



Shanti Education Society's

**A. G. PATIL INSTITUTE OF TECHNOLOGY, SOLAPUR**

Affiliated to DBATU, Lonere/Solapur University, Approved by AICTE & Recognized by Govt of Maharashtra,  
(NAAC Accredited – "B+" Grade)

**Report on**  
**Technical Training on Python Programming**

**From 22/02/2020 to 27/02/2020**

**Held at**

**A.G.Patil Institute of Technology, Solapur**



**Organized by**

**TPO,A.G.Patil Institute Of Technology,Solapur**

## **REPORT Python Programming Training**

**From 22/02/2020 to 27/02/2020**

With the blessings of AGPIT Management, having constant encouragements of our TPO Prof. S.V.Patil, Technical Training On Python Programming was conducted. Following schedule and the topic:

**Topic:** “Technical Training On Python Programming ”

**Date:** 22/02/2020 to 27/02/2020

**Resource person:** TPO,AGPIT Solapur

**Time:** 10:00am to 05:00pm.

**Duration of the Training:** 42hrs

**Program:** This session was conducted for final year CSE students at Smart Classroom. Total 70 Students have enrolled among these 51 students have participated. At the end of the session questions were given for the assessment. All 51 students have successfully completed the test.

## Syllabus - Core Python

- Introduction of Python
- Installing Python IDEs – Python IDLE and Anaconda
- Writing Your First Python Program
- Data-types in Python
- Variables in Python – Declaration and Use
- Typecasting in Python
- Operators in Python – Assignment, Logical, Arithmetic etc.
- Taking User Input (Console)
- Conditional Statements – If else and Nested If else and elif
- Python Collections (Arrays) – List, Tuple, Sets and Dictionary
- Loops in Python – For Loop, While Loop & Nested Loops
- String Manipulation – Basic Operations, Slicing & Functions and Methods
- User Defined Functions – Defining, Calling, Types of Functions, Arguments
- Lambda Function
- Importing Modules – Math Module

## Syllabus - Object Oriented Programming in Python

- Basics of Object Oriented Programming
- Creating Class and Object
- Constructors in Python – Parameterized and Non-parameterized
- Inheritance in Python
- In built class methods and attributes
- Multi-Level and Multiple Inheritance
- Method Overriding and Data Abstraction
- Encapsulation and Polymorphism

### 1. Introduction to SQL

**(What is SQL?, Purpose of SQL, Who should learn SQL?, What are the subsets of SQL?, Data Definition Language, Data Manipulation Language, Data Control Language, and SQL vs. NoSQL)**

### 2. Introduction to Databases and RDMBS

**(What is a Database?, Database Objects, Database Tables, Table Records, Types of Database Management Systems, Relational Database Management Systems, and SQL/Relational Databases vs. No SQL Databases)**

### 3. Install a Database Engine

**(Download MS SQL Server or Oracle or MySQL Database Engine, and Install. Launch SQL Server Management Studio, Select New Query, and launch SQL Query. Type SQL Commands and Execute.)**

#### 4. SQL Syntax

**(Focus on SQL Syntax, SQL keywords, SQL is not case sensitive, SQL Comments, SQL Commands, and writing SQL Statements.)**

#### 5. SQL Data Types

**(SQL Numeric data types, Date and Time data types, Character and String data types, Unicode character string data types, Binary data types, and Miscellaneous data types.)**

#### 6. SQL Operators

**(SQL Arithmetic Operators, Comparison Operators, Logical Operators, and Bitwise Operators)**

#### 7. SQL Expressions

**(SQL Boolean Expression, SQL Numeric Expression, and SQL Date Expression)**

#### 8. SQL Comments

**(SQL Comments, Comments are used to explain sections of SQL statements, or to prevent the execution of SQL statements. Single-Line Comments, and Multi-line Comments)**

#### 9. SQL – Data Definition Language Commands and Operations.

**(SQL Data Definition Language Commands, Create, Alter, Drop, Truncate, and Rename.**

**Data Definition Language Operations, Create a Database, Use Database, Rename a Database, Drop Database, Create a Table, Rename Table, Add a Column to existing Table, Add multiple columns to existing Table, Modify an existing column, Rename a Column, Drop a Column, Truncate a Table, and Drop a Table.)**

#### 10. SQL – Data Manipulation Language Commands and Operations

**(Data Manipulation Language Commands, SELECT, INSERT, UPDATE, and DELETE.**

**Data Manipulation Language Operations, Retrieving data from a table, Inserting data into a table, Updating existing data into a table, and Deleting all records from a table.)**

