

## BASH

## Navigating

```
cd name_of_directory
cd ..      # Go up one
cd ~       # Go to home
pwd        # Where am I?
```

## Listing files

```
ls          # List files
ls -a       # See hidden
ls -l       # See more info
ls -R       # Recursive
```

## Moving and renaming

```
mv file.txt new_name.txt
mv file.txt ../new/place/
```

## Copying

```
cp file.txt file_backup.txt
cp -r directory/ backup/
```

## Deleting

```
rm file.txt
rmdir empty_directory/
rm -r full_directory/
```

## Creating

```
mkdir my_directory
touch empty_file.py
```

## Reading data from file

```
cat filename.txt
cat file1 file2 file3
```

## Piping data into file

```
ls -R > all_files.txt
cat a.html b.html > c.html
```

## Running file as bash script

```
# Save commands to script.sh
bash script.sh
```

## BASH TRICKS

**Auto complete** Start typing then hit <Tab>. Hit twice for options.

**Re-run previous command** <Up> then <Enter>

**Search previous history** <Ctrl+R> then start typing, <Ctrl+R> to cycle back, <Enter> to run.

## Advanced piping

```
# Search process for "chrome"
ps -e | grep chrome
# Search dir for ".py" files
find . | grep .py$
```

## GIT

## Starting repo

```
git init
```

## Adding changes and committing

```
git add -A
git commit -m "Fixed :)"
```

## Finding out status

```
git status
git log
```

## Learning about past

```
git log          # Q to quit
git show f85bfcf
git diff f85bfcf master
git checkout f85bfcf
```

## Branch workflow

```
git branch my-stuff
git checkout my-stuff
# Do some work...
git add -A
git commit -m "did stuff"
git checkout master
git merge my-stuff
```

## Interacting with GitHub

```
git pull # get updates
# Do some work...
git add -A
git commit -m "it works!"
git push # share updates
```

## KEY TERMS

**Repository** A repo is a “git enabled” directory, stores “undo-history” (commit log) and enables collaboration (via *push* and *pull*).

**Commit** The state of a *repo*, as if frozen-in-time, uniquely designated by a *hash* (long series of letters and numbers).

## PYTHON

You can test Python with the interactive prompt (aka “REPL”):

## Interactive prompt

```
python3
>>> print("Hello world")
Hello world
>>> 5 + 5
10
>>> exit()
```

## Running code from file

```
# Save code as mycode.py
python3 mycode.py
```

## Hello world program

```
print("Hello world")
menu = "Spam spam spam"
print(menu)
```

## PYTHON I/O

## Reading text from file

```
a = open("file.txt").read()
print("file1.txt has: ", a)
```

## Writing to file

```
a = "Some text for o.txt"
open("o.txt", "w+").write(a)
```

## Appending to file

```
b = "Repeat this text x3"
open("o.txt", "a+").write(b)
open("o.txt", "a+").write(b)
open("o.txt", "a+").write(b)
```

## Combining files

```
a = open("f1.txt").read()
b = open("f2.txt").read()
c = a + b
open("f3.txt", "w+").write(c)
```

## KEY TERMS

**Variable** A named “bucket” that holds data. Can be updated with *assignment* =

**String** A data type that represents text (the term comes from “a string of characters”)

**Operator** A symbol that can perform arithmetic, modifying and combines data in variables, such as + and -.