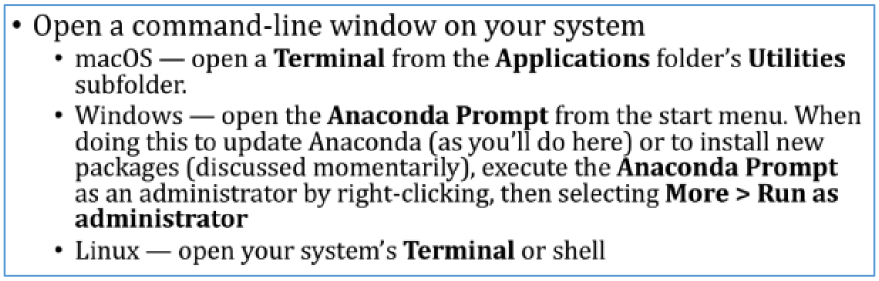
CS121 Chapter 2 Lab – Python Fundamentals

Name: Jake Gadaleta

****

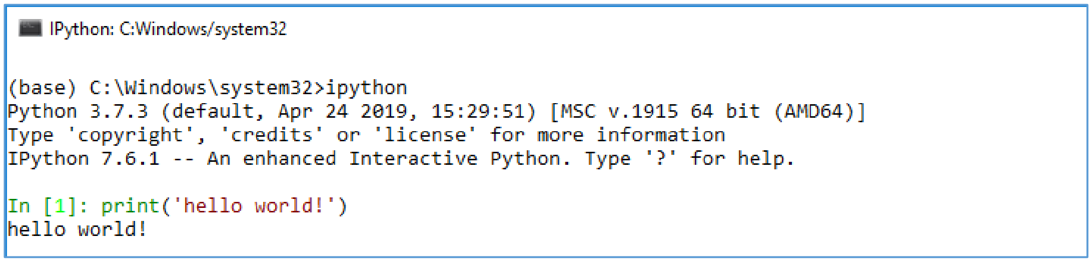
Parts 1 - 8 of this lab will use the IPython Interactive Mode to execute short snippets, thus immediately showing you your results. Here is a reminder on how to get to this mode.

- Open an anaconda prompt terminal window.



- Open a command-line window on your system
  - macOS — open a **Terminal** from the **Applications** folder's **Utilities** subfolder.
  - Windows — open the **Anaconda Prompt** from the start menu. When doing this to update Anaconda (as you'll do here) or to install new packages (discussed momentarily), execute the **Anaconda Prompt** as an administrator by right-clicking, then selecting **More > Run as administrator**
  - Linux — open your system's **Terminal** or shell

- Type ipython. Then you are able to execute commands in python, like I have done below.



```
(base) C:\Windows\system32>ipython
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.6.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello world!')
hello world!
```

****

To get full credit for this lab, you must fill in every blank (using the Adobe "fill and sign" - use the "me" option) and you must submit the code you create. More instructions are at the end of this lab.

1 GETTING COMFORTABLE WITH PRINT STATEMENTS

On your thumb drive (if you are using a lab computer) OR on your laptop, create a folder called "LabCh2". Save a copy of this pdf to that folder. Then open it with ADOBE!!

2 GETTING COMFORTABLE WITH PRINT STATEMENTS

1. READ (DO NOT TYPE CODE YET). If x = 2 and y = 3, guess what will be printed out by the code below. Put your guess on the lines below

   `print ('x =', x)` x= 2

   `print('Value of', x , '+', x, 'is', (x+x))` Value of 2 + 2 is 4

   `print('x = ')` x =

   `print((x+y), '=', (y+x))` 5=5

2. Now use the IPython Interactive Mode editor to create variables x and y which should be set to 2 and 3, respectively. Then type in the code shown to you in 1. Fix any incorrect guesses.

3. We are going to move on to the next part but please don't delete/clear your command prompt because I plan to have you save the contents later in this lab.

3 GETTING COMFORTABLE WITH ESCAPE SEQUENCES

1. When the backslash character appears within quotes it is called the *escape character*. If another character follows it immediately without a space, it is called an *escape sequence*. The main escape sequences are below.

