Data, BigData. Cloud. .
Anaytics & Virtualizaton
HadoopExam.com

230+ Python Interview Questions (Core Language)

1st Edition

QuickTechie

About Us and Book: This Book is published by www.HadoopExam.com (**HadoopExam Learning Resources**). Where you can find material and training's for preparing for BigData, Cloud Computing, Analytics, Data Science and popular Programming Language. This Book will contain popular Python Core language Interview Questions usually being asked during selecting the candidate for Python Development or Starting Career on Analytics or will be working as Data Scientists. These book covers only core language interview questions and does not give any guarantee that you will be selected in interview process. This book will help you to gain more Python programming language as well. There will be in total 90+ selected Questions. There is a Python Programming training which is provided by HadoopExam.com and that will be less than 8 hours training, which can help you to become Python programmer. This is the much faster way to learn Python, whether you have any programming skill or not. It is really recommended that you will go with that training to learn the Core Python Language and complete the Hands on session as well. HadoopExam is providing Big Data, Cloud Computing, Programming and Analytics learning since last 5 years. Following are the other popular products.

TRAINING'S (AVAILABLE)

- Hadoop BigData Professional Training
- HBase (NoSQL) Professional Training
- Apache Spark Professional Training
- Apache OOZie (Hadoop workflow) Professional Training
- Core Java 1z0-808 Exam training
- Scala Programming Training
- Python Programming Training
- Cloudera® Administration Professional Trainings

Acknowledgement: We really wish to thanks our learners who helped us to reach at this level in just short span (4.5 Years). We have 1000's of learners who subscribed our products and continuously provides the feedback on the material which we create. And based on their feedback we upgrade/update our material so that other learners can get benefit out of this. Without their feedback, we cannot provide this material. As there are many 1000's of learners who provides feedback, so we cannot write all the names here. But combined effort lead us to quality and low cost material. In future we are similarly expecting the same help from our network. Please subscribe here , if you want to keep in touch with HadoopExam Learning Resources.

Who all are the user of the Python Programming?

Ans: Python programming is not only for the IT developers, it is mainly used by professionals who are working in Finance domain, Analytics, Data Scientists, Researchers and IT Developers etc.

QUESTION-1. Is Python High level or low level language?

Ans: Python is a high level language as other programming language C, C++, Java, Scala etc.

QUESTION-2. What do you mean by low level language?

Ans: Low level language which can be understood by underline machine, you can say machine language or assembly language are low level language. And computers can run the program which is written in low language only.

QUESTION-3. How high level language are executed by computers?

Ans: High level language are first needs to be converted to low level language e.g. Java Virtual Machine does of java language. This is an extra step which makes executing code slower compare to low level language.

QUESTION-4. What are the benefits of high level language?

Ans : Writing program using high level language can have following advantages.

- Easier to program
- They are portable, e.g. write program on Windows OS that can be executed on IOS without any or very less change. In case of low level language, you have to write program again for different OS.

QUESTION-5. How high level language are converted to low level language?

Ans: There are two way by which high level languages are converted into low level language.

- Interpreter
- Compiler

QUESTION-6. What is interpreter?

Ans: It does not do lot of conversion, it reads the program and directly execute on the computer. Hence, this is relatively faster than compiler.

QUESTION-7. What is compiler?

Ans: Compiler reads your program written in high level language e.g. Java Code and it is known as source file and once source file is compiled it will be converted into object code(also executable). This executable need not be converted in any other form. Now you will have run time environment where you can run this executable on any operating system.

QUESTION-8. Is Python an Interpreter or compiler language?

Ans: Python is an interpreter language. Hence, it is not required to be compiled and then executer (as in case of Java you have to do this).

QUESTION-9. What is the extension of the Python source file?

Ans : Python source file has .py extension like test.py , is a python file.

QUESTION-10. What all types of errors in a Python program?

Ans: There are mainly three types of errors.

- Syntax Error: It is any syntactical error in the program. E.g. (8.value, is not correct)
- Runtime Error
- Semantic Error

QUESTION-11. What is a runtime error?

Ans: These errors only appears when you run the program. This is also known as exceptions.

QUESTION-12. What is semantic errors?

Ans: Catching such errors are not easy. Until and unless you test the output. It means your program will run without any issue. It will successfully complete. But when you see the result than only you can say that there is an issue e.g. You want to add two numbers 2 and 3 and expected answer is 5, but program generate 6 (because instead of + sign, * has been used.).

QUESTION-13. What is the difference between interpreter and compiler?

Ans: Interpreter convert high level language line by line but for compiler entire program will be translated in low level language.

Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-14. What do you mean by data type in Python?

Ans: This is a category of the data e.g. 23 is an int , 23.7 id a float and ‘Amit’ is a string type.

QUESTION-15. What is the variable in a python?

Ans: Its name to a value. You can refer a particular value using variable name like ‘Amit’ it is a value but will be referred using a variable called ‘name’

QUESTION-16. What is a function?

Ans: Function is a collection of statements, which is defined once and then call it later to do some executions. Below is the function definition example

```
def total_course_durations(c1,c2):
    training=curse_duration()
    training.hours=c1.hours+c2.hours
    training.minutes=c1.minutes+c2.minutes
    training.seconds=c1.seconds+c2.seconds
    return training

total_duration=total_course_durations(hadoop_training,spark_training)
```

Name of the function: total_course_duration

Statements: All orange color lines are statements.

return: It is a return statements.

total_duration: It is a variable, holding the values return by function call.

Function Arguments: These two variables are known as function arguments hadoop_training,spark_training

QUESTION-17. How do you convert data from one data type to another data types?

Ans: There are various functions available, which can help you convert data from one data types to another data type. See the example below

int(‘100’) -> 100 #Converting string to int

flaot(‘100.5’) -> 100.5 #Converting string to float

str(3.02) -> ‘3.02’ #Converting float to string

QUESTION-18. What is the module in Python?

Ans: Python module is any python file with .py extension, which contains functions and variables are known as Python Module. To use module in our current program we have to import it first. There are in-built modules which are provided by Python itself e.g. import math (Here, math is a Python module)

QUESTION-19. What do you mean by a fruitful functions?

Ans: Function, which return a value is known as fruitful function. E.g math.sqrt(4), it will return 2

QUESTION-20. What is void function?

Ans: Function which does not return a value is known as void function.

QUESTION-21. What are the all ways, by which you can import a module in Python?

Ans: Python provides two ways by which you can import the Python module.

- Using import statement e.g. import math, it will give you an object of math module.
- Another way, import only, which you want from module e.g**. from math import pi** now you can use pi directly without dot notation.

QUESTION-22. How can I import the all the functions and variable from a module?

Ans: You can import everything from your module using * operator as below.

from math import *

However, this is a bad practice. It will overwrite all the variables and functions which you have defined in your program. So it is suggested, to avoid this.

QUESTION-23. What is function definition and function object?

