15-110: Principles of Computing, Spring 2018

Lab 2 – Thursday, January 25

Goals

- Review the various sites we're using: Piazza, Autolab, Gradescope.
- Configure gedit.
- Learn how to interact with Python3
- Create and test simple Python programs

Part 1 : General Information

1.1 Piazza

a. Make sure you set up your access to Piazza. You should have received an email from Piazza if you're registered for the course.

1.2 Autolab

a. You should make all submissions for Labs and Programming Assignments (PA) on Autolab (`https://autolab.andrew.cmu.edu`).
b. In order to see your submissions, click the 'View handin history' button. Note that only your **last submission will be graded**.
c. If you click the magnifying glass icon next to your submission, you should be able to see all files in the zipped folder that you have submitted. If you click the magnifying glass icon next to each file, you should be able to see the content of the file.
d. For each submission, make sure you double check if you have submitted the right work.

1.3 Gradescope

a. You should make all submissions for Problem Sets (PS) to gradescope (`https://www.gradescope.com`). We suggest you find an editor for PDF files so you can enter your answers directly on the PDF file that we supply. After you hand in your answered PDF file to gradescope, be sure you submit the correct file by checking your handin.

Can't access Piazza, Autolab or Gradescope? Email richasin@andrew.cmu.edu immediately!

1.4 Preparing for lab2

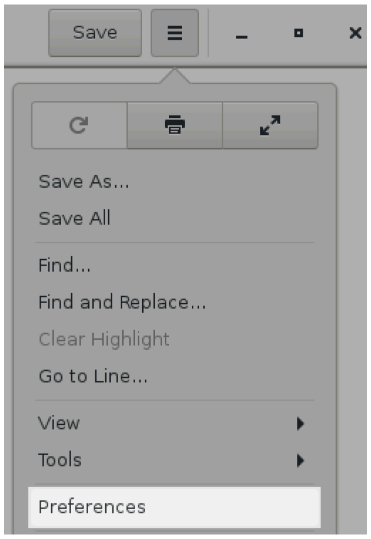
```
cd private/15110
mkdir lab2
cd lab2
```

Part 2 : Using Gedit in the lab

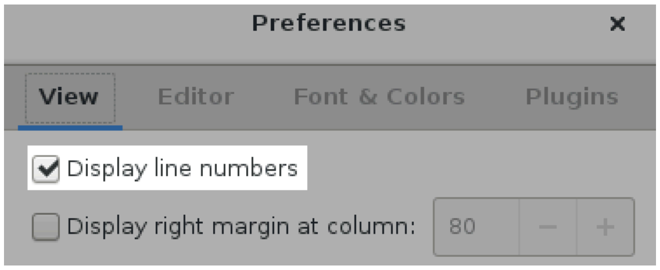
2.1 Gedit Setup

a. Open gedit by typing '`gedit &`' in Terminal (without the quotes). Your terminal window should still be active. The `&` means run `gedit` in the background.

b. Click the 'Preferences' button.

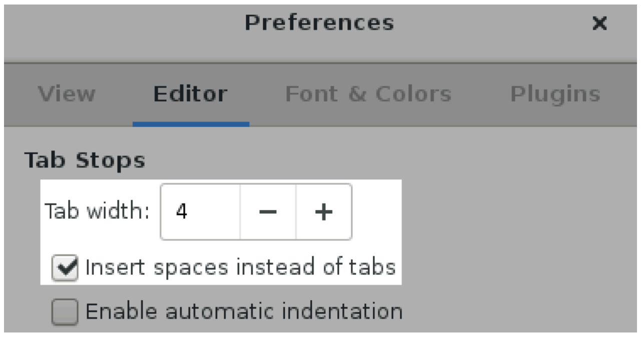


c. Turn on the 'Display line numbers' option under the 'View' tab.



☑ Display line numbers
☐ Display right margin at column: 80

d. Click the 'Editor' tab. Set tab width to 4 spaces and turn on the 'Insert spaces instead of tabs' option.

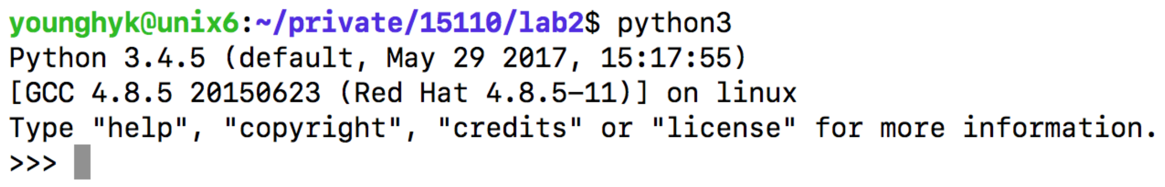


Tab Stops
Tab width: 4
☑ Insert spaces instead of tabs
☐ Enable automatic indentation

Part 3 : Basic Expressions in Python Shell

3.1 Running Python3 in Terminal [TA Demonstrations]

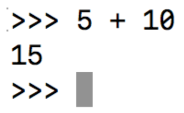
a. Type ‘`python3`’ in Terminal.



```
younghyk@unix6:~/private/15110/lab2$ python3
Python 3.4.5 (default, May 29 2017, 15:17:55)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

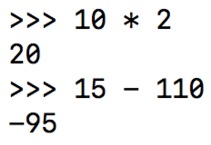
The prompt >>> on the last line indicates that you are now in an interactive Python interpreter session, also called the Python shell. This is **different from the normal terminal command prompt**!

b. You can now enter some code for Python to run. Type ’`5 + 10`’ and press the `Enter` key.



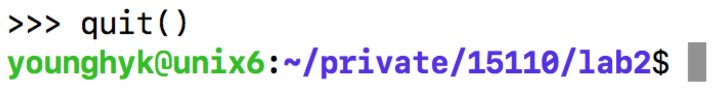
```
>>> 5 + 10
15
>>> 
```

Note that after showing the result, Python brings you back to the interactive prompt, where you can enter another command.