\n	End line
\t	Tab
\\	Display the backslash character
\’	Display single quotes
\”	Display double quotes

2. Type the following command and see what you get. Does the result make sense? yes

```
print ("We\n\thave\n\t\toff\n\t\t\tMonday!\n\t\t\t\tYay!")
```

3. Using escape sequences, set the variable x below equal to a string so that this text appears on the console:
Save the file at C://MyFolder/MySubfolder.

```
x = ???
print (x)
```

x = "C:\\\\MyFolder\\MySubfolder

4. Python does allow you to get out of using escape sequences when printing single/double quotes in some scenarios. Run the following code. It will fail because you cannot put single quotes inside single quotes.

```
print(‘Her name is Mary O'Bono.’)  //THIS CAUSES AN ERROR
```

To correct this, since we want single quotes in the string, use double quotes on the outside, like this.

```
print("Her name is Mary O'Bono.") //THIS DOES NOT
```

5. Now, the trick in #4 won’t help us here because in this example we want to print a statement with BOTH a single and double quote.

So using escape sequences, set the variable x below equal to a string so that this text appears on the console:
Mary O’Bono screamed, “Run Spot! Run!!” and then started running as well.

```
x = ???
print (x)
```

x = "Mary O'Bono screamed, \"Run spot! Run!!\" and then started running as well"

6. Luckily, here’s another trick. Single/double quotes pop up often in print statements, so you can also use the triple-quoted strings. In triple quoted strings, you won’t need escape sequences. Print this command to the screen and notice there are no escape sequences.

```
print("""Mary O’Bono screamed, “Run Spot! Run!!” and then started running as well.""")
```

4 GETTING INPUT FROM THE CONSOLE/USER

1. The following code should read an INTEGER into the variable rating. What is wrong with this code?

```
rating = input('Enter an integer rating between 1 and 10: ')
```

You must wrap the input in an int() cast

input will always come in as a string

2. FIX what is wrong with the above line of code and type your code into the ipython interactive mode editor. Then use the "type(…)" method to show that you have applied the correct fix.

rating = int(input('Enter an integer rating between 1 and 10: '))

5 ARITHMETIC REVIEW

1. GUESS the output from executing the lines of code below.

`27.5 + 2` 29.5

`27.5 - 2` 25.5

`27.5 * 2` 55

`27.5 / 2` 13.75

`27.5 // 2` 13

`27.5 ** 2` 756.25

`27.5 ** 1/2` 13.75

`27.5 ** (1/2)` 5.2440

2. Now run this code in the IPython Interactive Editor. Fix any incorrect guesses.

6 IF STATEMENTS

Remember that the syntax for a generic if statement is as follows.

```
if condition-to-check:

    what-to-do-if-true //Remember this line MUST be indented to b
                       //part of the “if” suite
```

1. If a variable x is set to 7, first GUESS: what will be printed out below?

```
if x >= 5:

    print (x, 'is at least 5!')

    print ('The square of', x, 'must therefore be at least 25!')
```

7 is at leasat 5

The square of 7 must therefore be at least 25

2. Create a variable set to 7, type in the code above into the IPython editor. To end multi-line input in the Python interpreter, you’ll have to hit enter an additional time. Was your guess right? If not, correct it above.

3. If a variable x is set to 3, first GUESS: what will be printed out below?

```
if x >= 5:

    print (x, 'is at least 5!')

print ('The square of', x, 'must therefore be at least 25!')
```

The square of 3 must therfore be at least 25

___

4. Create a variable set to 3, type in the code above in #3 into the IPython editor. Was your guess right? If not, correct it above.

5. Create a variable called grade and initialize it to 91.

6. Now create a variable grade that is initialized to 91. Replace the stars in the code below with a statement that will print out “Congratulations! Your grade of 91 earns you an A in this course!”

```
if grade >= 90:
        ***
```

7 Remainder Review

Remember that the % symbol tells us the remainder when division occurs. For example:

```
25 % 7 evaluates to 4.
```

1. Write an "if" statement that uses the reminader operator (%) to check if 1024 is a multiple of 4.

2. Write an "if" statement that uses the reminader operator (%) to check if 17 is a multiple of 10.

8 SAVE YOUR IPYTHON INTERACTIVE MODE WORK

1. Look at your command prompt. What is the final identifoer of your last prompt in Python prompt number? You will need this number in the next line.

