



# **Mawlana Bhashani Science and Technology University**

Santosh, Tangail-1902.

## **Lab Report**

### **Department of Information and Communication Technology**

**Report No:** 01

**Report Name:** Introduction to Python

**Course Title:** Network Planning and designing Lab.

**Course Code:** ICT-3208

Submitted By	Submitted To
Name: <b>Zafrul Hasan Khan &amp; Hasibul Islam Imon</b> ID: <b>IT-18003 &amp; IT-18047</b> Session: 2017-18 3rd Year 2nd Semester Dept. of Information & Communication Technology, MBSTU.	<b>Nazrul Islam</b> <b>Assistant Professor,</b> Dept. of Information & Communication Technology, MBSTU.

**Objectives:** The main objectives of this lab how to Setup python environment for programing, to Learn the basics of python, to Create and run basic examples using python.

**Theory :** Python is a simple and minimalistic language. This pseudo-code nature of Python is one of its greatest strengths. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. Python is extremely easy to get started with. Python has an extraordinarily simple syntax.

## **Methodology :**

### **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 below **figure 1**):

**a.** Go to Windows > Preferences > General > Network Connections.

**b.** Change Active Provider to Manual.

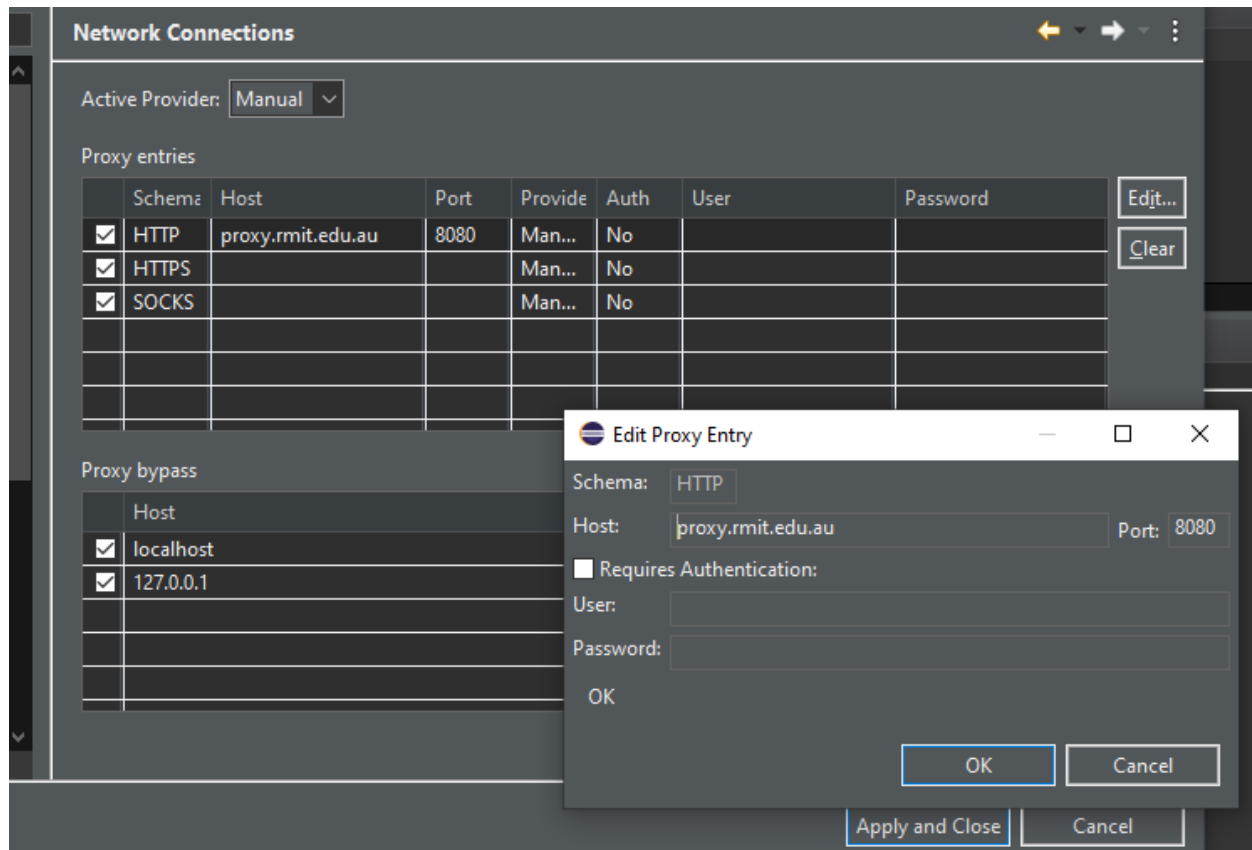
**c.** Input proxy details, including username/password if required.

