



Lab-Report

Report No : DNS_LAB_01

Course code : ICT-3208

Course title : Introduction to Python

Date of Performance : 08-01-21

Date of Submission: 08-01-21

Submitted by

Name: MD.Abdullah Al Mamun

ID:IT-18040

3th year 2nd semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Theory :

Python is an easy to learn, powerful programming language. It has efficient highlevel data structures and a simple but effective approach to object-oriented programming. 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.

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):

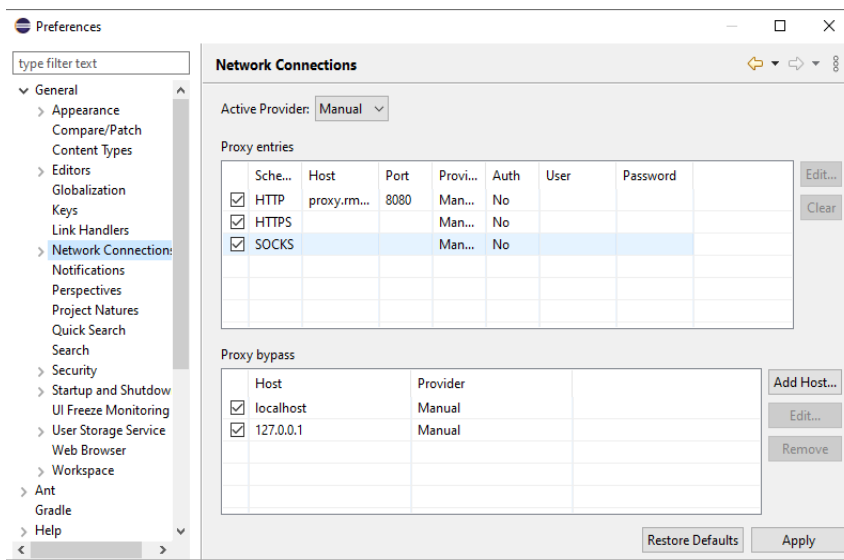
- Go to **Windows > Preferences > General > Network Connections**.
- Change Active Provider to Manual.
- Input proxy details, including username/password if required.

Host : proxy.rmit.edu.au

Port : 8080

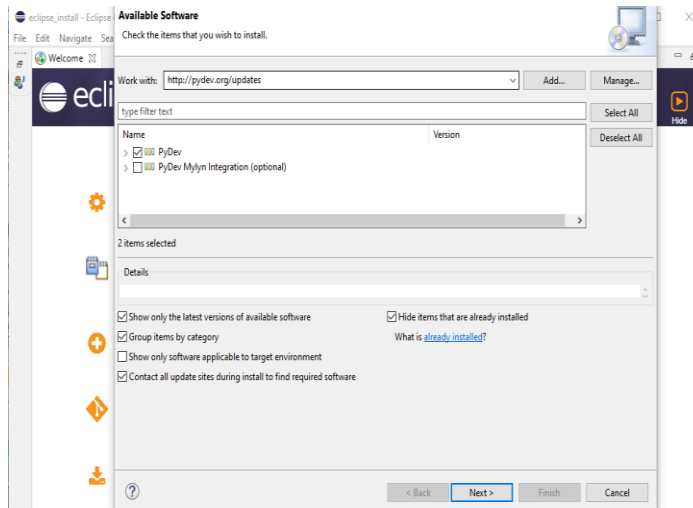
Username/password : No required

- Clear SOCKS proxy.
- Restart Eclipse.