#### 13. SQL – Data Control Language Commands

**DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.**

# A.G.Patil Institute of Technology, Solapur

6 Day's Fastrack Technical Training on "Python Programming" (Campus Recruitment Training)

Training & Placement Cell

R. N. Students Name	Sem II														Received Status	
	22.02.2020		23.02.2020		24.02.2020		25.02.2020		26.02.2020		27.02.2020		Total Present	Total Absent		Fine to be Collected
	M	E	M	E	M	E	M	E	M	E	M	E				
1. KOLE APURY SANJAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
2. ANGADI SWAPNIL SANJAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
3. ANGONDA SHWETA NARAYAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
4. ATTAR MAHAMADABDULKADARI	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
5. ATTAR NIGAR SULTANA ATIYUB ALI	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
6. BABALESHWAR SHRADDHA S.	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
7. BHOSALE SNEHA DATTAIRAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
8. BIRADAR PRATHAMESH SIDDHARAM	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
9. BOLADE PRIYANKA CHANDRAKANT	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
10. BOMDIYAL AKSHATA AMBAJI	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
11. CHAVARE NIKITA VILAS	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
12. CHIVADSHETTI PRAVIN S.	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
13. DHU'DHAGUNDI DIPAK TUKARAM	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
14. DONGAONKAR MADHA RAUF	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
15. ERABOTE GOVIND BALAJI	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
16. GAIKWAD RUSHIKESH BALASAHEB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
17. GHANATE SHUBHAM SIDDHESHWAR	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
18. GHONGADE AISHWARYA RAUKUMAR	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
19. GUNDETI MONIKA RAYESHAM	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
20. GUNDETI SANJANA ASHOK	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
21. HAMPE PRADNYA SANTOSH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
22. INDE MAYURI RAJENDRA	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
23. JAMMA ANKITA ANIL	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
24. JEURKAR ASHUTOSH AVINASH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
25. JOSHI VIVEK PRASAD	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
26. KAMBLE MOHNI MEENAKSHI	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
27. KESH AISHWARYA PADMARAJ	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
28. KHAN FAISAL YUSUF	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
29. KHIRSAGAR SANJANA VITHAL	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
30. KULKARNI AJINKYA DINKAR	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
31. KULKARNI PRIYA PRAVEEN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
32. KUMBHAR ASHWINI BASAVRAJ	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
33. LONDHE POONAM LAHU	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
34. MADAS ROHIT SRINIVAS	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
35. NADAF SUMAN A BUDAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
36. NAGOJI MANALI MALLINATH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
37. NAMA SANJOT VILAS	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
38. NANNAKAR CHINMAYEE ATUL	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
39. NAYAR SUSMITA RAJAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
40. PANCHWAGH SHREYAS ARUN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
41. PARCHANDE MALLIKARJUN SHIVANAND	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
42. PATHAN SALMA ANWAR KHAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				
43. PATIL ABHISHEK SANJAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB				

R.N.	Students Name	22.02.2020		23.02.2020		24.02.2020		25.02.2020		26.02.2020		27.02.2020		Total Present	Total Absent	Fine to be Collected	Remarks
		M	E	M	E	M	E	M	E	M	E						
44	PAWAR TRUPTI SHAM	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
45	SALUNKE VIKAS ARJUN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
46	SAMBHARAM TEJAS VIJAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
47	SANDUPATLA AKHIL PRABHAKAR	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
48	SHAIKH SHAHID KHAJA	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
49	SHAIKH USAAMA ABDUL REHMAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
50	SHINDE PUNAM BALASAHEB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
51	SOMA SNEHA GANESH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
52	TAMBAKE VAIBHAVI SUBHASH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
53	TORANE PRATHMESH DATTATRAY	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
54	UDANSHIVE PRABUDHA A.	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
55	VALSANGKAR MID FAIZAN SAJJAD HUSAIN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
56	VANGARI UMESH PANDURANG	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
57	VARHADE PRATHMESH PANDIT	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
58	WADNE GEETA JANAK	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
59	YEDUR PRIYA SIDRAM	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
60	SURADE INDRAJEET VILAS	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
61	GIDGONDE SUPRIYA RAOSAHEB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
62	KARANDE PRANALI PADMAKAR	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
63	GADDAM SURESH ANAND	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
64	SARGAM MOHNIISH	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
65	GOGI ROHAN	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
67	PATIL OMKAR DHANARAJ	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
68	GAIKWAD ASHISH ASHOK	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
69	KOKATE SANDYA J	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
70	MANE DYANAND S	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB	AB					
<b>Total Amount</b>		40	40	35	35	45	31	37	38	35	35	40	40				

Note: Fine will be collected from absent students.

*Manoj*  
 Departmental T&P Coordinator  
 (Prb.kenboard.v.d)  
 CSE DEPT.

## Assessment on Python Programming

### Q1. What is Python?

Python is nothing but an advanced and smart computer programming language that has exceptions, objects, modules, threads, and automatic memory management. The language is quite simple and easy to use, transferrable, extensible, open-source, and build-in data structure.