# Host: proxy.rmit.edu.au

# Port: 8080

# Username/password: No required

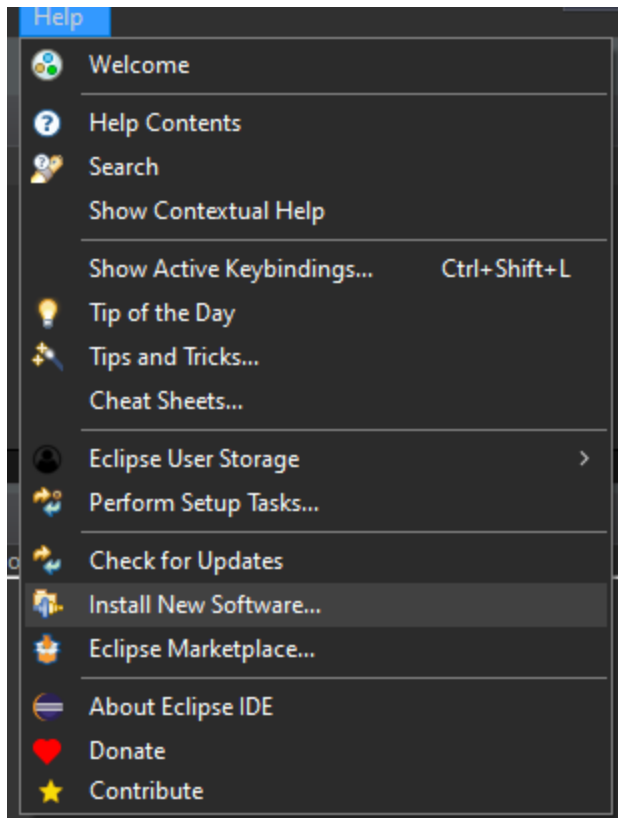
**d.** Clear SOCKS proxy. **e.** Restart Eclipse.