Ans : Function definition, is the way by which you define a function as below.

```
def total_course_durations(c1,c2):
    training=curse_duration()
    training.hours=c1.hours+c2.hours
    training.minutes=c1.minutes+c2.minutes
    training.seconds=c1.seconds+c2.seconds
    return training
```

Function Object: This is created by a function definition. Name of the function '**total_course_durations**' is a variable, which points to function object.

QUESTION-24. What do you mean by function arguments and parameters?

Ans: Parameters are defined by the names that appear in a function definition, whereas arguments are the values actually passed to a function when calling it. Parameters define what types of arguments a function can accept. For example, given the function definition:

```
def func1(value1, value2=None, **kwargs):
```

Value1, value2 and kwargs are parameters of func1. However, when calling func1, for example:

Learn Python in Less than 8 Hours sitting at Home/@Desk

```
func(100, value2=”hadoopexam”, new_name=anyvar)
```

the values 100, “hadoopexam” and anyvar are arguments.

QUESTION-25. What do you mean by local variables?

Ans: A variable which you define inside the function is known as local variable, you can not use local variable outside the function. Below two variables course1 and course2 are local and cannot be accessed outside the function.

```
def total_course_durations(c1,c2):
    course1=c1
    course2=c2
```

QUESTION-26. What do you mean by module object?

Ans: A value created by an import statement that provides access to the values defined in a module.
e.g import math , here math is a module object and using dot notation you can access the variables defined in this module for instance math.pi

QUESTION-27. What is the encapsulation?
Ans: It is the process of transforming a sequence of statements into a function definition. Everything is encapsulated inside the function.

QUESTION-28. What is the docstring in python?
Ans: Python documentation strings (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.

An object's docstring is defined by including a string constant as the first statement in the object's definition.

```
def my_function():
   """Do nothing, but document it.
   No, really, it doesn't do anything.
   """
```

You can print this doc string using: print my_function.__doc__

QUESTION-29. Which operator will help you get the remainder and quotient value in Python?
Ans: There are two mathematical operator to help you get quotient and remainder.

Getting remainder using '%' modulus operator and '/' to get the quotient value for example

```
quotient = 10 / 3 # will result 3
remainder = 10 % 3 #will result 1
```

QUESTION-30. What all are the relational operators?
Learn Python in Less than 8 Hours sitting at Home/@Desk

Ans: Below are all considered relational operator to do the comparison.

```
A !=B # A is not equal to B
A>B # A is greater than B
A<B # A is less than B
A>=B # A is greater and equal to B
A<=B #A is less than or equal to B
```

QUESTION-31. What all are logical operators in Python?
Ans: There are three logical operators in Python **and, or** and **not.** See example below

- a>0 and a<10 # This condition is true only if a is greater than 0 and less than 10 . If a=100 than it will return false

- a>0 or a<10 # this condition will return true, if either a is greater than 0 or less than 10 e.g. a=100 than also it will return true.

- not (10 >100) # It will return true. Because first 10 >100, it means it is false but we are negating this condition. Hence, it will become true.

QUESTION-32. What is recursion?
Ans: A function calling itself. If you don't have a base condition (to break the recursion). It can be infinite call which can lead your program to run indefinitely and then finally crash.

QUESTION-33. What do you mean by dead code?
Ans: A part of the code, which is never executed is known as dead code. This is generally written after the return statement and will never be executed.

QUESTION-34. What is the length of empty string?
Ans: Length of empty string will always be 0 and it will be represented as ''.

QUESTION-35. What does it mean string is an immutable object?
Ans : String is considered an immutable object because. Once, you create it you cannot change its value.

A='Amit'
a.replace('A', 'S') # this will not work and give error.

QUESTION-36. What is the method?
Ans: Method is a function only, but it is associated with the object. And can be called using dot notation.
object.method()

QUESTION-37. How do you represent file in Python?
Ans: A file is represented using file object. For instance you have a file "c:/hadoopexam.txt" . If you want to read or write in this file you have to use Python file object.

QUESTION-38. What is the list in Python?
Ans: In python list is a sequence of values e.g. ["Amit","Rakesh","Rohit","Umesh"] is a list and contains 4 elements. List is represented using [] bracket. And it's indexing start with 0. So here value at index 0 is "Amit" and value at index 3 is "Umesh" and there is no value at index 4.
Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-39. Can a list contain another list?
Ans: Yes, a list can contain another list as below

["Amit","Rakesh","Rohit","Umesh" , ["Mumbai" , "Delhi" , "Banglore" ,"Chennai"]]

QUESTION-40. What is an accumulator?
Ans : Accumulator is a variable, which generally used in the loop to accumulate the values in each iteration. E.g. if you want to add all the values in a list [1,2,3,4] . Then, you will be creating an accumulator which will hold the sum till previous iteration.

QUESTION-41. What is the dictionary in Python?
Ans: Dictionary is a key value pair and will be used inside this {}. For instance, see below name as key and age as value.

{"Amit":32, "Rakesh":37, "Jayesh":42}

You cannot have duplicate keys in dictionary. Also, key should be immutable.

QUESTION-42. What is global variable?
Ans: A variable, which can be defined outside the function and can be accessed from any function is known as global variables.

QUESTION-43. Which module, you can use to debug Python code?
Ans : There is a module called pdb, which you can use for debugging.

QUESTION-44. Which tool you can use, which can help you to find the bugs in your Python source code?
Ans: You can use PyChecker tool, this will help you find the issue with your source code and also help you to do static analysis for Python.

QUESTION-45. Is there any tool or library which I can use to convert my Python code as windows executables or windows exe?
Ans: You can use py2exe tool, it convert the Python scripts into windows executables or exe and to run it you don't have to explicitly install the python on your windows machine.

QUESTION-46. What is UnboundLocalError and why you get it?
Ans: UnboundLocalError is a error when , you try to assign a local variable which is not initialized.

Let's check following steps

```
name='Amit'
def fullname():
   print name

fullname() # call the function, it will work fine.
```

Now define the function as below

```
name='Amit'
def fullname():
   print name
   name= name+'Khanna'

fullname() # call the function, it will give error 'UnboundLocalError'
```

Reason: This is because when you make an assignment to a variable in a scope, that variable becomes local to that scope and shadows any similarly named variable in the outer scope. Since the last statement in fullname assigns a new value to name, the compiler recognizes it as a local variable. Consequently when the earlier print name attempts to print the uninitialized local variable and an error results.
Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-47. How can be avoided 'UnboundLocalError' with the global variable?
Ans: You have to mark the variable global explicitly

```
def fullname():
    global name
    print name
    name= name+'Khanna'
    print name

fullname()
```

QUESTION-48. I am still confused with the local and global variables of Python?
Ans : If you have defined variable outside the function and only referencing them inside the function is considered global variable. But if you try to assign them with same or new value then it will be considered a local variable and local variable must be initialized first to use it else you will get **UnboundLocalError'**

QUESTION-49. You are having below code with the Lambda

```
doubled= []
for v1 in range(5) :
            doubled.append(lambda:2*v1)
print(doubled[0](), doubled[1](), doubled[2](), doubled[3]() , doubled[4]())
```

Why all the elements in doubled list are 10

Ans: Because v1 is not a local variable and it is defined in the outer scope and will be accessed in lambda function. Hence, when the loop ends, value of the v1 variable will be 4. Hence, all the values in the list will become 8.

