Shell, Emacs, & Python (oh my!) Cheat Sheet

Lawrence David

09.07.06

1 Shell

To run these commands, only type in the part after the dollar sign. Replace both arguments *and* square brackets (don't leave square brackets in).

$ cp [oldfile] [newfile]	copies
$ mv [oldfile] [newfile]	moves
$ rm [oldfile]	deletes (no undo!)
$ mkdir [newdir]	new directory
$ rm -r [olddir]	remove directory
$ cd [destination]	change directory
$ cd ..	up one directory
$ ls	list contents of directory
$ cat [file]	print every line of file
$ more [file]	print one screen of file
$ emacs [file]	edit file
$ python [file]	execute python file

2 Emacs

Below, 'C' is short for the `control` key and 'M' denotes the `meta` key. On most operating systems, the meta key defaults to the `Esc` button; however, you can usually futz with your terminal application to map `meta` to something more ergonomically reasonable, like the `alt` button.

To execute one of these commands (e.g. C-d), hold down `control`, press d with a free finger, and then release `control`. To execute something like C-x,C-s, hold down `control`, press x, then press s, and finally release `control`.

C-d	delete one character
M-d	delete one word
C-k	delete one line
M-k	delete one paragraph
C-_	undo
C-x,C-c	save and quit
C-x,C-w	save as
C-x,C-s	save
C-s	find as you type
M-%	find and replace
C-space	set mark
M-w	copy from mark to current position
C-w	cut from mark to current position
C-y	paste
C-a	jump to beginning of line
C-e	jump to end of line
M-¡	jump to beginning of document
M-¿	jump to end of document
C-v	page down
M-v	page up

3 Python

3.1 Comments

Comment out lines with the pound-sign: #.

3.2 Lists

List is python-speak for array or ordered collection of items.

L = []	initiate a list
L = [expr1,expr2,...]	list creation
L.append(x)	add an item x
L.remove(x)	remove first occurence of item x
L[i]	return i'th item of list
L[0]	return first item of list
L[-1]	return last item of list
L[0:2]	return first 3 items of list
L.extend(M)	merge list M into L
L.reverse()	flip ordering of list
L.sort()	sort items of L
len(L)	return length of list

3.3 Dictionaries

Dictionary is python-speak for hash table or data structure where *values* are stored according to *keys*

D =	initiate a dict
D[k] = i	store item i in dict with key k
D = k1:v1, k2:v2, ...	another way to populate dict
D.has_key(k)	return true if k is a key in D
D.keys()	return a list of keys
D.values()	return a list of values

3.4 Logic

a == b	return true if a equals b
a > b	return true if a greater than b
a < b	return true if a greater than b
a and b	return true if both a and b are true
a or b	return true if either a or b are true
not a	return true if a is false

3.5 Loops

Be sure to indent all lines of code to be executed in a loop! Note that you don't need any brackets or end statements.

```
for i in L:
 print i
for i in range(0,len(L)):
 print L[i]
```

```
while i == True:
 print i
```

3.6 Conditionals

Like loops, code executed within conditions must be indented!

```
if i == 0:
 print "i equals 0"
elif i == 1:
 print "i equals 1"
else:
 print "who knows what i equals"
```

break	will terminate the loop encompassing the conditional
continue	will skip encompassing loop to next iteration

3.7 Reading and Writing to Files

f = open('filename','r')	open file for reading
f = open('filename','w')	open file for writing
f.close()	close a file
f.read()	return the file's entire contents in a string
f.readlines()	return a list containing each line of the file
f.write(s)	write string s to file f

3.8 Math

For commands that are preceded by 'math', you will need to import the math module.

import math	imports math module
a*b	get product of a and b
a/b	divide a by b
a+b	sum a and b
a-b	difference of a and b
a += b	same as a = a + b
math.log(n)	take log of n
math.exp(n)	e raised to n

3.9 Defining Functions

Again, you'll need to indent all the function's code.

```
def myfunction(arg1,arg2):
 myprod = arg1*arg2
 return myprod
```

3.10 Modules

Note that imported modules need to live in Python's path. If that doesn't make any sense to you, it means that modules you write or download from online should be placed in the same directory as your code, to prevent import errors.

import m	import module m
m.myfun()	call function 'my fun' in m

3.11 Miscellany

print string	display string on terminal output
import sys; sys.argv[1]	first command-line argument