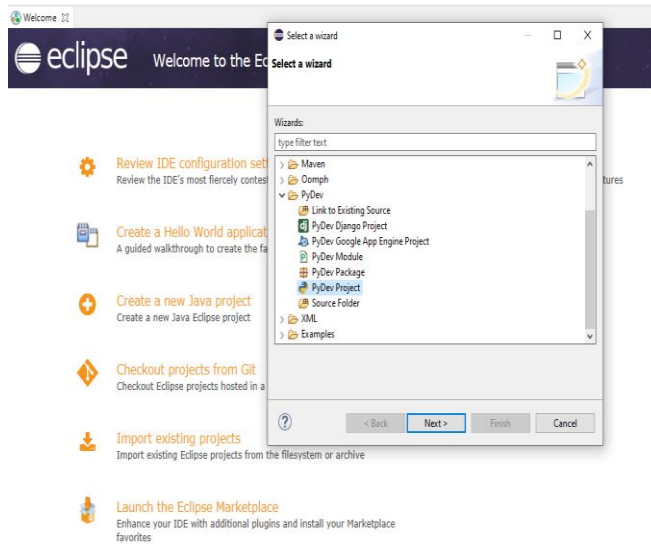
STEP 2 : Installing python environment using Eclipse Graphical Interface1.

- 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:



- 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
- D. Then, UNCHECK the 'Contact all update sites during install to find required software' and press 'NEXT' again to confirm your selection
- E. And finally, read the license agreement if you accept, select the accept radio button and click 'Finish'

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):

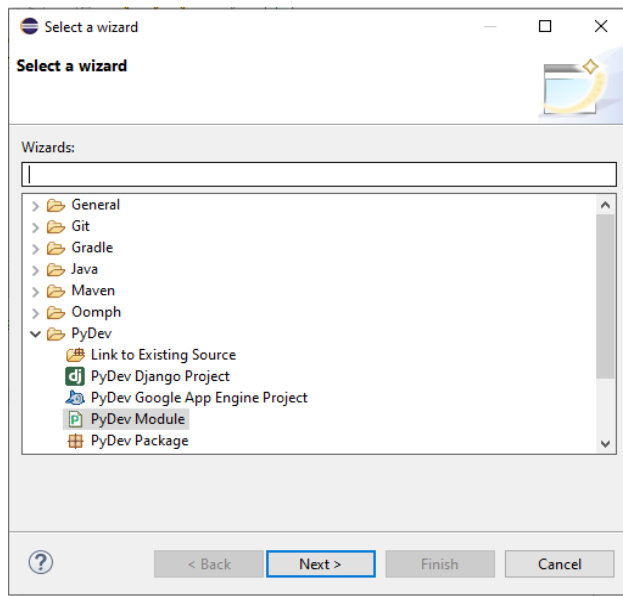


Exercises

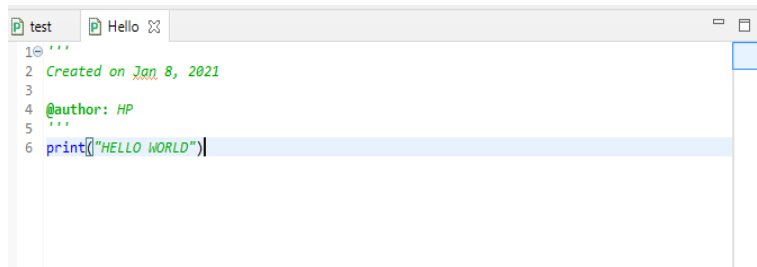
Section 4.1: Basics of python and programming

Exercise 4.1.1: Create a python project.

Answer:

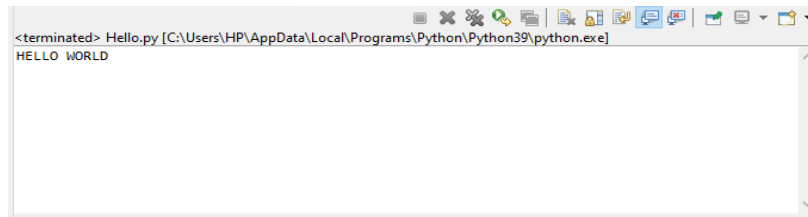


Exercise: 4.1.2 : Write a Hello World Program.