To avoid this, you have to define a new variable that is local to the lambda as below

```
doubled= []
for v1 in range(5) :
            doubled.append(lambda x=v1:2*x)
print(doubled[0](), doubled[1](), doubled[2](), doubled[3]() , doubled[4]())
```

Now, here x is local variable to lambda, which holds the current value and then doubled and then appended to list.

QUESTION-50. How to make a variable shared across all the modules in your application and it should be global?
Ans: If you want to make a variable global across all the modules in your application than you have to define a new module usually config.py and then define that global variable in that module.

Import the config.py module in each of your application module. And as you know, only one instance of a particular module will be created. Hence, any change you make to global variable in any module will be reflected in each module. See below example with three different module files

config.py

```
price=1000 # Default price for the course will be 1000INR
```

hadoop.py

```
import config
config.price=2000 #Increase the price to 2000INR and same will be reflected in each module
```

course_detail.py

```
import config
import hadoop
print(config.price)
```

QUESTION-51. Why should you avoid below import statement
*from nameofmodule import **

Ans : Because, if you have created a variable with the same name as defined in import module as well, then it will overwrite your variable and it will be difficult to debug, what happened to the variable.

QUESTION-52. What is the best order to import the modules?
Ans : It is considered best practice, if you import the modules in following order

- First import standard library module e.g. sys, os, re
- Then third-party modules e.g. mx.DateTime
- Then modules developed by you locally.

QUESTION-53. What is the circular import and which form of circular import will not create a problem?
Ans : When you have import statement at the top and uses the "import nameofmodule" it is fine.

a.py

```
import b
```

b.py

```
import a
```

This is circular import and they will work fine, without any issue.

Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-54. Which form of circular import will not work?
Ans : When you put circular import in the form of “from nameofmodule import something” and you will have circular dependencies than it will fail.

a.py

```
from b import x
```

b.py

```
from a import y
```

Here, module a is busy importing module b and in module b value y from module a is not yet available. Hence, this kind of circular import should be avoided at the top level and needs to move import to function or class level.

Only move imports into a local scope, such as inside a function definition, if it’s necessary to solve a problem such as avoiding a circular import or are trying to reduce the initialization time of a module. This technique is especially helpful if many of the imports are unnecessary depending on how the program executes. You may also want to move imports into a function if the modules are only ever used in that function. Note that loading a module the first time may be expensive because of the one time initialization of the module, but loading a module multiple times is virtually free, costing only a couple of dictionary lookups. Even if the module name has gone out of scope, the module is probably available in sys.modules.

QUESTION-55. When you define a function with default shared value like “def function_name(mydict={}):” what issue you will see?

Ans: When you define a function as below

```
def fun1(my_dict={}):
        … do something…
        my_dict[key1]=”val1”
        return my_dict
```

Here, it is expected that every time you call the function fun1, it will create a new my_dict instance with no values. But this is not correct, because default values are created only once. And any subsequent call to that function will use the already created my_dict object.

To avoid such scenario, you should have used None as a default value and inside the function you should check whether the value is None then create a new my_dict.

```
def fun1(my_dict=None):
        if my_dict is None:
                my_dict={} #now it is a local dictionary object
```

QUESTION-56. How, can you pass optional or keyword parameters from one function to another function?

Ans: You have to collect the arguments as * and ** , which will give you positional arguments as a tuple and the keyword arguments as a dictionary. You can pass these arguments when calling another function using the * and **

```
def fun1(param1, *args, *kwargs):
      kwargs[key]=value
      fun2(param1, *args, *kwargs) # Calling another function
```

QUESTION-57. You have the following code

```
list1 = [ ]
list2 = list1

list2.append("hadoopexam.com")
```

What all the values are in list1 and list2?

Ans : Both list1 and list2 will hold the same value. Because, list2 is pointing to the same list as created by list1. Hence, any modification you do either through list1 or list2 , changes will be reflected in both. In fact there is only one list and pointed by two variables named list1 and list2.

QUESTION-58. You have been given below code

```
X=100
y=x
x=x+1 #there new object of x will be created as x itself cannot be mutated.
```

What is the value hold by variable x and y?

Ans : Here x will have 101 and y will be 100. Because, integers are immutable and when you do x=x+1 , it will be creating new int object and variable x will be assigned that variable. However, y will still hold the old value.

QUESTION-59. You have a list y as below

```
y=["hadoopexam" , "learning" ,"resource" , "spark" , "learning" , "resources"]
x=y.sort()
```

When you call y.sort() , what will happen?

Ans: In this case y will hold the sorted list , but variable x is pointing to None. Most of the cases the operation which mutate object itself return None. Here, y itself sorted.

[Learn Python in Less than 8 Hours sitting at Home/@Desk]

QUESTION-60. You have below code

```
x=(1,2,3)
x+=(100,)
```

What will happen to the x?

Ans : In this case x is immutable, hence new object of x will be created.

QUESTION-61. How do you find, if two variables are pointing to the same object or not?
Ans: You should have used is operator or build-in function id().

QUESTION-62. How can you return more than 1 value from a function?
Ans : You should have used tuple as a return value. If you want more than 1 value as a return value.

QUESTION-63. What do you mean by Higher-order function?
Ans : A function, which can accept other function as an arguments and return function to the caller is known as higher order fi=unction. See the example below.

```
def doOperation(a,b):
    def operation(x):
        return (a+x) *(b+x)
    return operation
```

Here, doOperation function take two arguments a and b as input and return operation function as an output. So you call the function as below.

```
result=doOperation(5,10)
result(10)
```

It will give output as below
(5+10)*(10+10)=(15)*(20)=300

QUESTION-64. What is the ideal way to copy sequences?
Ans : You should use slicing to copy sequence as below

```
New_list=old_list[:]
```

QUESTION-65. Which method will you use to find all the methods and attributes of an Object of a class?
Ans : We will be using either help() or dir() method.

QUESTION-66. How is the below statement will be evaluated?

```
“x” in “a” , “x”
```

Ans : Please note that comma is not an operator. It is a separator. Hence, above statement will be evaluated as below.

```
(“x” in “a”) , “x”
```

And result will be as below

```
(False, “x”)
```

QUESTION-67. What is the value of below expression?

```
X=100
Y=20

Val = X if X < Y else Y
```

Ans: Here Val will have 20. Because first condition is false, which will result in Y, which is 20.

QUESTION-68. How will you remove the duplicate elements from the list?
Ans: It’s very simple, as we know set contain only unique values. Hence, first convert list to set and then set back to list.

```
list(set(list_with_duplicates))
```

QUESTION-69. What is a Class?
Ans : You can say, it is a user defined object. It works as a template and hold attributes and functions.

Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-70. What is self?

Ans: Self is a first argument of a method. A method defined as do_something(self, a, b, c) should be called as object.do_something(a, b, c) for some instance object of the class in which the definition occurs; the called method will think it is called as do_something(self, a, b, c).

