

Mawlana Bhashani Science and Technology University



Lab Report

Lab Report No: 01

Lab Report Name : Introduction to Python

Course code: ICT-3208

Course title: Computer Network

Date of Performance: 7-01-2021

Date of Submission: 11-01-2021

Submitted by

Name: Md Sohanur Islam

ID: IT-18011

3rd year 2nd semester

Session: 2017-2018

Dept. of ICT

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Theory:

Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and scientific applications.

Setup of Python Environment :

STEP 1:

Open Eclipse and setup a correct access to Internet (This is required only in RMIT network). In order to set up Manual Proxy follow the instructions (see also figure 1):

- a. Go to Windows > Preferences > General > Network Connections.
- b. Change Active Provider to Manual.
- c. Input proxy details, including username/password if required.
 - Host: proxy.rmit.edu.au
 - Port: 8080
 - Username/password: No required
- d. Clear SOCKS proxy.
- e. Restart Eclipse.

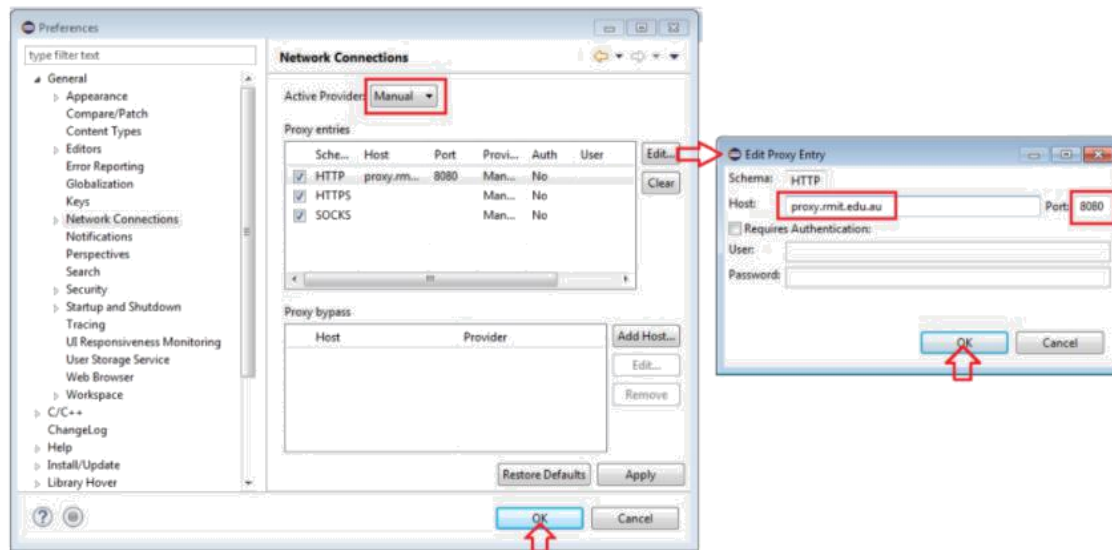


Figure 3-1. Eclipse setup for Internet.

STEP 2:

Installing python environment using Eclipse Graphical Interface1.

- a. To install PyDev and PyDev Extensions using the Eclipse Update Manager, you need to use the Help > Install New Software... menu (note that in older versions, this would be the 'Find and Install' menu) as shown in the following figure:

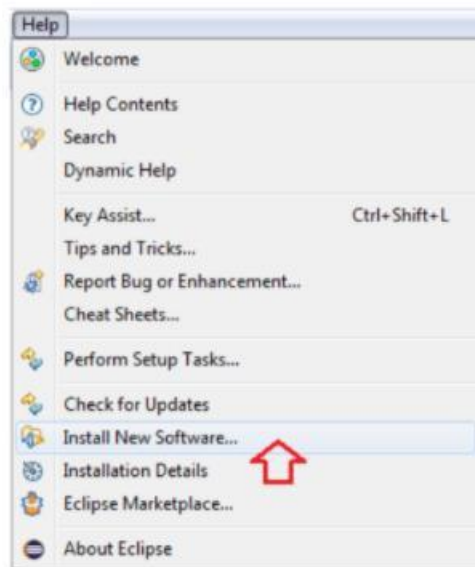


Figure 3-2. Step 2.

- b. . In the next screen, add the update site(s) you want to work with (see the figure below). The available update sites are :

<http://www.pydev.org/updates>

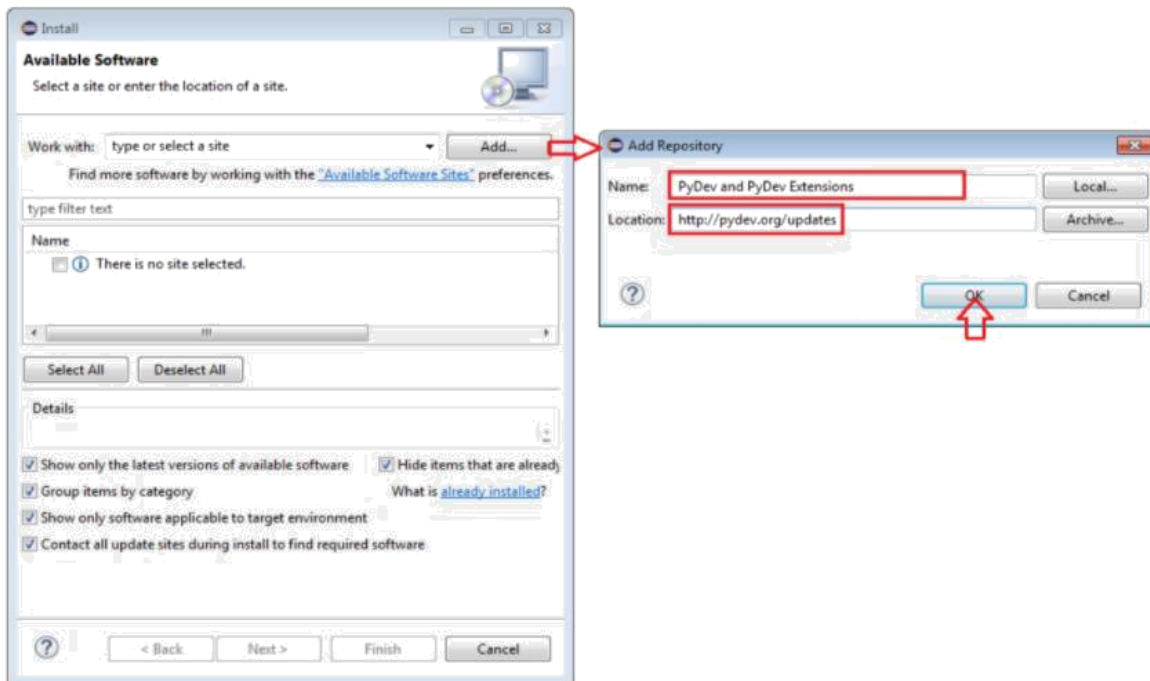


Figure 3-3. Set up Python on Eclipse.