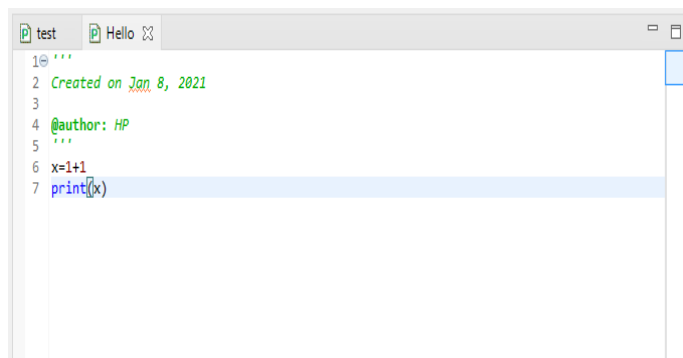
```
1 '''  
2 Created on Jan 8, 2021  
3  
4 @author: HP  
5 '''  
6 print("HELLO WORLD")
```

Output:



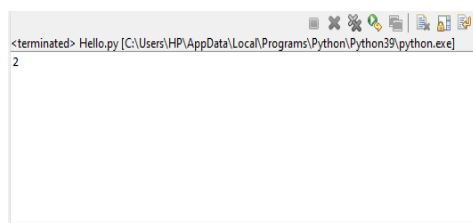
```
<terminated> Hello.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]  
HELLO WORLD
```

Exercise 4.1.3: Compute 1+1



```
1 '''  
2 Created on Jan 8, 2021  
3  
4 @author: HP  
5 '''  
6 x=1+1  
7 print(x)
```

Output:



```
<terminated> Hello.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]  
2
```

Exercise 4.1.4: Type in program text.

```
test Hello formulaus_shapse
10 area_parallelogon=h*r
11 print("the area of parallelogram is ")
12 print(area_parallelogon)
13
14 area_squre= r**2
15 print("the area of squre is ")
16 print(area_squre)
17
18
19 area_circul = pi*r**2
20 print("the area of circul is ")
21 print(area_circul)
22
23 area_volume =(1.0/3)*pi*r**2*h
24 print(" the area of volume is ")
25 print(area_volume)
26
27
28
29
30
```

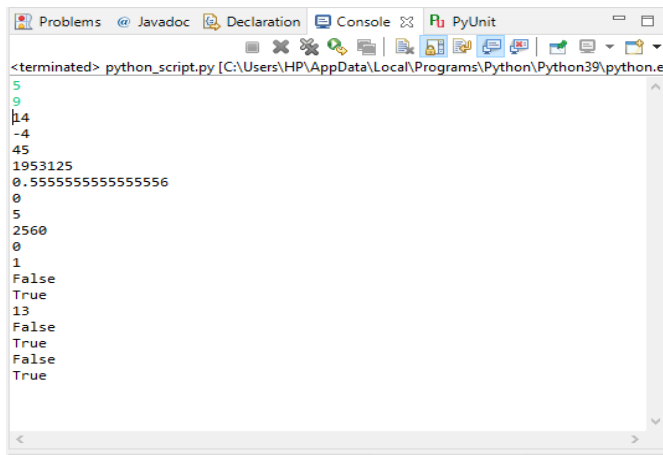
Output:

```
<terminated> formulaus_shapse.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\pyth
the area of parallelogram is
7.5
the area of squre is
2.25
the area of circul is
7.0686
 the area of volume is
11.780999999999999
```

Section 4.1: Create and run basic example.

```
test Hello formulaus_shapse python_script
2 Created on Jan 8, 2021
3
4 @author: HP
5 ...
6 a= int(input())
7 b=int(input())
8
9
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)
21 print(a<b)
22 print(a|b)
23 print(a>=b)
24
25 print(a<=b)
26 print(a==b)
27
28 print(a!=b)
29
30
31
32
33
```

