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
```

Redirecting 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.

Previous command history <Up>

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

Wildcards

```
rm *.jpg # Delete jpg files
```

Advanced piping

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

Git

Starting GitHub repo Click button on GitHub. Check box to create with "README". Then clone locally (`U` and `R` should be username and repo name):

```
git clone http://github.com/U/R.git
```

Starting (local-only) 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

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

Writing to file

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

Appending to file

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

Combining files

```
start = open("f1.txt").read()
end = open("f2.txt").read()
full = start + end
open("f3.txt", "w+").write(full)
```

Key Terms

Variable A named "bucket" that holds data. Can be updated with *assignment operator* (equals sign `=`)

String Text data, term comes from "string of characters"

Operator A symbol that can perform arithmetic, modifying and combines data in variables, e.g. `+/-`.