Lab 0 – Getting started with Python

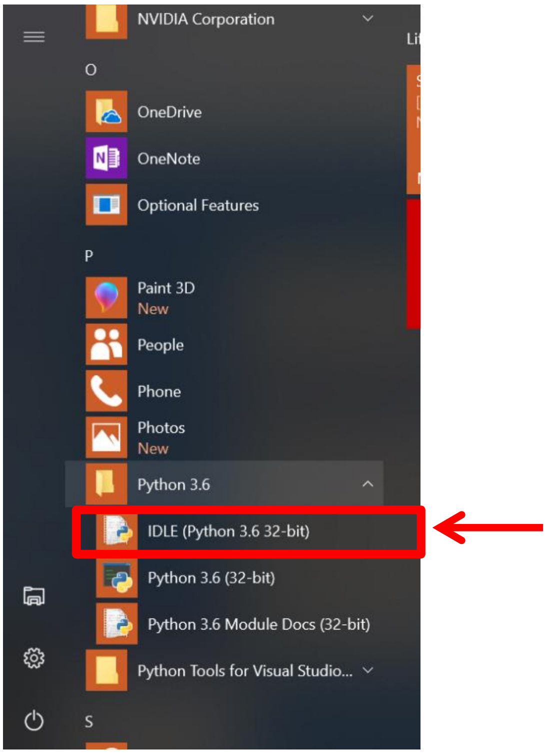
Welcome to your first Python lab session! The next four laboratory sessions are designed to be your opportunity to experiment with Python and gain hands-on experience.

The primary goal of this lab is to introduce you to the Python Development Environment (i.e. IDLE), and gives you the chance to write what might be your first program in Python. You will be able to experiment with both editor and interactive interpreter (or 'shell').

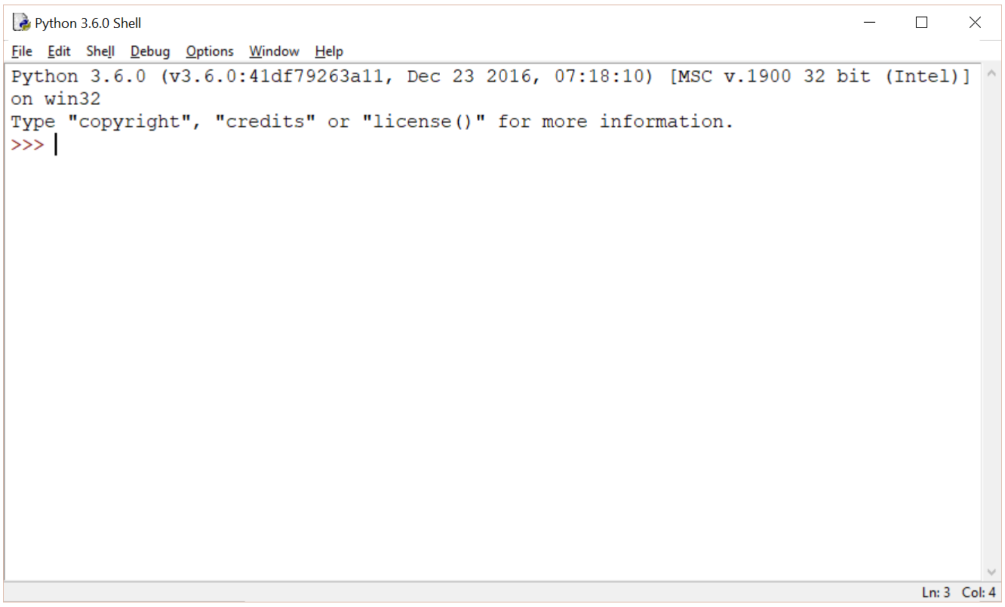
Part 1: Python Development Environment - IDLE

* In this lab, we will be using IDLE exclusively. However, there are many development environments for Python, some are better than others depending on how you plan to use Python. So feel free to explore any development environment on your own.

1. The first thing we would like to do is start running the Python program development tool named IDLE, which should be listed in the installed programs under Python tab. Note: We will be using Python 3.4.0 and above for this course.



