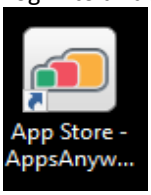
Tutorial Week 2

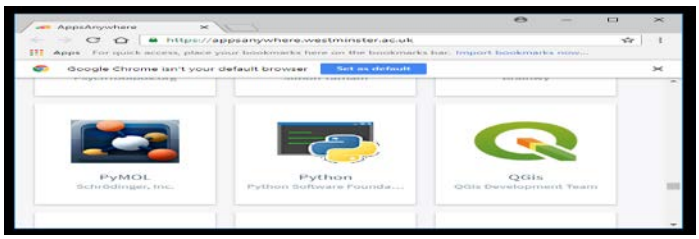
Aims:

- Launch IDLE on the lab PCs.
- Review lecture 1 content
- Use built-in functions - print(), int(), str() and input()
- Translate your Pseudocode into Python programs (Lecture 2)
- Experiment with Python Exception Handling

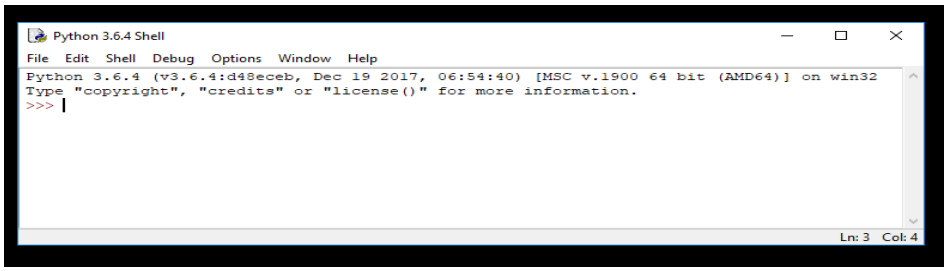
1. **Launching IDLE**
   a) Log in to a Lab PC and then click on the AppsAnywhere icon.



   b) Then scroll down, and click on the Python icon.

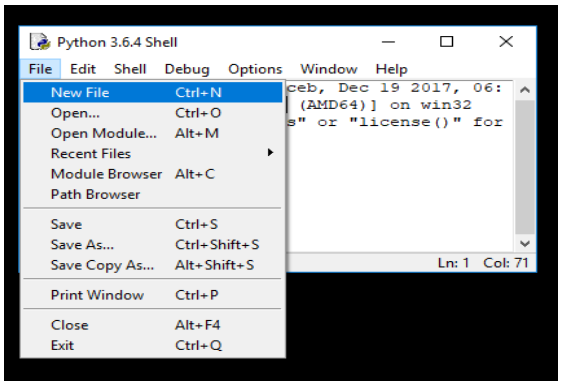


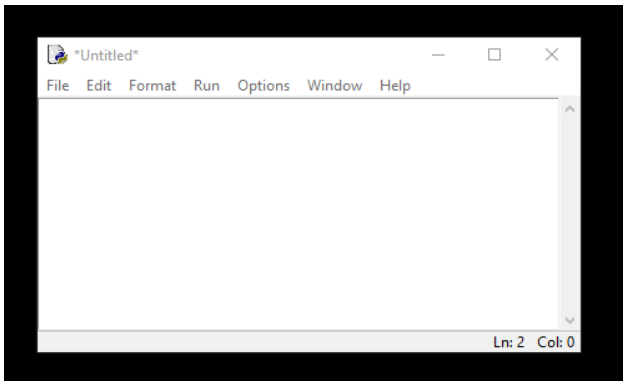
   c) When you launch Python / IDLE the Python Shell window will display and will look like this:



This window lets us experiment with Python commands and also lets us open a program editing window.

2. To write Python programs we need to start IDLE's editor. From the Python Shell window, select **New File** from the **File** menu. You will see a window entitled "Untitled".





- From the **File** menu, select **Save As**, and select a folder to save your Python program file. **You need to save to your H drive.** In the **File name:** text box, type: test1.py. Then click on the **Save** button.
- You will then see a blank editor window ready for you to type in the following Python program.

3. Review of Lecture 1 - Type in your solutions to the lecture questions 4 and 5.
   a) Lecture 1 question 4 - Write the code to put 4 into a variable called 'item1'. Then put 6 into a variable called 'item2'. Then write an instruction to add the two variable values together and put the answer into a variable 'item3'.

   b) Lecture 2 question 5 - A meal costs £56. Write the code to set 56 into a variable. Then multiply whatever is in the variable by 1/10 to work out the 10% tip (store the answer in a variable).

