LAB 2

Terminal / Gedit

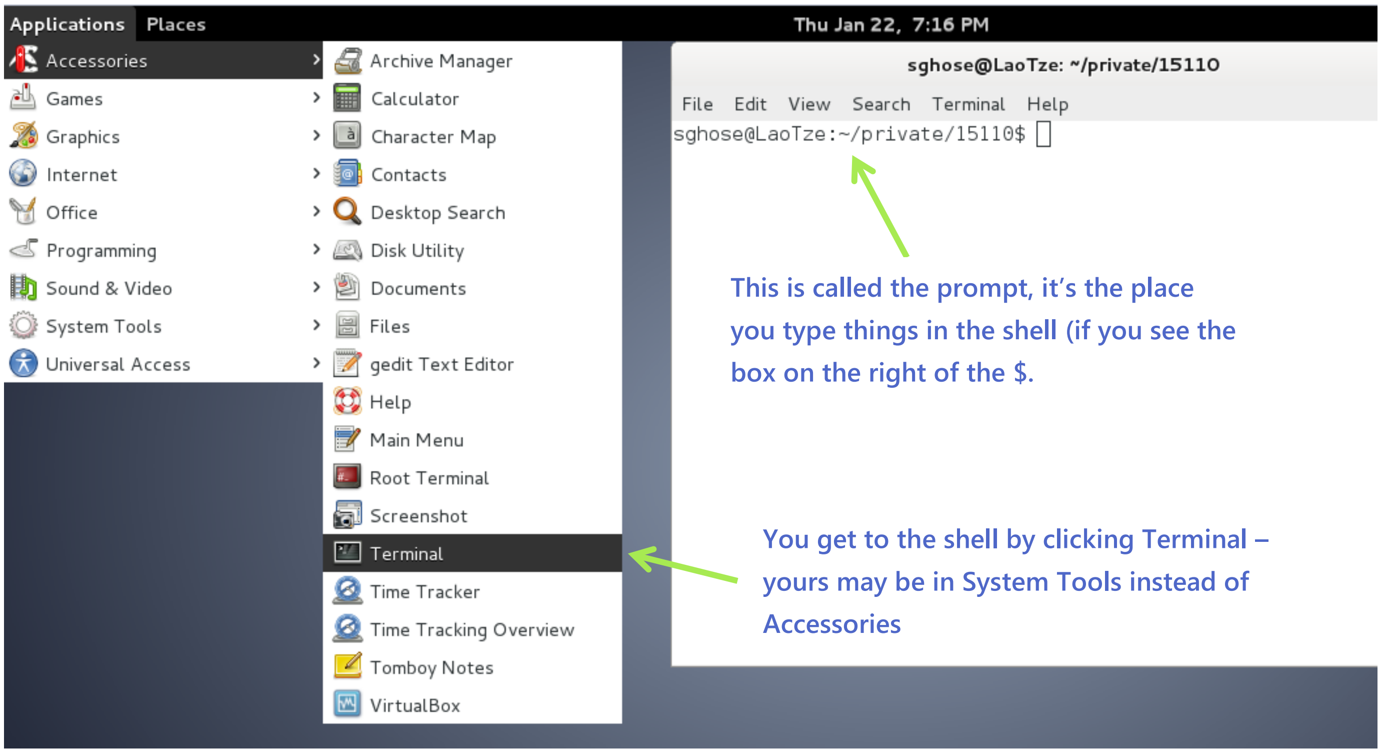
Python

Ronnie Ghose

Shell Commands

Command	**‘Acronym’**	**What does it do?**
ls	List	Lists the files and directories/folders in your current folder
mkdir <directory name>	Make directory	Make a folder in your current directory
pwd	Print working directory	Print where you are!
cd	Change Directory	Change directories – Like clicking a folder Finder (mac) or Explorer (windows)

So what's a shell?