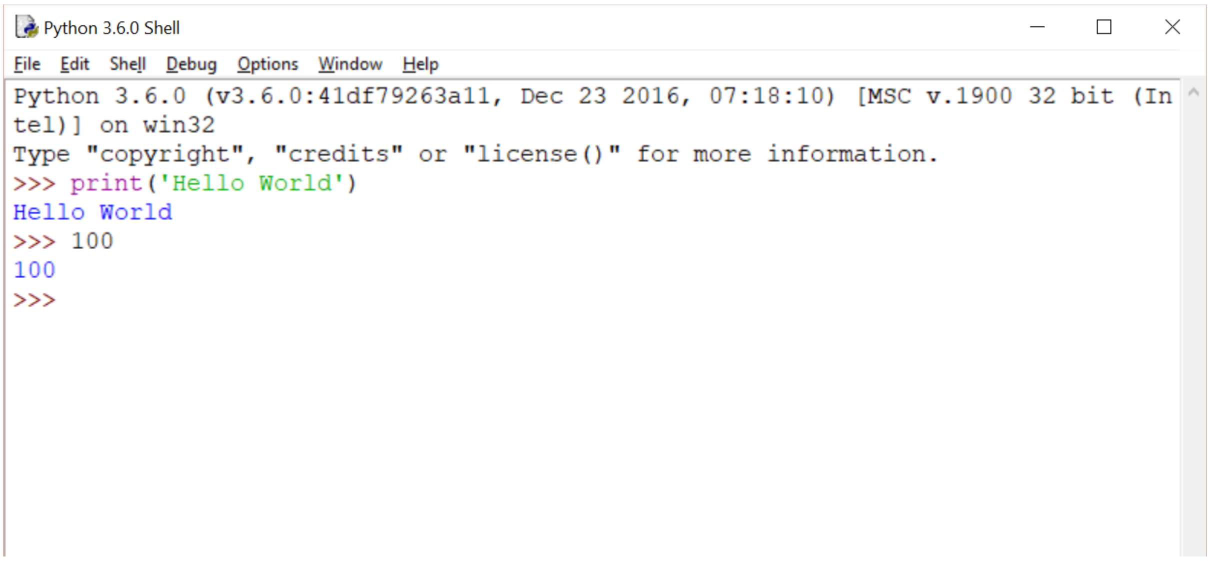
2. We will see that a new window will open up.



```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
```

3. This is the main window to IDLE, and what we see right now is called the interactive interpreter (or ‘shell’). When we type something with value, the interactive interpreter prints its value automatically.
   Play with the interactive interpreter a little.
   - As per tradition, let’s get Python to say the immortal words, “Hello World”.
   - Type 100 in the interactive interpreter, it will be echoed to your terminal.
   - It can also be used like a calculator, for instance type 8 * 9. Press **Enter** key to see the result.

Those ‘>>>’ signs act as a prompt for us: Python is ready to read in a new command by giving us that visual cue.



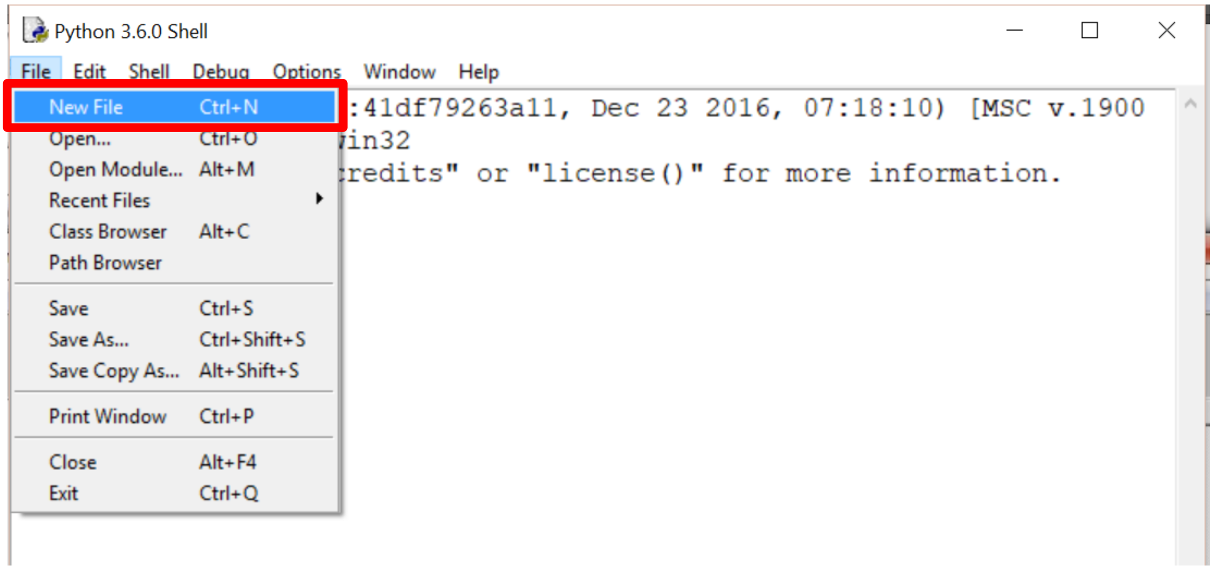
```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello World')
Hello World
>>> 100
100
>>> 
```

