

## **Bash Cheat Sheet**

---

## 1 Navigating Directories

```
pwd                # Print current directory path
ls                 # List directories
ls -l              # List directories in long form
ls -l -h|--human-readable # List directories in long form with human readable sizes
ls -ltr            # List directories by modification time, new files at bottom
stat foo.txt       # List size, created and modified timestamps for a file
cd mydir           # Go to directory mydir
cd ..              # Go one directory up
cd ../..          # Go two directories up
cd ~               # Go to home directory
```

## 2 Creating, Move, Delete Directories

```
mkdir foo          # Create a directory
mktemp             # Create a temporary file
cp -R foo bar     # Copy directory
mv foo bar        # Move directory
rmdir foo         # Delete non-empty directory
rm -r foo         # Delete directory including contents
```

## 3 Creating, Moving, Deleting Files

```
touch foo.txt     # Create file or update existing files modified timestamp
touch test{1..3}  # Create test1, test2 and test3 files
cp foo.txt bar.txt # Copy file
mv foo.txt bar.txt # Move file
rm foo.txt        # Delete file
rm -f foo.txt     # Delete file, ignore nonexistent files and never prompt
```

## 4 Reading and Editing Files

```
cat foo.txt       # Print all contents
head foo.txt      # Print top 10 lines of file
head -n30 foo.txt # print top 30 lines
tail foo.txt      # Print bottom 10 lines of file
gedit foo.txt     # Open file in the default editor
nano foo.txt      # Open in editor Nano
```

## 5 Standard Output, Standard Error and Standard Input

```
echo "foo" > bar.txt      # Overwrite file with content
echo "foo" >> bar.txt     # Append to file with content

ls exists 1> stdout.txt   # Redirect the standard output to a file
ls noexist 2> stderr.txt # Redirect the standard error output to a file
ls 2>&1 out.txt          # Redirect standard output and error to a file
ls &> /dev/null          # Discard both standard output and error
grep 'foo' bar.txt | head # Pipe, redirect stdout of first command to stdin of second command
```

---

## 6 Finding Files

```
find /path -name foo.txt          # Find a file
find /path -iname foo.txt        # Find a file with case insensitive search
find /path -name "*.txt"         # Find all text files
find /path -name foo.txt -delete # Find a file and delete it
find /path -type f -name foo.txt # Find a file
find /path -type d -name foo     # Find a directory
```

## 7 Find Patterns in Files

```
grep 'foo' bar.txt                # Search for 'foo' in file 'bar.txt'
grep 'foo' -r bar                 # Search for 'foo' in directory 'bar'
grep 'foo' -r -l bar              # Show only files that match
grep 'foo' -r -L bar              # Show only files that don't match
grep -i 'Foo' bar.txt             # Case insensitive search
grep -v 'foo' bar.txt             # invert selection, find lines without 'foo'
grep -c 'foo' bar.txt             # Count number of lines with 'foo'
grep -E 'foo|baz' bar.txt         # Search foo OR baz, same as egrep
```

## 8 Replace in Files

```
sed 's/fox/bear/g' foo.txt        # Replace fox with bear in foo.txt and output to console
sed 's/red/fox/g;s/blue/bear/g' foo.txt # Replace red with fox and blue with bear in foo.txt
sed 's/fox/bear/g' foo.txt > bar.txt # Replace fox with bear in foo.txt and save in bar.txt
sed -i 's/fox/bear/g' foo.txt     # Replace fox with bear and overwrite foo.txt
```

## 9 Command History

```
history          # History of previous commands
!!              # Run the last command

touch foo.sh
chmod +x !$     # !$ is the last argument of the last command i.e. foo.sh
```

## 10 Print Text or Numbers

```
echo "Hello World"          # Write text to standard out
echo -e "Hello\tWorld\n"   # Write with Tabs \t or Newlines \n
printf "%-20s %8.3f %8d\n" Name 6.1 8 # Write formatted output for text, floats, or integers
```