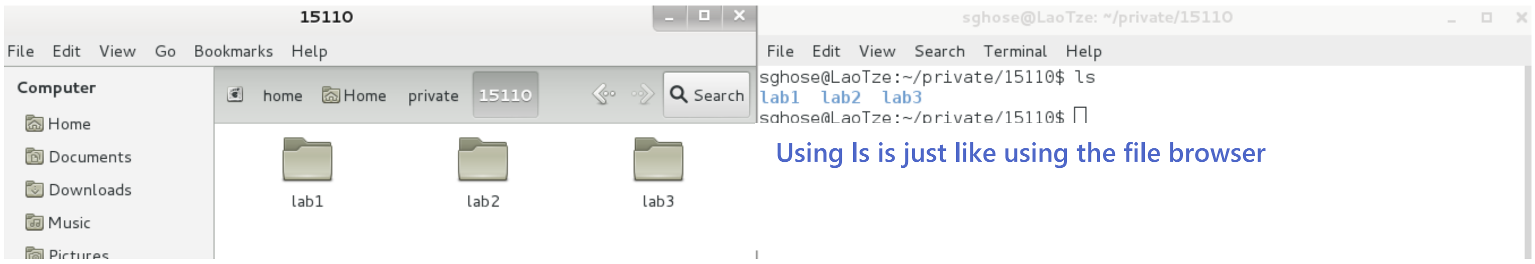
```
sghose@LaoTze:~/private/15110$
```

This is called the prompt, it's the place you type things in the shell (if you see the box on the right of the $.

You get to the shell by clicking Terminal – yours may be in System Tools instead of Accessories

Shells are programs for running other programs. We're going to use them to run **gedit** and **python3** in this class, but you can run any program you have installed on your computer through it!

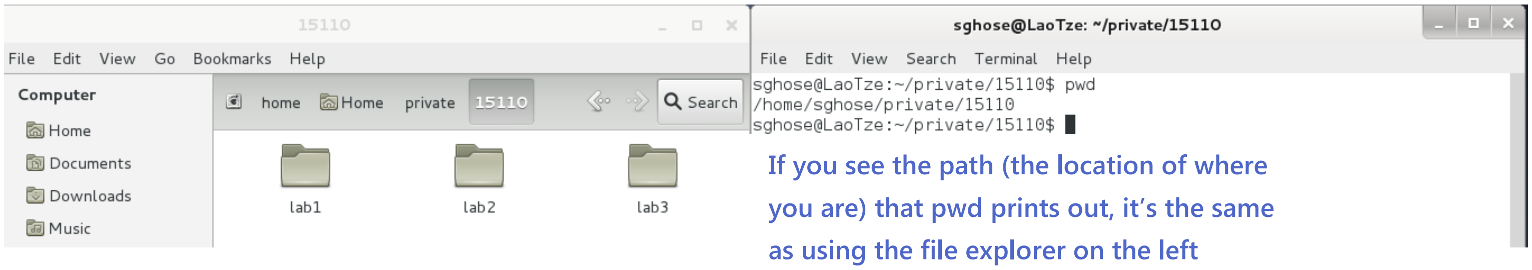
LS


15110
File Edit View Go Bookmarks Help
Computer
Home
Documents
Downloads
Music
Pictures
home Home private 15110 Search
lab1
lab2
lab3
sghose@LaoTze: ~/private/15110
File Edit View Search Terminal Help

```
sghose@LaoTze:~/private/15110$ ls
lab1  lab2  lab3
sghose@LaoTze:~/private/15110$ 
```

