 # less than
-le                 # less than or equal
-eq                 # equal
-ge                 # greater than or equal
-gt                 # greater than
-ne                 # not equal
```

INPUT/OUTPUT REDIRECTORS

```
cmd1|cmd2   # pipe; takes standard output of cmd1 as standard input to cmd2

< file      # takes standard input from file
> file      # directs standard output to file
>> file     # directs standard output to file; append to existed file
<> file     # uses file as both standard input and standard output

n<>file     # uses file as both input and output for file descriptor n
n>file      # directs file descriptor n to file
n<file      # takes file descriptor n from file
n>>file     # directs file n to file; append to already existed file

n>&         # duplicates standard output to file descriptor n
n<&         # duplicates standard input from file descriptor n
n>&m        # file n is made to be a copy of the output file descriptor
n<&m        # file n is made to be a copy of the input file descriptor
&>file      # directs standard output and standard error to file

<&-         # closes the standard input
>&-         # closes the standard output
n>&-        # closes the ouput from file descriptor n
n<&-        # closes the input from file descripor n
```

VARIABLES

```
varname=value                # defines a variable
echo $varname                # display variable's value
export VARNAME=value         # defines an environment variable

${#varname}                  # returns the length of the value of variable
$(UNIX command)              # command substitution
```