- c. After entering the update sites, select the update site you entered or select "All available sites" and add a filter for PyDev, so that it shows the contents of all the update sites that have PyDev, then select what you want to install and click 'Next' (See the figure below):

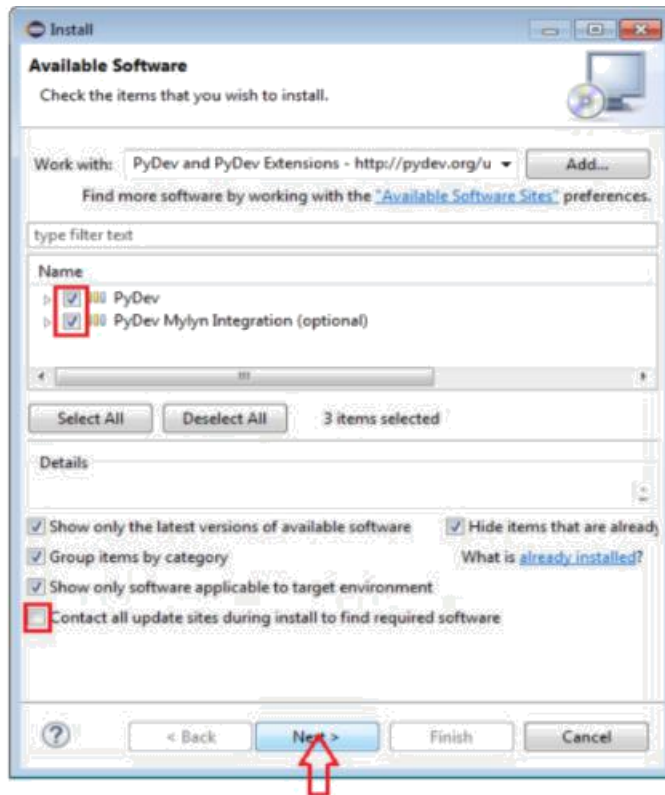


Figure 3-4. Set up Python on Eclipse.

- d. Then, UNCHECK the 'Contact all update sites during install to find required software' and press 'NEXT' again to confirm your selection(see the figure below):

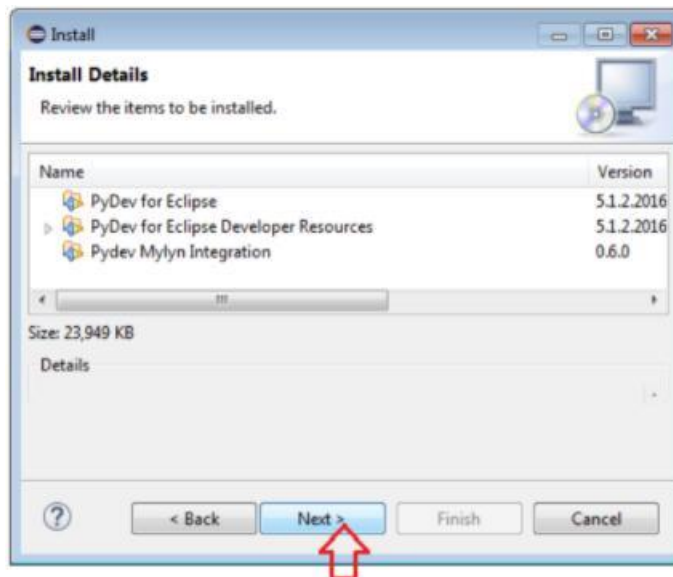


Figure 3-5. Set up Python on Eclipse.

- e. And finally, read the license agreement if you accept, select the accept radio button and click 'Finish'(see the figure below):

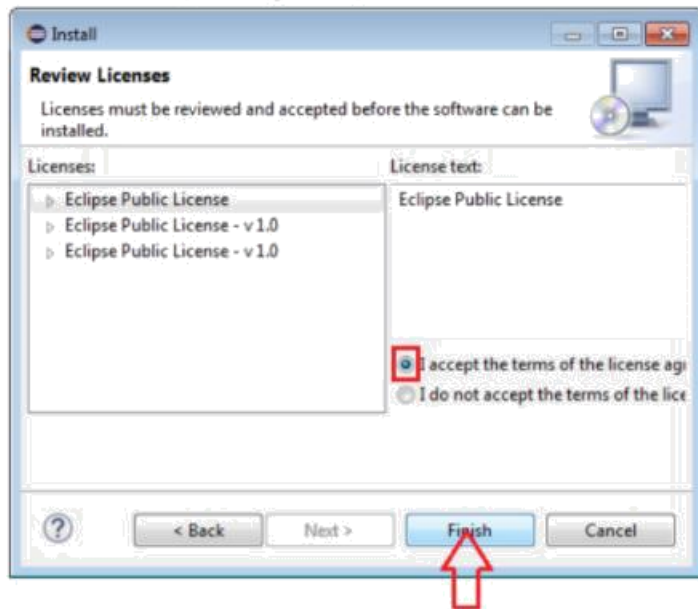


Figure 3-6. Set up Python on Eclipse.

- f. At the point, Eclipse should automatically download the plugin contents and present you to a dialog asking if you want to restart (to which you should say yes).

