



# MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

Department of ICT

LAB REPORT NO-04

Course Code : ICT – 3207

Course Title : Computer Networks

**Submitted By:**

Name : Md. Al- Mamun

ID : IT-17012

Session : 2016-2017

**Submitted To:**

Nazrul Islam

*Assistant Professor, Department of ICT*

MBSTU

**Lab Report No : 04**  
**Report Name : Introduction to Python**  
**Group Member : MD.AL-MAMUN ( IT-17012 )**  
**Monir John Rakib ( IT-17006 )**

## **Introduction to Python**

### **Objective :**

- Setup python environment for programing.
- Learn the basics of python.
- Create and run basic examples using python.

### **Theory :**

Definition of Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Main Features of Python:

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted Language
- Large Standard Library
- Dynamically Typed Language


### **Setup of Python Environment :**


**Step 1:** Open Eclipse and setup a correct access to Internet.

**Step 2:** Installing python environment using Eclipse Graphical Interface.

To install PyDev, we need to use **Help > Eclipse Marketplace** and installed PyDev

– Python IDE for Eclipse 7.6.0.

 Eclipse Marketplace




Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.  
Press the "more info" link to learn more about a solution.

Search Recent Popular Favorites Installed Giving IoT an Edge

Find: PyDev × All Markets All Categories Go




**PyDev - Python IDE for Eclipse 7.6.0**  
PyDev is a plugin that enables Eclipse to be used as a Python IDE (supporting also Jython and IronPython). It uses advanced type inference techniques which allow... [more info](#)  
by [Brainwy Software](#), EPL

★ 1864

Installs: 1.41M (11,381 last month)

Installed



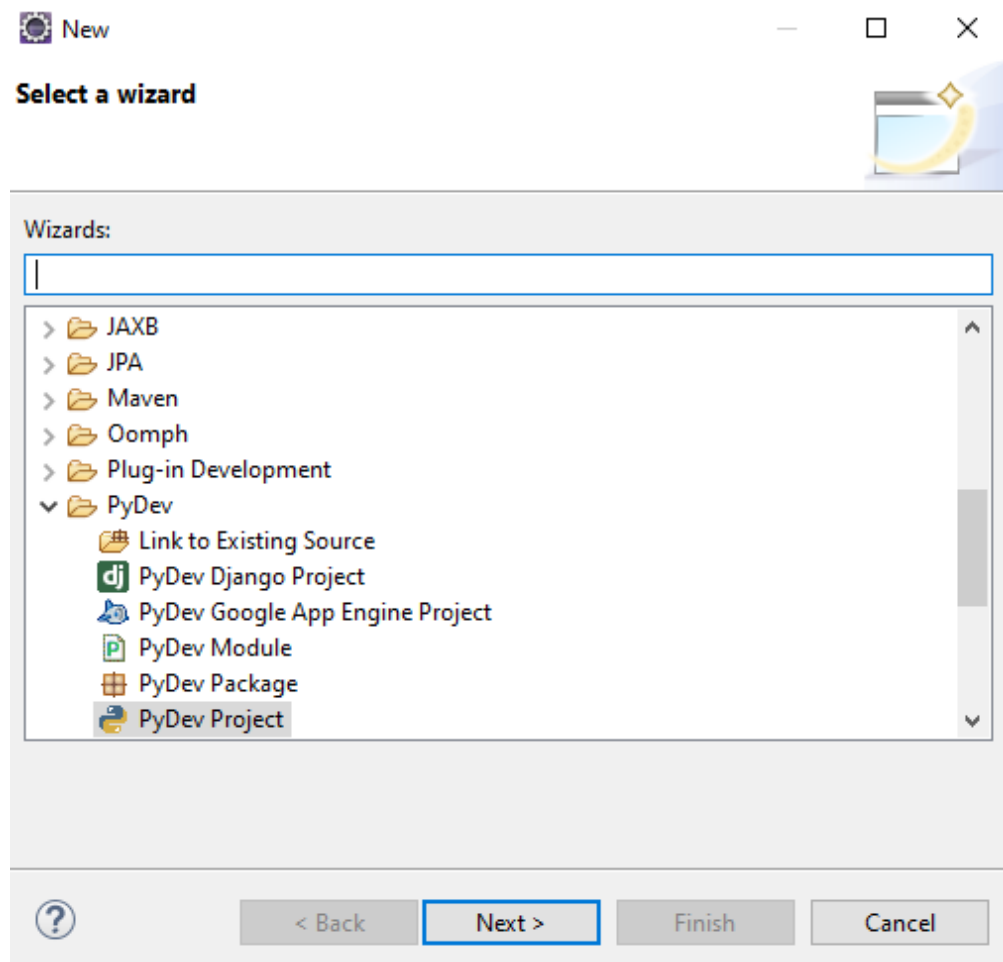
**Vrapper (Vim) 0.74.0**  
Vrapper acts as a wrapper for Eclipse text editors to provide a Vim-like input scheme for moving around and editing text. Unlike other plugins which embed Vim in... [more info](#)  
by [Vrapper Team](#), GPL


★ 382

Installs: 188K (1,331 last month)

Install

**Step 3 :** After installing PyDev, have to go **File > New > Other > PyDev > PyDev Project**.



— □ ×

## PyDev Project

Create a new PyDev Project.