```
>>> 10 * 2
20
>>> 15 - 110
-95
```

c. You can exit the Python shell by typing ‘`quit()`’ or holding down the `Ctrl` and `D` keys at the same time.



```
>>> quit()
younghyk@unix6:~/private/15110/lab2$ 
```

3.2 Basic Expressions in Python Shell [Student Activities]

First, create a text file `answers.txt` using gedit. Then, type each of the following expressions into the Python shell. What **value** does each of the following Python expressions evaluate to? Is the value an **integer** or a **floating point**? Write your answers in `answers.txt`.

a. `28 % 5`

b. First, type ‘`20 + 35 * 2`’ and press the `Enter` key. What do you get?
Now, type ‘`(20 + 35) * 2`’ and press the `Enter` key. What do you get?
Why are they different?

c. First, type ‘`2 / 3 * 3`’ and press the `Enter` key. What do you get?
   Now, type ‘`2 // 3 * 3`’ and press the `Enter` key. What do you get?
   Why are they different?

d. `2.0 ** 6`

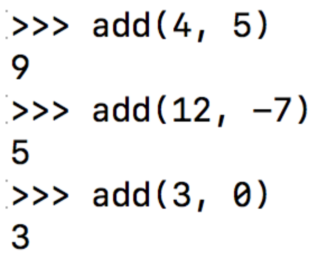
Part 4 : Basic Functions in Python

4.1 Creating and Using Functions [TA Demonstrations]

a. Create a file `add.py` using gedit. In `add.py`, define a function `add(x, y)` that has two parameters `x` and `y` and calculates their sum.

b. Load `add.py` by typing ‘`python3 –i add.py`’ in Terminal.

c. Test the function using different arguments.



```
>>> add(4, 5)
9
>>> add(12, -7)
5
>>> add(3, 0)
3
```

4.2 Using Built-In Functions [TA Demonstrations]

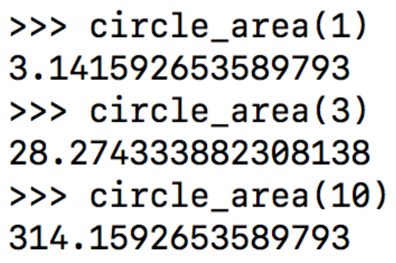
a. Create a file `circle.py` using gedit. In `circle.py`, define a function `circle_area(r)` that takes the radius `r` to calculate the area of the circle whose radius is `r`. (Note that the mathematical formula to find a circle’s area is $\pi \cdot r^2$.)

b. What do we write to use π in python3?

c. Add a comment to explain what `r` means by typing # at the beginning of the line.

d. Load `circle.py` by typing ‘`python3 –i circle.py`’ in Terminal.

e. Test the function using different arguments.



```
>>> circle_area(1)
3.141592653589793
>>> circle_area(3)
28.274333882308138
>>> circle_area(10)
314.1592653589793
```

4.3 Creating and Using Functions [Student Activities]

a. Create a file `sphere.py` using gedit. In `sphere.py`, define a function `sphere_volume(r)` that has one parameter representing the radius `r` and calculates and returns the volume of the sphere whose radius is `r`. (Note that the mathematical formula to find a sphere's volume is $\frac{4}{3} \cdot \pi \cdot r^3$.) Test your function in `python3`.

   In `answers.txt`, place a copy of your interaction with `python3`. That is, load `sphere.py`, call `sphere_volume(r)` with any radius, and copy and paste what you see (including the result) in Terminal to `answers.txt`.'

b. Create a file `hypotenuse.py` using gedit. In `hypotenuse.py`, define a function `hypotenuse(a, b)` that has two parameters representing the side lengths `a` and `b` of a right triangle and calculates and returns the length of the hypotenuse `c`. (Note that the Pythagorean Theorem states that $c^2 = a^2 + b^2$.) Test your function in `python3`.

   In `answers.txt`, place a copy of your interaction with Python3. That is, load `hypotenuse.py`, call `hypotenuse(a, b)` with any side lengths, and copy and paste what you see (including the result) in Terminal to `answers.txt`.

c. [Optional] Pick a mathematical formula that is used in your major or that interests you. Create a file `my_function.py` using gedit. In `my_function.py`, define a function that calculates some value using the formula.

   At the beginning of your `my_function.py`, include comments that describe what the function computes and what each parameter is.

Submission

When you finish the lab, you should be inside the `lab2` folder, which is inside the `private/15110` directory. When you type '`ls`' and press the `Enter` key, you should see the following files:

a. `answers.txt`
b. `add.py`
c. `circle.py`
d. `sphere.py`
e. `hypotenuse.py`
f. [Optional] `my_function.py`

Once you see all files, please type '`cd ..`' and press the `Enter` key.
Then, zip your `lab2` folder by typing

```
zip –r lab2.zip lab2
```

Please submit the zipped file `lab2.zip` on Autolab.
Check your submission to make sure it was uploaded correctly!