STEP 2: Checking the installation: You can verify if it is correctly installed going to the menu 'window> preferences' and checking if there is a PyDev item under that (see Figure 7). After that eclipse will display the graphical interface for python perspective, the main components are (see Figure 8):

- Project space is the section where all your python projects are visualized,
- Project Editor is the section where python scripts can be edited,
- Console allows the visualization of results father running a python script,
- Run bottom allows to run a python script,

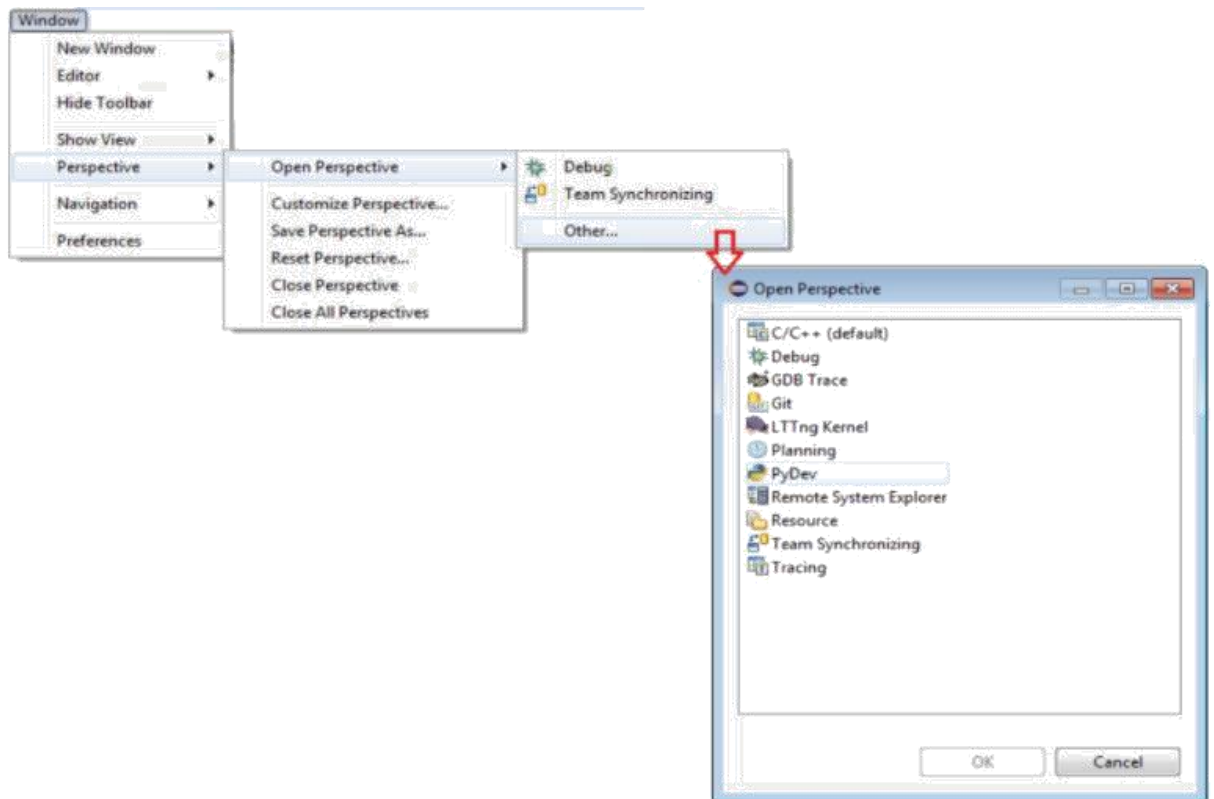


Figure 3-7. Python perspective in Eclipse.

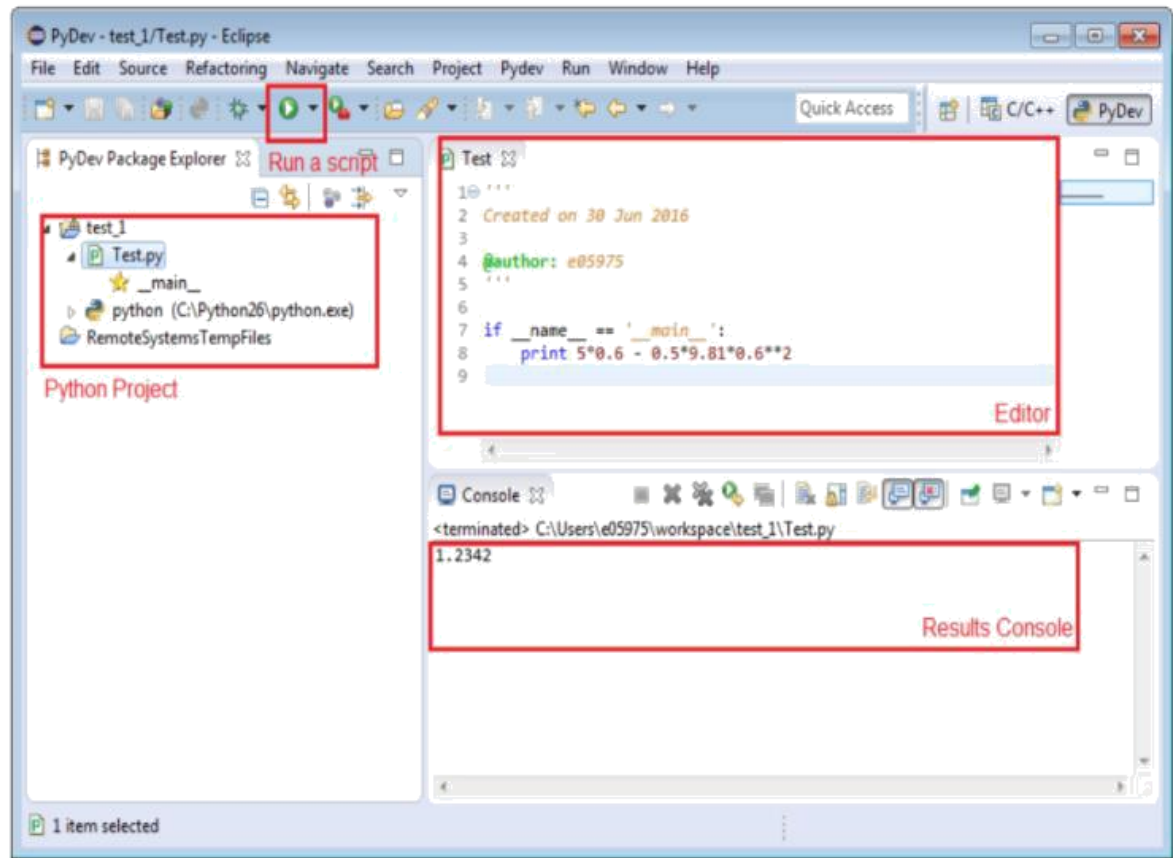


Figure 3-8. Python perspective environment.