Using ls is just like using the file browser

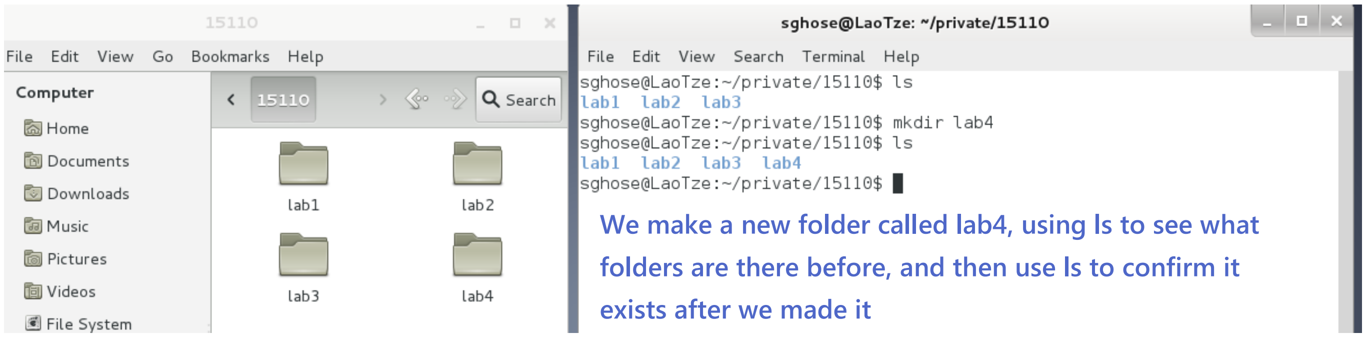
PWD


15110
File Edit View Go Bookmarks Help
Computer
Home
Documents
Downloads
Music
home Home private 15110 Search
lab1
lab2
lab3
sghose@LaoTze: ~/private/15110
File Edit View Search Terminal Help

```
sghose@LaoTze:~/private/15110$ pwd
/home/sghose/private/15110
sghose@LaoTze:~/private/15110$ 
```

If you see the path (the location of where you are) that pwd prints out, it's the same as using the file explorer on the left

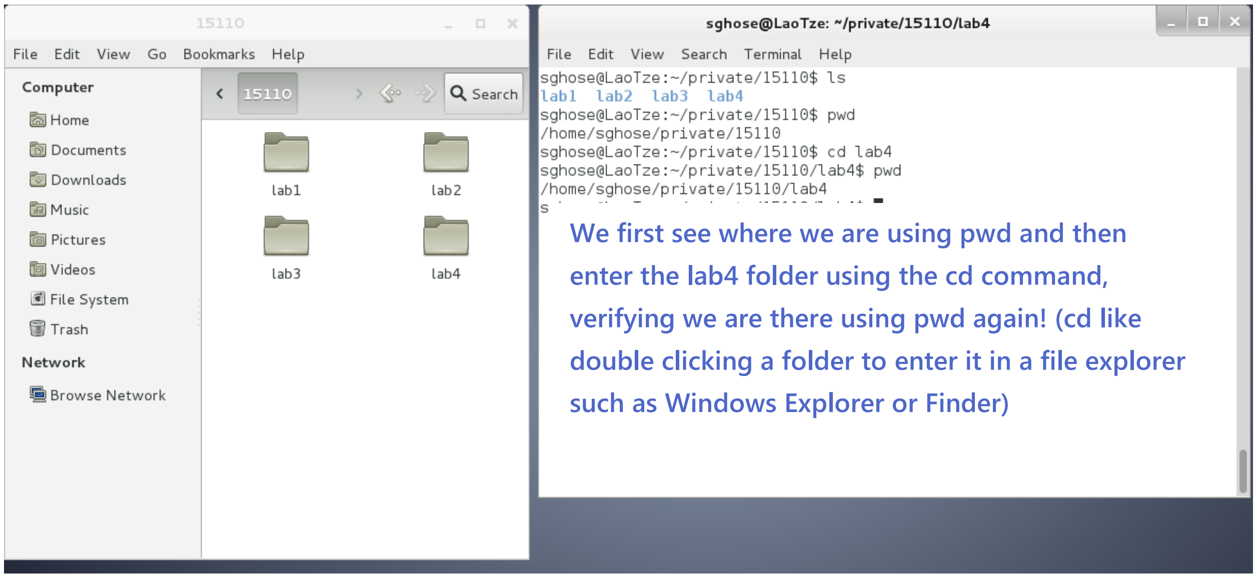
MKDIR


15110
File Edit View Go Bookmarks Help
Computer
Home
Documents
Downloads
Music
Pictures
Videos
File System
15110 Search
lab1
lab2
lab3
lab4
sghose@LaoTze: ~/private/15110
File Edit View Search Terminal Help

```
sghose@LaoTze:~/private/15110$ ls
lab1  lab2  lab3
sghose@LaoTze:~/private/15110$ mkdir lab4
sghose@LaoTze:~/private/15110$ ls
lab1  lab2  lab3  lab4
sghose@LaoTze:~/private/15110$ 
```