The idea is that we can experiment with Python by typing in commands. If things don’t work, we can always correct the mistake and try again.

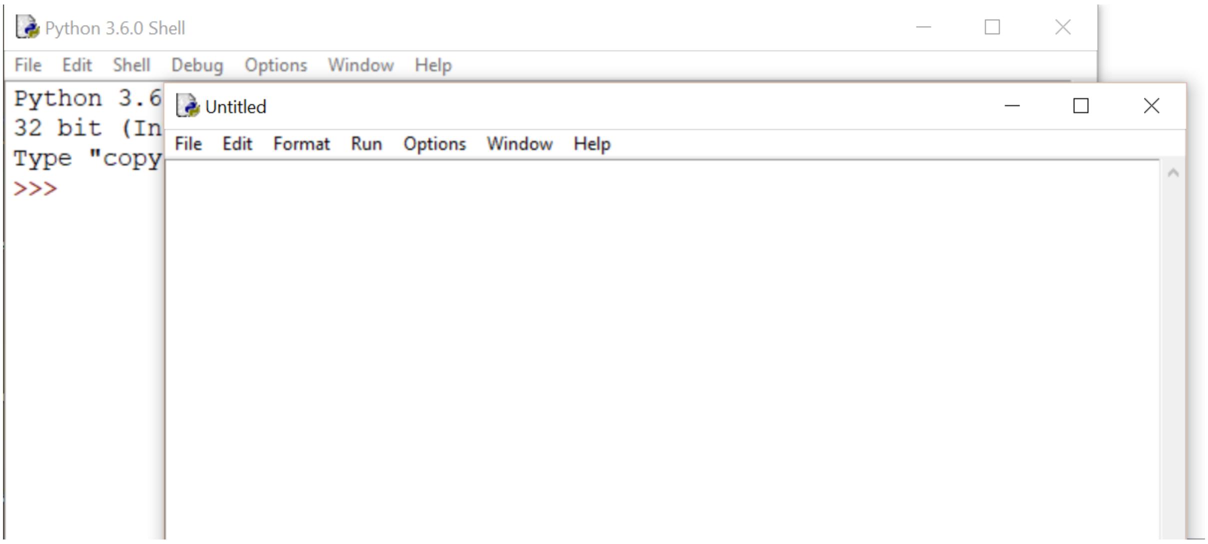
4. The thing with interactive interpreter is that if we close down Python and start it up again, how do we get the computer to remember what we have typed?

   The solution is a little subtle: we can't directly save what is on the interpreter because it will include both our commands and the system's responses. What we would like is to make a prepared file, with just our own commands, and to be able to save that file as a document. We can later open that file and "run" Python over it, saving us the time of retyping the whole thing over again.

   So let’s do that. Now open up a new file.



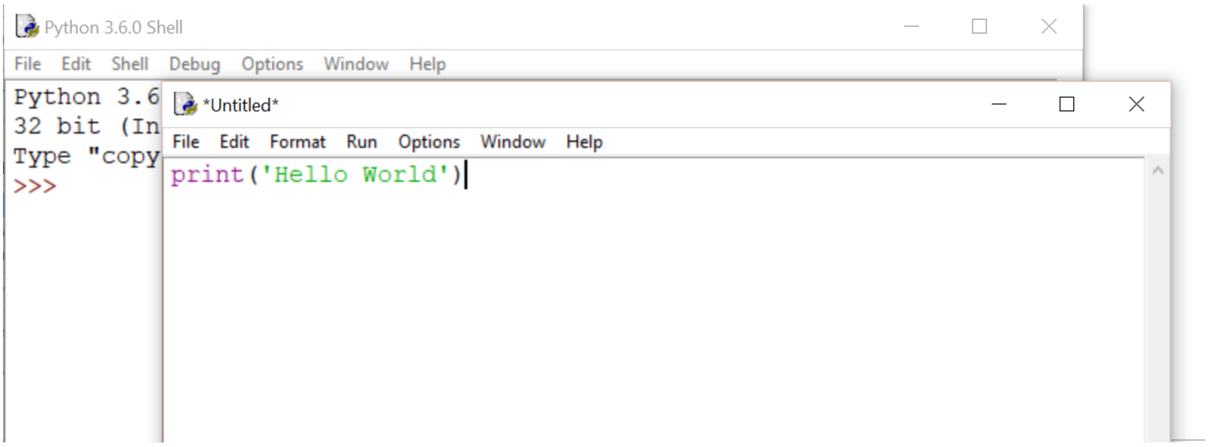
```
:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900
in32
redits" or "license()" for more information.
```



```
Python 3.6
32 bit (In
Type "copy
>>>
```