### Q2. What is the shortest and the easiest way to read a file?

2 memory- effective way which is ranked in order where the first one is the best. It is exclusively supported from 2.5 Python and above it.

2. If you eventually would like to have control over how much is to be read, you can make use of yield.

3. Use of WITH

This can be the best as well as the most effective pythonic way to read huge files.

#### Advantages:

Let us check out a few of these advantages

- The object of the file is automatically closed once the exciting form is closed along with the execution block.
- Handling the exception inside the with block
- The memory of the loop iterates along, through the file f with an object line by line. It internally buffers the optimization on costly IO operations as well as with the memory management with an open (“x.txt”) as f for the line in f as something to do with data.
- The statement handles and takes care of the opening as well as the closing of the file, along with the exception which is raised with the inner block.

### Q3. What is the reason behind using yield in Python?

There are situations where sometimes people wish to have more thorough and fine access control over how much is to be read over in each iteration. To get better control in these cases it's always the best idea to use iter and yield.

### Q4. What is the use of the <\_\_init\_\_.py> module in Python?

The use of the <\_\_init\_\_.py> module can help in several ways for fulfilling the following objectives.

- It allows Python to interpret the directories with the package containments by not including the things which contain names in common like a string.
- Apart from all this, it additionally allows all the programming experts to take control and decide whether which directory is a package and which one is not.
- Likewise, the <\_\_init\_\_.py> can be an empty file as well which can further help in code initialization and execution for setting the variable of <\_\_all\_\_> as well as for a package.

### Q5. Which are the various methods that Python provides for copying an object?

Either we can use the Shallow Copy method or the Deep Copy method to approach copying an object.

Shallow Copy method.

In this method, the content of the object like a dictionary does not get copied by the values but it does by creating the new reference.

SHALLOW COPY METHOD.

```
>>> a = {1: [1,2,3]}
>>> b = a.copy()
>>> a, b
({1: [1, 2, 3]}, {1: [1, 2, 3]})
>>> a[1].append(4)
>>> a, b
({1: [1, 2, 3, 4]}, {1: [1, 2, 3, 4]})
```

1  
2  
3  
4  
5  
6  
7

```
>>> a = {1: [1,2,3]}
>>> b = a.copy()
>>> a, b
({1: [1, 2, 3]}, {1: [1, 2, 3]})
>>> a[1].append(4)
>>> a, b
({1: [1, 2, 3, 4]}, {1: [1, 2, 3, 4]})
```

DEEP COPY METHOD.

It copies all the contents by value.

```
>>> c = copy.deepcopy(a)
>>> a, c
({1: [1, 2, 3, 4]}, {1: [1, 2, 3, 4]})
>>> a[1].append(5)
>>> a, c
({1: [1, 2, 3, 4, 5]}, {1: [1, 2, 3, 4]})
```

1  
2  
3  
4  
5  
6

```
>>> c = copy.deepcopy(a)
>>> a, c
({1: [1, 2, 3, 4]}, {1: [1, 2, 3, 4]})
>>> a[1].append(5)
>>> a, c
({1: [1, 2, 3, 4, 5]}, {1: [1, 2, 3, 4]})
```

**Q6. How can one set a global variable inside a function?**



One can easily use the global variable for various functions by presenting it as global in every function which gets assigned to it:

```
globvar = 0
def set_globvar_to_one():
    global globvar # Needed to modify global
    copy of globvar
    globvar = 1
def print_globvar():
    print globvar # No need for global
    declaration to read the value of globvar
set_globvar_to_one()
print_globvar() # Prints 1
1
2
3
4
5
6
7
8
```

I can only imagine all the reasons behind it that, as the global variables happen to be so dangerous, Python tries and makes sure that you should know about what you actually playing with explicitly needs the global keyword.

#### **Q7. For what is forelse used in Python language?**

The language Python is very interesting which easily allows you to specify an else suite:

Here is a clear example of this type.

```
for I in foo:
```

```
    if bar(i):
```

```
        break
```

```
    else:
```

```
        baz()
```

the else suite is executed after the for, but only if the for terminates normally (not by a break).

Here's some code is written without for...else:

```
def contains_even_number(l):
```

```
    "Prints whether or not the list l contains an even number."
```

```
    has_even_number = False
```

```
    for elt in l:
```

```
        if elt % 2 == 0:
```

```
            has_even_number = True
```

```
            break
```

```
    if has_even_number:
```

```
        print "list contains an even number"
```

```
    else:
```

```
print "list does not contain an even number"
```

The same code snippet listed below shows how to use of for...else. This allows you to remove the extraneous flag variable which is in the loop:

```
def contains_even_number(l):  
    "Prints whether or not the list l contains an even number."  
    for elt in l:  
        if elt % 2 == 0:  
            print "list contains an even number"  
            break  
    else:
```

```
    print "list does not contain an even number"
```

Decide wisely while judging whether to use for...else construct or not. It is not really unequivocally better but yes, when there is an asymmetry between any two possibilities, you can easily make your code more readable to use it for...else to follow the happy path logic at the top avoiding the errors at the bottom. Learn at [3RI Technologies](#).