QUESTION-71. How would you find the current module name?

Ans : You can use __name__ to find the name of current module. But if it is run by script than , __name__ will be equal to “__main__ ”

QUESTION-72. What is the difference between dir and help function?

Ans : Both the function will help you get the detail about modules. There usage are below.

- If you write only dir, without any () braces , it will return all the names in current scope.

Using dir() function for each of the below behaves like

- For module: It will return all the module’s attributes and functions.
- For Class object: It will return all the attributes of the same class as well as the base class attributes.

- dir()   Returns the attributes of the object or module.
- help()   Returns the python built in documentation about the object.
- type()   Returns the type of object.
- __doc__   Returns the doc-string of object or module.

help() : Help function, name itself defines the usage of this function. This function returns the help related to python module, object or method if it is called with respective argument but without any argument it will return the help related to currently running programming module.

QUESTION-73. Can we override the dir() function behavior?

Ans: Yes, you can. By overriding __dir__ function.

QUESTION-74. How would you get builtin function using dir()

Ans: You can get all the builtin functions using dir(__builtins__)

QUESTION-75. Why Python does not de-allocate all the memory on exits?

Ans: Python does not de-allocate all the memory as soon as it exits, because some memory is reserved by C library and possible there could be some circular references exists. Hence, memory can not be de-allocated immediately in Python.

QUESTION-76. What zip() function does?

Ans : zip() function will combine multiple lists elements based on their position and create tuple out of this. See below example

```
l1=[1,2,3]
l2=['a','b','c','d']
l3=['hadoop' , 'exam' , 'learning']

zip(l1,l2,l3)
Out[11]:
[(1, 'a', 'hadoop'), (2, 'b', 'exam'), (3, 'c', 'learning')]
```

It will create tuple only the minimum length of the any list.

QUESTION-77. What is the Pass by Value and Pass by Reference?
Ans : Whenever arguments passed to the function they are always passed by reference. It means, if you change the value of the function inside the parameter, it will also change the value of the original parameter. However, if you are passing immutable variable then it will be pass by value, hence original copy of the variable will not be changed.

QUESTION-78. Is everything object in Python?
Ans : Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute __doc__, which returns the doc string defined in the function's source code. The sys module is an object which has (among other things) an attribute called path.

QUESTION-79. What is an id() function?
Ans : id() unction provide the unique integer value of an object.

QUESTION-80. What is the purpose of the __init__ method?
Ans : The __init__ method is run as soon as an object of a class is instantiated. The method is useful to do any *initialization* you want to do with your object.

QUESTION-81. What is the purpose of pass in Python?
Ans: Suppose you are designing a new class with some methods that you don't want to implement, yet.

```
class MyClass(object):
    def meth_a(self):
        pass
    def meth_b(self):
        print "I'm meth_b"
```

If you would leave out the pass, the code wouldn't run. To summarize, the pass statement does nothing particular but can act as a placeholder.
However, there are many more usage but in short that is.

QUESTION-82. Which all are Python web scrapping libraries?
Ans : There are following Python web scrapping libraries are available

- The Farm: Requests
- The Stew: Beautiful Soup 4
- The Salad: lxml

- The Restaurant: Selenium
- The Chef: Scrapy

QUESTION-83. What is the purpose of the with statement in Python?

Ans : In python the with keyword is used when working with unmanaged resources (like file streams). It allows you to ensure that a resource is "cleaned up" when the code that uses it finishes running, even if exceptions are thrown. It provides 'syntactic sugar' for try/finally blocks.

It's handy when you have two related operations which you'd like to execute as a pair, with a block of code in between. The classic example is opening a file, manipulating the file, then closing it:

```
with open('HadoopExam.txt', 'w') as f:
    f.write('Welcome to HadoopExam Learning Resources…')
```

The above with statement will automatically close the file after the nested block of code. (Continue reading to see exactly how the close occurs.) The advantage of using a with statement is that it is guaranteed to close the file no matter how the nested block exits. If an exception occurs before the end of the block, it will close the file before the exception is caught by an outer exception handler. If the nested block were to contain a return statement, or a continue or break statement, the with statement would automatically close the file in those cases, too.

QUESTION-84. What is the difference between {} dictionary and "OrderedDict"?

Ans: OrderedDict maintain the insertion order. In whatever order all the key value pairs are added to dictionary will be maintained and that cannot be possible with the regular dictionary.

QUESTION-85. Which is faster when we need to apply search operation on list and dictionary?

Ans : Dictionary will be faster than list, if you search based on the key.

Learn Python in Less than 8 Hours sitting at Home/@Desk

QUESTION-86. What is the purpose of the enumerate in Python?

Ans : The enumerate() function adds a counter to an iterable. So for each element in cursor, a tuple is produced with (counter, element); the for loop binds that to row_number and row, respectively.

```
elements = ('foo', 'bar', 'baz')
for count, elem in enumerate(elements):
...     print count, elem
...
0 foo
1 bar
2 baz
```

QUESTION-87. What is a regular expression?

Ans : A regular expression (regex or regexp for short) is a special text string for describing a search pattern. You can think of regular expressions as wildcards on steroids. You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager.

QUESTION-88. What is NumPy?

Ans: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

QUESTION-89. What is the matplotlib?
Ans: matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. SciPy makes use of matplotlib.

QUESTION-90. What is TkInter?
Ans: Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python. The name Tkinter comes from Tk interface.

QUESTION-91. What is the purpose of magic method __init__?

Ans : you don't have to invoke it directly. The invocation is realized behind the scenes. When you create an instance x of a class A with the statement "x = A()", Python will do the necessary calls to __new__ and __init__.

QUESTION-92. If Python fully object oriented language?
Ans : No, Python doesn't support strong encapsulation, which is only one of many features associated with the term "object-oriented".

The answer is simply philosophy. Guido doesn't like hiding things, and many in the Python community agree with him.

QUESTION-93. How can you ignore an exception?
Ans : Using pass , see below example

```
try:
  doSomething()
except:
  pass
```

Question-1: What do you mean by “Laxis” in any programming language?

Answer: a lexis: you can also say a dictionary, which is a set of words the in a language.

Question-2: What do you mean by Syntax?

Answer: Syntax is a set of rules for a programming language. For example, in natural English language your question must end with “?” Similarly, computer programming language has various rules too.

Question-3: Can computer directly understand programming language?

Answer: No, computer cannot understand Programming or High-Level Language directly. Computer can understand only Machine Language. Hence, programming languages needs to be converted into Machine Language.

Question-4: How can a computer convert a programming or high-level language to Machine Language?

Answer: There are two ways by which a programming language can be converted into Machine Language that is

- Compilation
- Interpretation

Question-5: What is compilation?

Answer: Converting high level programming language in a Machine Level language is known as compilation.

Question-6: What is interpreter?

Answer: In case of interpretation program has to be translated each time you want to run. It's not one-time activity as compiling once and run anywhere. There is interpreter required to translate your program each time it is translated. Hence, distributing just your source code would not help until and unless end user does not have interpreter.