Exercises Section 4.1:

Basics of python and programing Exercise 4.1.1: Create a python project.

Answer:

PC

Create Project

×

Location: C:\Users\HP\PycharmProjects\Lab1

▼ Python Interpreter: New Virtualenv environment

☒ New environment using

Virtualenv

Location: C:\Users\HP\PycharmProjects\Lab1\venv

Base interpreter: C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe

☐ Inherit global site-packages

☐ Make available to all projects

☐ Existing interpreter

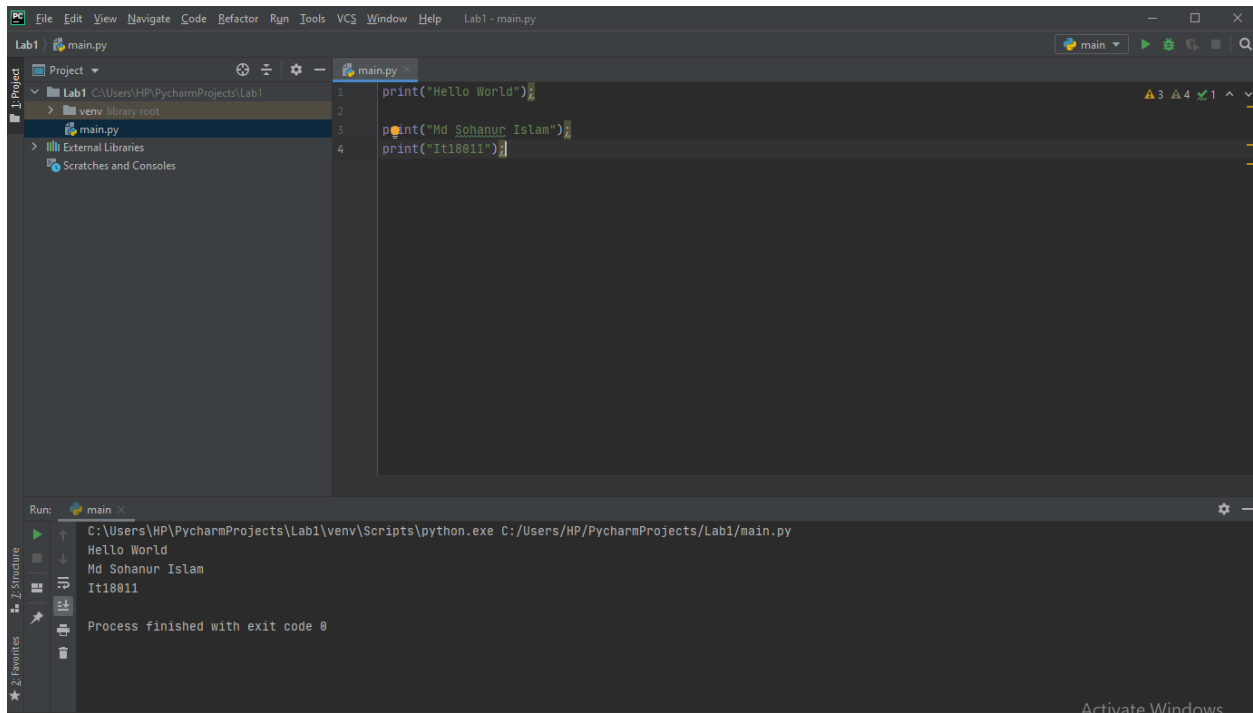
Interpreter: <No interpreter>

☒ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Create

Exercise: 4.1.2 : Write a Hello World Program.



The screenshot displays the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the Project view with a tree structure containing Lab1, venv, library root, main.py, External Libraries, and Scratches and Consoles. The main editor window shows the code in main.py:

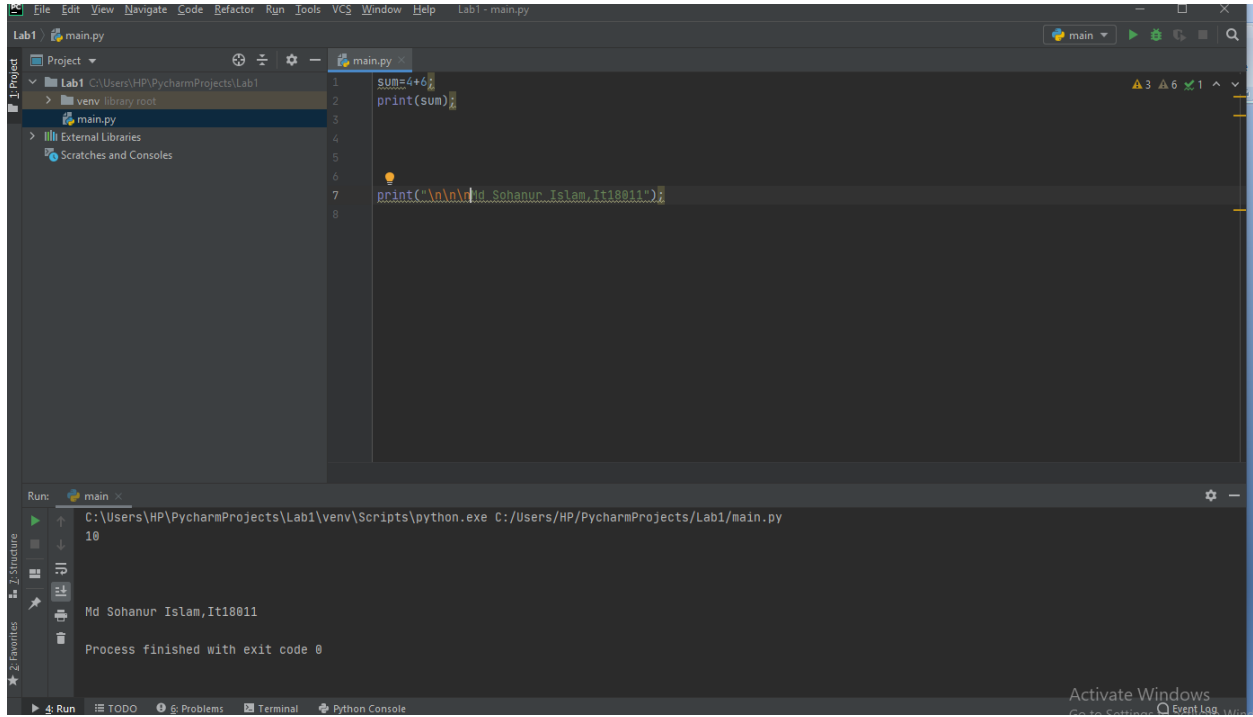
```
1 print("Hello World")
2
3 print("Md Sohanur Islam")
4 print("It18011")
```

The Run window at the bottom shows the execution command: `C:\Users\HP\PycharmProjects\Lab1\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/Lab1/main.py`. The output of the program is displayed as:

```
Hello World
Md Sohanur Islam
It18011
```

Below the output, it states "Process finished with exit code 0". The bottom right corner of the window has a watermark that says "Activate Windows".