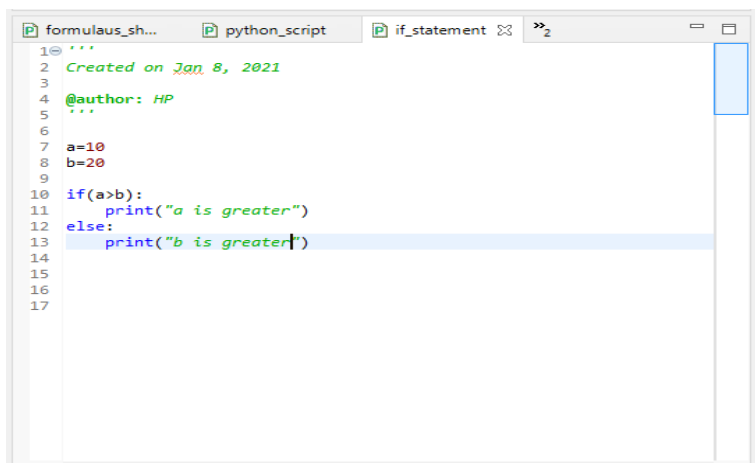
Output:



The screenshot shows an IDE window with a tab labeled 'PyUnit'. The console displays the output of a Python script, which includes numerical values and boolean results. The output is as follows:

```
<terminated> python_script.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]
5
9
14
-4
45
1953125
0.5555555555555556
0
5
2560
0
1
False
True
13
False
True
False
True
```

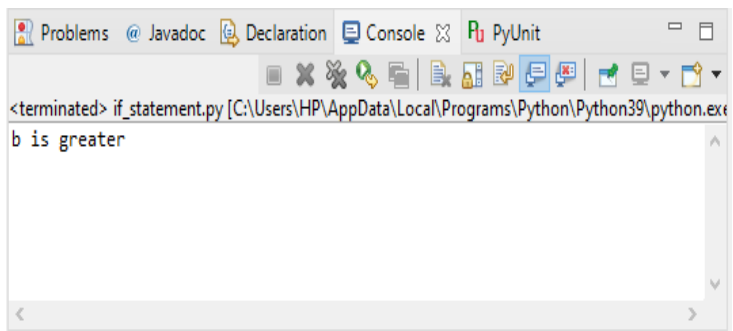
Exercise 4.2.2: The if statement:



The screenshot shows an IDE window with a tab labeled 'if_statement'. The editor displays a Python script that uses an if statement to compare two variables, a and b. The script is as follows:

```
1 '''
2 Created on Jan 8, 2021
3
4 @author: HP
5 '''
6
7 a=10
8 b=20
9
10 if(a>b):
11     print("a is greater")
12 else:
13     print("b is greater")
14
15
16
17
```

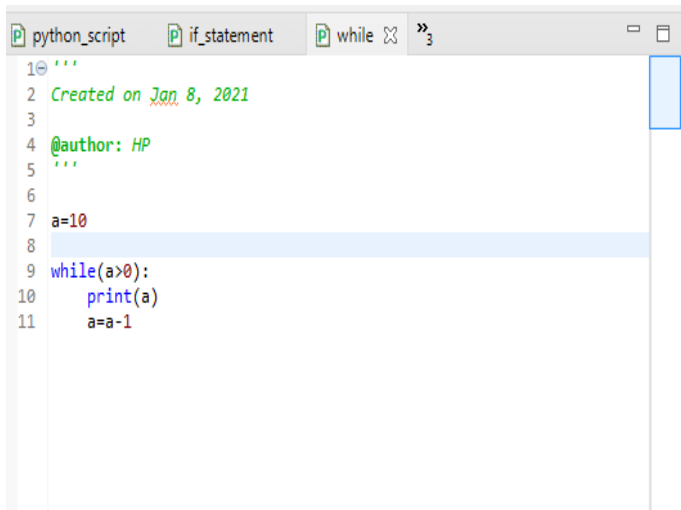
Output:



The screenshot shows an IDE window with a tab labeled 'PyUnit'. The console displays the output of the Python script, which is the string 'b is greater'.