Question-7: Any programming language does need both interpreter and compiler?

Answer: May be, programming language can fall either in compilation based on interpreter based. It can not be both. There are few languages which falls under both the category.

Question-8: How interpreter or compiler works?

Answer: Interpreter or compiler read your source file line by line. Check syntax and semantics of the code. If everything is fine, it moves to next line. In case there is any issue on a particular line. Compiler or interpreter will stop there and give error.

Question-9: Which one is better compiler based or interpreter-based programming language?

Answer: Both exists it means none of them is better than other. Both are having some advantages and disadvantages from other one.

Question-10: What are the advantages of compilation based high level programming language?

Answer: Following are the advantages of compilation-based languages

- Once code is translated it runs faster compare to interpreter-based translation.
- Person who wants to run or execute translated code, he does not need compiler.
- Once code is compiled it is stored as a Machine Language.

Question-11: What are the advantages of the Compilation based programming language?

Answer: Following are the advantages of the interpretation-based programming language

- In reality no need to translate code explicitly. It is taken care by programming language.
- As soon as you are done writing your program, it can be executed immediately, without you are explicitly translating it.
- Your code is stored as a simple text file only and not in the Machine Language format.
- There are different kind of computer architecture i.e., X86. You don't have write code for each different architecture.

Question-12: What are the disadvantages of the "compiler-based programming language"?

Answer: Following are the disadvantages of the compiler-based programming language

- Compiling a program is a time-consuming process.
- As soon as you change your code you first need to compile it then only you can run it.
- For each different computer architecture different compile is needed. For example, if you want to run your program on windows you need different compile, if you want to run on Mac, you need different compiler.

Question-13: What are the disadvantages of the interpreter-based programming language?

Answer: Following are the disadvantages of the interpreter-based programming language

- Interpreter based code does not run or executer very fast. Because it has to first translate your code, every time you want to execute it.
- It's slow.
- Anybody, who want to execute the code must have interpreter installed on their computer.

Question-14: Is Python interpreter-based programming language?

Answer: Yes, Python is interpreter-based programming language.

Question-15: How do you compare Python and Java for program translation?

Answer: These are the below important point you should know about Java and Python program translation

- Interpreter based programming language, while Java is a compiler-based language.
- Hence, you always see Java is faster than Python while you execute the code.
- To run Python code, you or whoever is executing it needs the interpreter installed on their computer.
- In case of Java only the program creator needs the compiler and he can give translated code to anybody and they can execute on their machine. However, Java runtime needs to be installed. (We can discuss this in Java course for more detail).

Question-16: What are the two version of Python exists, and are they compatible with each other?

Answer: There are two version of Python is available

- Python2
    - Any new development is not being done

    - Only fix or bugs are released.
  - Python3
    - Python2 and Python3 are not compatible with each other
    - It means program written in Python3 may not work on Python2 and vice versa.
    - If you want to upgrade Python version, then you have to re-write the entire code.
    - If you are writing new program you must do in Python3

Question-17: What is CPython?

Answer: Python Software Foundation maintain core Python programming language. This is also called Canonical Python. This Python is written using C Programming language. Hence, it is called CPython.

Question-18: What is Cython?

Answer: This automatically translate Python code into C programming language. Because C is much faster programming language. Hence, it can execute much faster.

Question-19: What is Jython?

Answer: It helps in communicate with existing Java infrastructure using Python. However, Jython follow Python 2 standards. There is no Jython conforming to Python3.

Question-20: When you run your program and you get EOF (End of File) error, what does that mean?

Answer: It means your code is not complete and it expect some more code to complete your program.

Question-21: What is a built-in function?

Answer: Function provided by programming language itself is called function for example “print” function. It does not require any module import. In Python 3.9 there are 69 built in functions.

Question-22: What are a Module functions?

Answer: Function can come from some additional component that is called Modules. Some of the modules provided by Python programming language itself and others you may have to install additionally to use them.

Question-23: What is Custom function in Python?

Answer: Sometime there are no functions available which fulfil your need. Hence, you create your own function and use them.

Question-24: What do you know about the function arguments in Python?

Answer: Argument is a part of functions. A function may or may not need arguments. And there is no limit how many arguments you can use for a function. Whether you use arguments or not, function must have a pair of parentheses. Arguments must be in between the pair of parentheses.

Question-25: What is the type of argument passed in print() function?

Answer: Print function can expect “Any” type of arguments. It means it can be String, Boolean, Number, Object etc.

Question-26: Does print function return any value?

Answer: No, print function does not return any value. It just creates a print effect.

Question-27: What is a function in a Python?

Answer: Python function is an instruction

Question-28: What do you know about Python single line instructions?

Answer:

- In Python there cannot be more than one instruction in a line.
- However, your program can have empty line
- However, your single instructions can be possible across line as below

Question-29: What is the use of escape character?

Answer: If we want to use any special character then we use escape character. As in below example

```
Print(“Hadoop Exam \n Learning resources”)
```

We want to generate new line middle of the string. Hence, we have used “\” escape character which says character “n” comes after “\” then it should generate new line string. There are similarly different behavior can be generated for various character by escaping them with “\”

Question-30: What are the Keyword arguments in Python?

Answer: Python functions support various keyword-based arguments. And this keyword has special meaning. For example, lets consider the print function.

```
print("Hadoop", "Exam", "Learning", "Resources", sep="$")

print("Hadoop", "Exam", "Learning", "Resources", end=" $ ")
```

In above example sep and end are keyword arguments. Where sep means, while print each word should be separated by the given string in this case “$” and end means entire string should end with the $. If we run above instructions, it would generate below output.

```
Hadoop$Exam$Learning$Resources
Hadoop Exam Learning Resources $
```

end=”\n” is a default behavior in print function.

Question-31: What is the “literal” in a Python Programming?

Answer: Literal means a data which represent a value by itself, and this has a fixed values. Following are the few examples of the literals. Which are of different types like int, float, String and Boolean

```
123
1000
9.99
“HadoopExam"
True
```

Question-32: When you have been given two values in Python “99” and 99. What is the difference?

Answer: In this “99” is a string type value while 99 without quote is representing integer type value. They both look similar but inside computer memory these both are stored differently. Numbers are stored in bit combinations while string is stored as series of characters. However, each character is also a series of bits only.

Question-33: What are the different types of number possible in Python?

Answer: There are two types of Number possible in Python programming as below

- Integers i.e. 100, 1, 7 etc.
- Floating-point numbers i.e. 9.9, 2.1, 2.1473743 etc.

Question-34: Is there any limit of maximum possible size of int and float number a computer can store?

Answer: Yes, there is a limit based on computer hardware architecture whether its 32 bit or 64 bits. And based on that minimum and maximum value of a numeric type is decided. Keep in mind both int and float value is stored differently in computer memory.

Question-35: What is the meaning of Type in a programming language?

Answer: Type is characteristic in a programming language which determine the numerical maximum and minimum possible values. Based on the type determined by Python will decide, how that literal should be stored in Memory.