Exercise 4.1.3: Compute 4+6



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `main.py` with the following code:

```
1 sum=4+6
2 print(sum)
3
4
5
6
7 print("\n\nMd_Sohanur_Islam,It18011")
8
```

The Run tool window at the bottom shows the execution output:

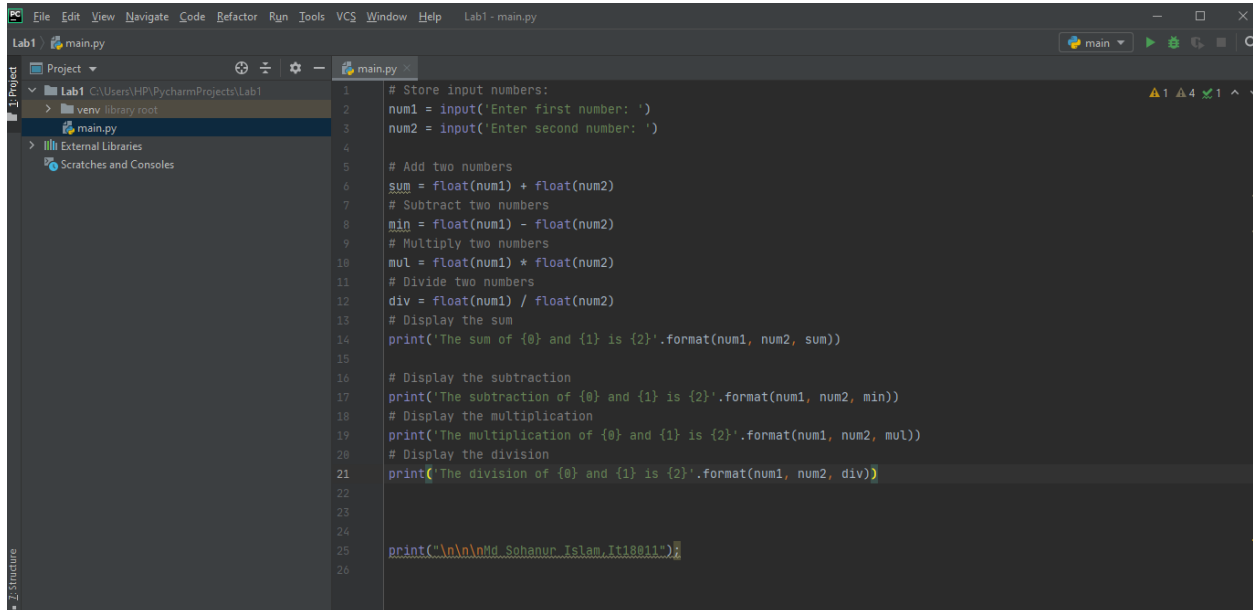
```
Run: main
C:\Users\HP\PycharmProjects\Lab1\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/Lab1/main.py
10

Md Sohanur Islam,It18011

Process finished with exit code 0
```

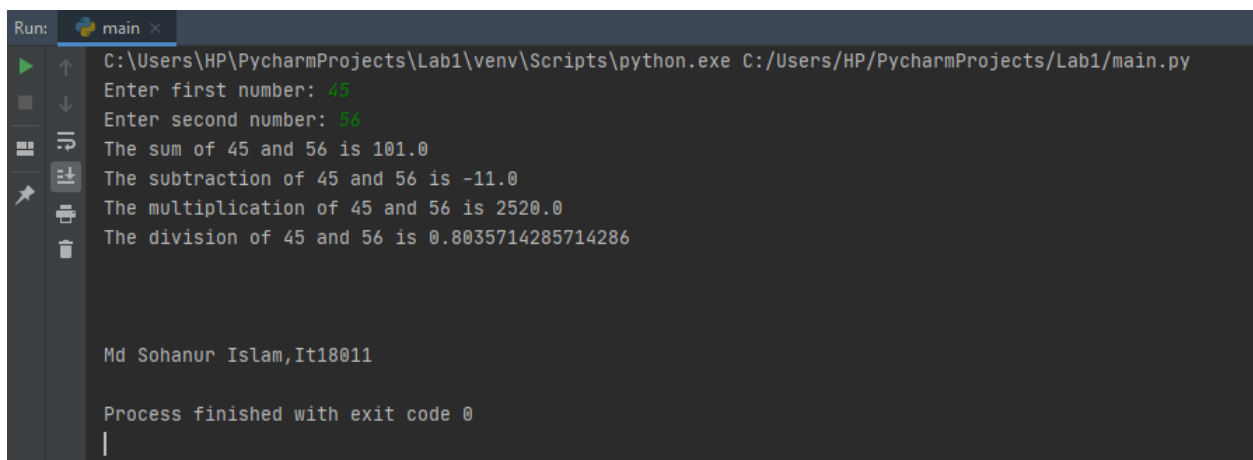
The status bar at the bottom indicates the Run tab is active, along with icons for TODO, Problems, Terminal, and Python Console. An "Activate Windows" watermark is visible in the bottom right corner.

Exercise 4.1.4: Type in program text.