We make a new folder called lab4, using ls to see what folders are there before, and then use ls to confirm it exists after we made it

CD



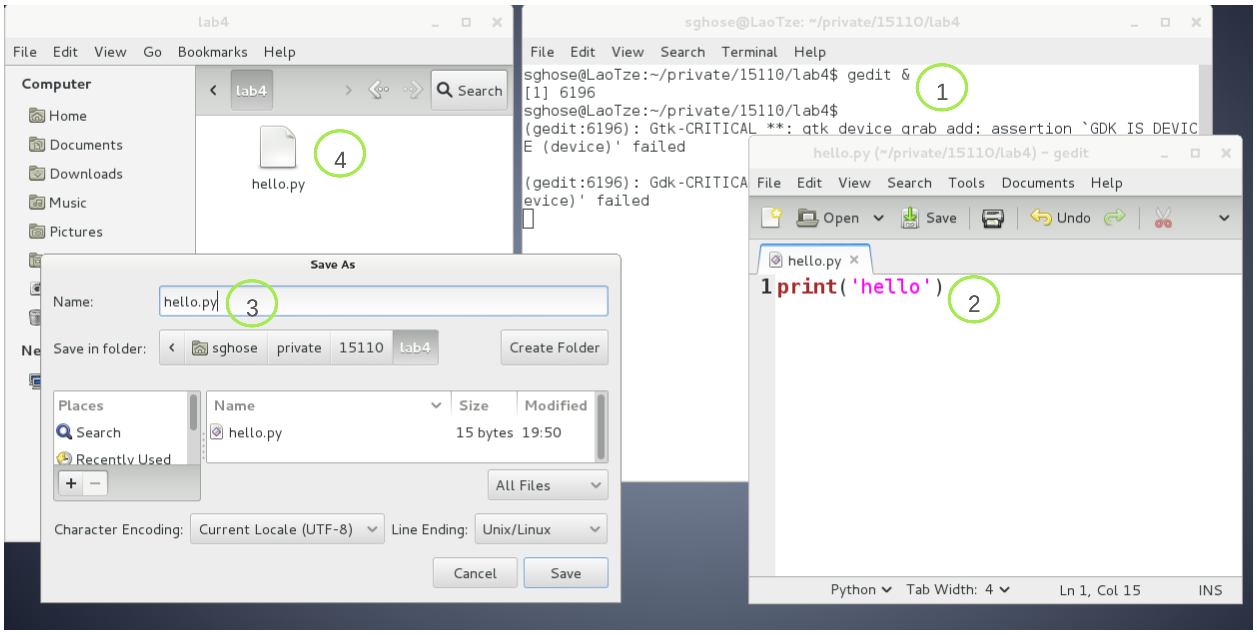
```
sghose@LaoTze:~/private/15110$ ls
lab1  lab2  lab3  lab4
sghose@LaoTze:~/private/15110$ pwd
/home/sghose/private/15110
sghose@LaoTze:~/private/15110$ cd lab4
sghose@LaoTze:~/private/15110/lab4$ pwd
/home/sghose/private/15110/lab4
```

We first see where we are using pwd and then enter the lab4 folder using the cd command, verifying we are there using pwd again! (cd like double clicking a folder to enter it in a file explorer such as Windows Explorer or Finder)

You can cd <folder name> to go into a directory/folder and cd .. if you want to *exit* a folder.