**Figure 1: Eclipse Setup for Internet**

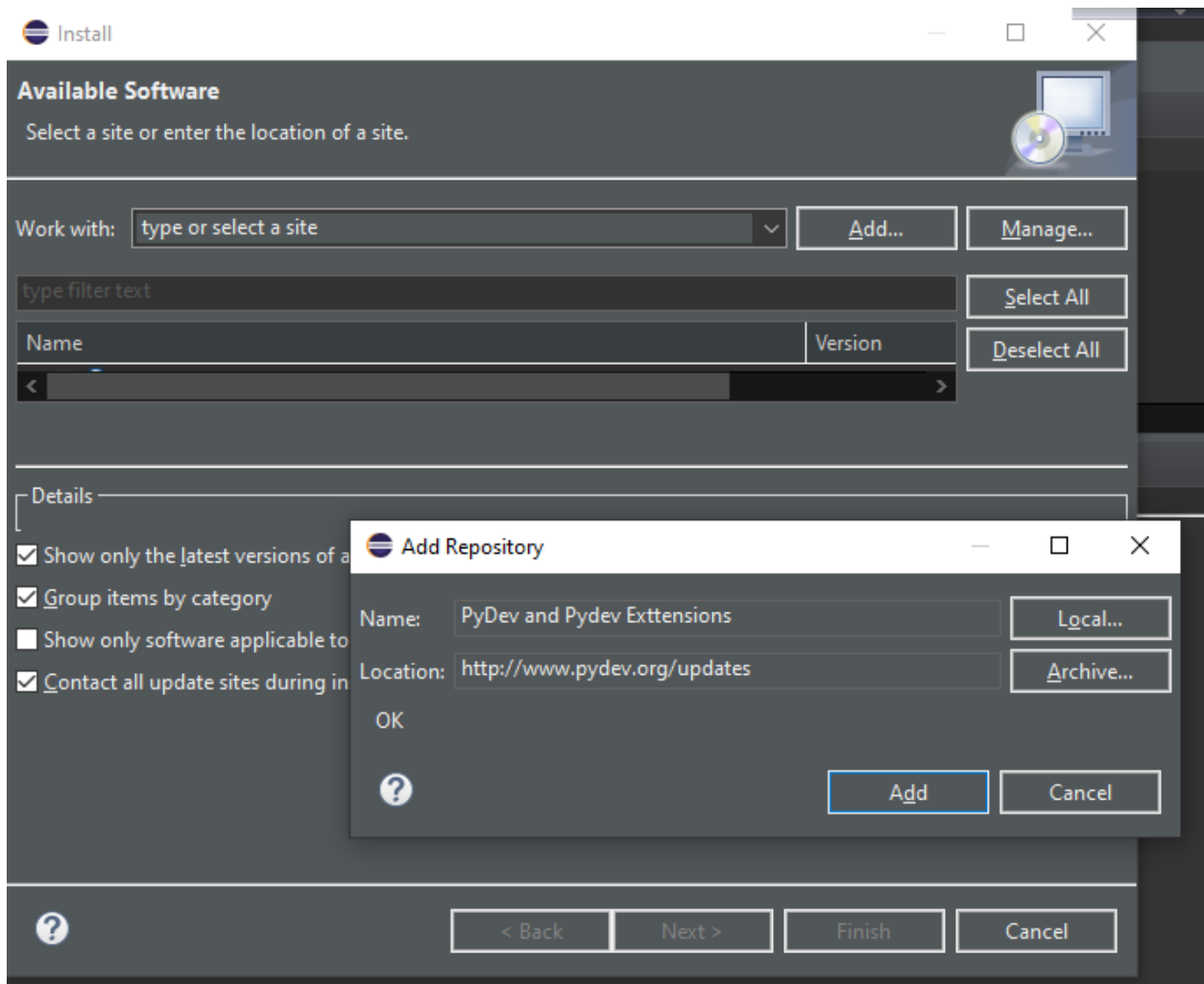
## **STEP 2:** Installing python environment using Eclipse Graphical Interface .

- 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 2**:



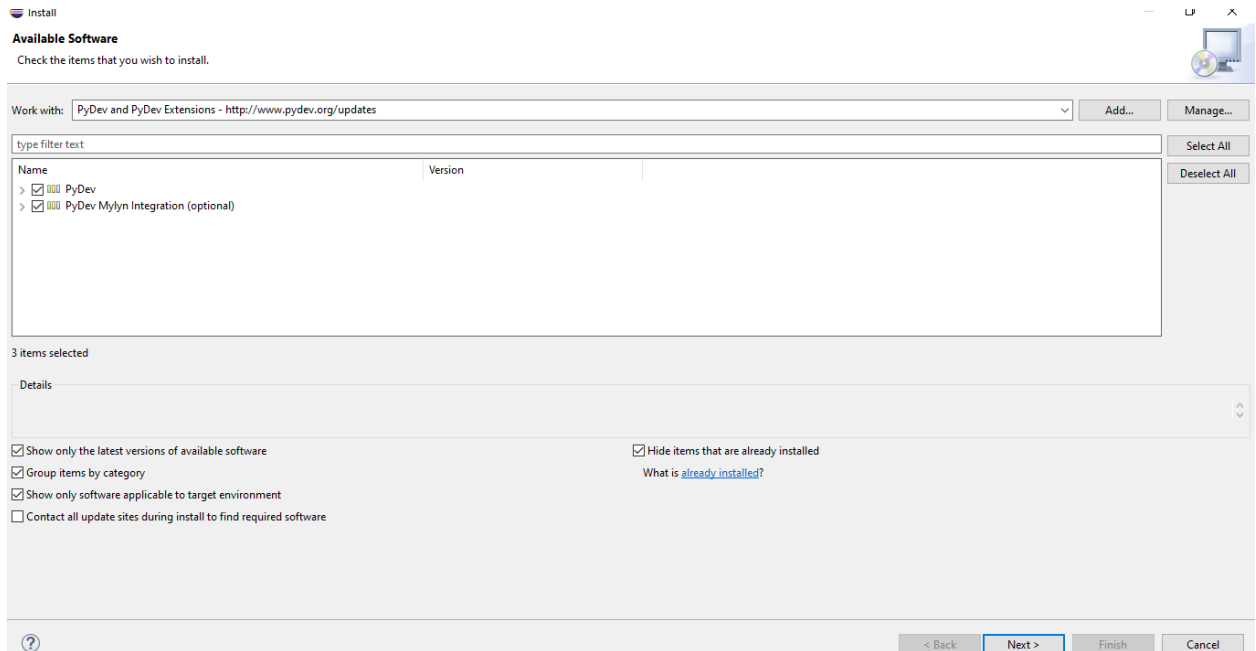
**Figure 2: setup for install.**

- b.** In the next screen, add the update site(s) you want to work with (see the figure below). The available update sites are (see **Figure 3**):
- <http://pydev.org/updates>



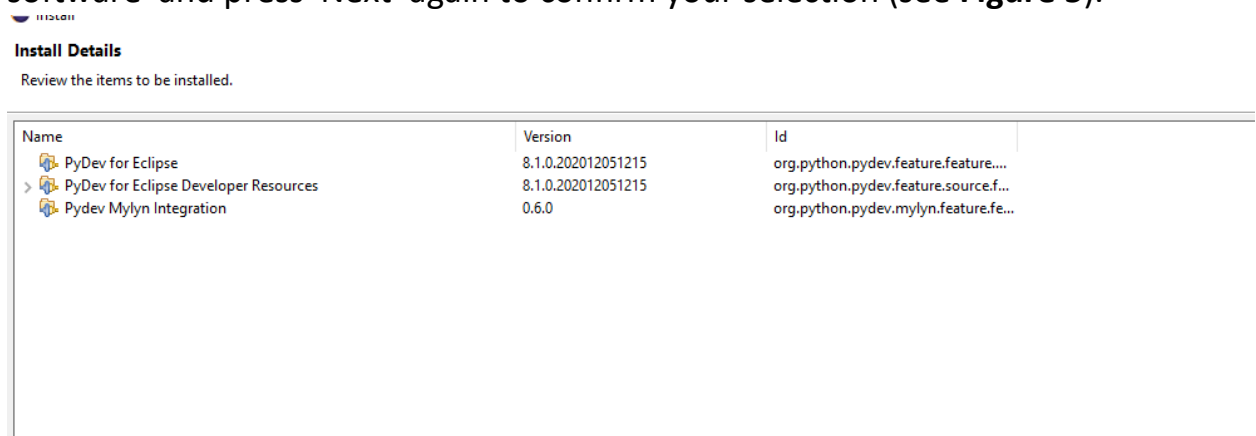
**Figure 3 : Setup Python on Eclipse**

- 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 **Figure 4**)



