



MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

Department of ICT

Course Code : ICT – 3207

Course Title : Computer Networks

Lab Report No : 04

Report Name : Introduction to Python

Submitted By: Name : Ishrath Mostakin ID : IT-17018 Session : 2016-2017	Submitted To: Nazrul Islam Assistant Professor, Department of ICT MBSTU
--	--

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 - 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)




Vrapper (Vim) 0.74.0

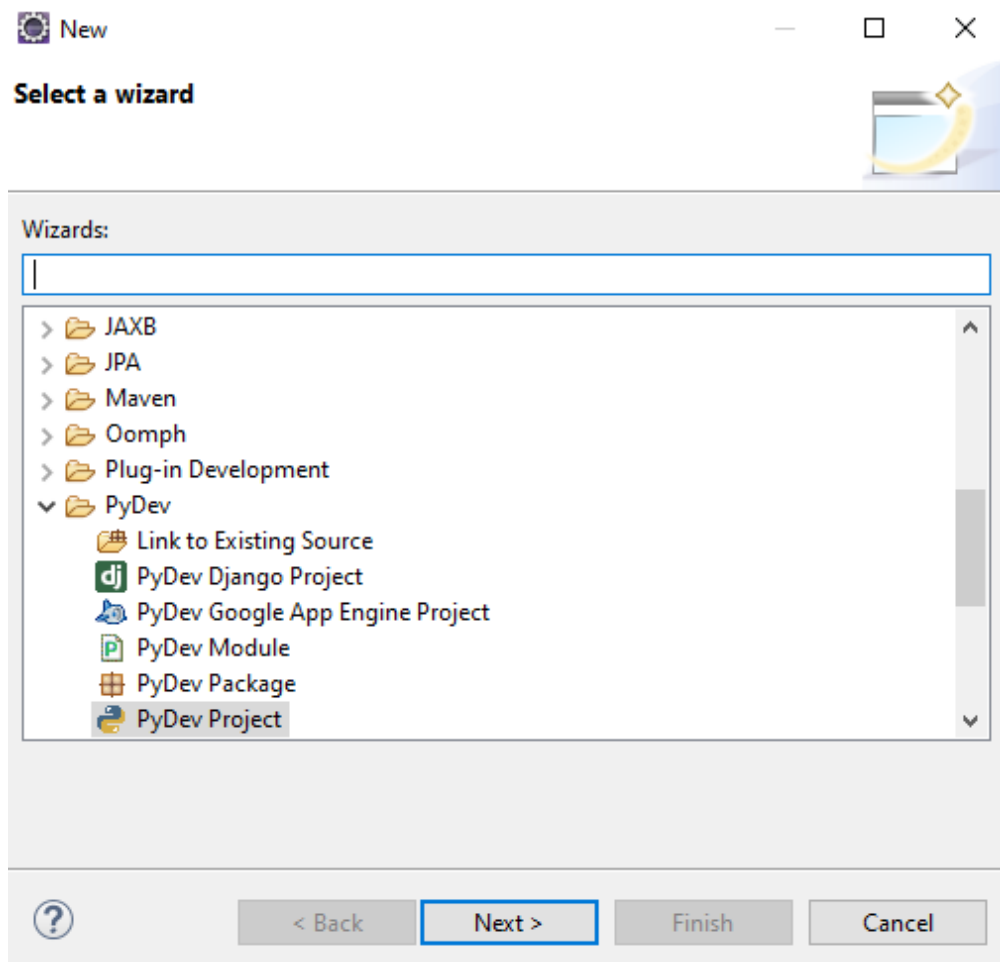
Vrapper acts as a wrapper for Eclipse text editors to provide a Vim-like input scheme for moving around and editing text. Unlike other plugins which embed Vim in... [more info](#)


by [Vrapper Team](#), GPL

★ 382

 Installs: **188K** (1,331 last month)

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)