Question-36: Which character is supported between numbers in Python , or _?

Answer: Do you know Python allows “_” in number value? This feature is specific to Python 3.6 and above. It was done to improve readability.

Question-37: How octal and Hexa-decimal number are presented in Python?

Answer: If Number value is preceded by (0-o), which is zero and “O” . It represents octal values. Octal values can hold each integer value between 0 to 7. Hexa-decimal should be preceded by 0X (Zero-X).

Question-38: What is the scientific notation in Python?

Answer: If your values is very big or small. You can use scientific notation to shorten them in program as below.

- All below are same except the last one
- Exponent always has to be an integer.

- Base value can be integer or may not be.

```
print(9e8)
print(90e7)
print(900e6)
print (900000000.0)
print (9.9e8)
print (900000000)
```

- Printing very small and big number.
- Last one is not a floating-point number.

```
print (0.00000000000000000000000000001)
print (10000000000000000000000000000.0)
print (10000000000000000000000000000)
```

Question-39: Can you please provide some features about the String in Python?

Answer: Below are some features you should remember about the string in Python

- String always needs a double quote.
- If your string itself require quotes then? Us escape character Or use apostrophe to create your string. If you want only one apostrophe in string, you have to escape it.
- Empty String, this does not have any character in it.

Question-40: What is None value in Python?

Answer: Any value which is absent can be represent using "None" in Python. This is called NoneType

Question-41: What is the difference between "/" and "//" as a mathematical operator in Python?

Answer:

- **"/" for Division**
  - It produces always float value.
- **"//" for division**
  - If you want to get Integer value after division use this.
  - Results are always rounded.
  - If you divide an integer by an integer, you will always get integer value else float.
  - Always rounded to nearest integer.

Question-42: What is Unary and Binary operator in Python?

Answer:

- Any operator which needs at least two values are binary Operator
- Add and Minus both works as Binary and Unary operator

Question-43: What do you mean by Expression in Python programing?

Answer: When data and operator both are combined, that is called expression.

Question-44: What do you mean by operator binding in Python?

Answer:

- When there are same operator multiple times continuously in same expression.
- Then Operator binding happens
- This will work from left to right direction
- Check below expression.
- What happen if you go from right to left. (**Fatal error**)
- That is called left side binding.
- However, there is an exception for exponential, which moves from right to left.

```
print (26 % 9 % 3)
print (27 % 9 % 3)
print (2 ** 2 ** 3)
```

Question-45: When you write an expression, and if there are operator with the same precedence, then how Python handle it?

Answer: You should always consider operator binding, when there are same precedence operators.

Question-46: What is the variable in Python programming?

Answer: All literals, if we want to hold them in memory then a container is required. That container is called "Variable".

Question-47: What is the feature you know for a variable?

Answer:

- All literals, if we want to hold them in memory then a container is required.
- That container is called "Variable"
- You have to give name to that container.
- You or developer has to create a variable.
- You can put anything in a variable, which is supported by Python Programming language.
- As soon as you assign a value to a variable, it would be created immediately.
- A variable represents a location in a memory area. Which can be used to store some values.

Question-48: What is the restriction for a variable in a Python Programming?

Answer: Below are some restrictions, you should remember while crating a variable

- Variable name can have upper- and lower-case letters.
- Variable name can have digit in it
- Variable name can have "_" underscore in it
- Name of the variable must start with a letter.
- Variable names are case sensitive

- Variable name cannot be from reserved keyword by Python.
- Space is not allowed in variable name.
- Variable name must not start with a number.

Question-49: What do you mean by a Keyword in a Python Programming?

Answer: These are the reserved keywords by Python programming and you should not not use them as a variable.

- Even your function name should not be using keyword
- Meaning of the Keyword is pre-defined by Python Programming language and should not be altered.

Question-50: Is Python static type or Dynamic type?

Answer: Python is a Dynamic type programming language.

Question-51: What do you mean by Python is a Dynamic programming language?

Answer: It means you don't have to provide type explicitly for a variable. And Python itself derive its type by looking at the variable.

- Python is a dynamic type language.
- Like Java is a static type language. Compiler need to know the type of each variable.

```
#In Java
int i=10
String name=" Amit"
```

```
#In Python
i=10
name=" Amit"
```

Question-52: What is the purpose and feature of the input function in Python and how it differ from print() function?

Answer:

- As we discussed, print() function does not return any value.
- It just print the calculated expression value as string on console.
- Based on function name, you would like to guess what it can do.
- Input() function reads the value provided and in program we can use it.
- Input() function makes your program more interactive.
- Now, you can say this can take input from voice, file, images etc.
- You have to store, input in a variable. If you want to use it further in your program.
- Input function can take argument as input as well.
- Your argument in a function will be printed on the console.
- Input to a function is always a String. And you should not directly use it in calculation.

```
myPlatform = input("What is the name of this learning platform \n")
```

```
print ("This is \"", myPlatform, " \" Provides various stuff")
```

Question-53: What is type casting in Python programming?

Answer: When you use input () function in Python it always consider it as a string, even you provide, int, float or Boolean value.

- Type casting will help you to convert your input to any other possible types.
- In above program let's use type casting to convert string into an integer.
- Try below code with various different inputs.
- **Input is always a String**

```
number = int (input ("Multiple Given Number \n"))
number=number*2
print(number)
```

```
number = float (input ("Multiple Given Number \n"))
number=number*2
print(number)
```

- You can cast number to string as well.

Question-54: How can you concatenate string in Python?

Answer: One option is we can use "+" operator.

Question-55: If you want to repeat same string multiple times in Python program, how can you do that?

Answer: If you want to print same string n number of times, you have to use "*" and number.

```
print ("Namaste !" *3 ) # Print "Namaste!" three times
```

Question-56: Equality operator is a left side binding or right-side binding?

Answer: Equality operator ("==") is a left side binding operator.

Question-57: What all are the possible indentation for loop or if condition?

Answer: You can use either tab character or n-number of spaces. However, four spaces is recommended. You should never mix the indentation.

Question-58: In Boolean operator, what is the 0 means?

Answer: If expression evaluates to 0 then if condition would consider as False. And any non-zero value would be considered as True.

Question-59: What is elif in Python?

Answer: It is a shorter form of else if.

Question-60: Is else block mandatory?

Answer: No, else block is not mandatory.

Question-61: If you have multiple elif in your code flow. And more than one conditions are true. Then what would happen?

```
if A>B:
   print ("A>B")
elif B>C:
   print ("B>C")
elif A>C:
   print ("A>C")
elif C>A:
   print ("C>A")
```

Answer: In this case, only one of the blocks in entire flow would be executed. Even multiple conditions are true.

Question-62: What is the iteration and possible ways of iteration in Python?

Answer:

- You can iterate particular statement as many times as your condition remain in force.
- There are multiple ways for iterations.
    - While Loop
    - For Loop

Question-63: How can you create infinite loop using while?

Answer: Its Simple

```
While(True):
```

This would be an infinite loop, as there is no condition to break the loop. Avoid doing this in your real program. And always use some condition to break the loop.