Opening Programs In Shell

We are going to make a file called **hello.py** in the lab4 folder using gedit after opening it from the shell



```
sghose@LaoTze:~/private/15110/lab4$ gedit &
[1] 6196
sghose@LaoTze:~/private/15110/lab4$
(gedit:6196): Gtk-CRITICAL **: gtk_device_grab_add: assertion `GDK_IS_DEVIC
E (device)' failed

(gedit:6196): Gdk-CRITICA
evice)' failed
```

```
print('hello')
```

Name: hello.py

Save in folder: sghose private 15110 lab4

Name	Size	Modified
hello.py	15 bytes	19:50

Character Encoding: Current Locale (UTF-8) Line Ending: Unix/Linux

1. Open gedit in terminal with the & (it means you want to use the terminal even after you launched the program, otherwise the terminal will only run **gedit** and will not be available for further commands – like if you want to run python in the same terminal)

2. Enter your code!

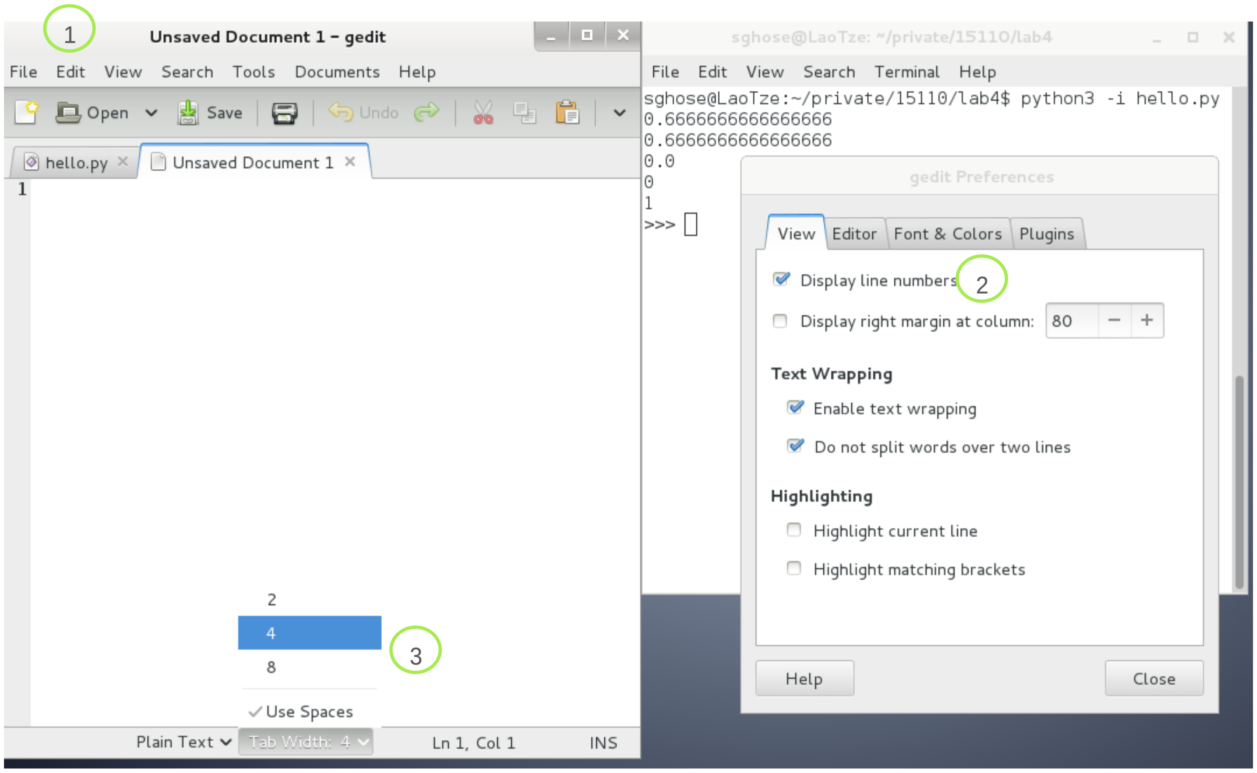
3. Save it in the lab4 folder as hello.py (you don’t need to change “character encoding” or “line encoding”)

4. Verify it’s in the folder using the file browser *and* the ls command

Python

Fun fact: Python is named after Monty Python

Setting up Gedit


1
Unsaved Document 1 – gedit
File Edit View Search Tools Documents Help
Open Save Undo
hello.py
Unsaved Document 1
2
3
Plain Text
Ln 1, Col 1
INS
sghose@LaoTze: ~/private/15110/lab4
File Edit View Search Terminal Help
gedit Preferences
View Editor Font & Colors Plugins
Help
Close

```
sghose@LaoTze:~/private/15110/lab4$ python3 -i hello.py
0.6666666666666666
0.6666666666666666
0.0
0
1
>>> 
```