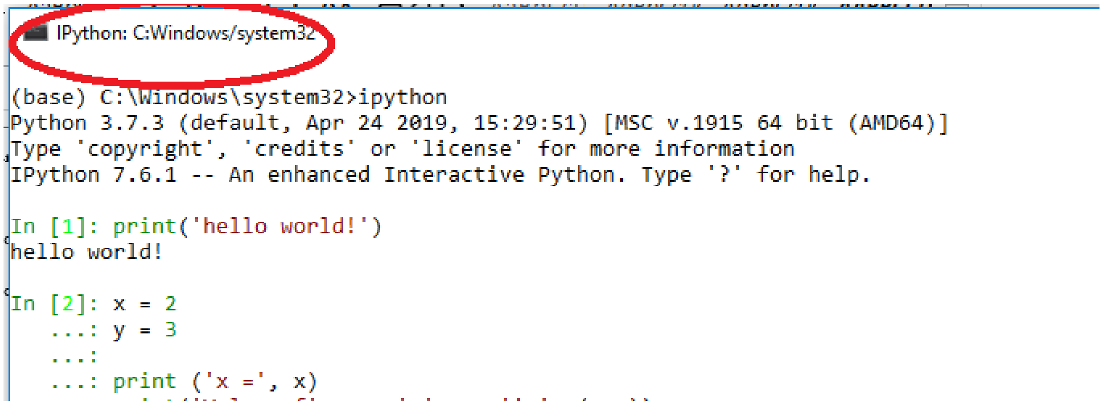
2. If your last prompt identifier is 10, then type the following command to save your work into a script. You must include the % sign. This will create a Python script from your interactive IPython prompt.

   %save LabCh2InteractiveMode_YourLastName 1-10

   If you only want me to see lines 1-5, 6, and 10, then you could also indicate that, but I don't mind if you just give me your whole script for now.

   %save LabCh2InteractiveMode_YourLastName 1-5 6 10

3. The above command ill create a file called LabCh2InteractiveMode_YourLastName.py. Because the above command did not specify where to save the file, it will be saved at whatever location is indicated at the top of the command prompt window. (See the red below – for me this was C:Windows/system32.) Go to this location, search for the file, and then paste it into your "LabCh2" folder so that you can submit it later.



```
(base) C:\Windows\system32>ipython
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.6.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello world!')
hello world!

In [2]: x = 2
   ...: y = 3
   ...:
   ...: print ('x =', x)
```

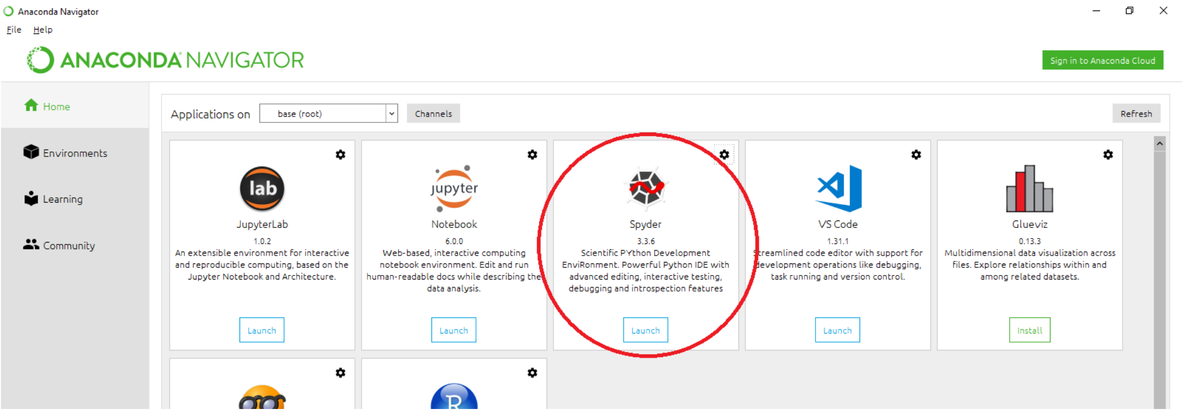
4. Right click on the file and "open with" notepad. Check out what is in the file. It's just your Python code!