Project name:

Project contents:

☒ Use default

Directory

Project type

Choose the project type

☒ Python ☐ Jython ☐ IronPython

Grammar Version

▾

Interpreter

▾

[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)

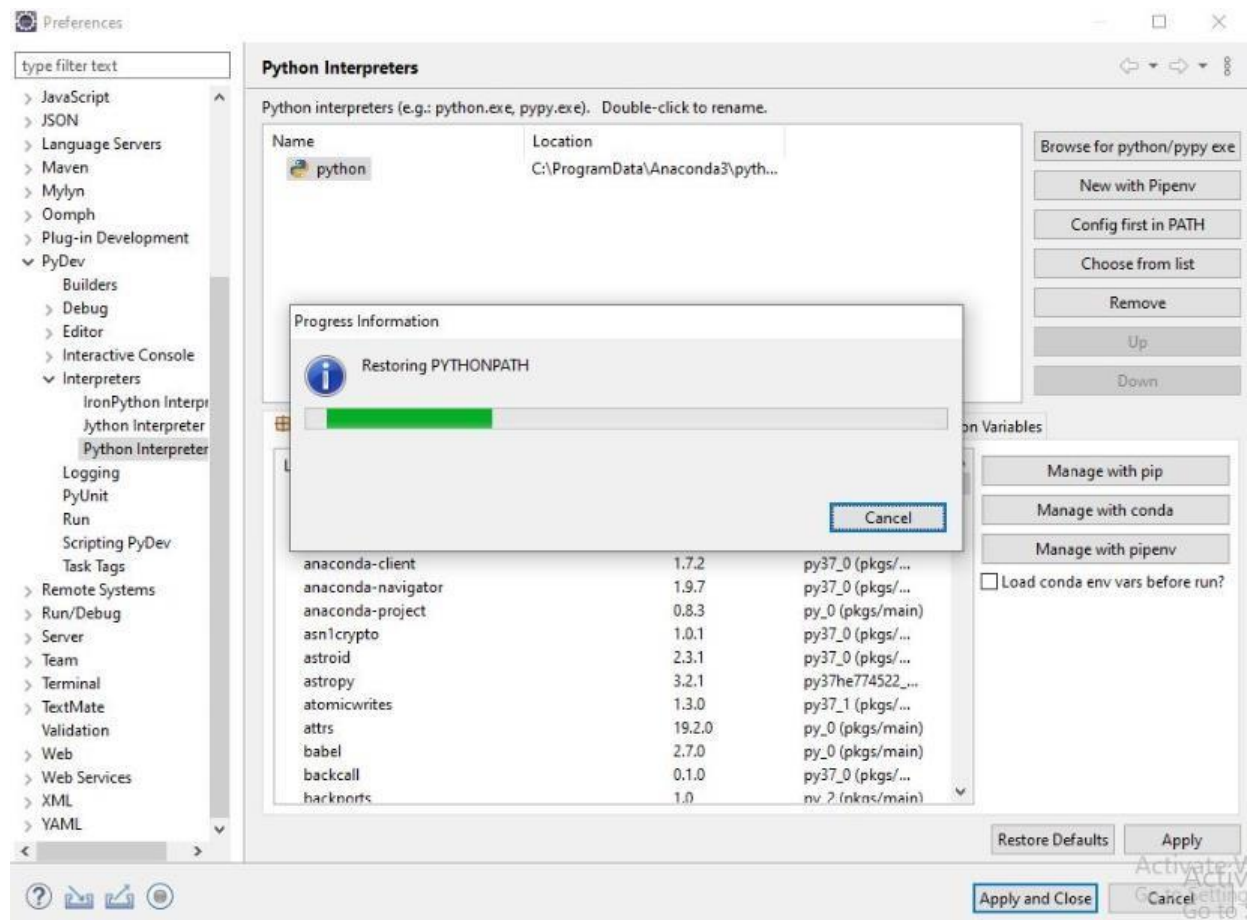
Working sets

☐ Add project to working sets

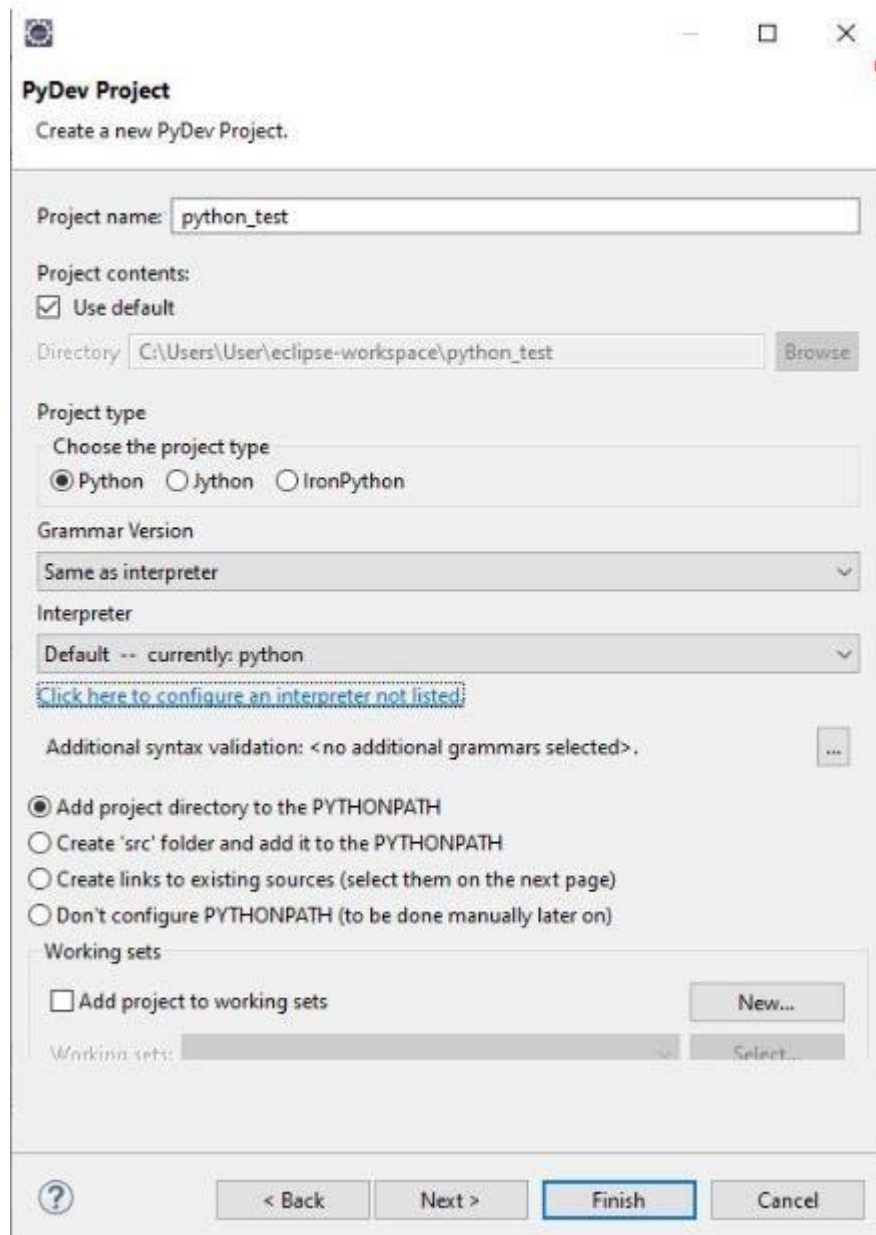
Working sets:  ▾



**Step 4 :** To configure an interpreter , have to go “click here to configure an interpreter not listed” and select “Config first in PATH”.



**Step 5 :** Then, give a project name and click “Finish” button.



The image shows the 'PyDev Project' dialog box in Eclipse. The title bar says 'PyDev Project' and the subtitle is 'Create a new PyDev Project.' The dialog is divided into several sections. The 'Project name' field contains 'python\_test'. The 'Project contents' section has a checked checkbox for 'Use default' and a 'Directory' field with the path 'C:\Users\User\eclipse-workspace\python\_test' and a 'Browse' button. The 'Project type' section has a label 'Choose the project type' and three radio buttons: 'Python' (selected), 'Jython', and 'IronPython'. The 'Grammar Version' section has a dropdown menu set to 'Same as interpreter'. The 'Interpreter' section has a dropdown menu set to 'Default -- currently: python' and a link 'Click here to configure an interpreter not listed'. The 'Additional syntax validation' section has a text field with '<no additional grammars selected>' and a button with three dots. The 'PYTHONPATH' section has four radio buttons: 'Add project directory to the PYTHONPATH' (selected), 'Create 'src' folder and add it to the PYTHONPATH', 'Create links to existing sources (select them on the next page)', and 'Don't configure PYTHONPATH (to be done manually later on)'. The 'Working sets' section has a checkbox 'Add project to working sets' and a 'New...' button. At the bottom, there is a 'Working sets' field with a 'Select...' button. The bottom of the dialog has a question mark icon, '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel' buttons.