Question-64: How many iterations below code would be having

```
while(100):
   print("Continuous Printing")
   print("******************")
```

Answer: Infinite iteration. Non-zero, value is considered as True.

Question-65: How many iterations below code would be having?

```
while(0):
    print("Continuous Printing")
    print("******************")
```

Answer: Zero-iteration. Because zero is considered as False.

Question-66: How many iterations, below code would have

```
n=5
while(n):
    print("Current Value = " , n )
   n=n-1
```

Answer: This would have 5 iterations. Because, in each iteration, you are reducing value with 1.

Question-67: In Python, how can you create multi-line string?

Answer: To create multi-line string in Python you can use “”” (three times quote) as below.

```
print("""
This is a multiline string
===============================
Name : Amit Jain
Dept : IT
Salary : 100K
Title : Software Engineer
""")
```

Question-68: If you want to do iteration using for loop between 0 to 100. But in each iteration, you want to have only 20 iterations. How can you do that?

Answer: We can step with 5 in each iteration as below.

```
for i in range(0,100,5):
    print("Current value of i==", i)
```

- Here 3rd argument is used to increment and provide next value in sequence.

- First argument, from where to start
- Second argument, upto this number. But does not include itself.
- Third argument is for specifying, how much it should increment for next number in series.

Question-69: In below two different cases, how many iterations you can have

for i in range(1,1):

for i in range(2,1):

Answer: In each case, we would be having zero iterations.

- In range function if both 1st and 2nd argument same. This would never be executed.
- 2nd Argument must be bigger than first one, if you want to have some iterations.
- In above case only 3rd for loop will create iterations.

Question-70: What is the difference between break and continue, with respect to iteration?

Answer: break will exit from the loop and no further iteration would be done in current for loop. In case of continue, only current iteration would be skipped and it iterations would continue from next iteration.

- Break statement will break your loop and exit from for loop.
- Break statement does not make you exit from program.
- Hence, the line which are not part of for loop would be executed.

Question-71: Can while loop have else block?

Answer: Yes. For loop and while loop both has the else block. Which would be executed as soon as all iterations are done.

Question-72: What is Conjunction and Disjunction are in Python?

Answer: Both logical operator AND and OR are

- **AND logical operator (Conjunction)**
- **OR logical operator (Disjunction)**

Question-73: What is special about AND operator?

Answer:

AND operator

- If there are more than one condition. All of them satisfied or passed. Program would move ahead.
- To not pass, any condition can fail.

Question-74: What is special about OR operator?

Answer:

OR operator

- If Any of the condition is satisfied, statements in if condition would be executed.

Question-75: Which operator in Python is called toggling operator?

Answer: not operator is called toggling operator.

Question-76: What all are the bitwise operator?

Answer: Below all 4 are considered bitwise operator.

- & - Bitwise And
- | - Bitwise Or
- ^ - Bitwise exclusive or
- ~ - Bitwise negation

Question-77: Can you explain bitwise operator with example?

Answer: Yes, as below

X	Y	X & B	X | B	X ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

X	~X
1	0
0	1

- In case of & if both are 1, then output would be 1
- In case of | , if any 1 value is 1. Then it would generate 1
- In case of ^, if both are 1 or 0 then result would be 0.
- In case of ^ , if any one value is 1 then result would be 1

- Bitwise operator should be int and not float.
- Logical operator does not check on bit level.
- Bitwise operator considers each bit separately.

Question-78: What is bit shift operator in Python?

Answer:

Bit Shifting

- If you shift a bit by 1 place. It would either multiply that value or divide it.
- >> Right Shift = Divide by 2
- << Left Shift = Multiply by 2

Question-79: What is scalar variable?

Answer: Variable which holds single value is called scalar. For instance

X=10

Y=20

Here, X and Y both are scalar variable.

Question-80: What is the basic feature you know about the list in Python?

Answer:

- Each element in a list can be accessed by its position.
- Position index would always start with 0.
- List can be empty
- List can have another list in it.
- List name is a pointer to a memory location of a list.

Question-81: Which function you can use to find the length of the list?

Answer: You can use len() function for that as below.

listOfNames=['Amit' , 'John' , 'Venkat' , 'Simon']

print("List Length ", len(listOfNames))

Question-82: How can you delete a particular element from a list?

Answer: You can use "del" operator for deleting a particular element or sublist from a list.

listOfNames=['Amit' , 'John' , 'Venkat' , 'Simon']

print("List Length ", len(listOfNames))

#Below line would delete 3rd element from list.

del listOfNames[2]

Question-83: How do you interpret -ve indices for a list?

Answer:

- -ve indices access elements in reverse order.

- -1 represent last element in list
- -2 represent 2nd last element in list.

Question-84: What is the difference between method and function in Python?

Answer:

- In most of the programming language, you will find that function and method refer the same thing.
- But in Python this is not the case.
- Method is specific kind of function.
- It looks and behave same as function.
- However, it differs in a way it acts and being invoked.
- **Method is owned by data and it works on it.**
- **Function is owned by entire program/Module or code.**
- **Function does not belong to any data.**
- **Function get data, create new data, and produces a result.**
- **Method also gets data, create new data, and produce the result. However, in addition it also change the state of selected entity.**
- **Invoking a method requires some specification of the data from which method can be invoked.**
- **If you already know, Java Programming or any other OOPs programming language. You will find you are using methods.**
- **Invocation style for a function and a method as below**

Function	result = nameOfFunctin(arguments)
Method	result = **entity.**nameOfFunctin(arguments)

- **As per above, you can see that method can change the state of entity itself.**
- **To call method you always need entity, in which it is defined.**

Question-85: Can you give example of method and function, which you can use with list?

Answer: append() and insert() are the method for the list. However, len() is a function. Both append() and insert() method belong to list but function len() does not.

Question-86: When you use insert method to insert an element in a list, what would happen if you given position bigger than list length?

Answer: That element is added at the end of the list.

listOfNames=['Amit' , 'John' , 'Venkat' , 'Simon']

```
listOfNames.insert(100, 'XXXX')
```

In this case ‘XXXX’ would be added at the end of the list.

Question-87: What is the simplest way in Python to swap two variables, without creating a third variable?

Answer:

- This is a special way, provided in Python to swap variable.
- You don’t need to use any third temporary variable.
- Simple and straight forward.

```
first = 'John'

second='Mike'

#Swap two variables. Without creating third variable.

first, second = second, first
```

Question-88: In Python, list can store different types of data?

Answer: Yes, it can. Below is a valid list in Python

```
list = ["Amit" , True, ['Wow' , 'This' , 'is' , 'nested' ] , 1 , 'A']
```

Question-89: What is the difference between scalar variable and list variable in Python?

Answer:

- Name of the scalar variable is a content of that variable.
- Name of the list is not a name for its content. It refers the memory location of that content.
- In below program both listA and listB are pointing to same list in memory.
- Hence, if you update content of the listA then content you access using variable listB, is also same.
- In below program you updated first element of listA, but you can see same is updated in listB as well.
- In memory there is only and only 1 list created in this case. And both variable listA and listB are pointing to the same list.

```
listA= [1,2,3,4,5]

listB=listA

listA[0]=999
```

```
print(listB[0])
```

Question-90: What is list slicing?