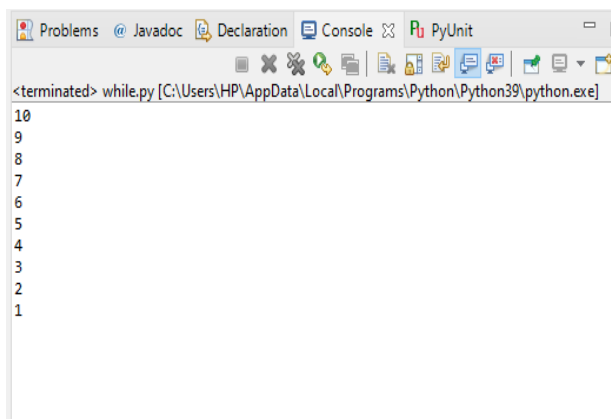
```
<terminated> if_statement.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]
b is greater
```

Exercise 4.2.3: The while Statement



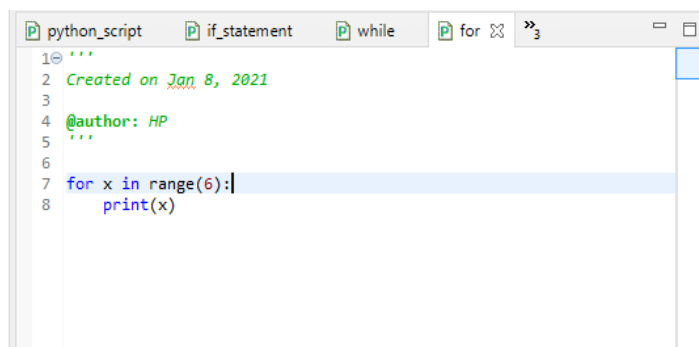
```
python_script  if_statement  while  »3
1  '''
2  Created on Jan 8, 2021
3
4  @author: HP
5  '''
6
7  a=10
8
9  while(a>0):
10     print(a)
11     a=a-1
```

Output:



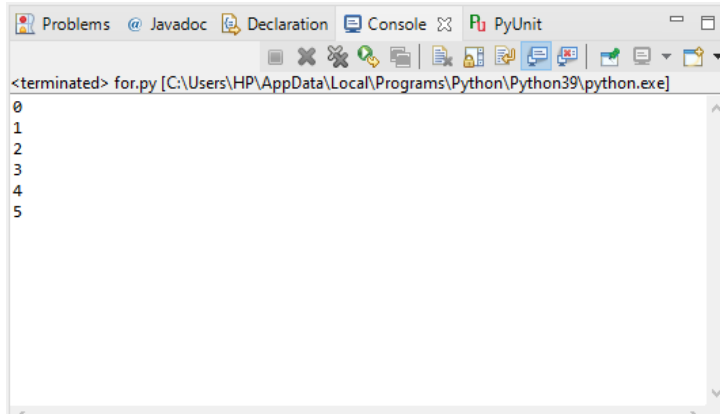
```
Problems  @ Javadoc  Declaration  Console  PyUnit
<terminated> while.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]
10
9
8
7
6
5
4
3
2
1
```

Exercise 4.2.4: The for Statement



```
python_script  if_statement  while  for  »3
1  '''
2  Created on Jan 8, 2021
3
4  @author: HP
5  '''
6
7  for x in range(6):
8     print(x)
```

Output:

A screenshot of a PyUnit test runner window. The window has a title bar with tabs for 'Problems', '@ Javadoc', 'Declaration', 'Console', and 'PyUnit'. The 'Console' tab is active, showing the command prompt output: '<terminated> for.py [C:\Users\HP\AppData\Local\Programs\Python\Python39\python.exe]'. Below the command prompt, a list of numbers is displayed: 0, 1, 2, 3, 4, 5. The window also features a toolbar with various icons for file operations and testing.

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.
- 4) Python is easy to read, even if you're not a skilled programmer. Anyone can begin working with the language, all it takes is a bit of patience and a lot of practice. Plus, this makes it an ideal candidate for use among multi-programmer and large development teams.
- 5) Python powers Django, a complete and open source web application framework. Frameworks – like Ruby on Rails – can be used to simplify the development process.
- 6) It has a massive support base thanks to the fact that it is open source and community developed. Millions of like-minded developers work with the language on a daily basis and continue to improve core functionality. The latest version of Python continues to receive enhancements and updates as time progresses. This is a great way to network with other developers.