5. We will see that there is nothing in this new file. This is the place where it is purely for our commands. Let's call this the 'editor'. In the editor, type:

```
print('Hello World')
```

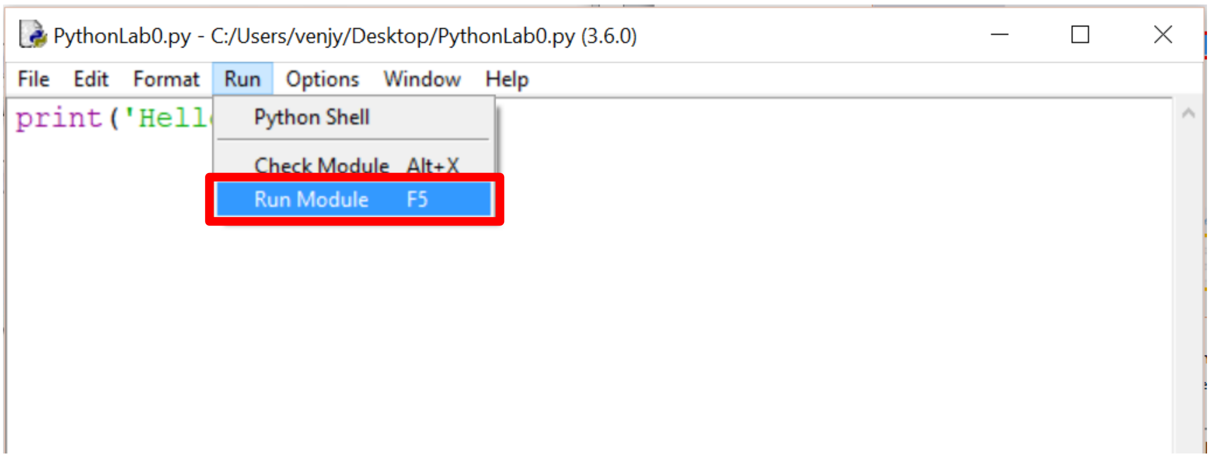


Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Untitled
File Edit Format Run Options Window Help

```
Python 3.6
32 bit (In
Type "copy
>>>
```

```
print('Hello World')
```

6. Remember to save your program. So now how do we run the program? Look at the menus on the editor window, there is a menu option 'Run Module'.



PythonLab0.py - C:/Users/venjy/Desktop/PythonLab0.py (3.6.0)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

```
print('Hell
```

If you have done all the steps correctly, you will be able to see 'Hello World' in the interactive interpreter.

7. Just before we wrap up, it is important to know that like other development environments, IDLE has a dedicated debugger. This YouTube video provide a good overview of the build in debugger in IDLE.

8. Now you're all set!

Bonus: Little secrets in Python

Despite being a serious language, Python is full of surprises, i.e. Easter eggs. Easter eggs are always a sweet surprise. Try them out and be surprised!

Type in the interactive interpreter:

# 1st Easter egg:

```
>>> import antigravity
```

This will instantly brought to you a webpage through your browser. This webpage will have a small stick-figure cartoon, basically to make fun of people who use Python.

#2nd Easter egg:

```
>>> import this
```

It will bring up a giant poem written by Tim Peters. It started out as a joke but later realized by many to actually represent Python.

Now continue to type the following in the interactive interpreter:

```
>>> love = this
>>> this is love
>>> love is True
>>> love is False
>>> love is not True or False
>>> love is not True or False; love is love
```

There are more Easter eggs waiting to be discovered. Have fun hunting!!! ☺