



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
Python Programming and SQL Training

From 23/12/2019 to 01/01/2020

Held at

A.G.Patil Institute of Technology, Solapur



Organized by

Giri's Tech Hub,Pune

REPORT Python and SQL Training

From 23/12/2019 to 01/01/2020

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

Topic: “Technical Training On Python Programming and SQL training ”

Date: 23/12/2019 to 01/01/2020

Resource person: Harish M Pawar

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

Duration of the Training: 60hrs (6hrs per day)

Program: This session was conducted for Second year CSE students at department. Total 37 Students have enrolled among these 35 students have participated. At the end of the session questions were given for the assessment. All 35 students have successfully completed the test.



1. Introduction
2. Conditional Statements
3. Looping
4. Control Statements
5. String Manipulation
6. Lists
7. Tuple
8. Dictionaries
9. Functions
10. Modules
11. Input-Output
12. Exception Handling
13. OOPs concept
14. Regular expressions
15. CGI
16. Database
17. Networking
18. Multithreading
19. GUI Programming

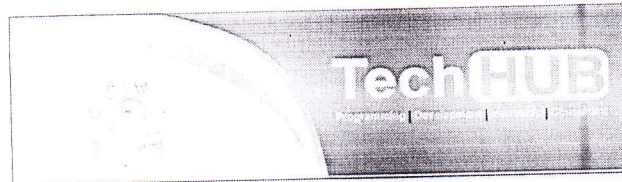
SQL:

Course Topics

Lesson 1: Introduction to SQL

Lesson 2: Retrieving Data

Lesson 3: Updating Data



Lesson 4: Inserting Data

Lesson 5: Deleting Data

Lesson 6: Sorting and Filtering Data

Lesson 7: Advanced Filtering

Lesson 8: Summarizing Data

Lesson 9: Grouping Data

Lesson 10: Using Subqueries

Lesson 11: Joining Tables

Lesson 12: Managing Tables

Lesson 13: Using Views

Lesson 14: Stored Procedures

Lesson 15: Using Cursors

Lesson 16: Using Transactions

And much more

Costing:

We propose to take the Technical Aptitude Training as per above details in following slabs rates-

Costing: Rs. 1200 per head

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 l 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.



Giri's Tech HUB

..... Talent Meet Opportunity

Programming - Aptitude - Soft Skills - Consultant

CERTIFICATE

This is to certify that Mr./Ms. Chakre Aishwarya Prabhuling
from A.G.Patil Institute of Technology, Solapur has
successfully completed 2 weeks in-house internship training
on Python Programming from 23/12/19 to 1/01/20.

Date: 1/01/2020

Location: A.G.P.I.T, Solapur

Director
Giri's Tech Hub