- [✓] Display line numbers
- [ ] Display right margin at column: 80

Text Wrapping

- [✓] Enable text wrapping
- [✓] Do not split words over two lines

Highlighting

- [ ] Highlight current line
- [ ] Highlight matching brackets

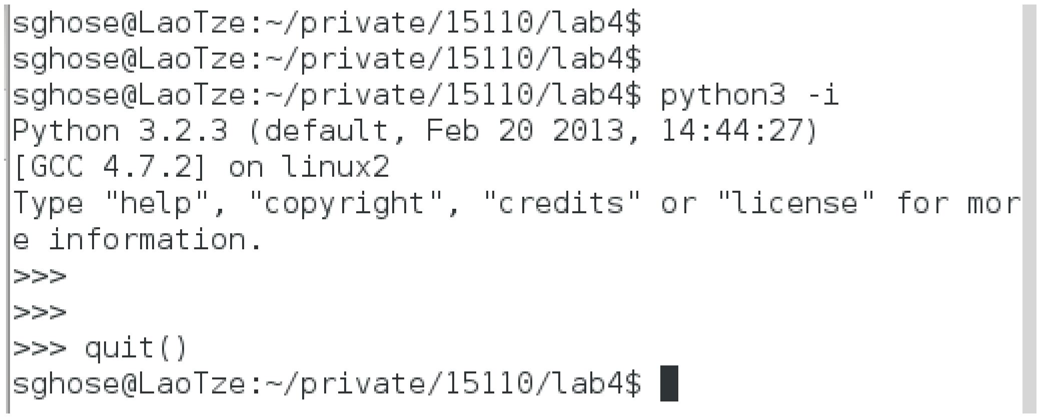
Tab Width: 4 — 2, [✓ 4], 8; ✓ Use Spaces

1,2. Check Edit -> Preferences -> Display line numbers if you want the line numbers next to each line of code

Also **Highlight matching brackets** is very useful when looking for that missing bracket!

3. Make sure at the bottom you are **using spaces** and there are **4 spaces for each tab** (that way you can hit tab instead of space 4 times for every python block – such as in functions).