**Figure 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 **Figure 5**).

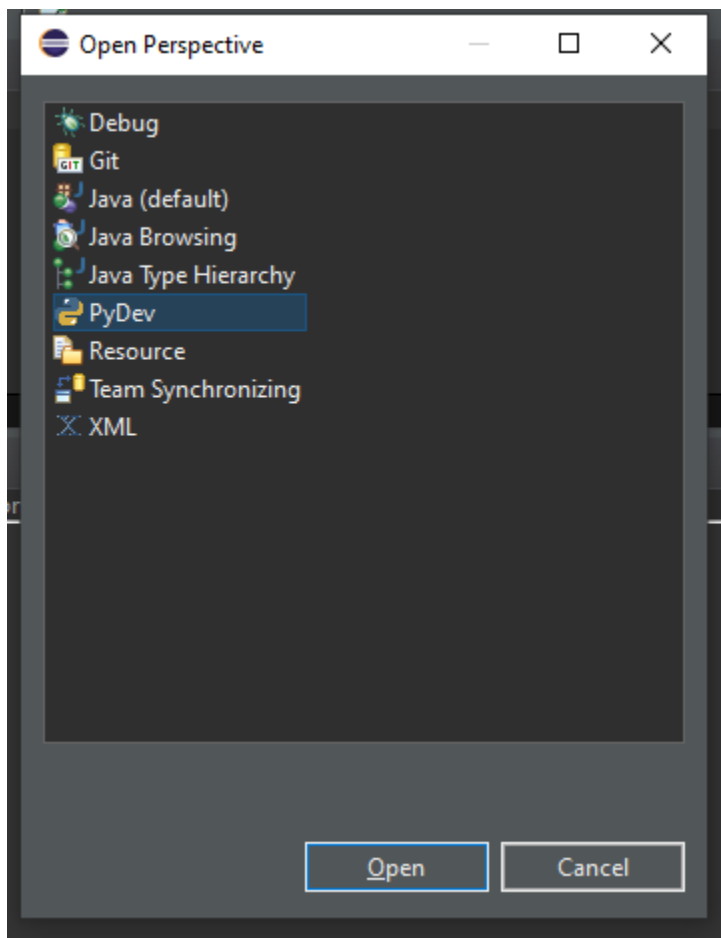


**Figure 5: Set up Python on Eclipse.**

- e. And finally, read the license agreement and if you accept, select the accept radio button and click 'Finish'.
- f. At that 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 3: 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 following Figure ):

- 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



**Figure : Python perspective in Eclipse.**

**Exercises :**

## Section 4.1:

### Basics of python and programing :

#### Exercise 4.1.1: Create a python project.

**Answer :** Create a python project, click in File > New > PyDev Project. Provide a name for the project ('NasimsProject' for the fits lab), then select the version of python to be used and select to add the project to working set as shown below:

The screenshot shows the 'PyDev Project Wizard' dialog box. The 'Project name' field is filled with 'NasimsProject'. Under 'Project contents', the 'Use default' checkbox is checked. The 'Directory' field shows 'F:\Programming code\Eclipse code\NasimsProject' with a 'Browse' button. In the 'Project type' section, 'Python' is selected with a radio button. The 'Grammar Version' dropdown is set to 'Same as interpreter'. The 'Interpreter' dropdown is set to 'Default -- currently: python'. A link 'Click here to configure an interpreter not listed.' is visible. Under 'Additional syntax validation', the option '<no additional grammars selected>' is chosen. In the 'Working sets' section, the checkbox 'Add project to working sets' is checked. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Project name: NasimsProject

Project contents:

☒ Use default

Directory: F:\Programming code\Eclipse code\NasimsProject Browse

Project type

Choose the project type

☐ Python ☒ Jython ☐ IronPython

Grammar Version

Same as interpreter

Interpreter

Default -- currently: python

[Click here to configure an interpreter not listed.](#)

Additional syntax validation: <no additional grammars selected>.

☐ Add project directory to the PYTHONPATH

☒ Create 'src' folder and add it to the PYTHONPATH

☐ Create links to existing sources (select them on the next page)

☐ Don't configure PYTHONPATH (to be done manually later on)