The screenshot shows the PyCharm IDE with a project named 'Lab1'. The file 'main.py' is open, displaying the following Python code:

```
1 # Store input numbers:
2 num1 = input('Enter first number: ')
3 num2 = input('Enter second number: ')
4
5 # Add two numbers
6 sum = float(num1) + float(num2)
7 # Subtract two numbers
8 min = float(num1) - float(num2)
9 # Multiply two numbers
10 mul = float(num1) * float(num2)
11 # Divide two numbers
12 div = float(num1) / float(num2)
13 # Display the sum
14 print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
15
16 # Display the subtraction
17 print('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))
18 # Display the multiplication
19 print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))
20 # Display the division
21 print('The division of {0} and {1} is {2}'.format(num1, num2, div))
22
23
24
25 print("\n\nMd.Sohanur Islam,It18011")
26
```



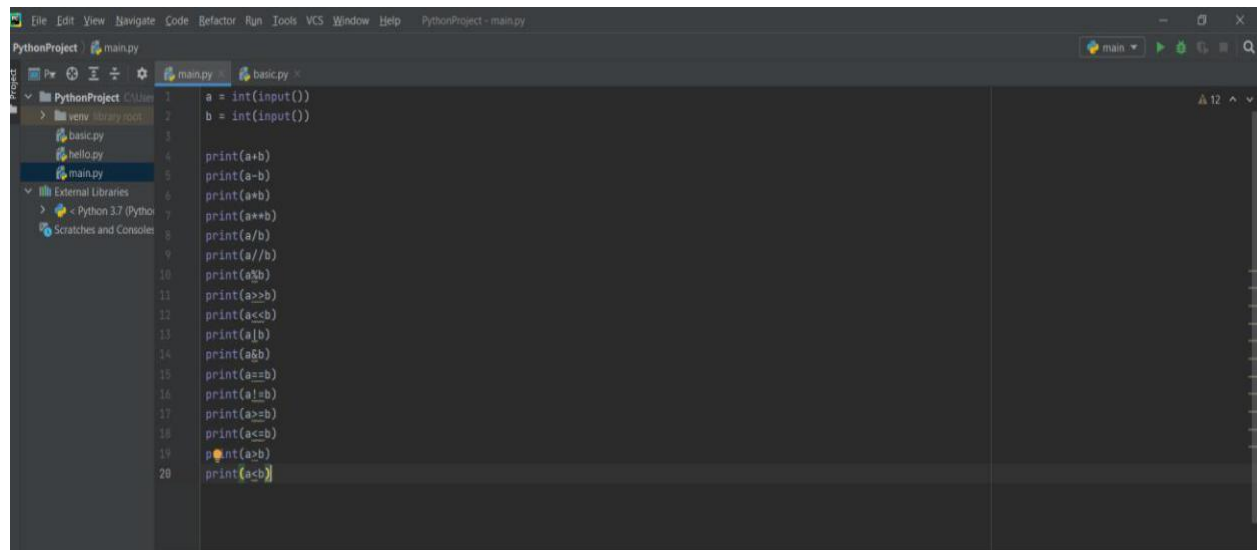
The screenshot shows the PyCharm Run console with the output of the program. The command prompt is 'C:\Users\HP\PycharmProjects\Lab1\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/Lab1/main.py'. The input values are 45 and 56. The output shows the sum, subtraction, multiplication, and division of these numbers, followed by a blank line and the author's name and ID.

```
Run: main x
C:\Users\HP\PycharmProjects\Lab1\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/Lab1/main.py
Enter first number: 45
Enter second number: 56
The sum of 45 and 56 is 101.0
The subtraction of 45 and 56 is -11.0
The multiplication of 45 and 56 is 2520.0
The division of 45 and 56 is 0.8035714285714286

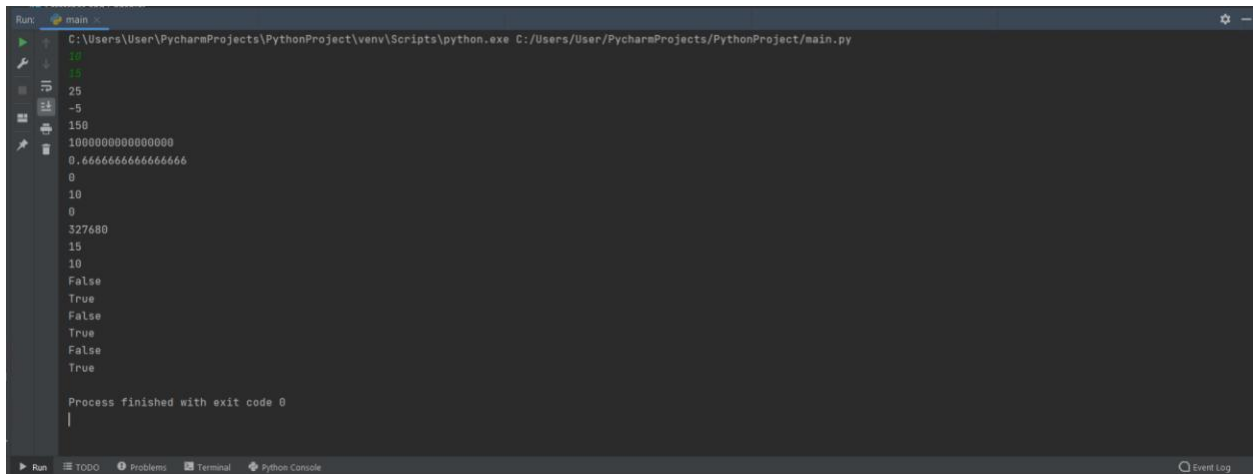
Md Sohanur Islam,It18011

Process finished with exit code 0
|
```

Section 4.1: Create and run basic example.