4. Arithmetic Operators
   a) Write down the output of the following program.

```
num = 5
print(num)              #Output?
num = num + 2
print(num)              #Output?
num = num // 3 * 6
print(num)              #Output?
print(7 + 15 % 4)       #Output?
num = 24 // 3 // 4
print(num)              #Output?
```

   b) Then create, save and run the above program.
   Did you get the right values for all of them? If you got any wrong, why was it?

5. **Built-in Functions -** print(), int(), str()
   a) Create, save and run the following program.

```
a = "Hello out there."
print(a)
b = 'Hello again.'
print(b)
```

   Then modify the program so that it concatenates (joins) variables a and b and prints the result.

   b) Why would the following produce an error?

```
total = 10
greet =  'Hello'
both = total + greet
print(both)
```

   - Note that you cannot concatenate a string and an integer.
   - Use the built-in str() function (converts to string) to fix the error.

   c) What is the output of the following program? Test the program to check your answer.

```
a = '10'
b = '99'
c = a + b
print (c)
c = int(c)
print (c)
```

   d) Modify the above program so that it prints out the value **109**.

6. **Keyboard input**
   a) Try the following program:

```
name = input('Please enter your name: \n')
print('Hello', name)
```

- Remember, `\n` within quote marks forces a new line to be printed.
- When you run the program, you should see the "Please enter your name:" message in the shell window. Type in your name, followed by the ENTER key. The program will greet you.

   b) Extend the program to get the user's age, and print out "your age is" (with the response).

7. How would you print the following so that it displays as shown: `test\test2\answers.txt`

8. Type the following. Check the differences between the three print statements.

```
the_text = input('Enter some text.\n')    # get some text!

#print - version 1
print('This is what you entered: ')
print(the_text)

#print - version 2
print('This is what you entered:', the_text)

#print - version 3 - To supress printing of a new line, use end=' '
print('This is what you entered:', end=' ')
print(the_text)
```

9. What will be displayed by the following code? Type and run the code to check your answer.

```
print("A", end = ' ')
print("B", end = ' ')
print("C", end = ' ')
print("D", end = ' ')
```

10. The following links to your solutions created for the **Lecture 2 Self-check Questions (1-5).**

Lecture 2 Question 1. Write the program to get and print the number of pets the user has.

Lecture 2 Question 2. Write the pseudocode to put zero into variable `running_total.` Then write separate instructions to add the following numbers onto what is in the variable, adding one number at a time 5, 8, 2, 3. Print `running_total.`

Lecture 2 Question 3. Write the Python program using the pseudocode shown.

```
total <- 0
INPUT num_1
INPUT num_2
total <- num_1 + num_2
```

Lecture 2 Question 4. Write the Python program using the pseudocode shown.

```
INPUT cost_of_item
INPUT cash_paid (e.g., 10 for £10)
CALCULATE change
```

Lecture 2 Question 5. Program to calculate the average of three numbers (pseudocode):

```
INPUT num_1
INPUT num_2
```

```
INPUT num_3
average <- (num_1 + num_2 + num_3) / 3
PRINT average
```

11. Temperature Program (part 1) – This will be extended in a later tutorial. Write a program that will convert a Centigrade temperature (c) entered as input into Fahrenheit (f) using the formula:

    f = (9/5)* c + 32

12. Write a program to calculate the volume of a box. Enter values for the length, height and width.

13. Write a program that changes meters (m) to centimeters (c). The program should allow you to enter in the number of meters and then print out the number of centimeters.

14. **Exception Handling in Python**

    In this example, we ask the user to enter a number using the input() method and cast the string input to an integer.

    ```
    n = int(input("Please enter a number: "))
    ```

    Type in the code, run it and enter 10.5 instead of a valid integer. If the input entered cannot be converted (cast) to an integer it will generate a ValueError.

    ValueError: invalid literal for int() with base 10: '10.5'

    With the aid of exception handling, we can write robust code for reading an integer from input:

    ```
    n = input("Please enter an integer: ")
    try:
      n = int(n)
    except ValueError:
      print("Requires a valid integer!")
    ```

    Example: The program accepts a value from the user. In the *try* clause the value in *n* is converted (cast) to an integer. If an exception occurs (the value cannot be converted to an integer) the *except* clause will be executed. The error, in this case a ValueError, has to match the name after except.

15. Test what type of error the following will produce.

    ```
    x = 2/0
    ```

    Then create a specific try except construct that will give the message 'Cannot divide by zero' when appropriate.