### **Q8 How to make an Iterator in Python?**

The objects in Python Iterator conform to the protocol of the iterator, which primarily offers 2 kinds of methods:

`__iter__()` and `next()`.

The `__iter__` returns the iterator object and is implicitly called at the start of loops.

The `next()` method returns the next value and is implicitly called at each loop increment.

`next()` raises a `StopIteration` exception when there is no more value to return, which is implicitly captured by looping constructs to stop iterating.

Here's a simple example of a counter:

```
class Counter:  
    def __init__(self, low, high):  
        self.current = low  
        self.high = high  
    def __iter__(self):  
        return self  
    def next(self): # Python 3: def  
        __next__(self)  
        if self.current > self.high:  
            raise StopIteration  
        else:  
            self.current += 1  
            return self.current - 1  
for c in Counter(3, 8):  
    print c
```

This will print:

```
3  
4
```

5  
6  
7  
8

**Q 9: What is the difference between lists and tuples?**

Lists are mutable, and it means these can be edited while tuples are immutable, it means these can be edited.

Lists are considered slower than tuples.

Syntax that is used in List is `list_1 = [10, 'Chelsea'], 20` while for tuples, syntax is `tup_1 = (10, 'Chelsea', 20)`

**Q 10: What is meant by pep 8?**

The term stands for Python Enhancement Proposal. Pep 8 is a set of rules that are used to specify the formatting of python code to get the maximum readability.

**Q 11: How can it possible to manage memory in Python?**

It is possible to manage memory management in Python with Python Private Heap Space. The fact, all the data structures, and python objects are located in a private heap. The programmer is not supposed to access this private heap. The inbuilt garbage collector is also found in Python that recycles all the unused memory thereby, and it can be used for heap space. Moreover, in python objects, the allocation of heap space is done through Python's Python's memory manager. Learn more at [Python Web Development Course](#)

**Q 12: What do you mean by namespace in Python?**

It is a naming system used to be sure that names are unique to avoid naming conflicts.

**Q 13: How can you explain the term Pythonpath?**

Python path is an environmental variable that is used in the case of the importing of a module. A module is introduced; the python path is used to check out the presence of the imported in all directories. With an interpreter, it is possible to decide which module is needed to load.

**Q 14: Differentiate the terms pickling and unpickling in Python?**

With the pickling module, it is possible to accept any module of python object to convert it into a string representation. After that, that object dumps into a file with a dump function that is known as pickling.

Moreover, the process where retrieving original Python objects is possible from the stored string representation is known as unpickling.

**Q 15: What do you mean by python modules? Which are built-in modules commonly used in Python?**

Python modules mean those files where python codes are available. Moreover, this code may be like function classes or variables. Kindly note that a python module is considered as a .py file that has executable code.

Commonly used built-in modules in Python are:

- os
- random
- data time
- os
- sys
- math
- JSON

**Q 16: In Python, what is the usage of local variables and global variables?**

**Global Variables:** When variables are considered either outside a function or in global space, it is known as global variables. The fact, it is possible to access these variables from any function in the program.

**Local Variables:** When variables are considered inside a function, it is known as local variables. The fact, this variable is presented in the local space rather than in the global space.

**Q 17: Names the tools used to perform static analysis or finding bugs?**

To find out errors or bugs in python source code, PyChecker is used that is a static analysis tool. It is also used to warn about the complexity and style of the virus.

Even Pylint is another tool that is verified to check whether the module fulfils the coding standard or not.

**Q 18: Explain about lambda in Python**

Lambda in Python is a single expression anonymous function that is frequently used as an inline function.

**Q 19: In Python, why lambda forms do not have statements?**

Yes! Lambda forms do not have statements in Python as it is used in case of a new function object and also get returned at runtime.

**Q 20: What do you mean by Python decorators?**

These are specific amendments that need to make in python syntax to alter functions flawlessly.

**Q 21: Explain the term self in Python?**

A python is an object of a class, explicitly considered as the first parameter. But this is not possible in the case of Java. This command assists in differentiating between attributes and methods of a class along with local variables.

# CERTIFICATE OF PARTICIPATION

THE FOLLOWING AWARD IS GIVEN TO

*Angonda Shweta Narayan*

This certificate is given to Angonda Shweta Narayan  
for her participation in the Python Programming held at A.G.Patil Institute of Technology  
from 22/02/2020 to 27/02/2020



**Managing Director**  
GIRI'S Tech Hub