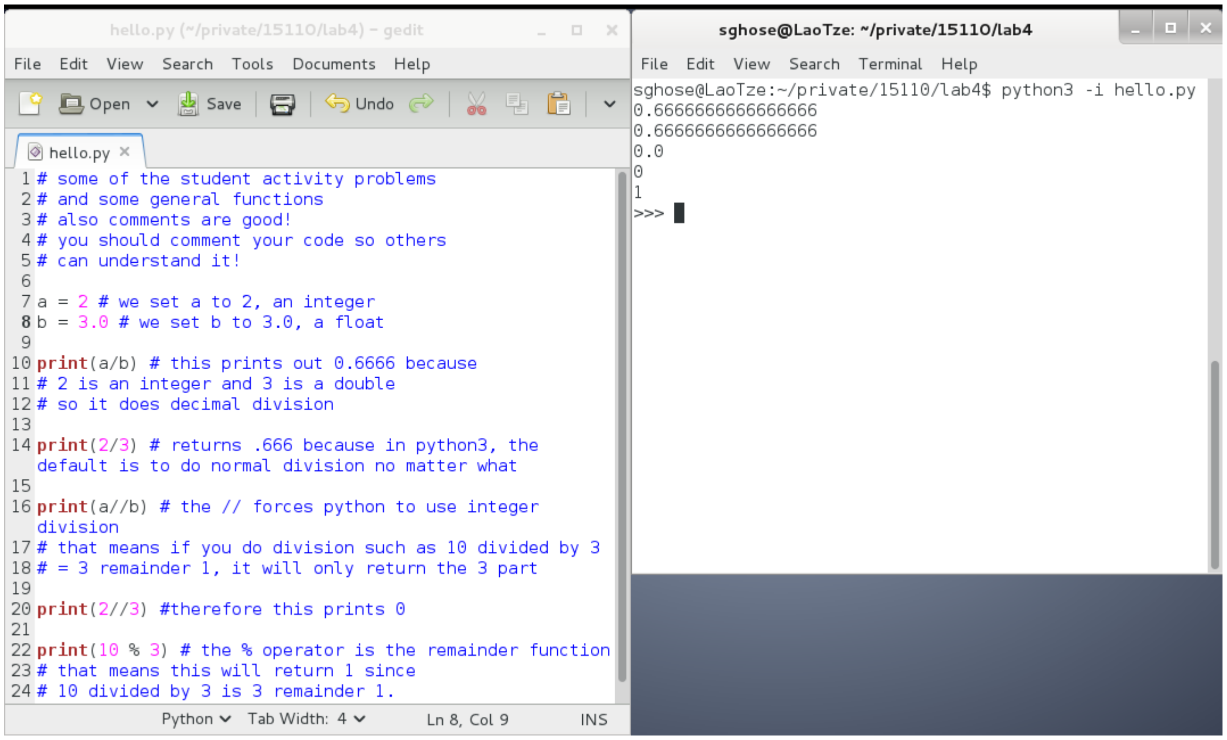
How do I know I’m in Python vs Shell?



```
sghose@LaoTze:~/private/15110/lab4$
sghose@LaoTze:~/private/15110/lab4$
sghose@LaoTze:~/private/15110/lab4$ python3 -i
Python 3.2.3 (default, Feb 20 2013, 14:44:27)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for mor
e information.
>>>
>>>
>>> quit()
sghose@LaoTze:~/private/15110/lab4$
```

Python has the >>> prompt, Shell has the bash-4.1 or your <Andrew id>@<the computer name>. You can exit Python using Ctrl-D or the quit() function

Assigning variables and some tricky operators in Python



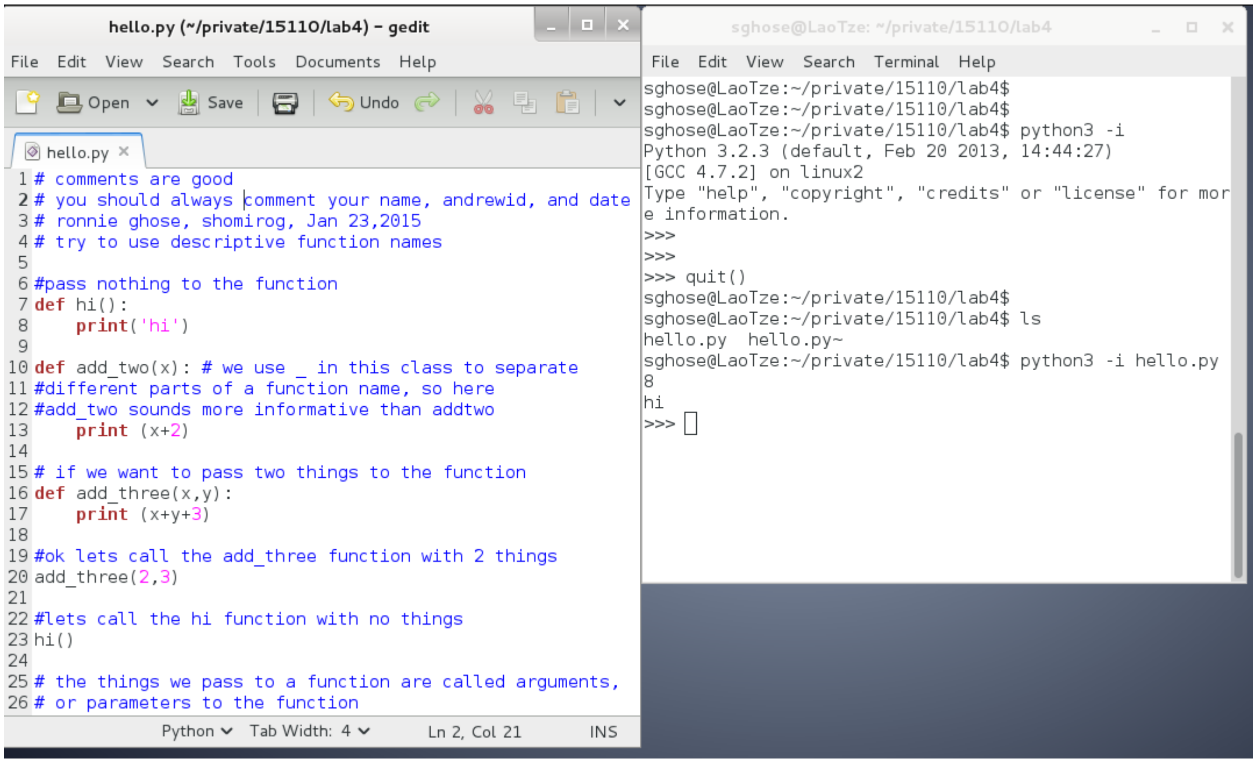
hello.py (~/private/15110/lab4) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
hello.py
Python Tab Width: 4 Ln 8, Col 9 INS
sghose@LaoTze: ~/private/15110/lab4
File Edit View Search Terminal Help

```
# some of the student activity problems
# and some general functions
# also comments are good!
# you should comment your code so others
# can understand it!

a = 2 # we set a to 2, an integer
b = 3.0 # we set b to 3.0, a float

print(a/b) # this prints out 0.6666 because
# 2 is an integer and 3 is a double
# so it does decimal division

print(2/3) # returns .666 because in python3, the
default is to do normal division no matter what

print(a//b) # the // forces python to use integer
division
# that means if you do division such as 10 divided by 3
# = 3 remainder 1, it will only return the 3 part

print(2//3) #therefore this prints 0

print(10 % 3) # the % operator is the remainder function
# that means this will return 1 since
# 10 divided by 3 is 3 remainder 1.
```

```
sghose@LaoTze:~/private/15110/lab4$ python3 -i hello.py
0.6666666666666666
0.6666666666666666
0.0
0
1
>>>
```