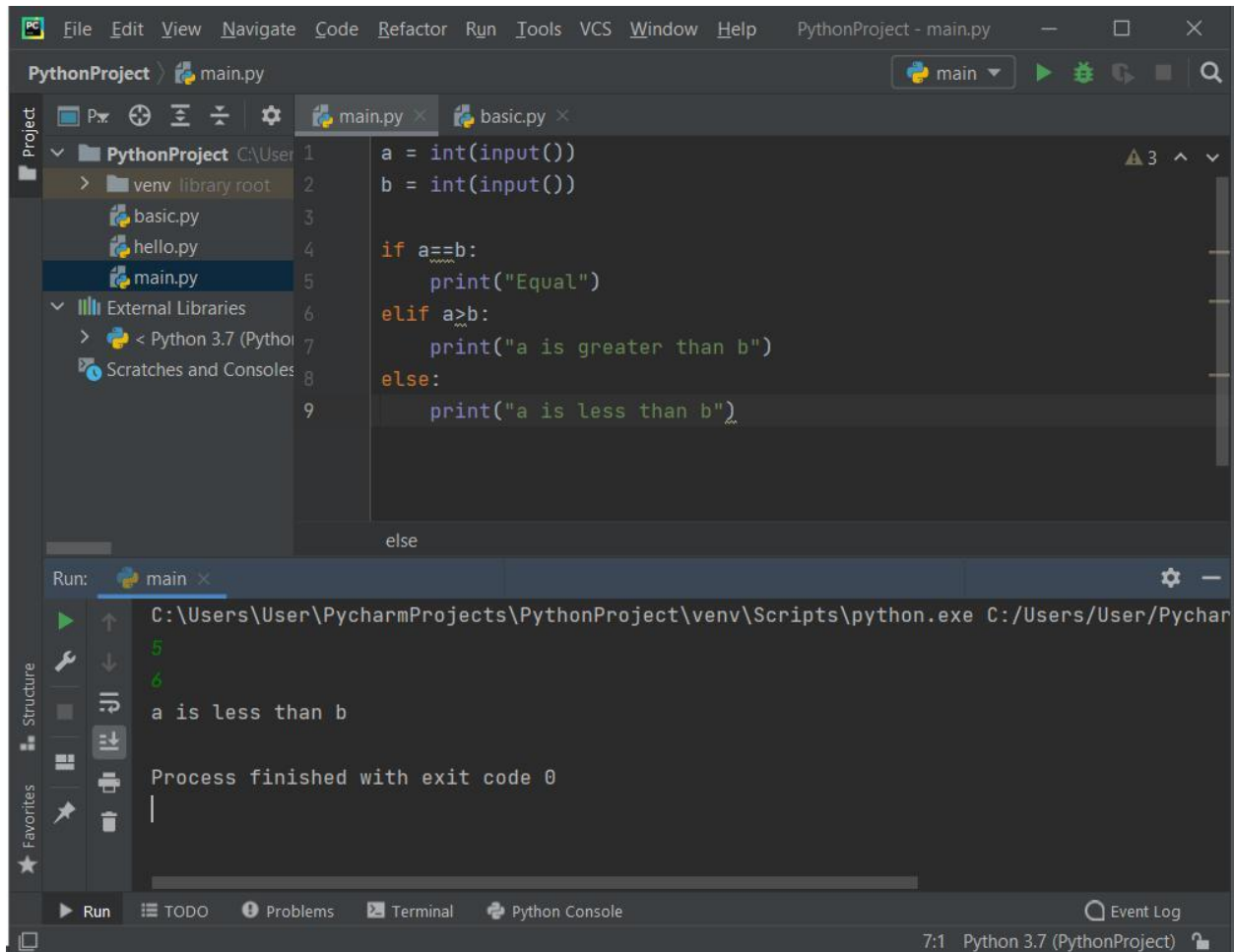


```
1 a = int(input())
2 b = int(input())
3
4 print(a+b)
5 print(a-b)
6 print(a*b)
7 print(a**b)
8 print(a/b)
9 print(a//b)
10 print(a%b)
11 print(a>b)
12 print(a<b)
13 print(a>=b)
14 print(a<=b)
15 print(a==b)
16 print(a!=b)
17 print(a>=b)
18 print(a<=b)
19 print(a>b)
20 print(a<b)
```



```
Run: main
C:\Users\User\PycharmProjects\PythonProject\venv\Scripts\python.exe C:/Users/User/PycharmProjects/PythonProject/main.py
10
15
25
-5
150
1000000000000000
0.6666666666666666
0
10
0
327680
15
10
False
True
False
True
False
True
True
Process finished with exit code 0
```

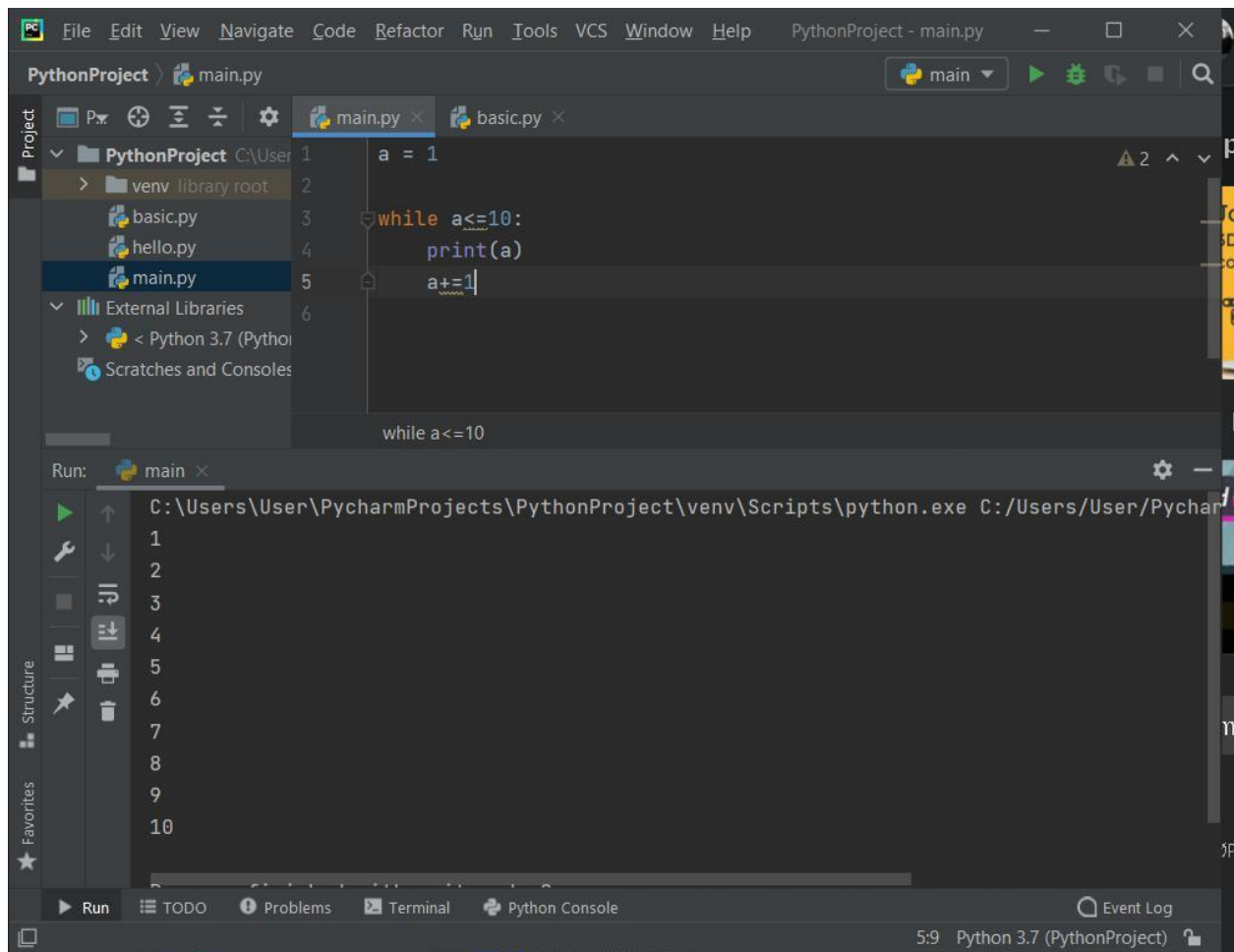
Exercise 4.2.2: The if statement:



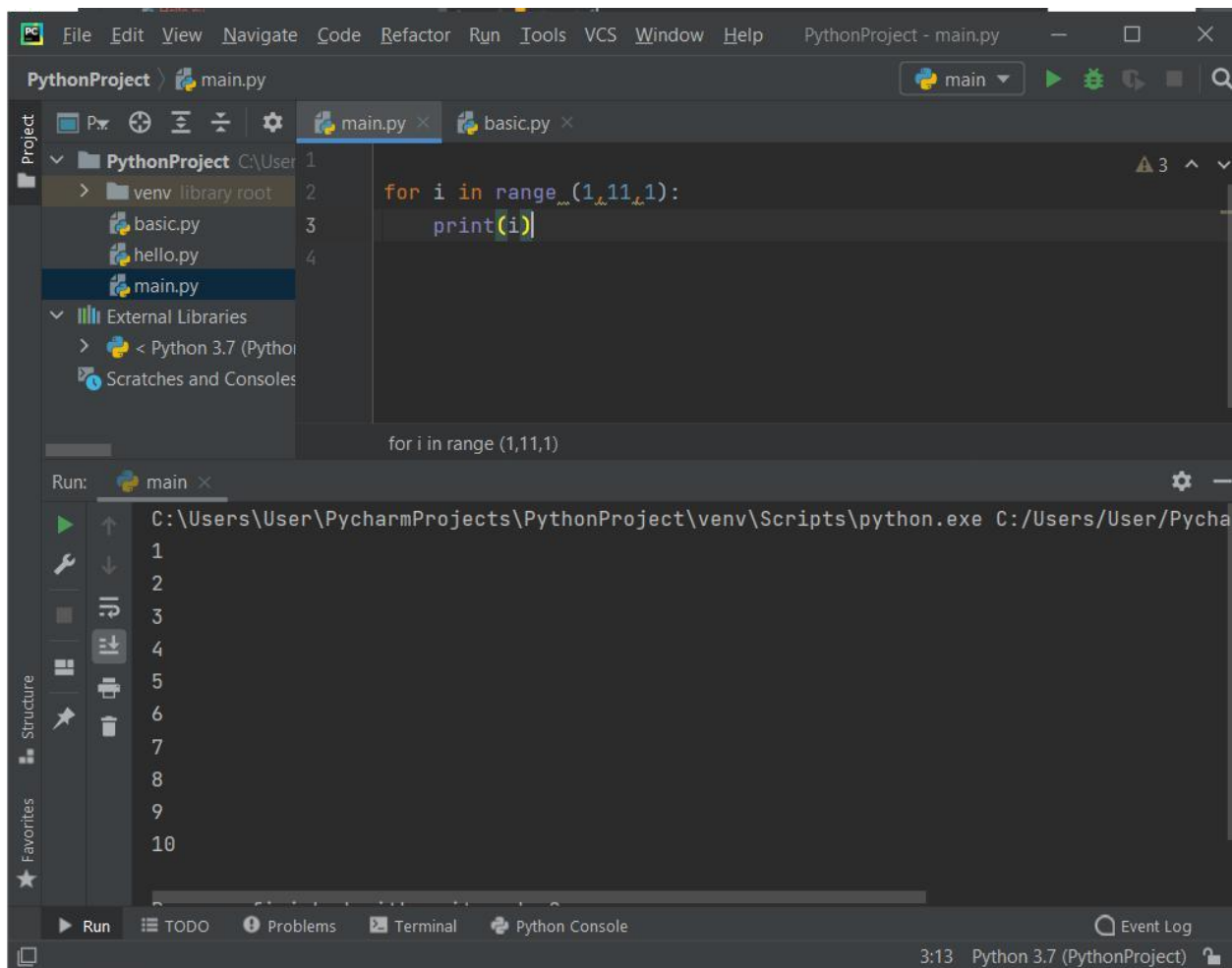
```
PythonProject - main.py
main.py
1 a = int(input())
2 b = int(input())
3
4 if a==b:
5     print("Equal")
6 elif a>b:
7     print("a is greater than b")
8 else:
9     print("a is less than b")
else
```

```
Run: main
C:\Users\User\PycharmProjects\PythonProject\venv\Scripts\python.exe C:/Users/User/Pychar
5
6
a is less than b
Process finished with exit code 0
```

Exercise 4.2.3: The while Statement



Exercise 4.2.4: The for Statement



Conclusion:

Python is a language that is remarkably easy to learn, and it can be used as a stepping stone into other programming languages and frameworks. If you're an absolute beginner and this is your first time working with any type of coding language, that's something you definitely want. Python is widely used, including by a number of big companies like Google, Pinterest, Instagram, Disney, Yahoo!, Nokia, IBM, and many others. The Raspberry Pi – which is a mini computer and DIY lover's dream – relies on Python as it's main programming language too. You're probably wondering why either of these things matter, and that's because once you learn Python, you'll never have a shortage of ways to utilize the skill. Not to mention, since a lot of big companies rely on the language, you can make good money as a Python developer.

- 1) Python can be used to develop prototypes, and quickly because it is so easy to work with and read.
- 2) Most automation, data mining, and big data platforms rely on Python. This is because it is the ideal language to work with for general purpose tasks.
- 3) Python allows for a more productive coding environment than massive languages like C# and Java. Experienced coders tend to stay more organized and productive when working with Python, as well.