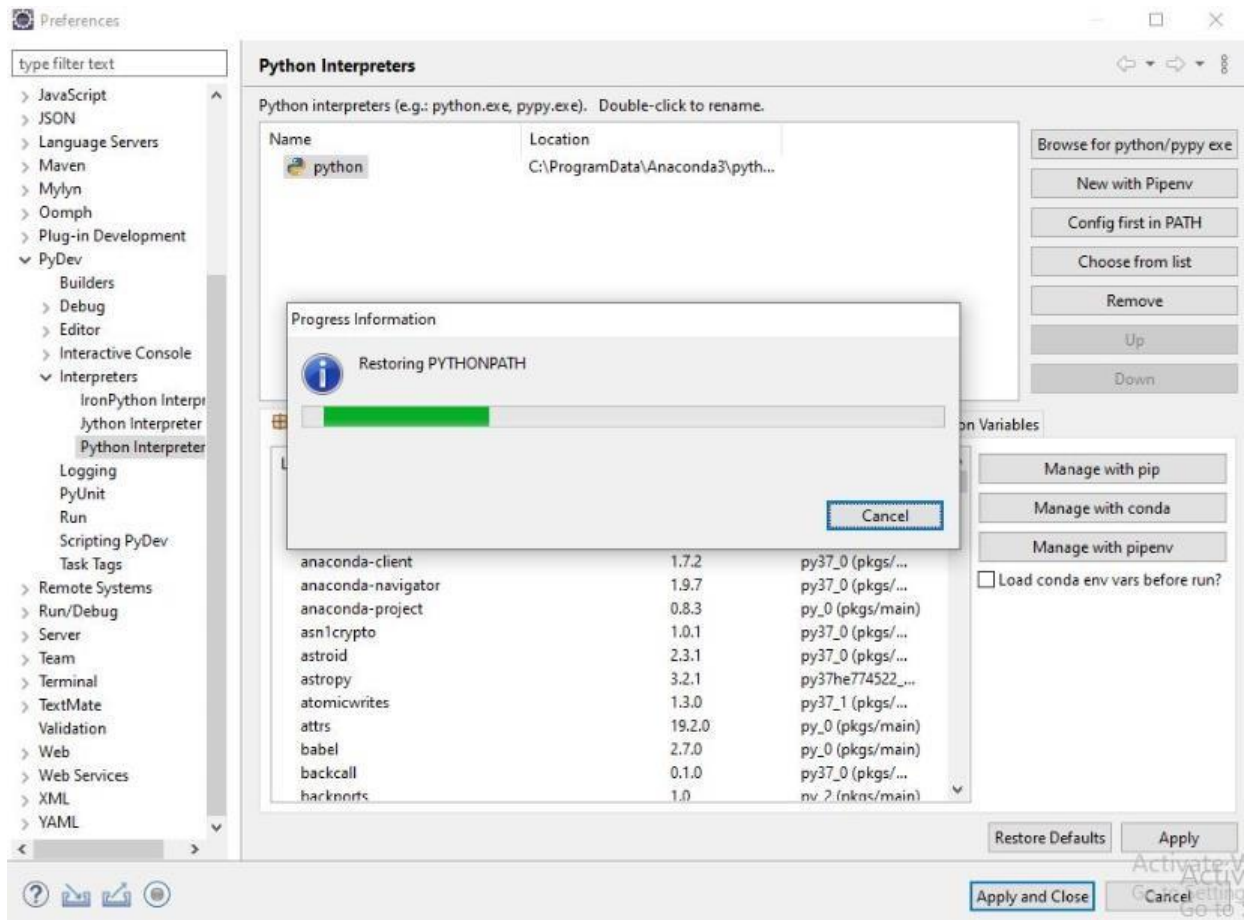
Working sets

☐ Add project to working sets

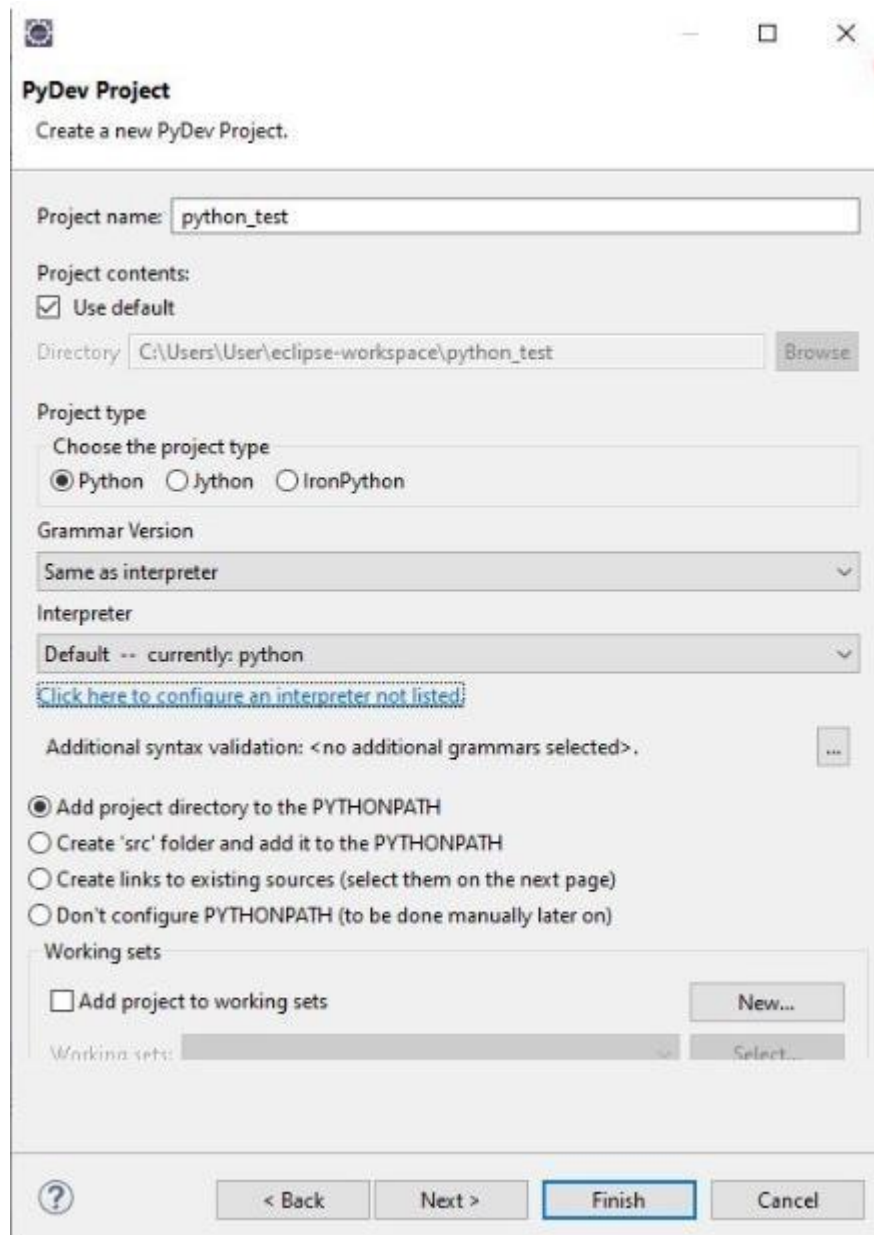
Working sets: ▾



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 'Project name' field contains 'python_test'. Under 'Project contents', the 'Use default' checkbox is checked. The 'Directory' field shows 'C:\Users\User\eclipse-workspace\python_test' with a 'Browse' button. The 'Project type' section has 'Choose the project type' with radio buttons for 'Python' (selected), 'Jython', and 'IronPython'. The 'Grammar Version' dropdown is set to 'Same as interpreter'. The 'Interpreter' dropdown is set to 'Default -- currently: python' with a link to 'Click here to configure an interpreter not listed'. The 'Additional syntax validation' section shows '<no additional grammars selected>' with a button to add more. 