Working sets

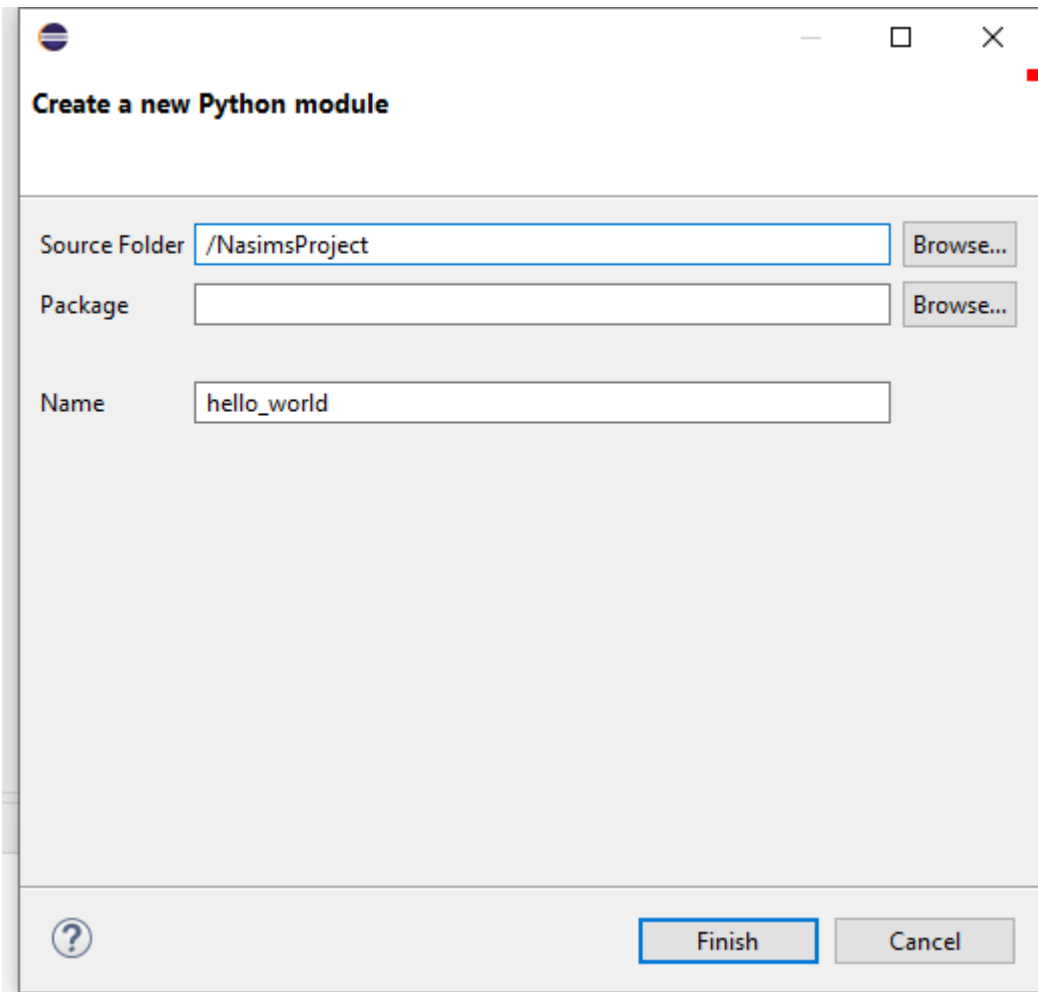
☒ Add project to working sets New...

Working sets: Select...

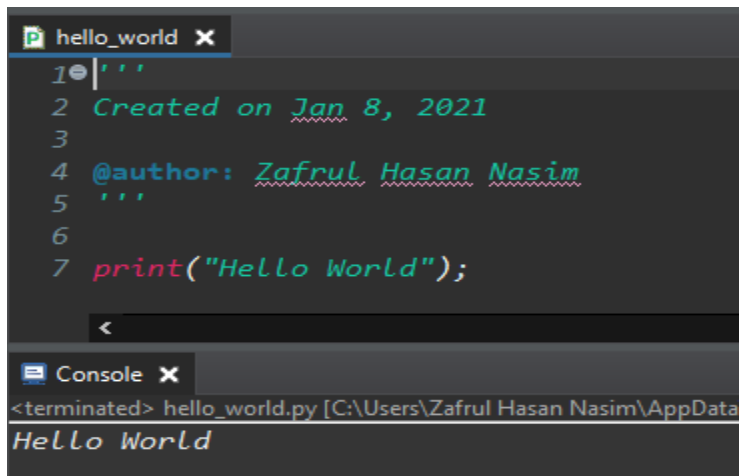
? < Back Next > Finish Cancel



Create a python script, click in **File > New > PyDev Module**. Select the folder source name. Then, provide a name for the project (Hello\_world), then select empty module or main module as shown below:



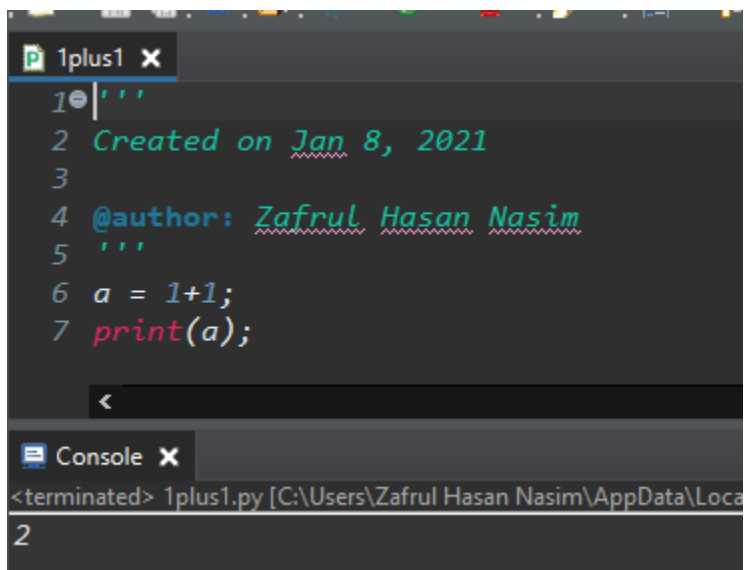
**Exercise-4.1.2 : Write a Hello World Program.**



```
hello_world x
1 '''
2 Created on Jan 8, 2021
3
4 @author: Zafrul Hasan Nasim
5 '''
6
7 print("Hello World");

<
Console x
<terminated> hello_world.py [C:\Users\Zafrul Hasan Nasim\AppData
Hello World
```

### Exercise-4.1.3: Compute 1+1



```
1plus1 x
1 '''
2 Created on Jan 8, 2021
3
4 @author: Zafrul Hasan Nasim
5 '''
6 a = 1+1;
7 print(a);

<
Console x
<terminated> 1plus1.py [C:\Users\Zafrul Hasan Nasim\AppData\Loca
2
```

### Exercise 4.1.4: Type in program text

```
*formulas-shapes X
2 Created on Jan 8, 2021
3
4 @author: Zafrul Hasan Nasim
5 '''
6 h = 5.0 # height
7 r = 1.5 # radius
8 b = 1.3
9 pi = 3.1416
10 if __name__ == '__main__':
11     area_parallelogram = h*b
12     print('The area of the parallelogram is %.3f' % area_parallelogram)
13     area_square = b**2
14     print('The area of the square is %g' % area_square)
15     area_circle = pi*r**2
16     print('The area of the circle is %.3f' % area_circle)
17     volume_cone = 1.0/3*pi*r**2*h
18     print('The volume of the cone is %.3f' % volume_cone)
19
<

Console X
<terminated> formulas-shapes.py [C:\Users\Zafrul Hasan Nasim\AppData\Local\Programs\Python\Python39\python.exe]
The area of the parallelogram is 6.500
The area of the square is 1.69
The area of the circle is 7.069
The volume of the cone is 11.781
```

## Section 4.2: Create and run basic example.

**Exercise 4.2.1:** Verify the use of the following operator. Execute the example code in python script and provide the output.

formulas-shapes hello\_world 1plus1 operator X

```
4 @author: Zafrul Hasan Nasim
5 '''
6 print("Enter the first integer value :")
7 a = int(input())
8 print("Enter the second integer value :")
9 b = int(input())
10
11 #verify the operator
12 print("plus :",a+b)
13 print("Minus",a-b)
14 print("Multiply:",a*b)
15 print("Power:",a**b)
16 print("Divide:",a/b)
17 print("Divide and floor:",a//b)
18 print("Modulo:",a%b)
19 print("Left shift:",a<<b)
20 print("Right shift:",a>>b)
21 print("Bit-Wise AND:",a&b)
22 print("Bit-wise OR:",a|b)
23 print("Bit-wise X-OR:",a^b)
24 print("Less than:",a<b)
25 print("Greater than:",a>b)
26 print("Less than or equal to :",a<=b)
27 print("Greater than or equal to :",a>=b)
28 print("equal to :",a==b)
29 print("Not equal to :",a!=b)
30
```

```
Console X
operator.py [C:\Users\Zafrul Hasan Nasim\AppData\Local\Pro
Enter the first integer value :
2
Enter the second integer value :
3
plus : 5
Minus -1
Multiply: 6
Power: 8
Divide: 0.6666666666666666
Divide and floor: 0
Modulo: 2
Left shift: 16
Right shift: 0
Bit-Wise AND: 2
Bit-wise OR: 3
Bit-wise X-OR: 1
Less than: True
Greater than: False
Less than or equal to : True
Greater than or equal to : False
equal to : False
Not equal to : True
<
```