5. You just created your first ",py" file. A ".py" file is what we call a python source file. It's really just a text file and the ".py" indicates that it has python code in it. This helps the Python compiler recognize the file.

****

For the rest of this lab, you will use the Spyder IDE to create Python Scripts. This is what you will use to create your HW scripts.

- Search for the Anaconda Navigator in your Windows Search Bar. Double click on it. It could take a while to load for the first time. Please select the "Launch" Button underneath the Spyder IDE.



JupyterLab 1.0.2 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook 6.0.0 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

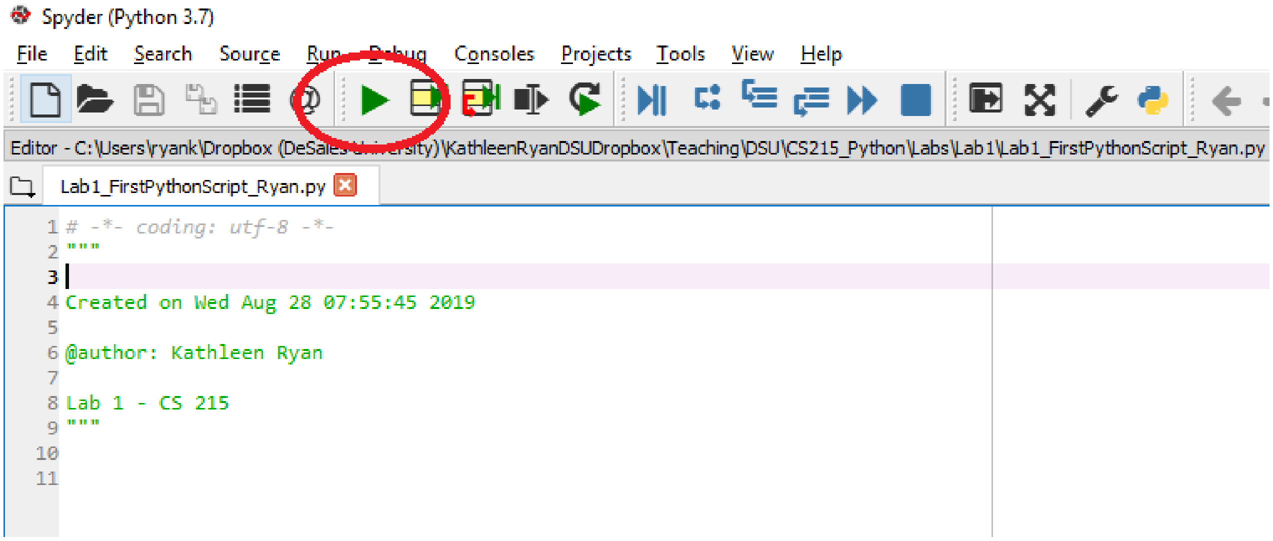
Spyder 3.3.6 Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code 1.31.1 Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz 0.13.3 Multidimensional data visualization across files. Explore relationships within and among related datasets.

- A basic Python script should appear with initial comments/header information. Please use File-Save As and save it as "LabCh2_FirstPythonScript_YourLastName" in your LabCh2 folder. Update the comments/header information with your information, like I did in the picture below.

- Circled in Red is a green triangle that you click on when you wish to execute your script.