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 an 'Add project to working sets' checkbox and a 'New...' button. At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

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)

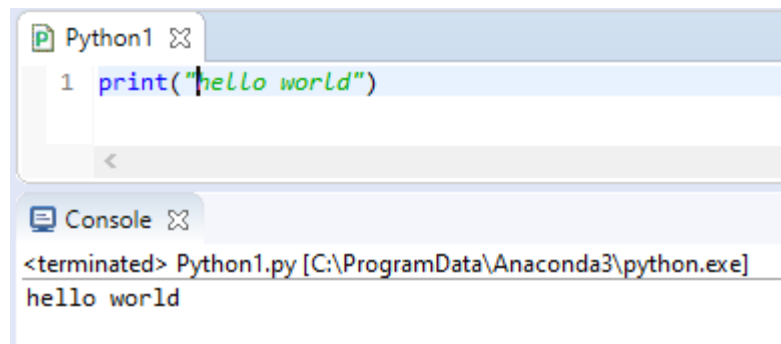
☐ Don't configure PYTHONPATH (to be done manually later on)

Working sets

☐ Add project to working sets

Working sets:

Exercise 4.1.2: Write a Hello World program

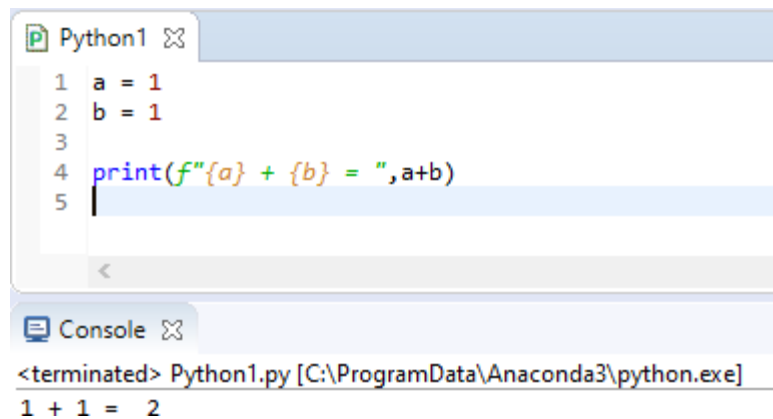


The screenshot shows a Python IDE with a file named 'Python1.py'. The code in the editor is a single line: `print("hello world")`. Below the editor is a console window showing the output: `<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]` followed by `hello world`.