#### Exercise 4.2.2: The if statement:

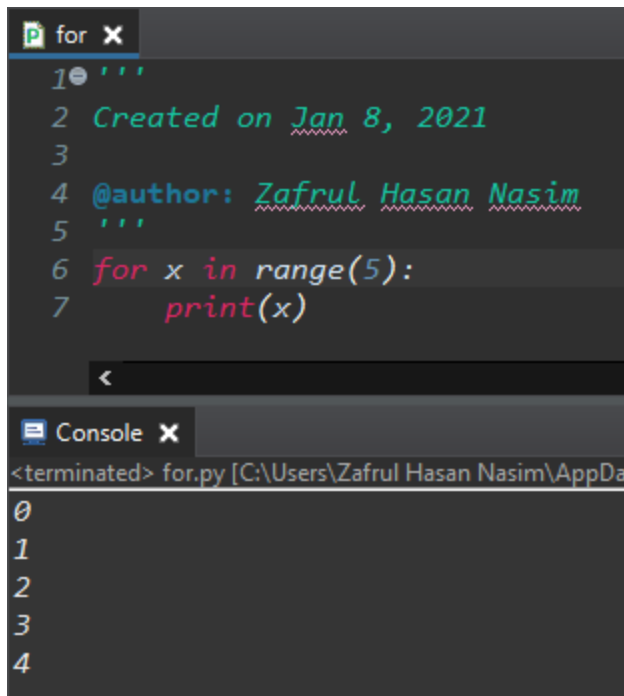
```
if x if2
1 '''
2 Created on Jan 8, 2021
3
4 @author: Zafrul Hasan Nasim
5 '''
6 print("Enter the first Integer value :")
7 x = int(input())
8 print("Enter the second integer value :")
9 y = int(input())
10
11 if(x==y):
12     print(x," is equal to ",y)
13 else:
14     print(x," is not equal to ",y)
<
Console X
<terminated> if.py [C:\Users\Zafrul Hasan Nasim\AppData\Local\Programs\Python\Python39
Enter the first Integer value :
2
Enter the second integer value :
3
2 is not equal to 3
```

### Exercise 4.2.3: The while Statement

```
while X
1 '''
2 Created on Jan 8, 2021
3
4 @author: Zafrul Hasan Nasim
5 '''
6 print("Enter the value for loop limit :")
7 x = int(input())
8
9 while(x>0):
10     print("Decresing value :",x)
11     x=x-1
<
```

```
Console X
<terminated> while.py [C:\Users\Zafrul Hasan Nasim\AppData\Local\Programs\
Enter the value for loop limit :
5
Decresing value : 5
Decresing value : 4
Decresing value : 3
Decresing value : 2
Decresing value : 1
```

#### Exercise 4.2.4: The for Statement



The image shows a screenshot of a Python IDE with two panels. The top panel, titled 'for.py', contains the following code:

```
1 '''  
2 Created on Jan 8, 2021  
3  
4 @author: Zafrul Hasan Nasim  
5 '''  
6 for x in range(5):  
7     print(x)
```

The bottom panel, titled 'Console', shows the output of the script:

```
<terminated> for.py [C:\Users\Zafrul Hasan Nasim\AppData  
0  
1  
2  
3  
4
```

**Conclusion:** From this lab , I have known that how to set up python environment for programing. I have Learnt the basics of python, to Create and run basic examples using python. Python is extremely easy to get started with. Python has an extraordinarily simple syntax. Python does not need compilation to binary. You just run the program directly from the source code. I have also known that how to set up python interpreter then Python converts the source code into an intermediate form called byte codes and then translates this into the native language of your computer and then runs it.