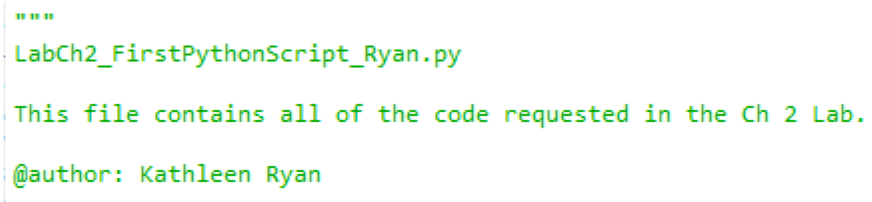
Editor - C:\Users\ryank\Dropbox (DeSales University)\KathleenRyanDSUDropbox\Teaching\DSU\CS215_Python\Labs\Lab1\Lab1_FirstPythonScript_Ryan.py

```
# -*- coding: utf-8 -*-
"""

Created on Wed Aug 28 07:55:45 2019

@author: Kathleen Ryan

Lab 1 - CS 215
"""


```

****

9 SETTING UP THE TOP OF THE SCRIPT FILE

1. Update the comments at top so that you see something like that at the right. The top of your .py files should always include, the name of the file, a description of the program, and your name, from today till the end of the class. You can keep the date created if you like or you can delete it.



```
"""
LabCh2_FirstPythonScript_Ryan.py

This file contains all of the code requested in the Ch 2 Lab.

@author: Kathleen Ryan
```

2. Notice that we used triple quotes here! This gives us another reason to use triple quotes: for making comments that span multiple lines.

10 CIRCLE AREA, DIAMETER, AND CIRCUMFERENCE

1. Add this to your script file.

```
#Calculate geometric values of a circle.
```

2. Create a variable called pi and set it to 3.1415926. (We will later learn how to use a more precise value for pi.)

3. Ask the user for a radius and save his/her response as a float variable called "radius".

4. Print out to the user what the diameter (2r), circumference (2πr), and area (πr^2) of a circle with the given area would be. Example output is below. You can test your program by running it, i.e., hitting the green triangle that was circled in red in the picture on the previous page.

   You should not use any functions from the Math package to do this. Your output should match mine below EXACTLY (even the spacing) and you should use the tab escape sequence to get the indentation.

```
Please enter a radius: 2
Here are the calculations for a circle with radius 2.0:
        Diameter:  4.0
        Circumference:  12.5663704
        Area:  12.5663704

Please enter a radius: 3
Here are the calculations for a circle with radius 3.0:
        Diameter:  6.0
        Circumference:  18.849555600000002
        Area:  28.2743334
```

11 CHECK FOR EVEN/ODD

3. Add this comment to your script file. "Parity" means "even or oddness" of a number. Even numbers are divisible by 2. Odd are not.

   ```
   #Determine the parity of an integer.
   ```

4. Ask the user for a number and save his/her response as an integer called x.

5. Use an if-statement to determine whether x is even or odd. HINT: Use the remainder operator to do this. Every even integer has a remainder of what when divided by 2? How about odd integers?

5. Example output is below. You should not use any functions from the Math package to do this. Your output should match mine below EXACTLY (even the spacing).

   ```
   Enter an integer: 11
   You entered 11 which is odd.

   Enter an integer: 16
   You entered 16 which is even.
   ```

12 TABLE OF SQUARES AND CUBES

1. Add this comment to your script file. "Parity" means "even or oddness" of a number. Even numbers are divisible by 2. Odd are not.

   ```
   #Create a table of squares and cubes.
   ```

2. Print out a table of squares and cubes of the numbers from 0 to 5. Print the results in a nicely formatted table, using necessary escape sequences.

   ```
   number  square  cube
   0       0       0
   1       1       1
   2       4       8
   3       9       27
   4       16      64
   ```

SUBMISSION INFO

TO GET CREDIT FOR THIS LAB, UPLOAD THESE 3 DOCUMENTS TO THE SUBMISSION AREA.

All of these should be in your LabCh2 folder if you followed the directions of this lab.

- Lab Ch 2.pdf (should have your guess work)
- LabCh2InteractiveMode_YourLastName.py (created in part 8)
- LabCh2_FirstPythonScript_YourLastName.py (has code from 9-12)

QUIZ INFO

ONCE YOU UPLOAD THE LAB, A LAB QUIZ WILL APPEAR IN THE SAME FOLDER WHICH YOU CAN TAKE TWICE.