(refer to lecture notes http://www.cs.cmu.edu/~./15110/schedule.html for more comprehensive notes)

Functions in Python


hello.py (~/private/15110/lab4) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
hello.py
Python Tab Width: 4 Ln 2, Col 21 INS
sghose@LaoTze: ~/private/15110/lab4
File Edit View Search Terminal Help

```python
# comments are good
# you should always comment your name, andrewid, and date
# ronnie ghose, shomirog, Jan 23,2015
# try to use descriptive function names

#pass nothing to the function
def hi():
    print('hi')

def add_two(x): # we use _ in this class to separate
#different parts of a function name, so here
#add_two sounds more informative than addtwo
    print (x+2)

# if we want to pass two things to the function
def add_three(x,y):
    print (x+y+3)

#ok lets call the add_three function with 2 things
add_three(2,3)

#lets call the hi function with no things
hi()

# the things we pass to a function are called arguments,
# or parameters to the function
```

```
sghose@LaoTze:~/private/15110/lab4$
sghose@LaoTze:~/private/15110/lab4$
sghose@LaoTze:~/private/15110/lab4$ python3 -i
Python 3.2.3 (default, Feb 20 2013, 14:44:27)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for mor
e information.
>>>
>>>
>>> quit()
sghose@LaoTze:~/private/15110/lab4$
sghose@LaoTze:~/private/15110/lab4$ ls
hello.py  hello.py~
sghose@LaoTze:~/private/15110/lab4$ python3 -i hello.py
8
hi
>>> 
```

Things to note:

Def means define, you're defining a function, telling it what to expect, and then ending with a semicolon.

This is just like in math where you say f(x) = 2*x, you would say

```
def f(x):
        return 2*x
```

Return means return a value! So if you do for example add_two(add_three(2)), you want add_three to return a value, not Just print out something. Refer to lecture notes here for more details and examples! Also if you ever want practice problems / have questions / want clarification about lab, remember we have office hours! ☺