**PyDev Project**  
Create a new PyDev Project.

Project name:

Project contents:  
☒ Use default  
Directory:

Project type  
Choose the project type  
☒ Python ☐ Jython ☐ IronPython

Grammar Version

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

Additional syntax validation:

☒ 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   
Working sets:

### Exercise 4.1.2: Write a Hello World program

```
Python1
1 print("hello world")

Console
<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]
hello world
```

### Exercise 4.1.3: Compute 1+1

```
Python1
1 a = 1
2 b = 1
3
4 print(f"{a} + {b} = ", a+b)
5

Console
<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]
1 + 1 = 2
```

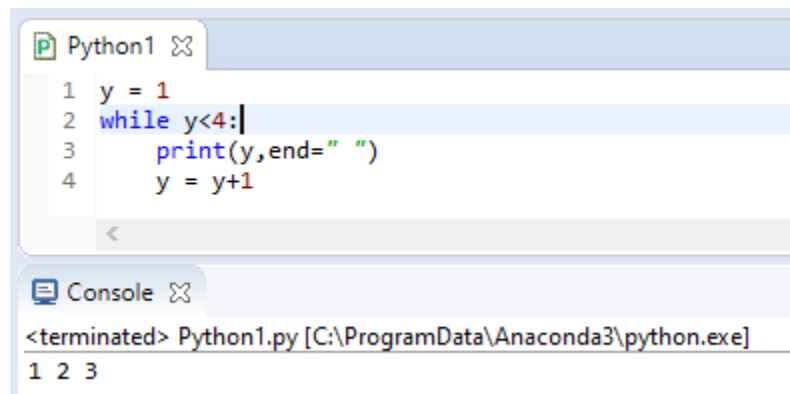
### Exercise 4.2.2: The if statement:

```
Python1
1 a = 4
2 b = 3
3
4 if a > b :
5     print("a is greater than b")
6 else:
7     print("b is greater than a")

Console
<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]
a is greater than b
```



### Exercise 4.2.3: The while Statement

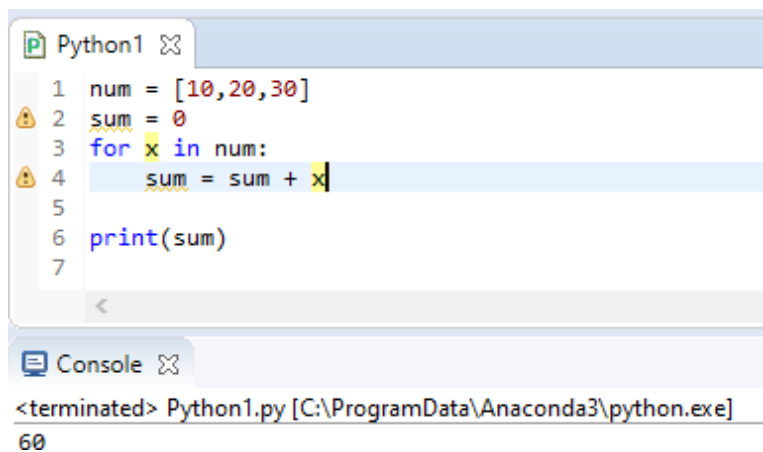


The screenshot shows an IDE window titled 'Python1'. The code editor contains the following Python code:

```
1 y = 1
2 while y<4:|
3     print(y,end=" ")
4     y = y+1
```

Below the code editor is a console window titled 'Console'. It shows the command prompt output: `<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]` followed by the output `1 2 3`.

### Exercise 4.2.4: The for Statem



The screenshot shows an IDE window titled 'Python1'. The code editor contains the following Python code:

```
1 num = [10,20,30]
2 sum = 0
3 for x in num:
4     sum = sum + x|
5
6 print(sum)
7
```

Below the code editor is a console window titled 'Console'. It shows the command prompt output: `<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]` followed by the output `60`.

## Conclusion

In this lab, we install python and eclipse ide and run some basic code with python. We didn't face any problem while doing this lab.