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

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

Exercise 4.1.3: Compute 1+1

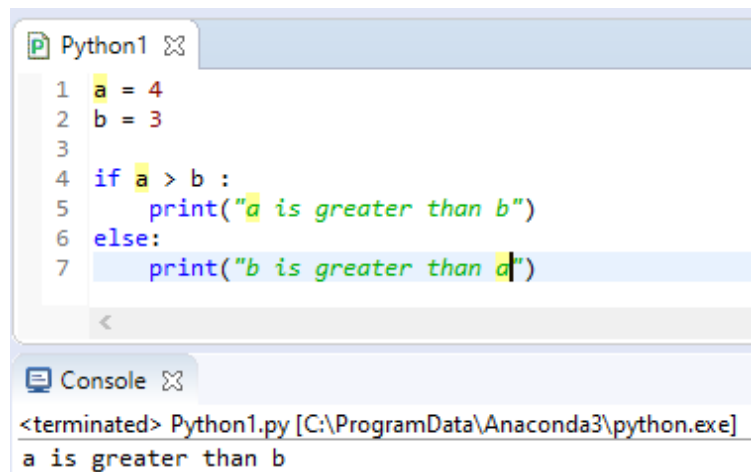


The screenshot shows a Python IDE with a file named 'Python1.py'. The code in the editor is: `a = 1`, `b = 1`, and `print(f"{a} + {b} = ", a+b)`. Below the editor is a console window showing the output: `<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]` followed by `1 + 1 = 2`.

```
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:

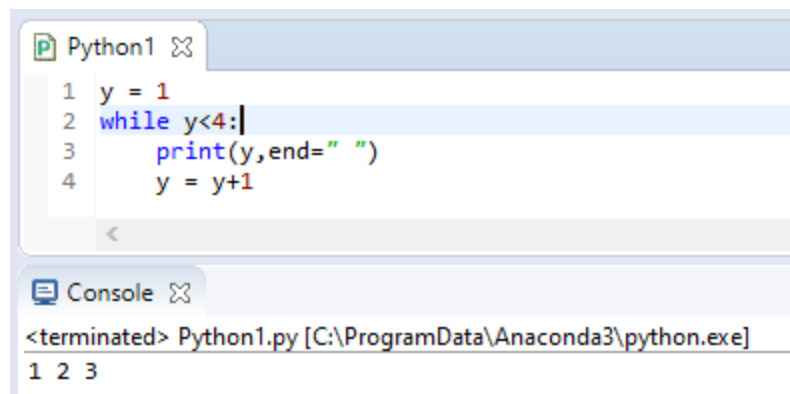


The screenshot shows a Python IDE with a file named 'Python1.py'. The code in the editor is: `a = 4`, `b = 3`, and an if-else statement: `if a > b :` followed by `print("a is greater than b")` and `else:` followed by `print("b is greater than a")`. Below the editor is a console window showing the output: `<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]` followed by `a is greater than b`.

```
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
```

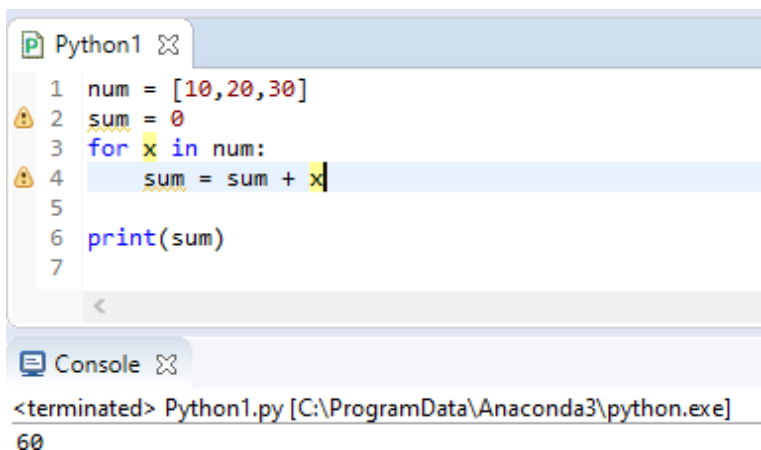

Exercise 4.2.3: The while Statement



```
Python1
1 y = 1
2 while y<4:
3     print(y,end=" ")
4     y = y+1

Console
<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]
1 2 3
```

Exercise 4.2.4: The for Statem



```
Python1
1 num = [10,20,30]
2 sum = 0
3 for x in num:
4     sum = sum + x
5
6 print(sum)
7

Console
<terminated> Python1.py [C:\ProgramData\Anaconda3\python.exe]
60
```

Conclusion :

The advantage is that we can see the results of changes in the program code much more quickly. By bypassing the need to compile the program before we run it, we greatly reduce the amount of work needed to try out small changes to a program. Another advantage is that we can interact directly with the interpreter; we can type commands directly into the interpreter and it will execute them and show the result immediately. This is a good way to begin learning a programming language.