Answer:

- If you want to create a new list from existing list then you can use slice.
- You can have exactly same copy of the list, which is duplicated in memory.
- If you want some portion of the list and create a new list then also you can use slice.
- In below program both listA and listB are different even in memory.

```
listA= [1,2,3,4,5]

#Creating a new list using existing list

listB=listA[:]

listA[0]=999
```

Question-91: What would happen when you do list slicing as below

```
listA= [1,2,3,4,5]

listB=listA[:]
```

Answer: Both listA and listB are having same content. But different list in Memory.

Question-92: Can you explain list slicing using -ve index?

Answer:

- If you see -1 is equal to first element in list.
- However, as per slicing logic -1 position would not be included.
- -1 (actually last element in list)
- 1 (represent 2nd element in list)
- So slice (1:-1) actually means.
- Starts with 2nd element and go till last element.
- However, last element would not be included.
- Hence, it would include 2,3,4

```
listA= [1,2,3,4,5]
#Creating a new list using existing list
listB=listA[1:-1]
print(listB)
```

Question-93: What would happen, while doing slicing you omit either start or end element?

Answer:

- If you omit start element in slice. Then it would consider the 0th index.
- If you omit last or end element in slice. Then sub-list would include all the elements till end from the start index.

```
listA= [1,2,3,4,5]

#Omiting first element

listB=listA[:3]

#Omiting last element

listC=listA[2:]
```

Question-94: What are the benefits of deleting list elements using del operator and slice?

Answer:

- If you want to delete some continuous elements from a list then you can use slice with del.
- Even using slice and del you can clean all the elements from the list.

```
listA= [1,2,3,4,5]


#Delete first 3 elements from list

del listA[:3]

print(listA)


#Now list has only 2 elements.

#This will delete 2nd element only

del listA[1:10]

print(listA)


#Re-create list

listA= [1,2,3,4,5]

print(listA)


#Delete all elements from list
```

```
del listA[:]

print(listA)


#Deleting list itself.

del listA

#print(listA)
```

Question-95: If you want to check an element exists in a list or not, then how can you do that?

Answer: In this case you can use in and not in operator.

```
listA= [1,2,3,4,5]

print (5 in listA)

print (6 not in listA)

print (7 in listA)

print (5 not in listA)
```

Question-96: What is the built-in functions?

Answer: These are pre-defined (it means their behavior is already implemented by Python) to make your life smooth.

Question-97: Can you explain function mathematically?

Answer:

- In maths, you have seen equation something like below

```
Y=f(x)
```

- In above f(x) is a function, which find the value y for given x.
- But what to do with X is defined in a function f(x)
- Suppose your function does below thing

```
f(x) = x*x + 10
```

- It multiple x with itself and then add 10.
- Similarly, you can have whatever function, you want to define you can do.
- Hence, in this case for x=2, y=2*2+10= 14

Question-98: What are the advantages of the function?

Answer:

- Once, you define a function, you can use as many times as you want.
- Usually, things you want to do more than once in your program. You should create function for that.
- Hence, you define that once and use wherever you want.
- This would keep your code clean. And tools would never complain, you have duplicate code.
- Even, sometime your algorithm is very complex, even it is called once. You should rather break or write a separate function and put your algorithm in that function.
- Having function in your code, is always considered good practice. And your manager and architect would remain happy with your coding style.
- To become a good programmer, you have to divide code in functions, with each related unit isolated.
- If you write each functionality in a separate function then it can be very easily unit tested.

Question-99: What is decomposition?

Answer: The process to divide your code in function is called decomposition. It means divide the problem statement in multiple parts, which can be individually developed and tested. If you have decomposed your problem very well, then it can be independently developed and different developer can work independently.

Question-100: What are the all-possible ways, you can get function in Python?

Answer: You can have functions for below three places

- **Built-in functions:** Python language: Python programming language has pre-defined functions. Like you have been using len(), print(), input() etc. without doing anything additional. In Python 3.0 there are around 69 built-in functions available.
- **Functions from modules**: If you install required module on your Python language then you can have access to those functions form those modules.
- **Custom functions**: You can have your own custom functions defined and use them wherever you want.
- **Lambda function**

Question-101: How do you define a function in Python?

Answer:

- You have to use a def keyword to define a function.
- You must have thought for the proper and valid function name.
- And you should know, what should be the behavior for your function.

```
def my_function_name():
      function_body
```

- In function body you are going to write the behavior of the function.
- Function name should follow the same naming convention as variables.
- Function name should end with () parenthesis.
- If your function requires some arguments then put them in () parenthesis.
- When you give function name, then that line should end with ":" colon.
- Function body is collection of instructions, and it can be as big as you want.
- But avoid creating long function body. Rather break them in multiple functions, definition.

Question-102: Can you invoke a function before you define it?

Answer: No, Remember before invoking function, you have to define it. If you do that then you will get error something like below

This would give error that "NameError: name 'calculate' is not defined"

Here, calculate is a function. Which you are trying to invoke before even defining it.

Question-103: In what order Python read the code?

Answer: Python, read code from top to bottom (that is not true for Java). Hence, in same file you have to define function at the top (or before you are using it) to use it later.

Question-104: Can you use same name of a variable in a function?

Answer: You can, but you should never do that.

- You should not use same name as function and variable in same program.
- In below code your function can never/ever be called. Because, you have created a variable with the same name as function. Which overwrites function as variable. (Not a good practice, don't do that).

```
def calculate():
  print("You are trying to calculate")

i=0
calculate=1

while(i<10):
  print(calculate)
  i=i+1
```

Question-105: What do you mean by Function signature?

Answer:

Function signature

- Function signature must match when you call a function.

- In below example, function would give an error.
- Because your function signature is expecting one argument, but while calling you are not providing any argument.

```
def calculate(x):
    print("Your calculated value is ", x*x+10)



i=0
while(i<10):
    x=int(input("Please enter some number : "))
    calculate()
    i=i+1
```

Question-106: What is a parameter and arguments for a function?

Answer:

- As we have seen in previous calculate(**x**) function, we are passing one parameter, named as x.
- There are two things, which you need to remember
  - **Parameters:** They are part of the function definitions. And remain only in function and not accessible outside function.
  - **Arguments**: While invoking a function, you must have to pass exact same number of arguments as number of parameters defined on function. In case of calculate(x) has one parameter, so while invoking (calling) this function should have pass one argument.

Question-107: What is parameter shadowing?

Answer:

Parameter's shadowing

- When your parameter's name and argument's name are same then that behavior is called shadowing.
- This is absolutely ok to have both parameters and argument's name as same.
- Shadowing happens only in the function name.

Question-108: What are Positional Arguments?

Answer:

Positional Arguments

- Most of the programmer by default consider parameters are based on positions.
- We also recommend you always follow that. So that anybody else looking at the code you have written would same thing in mind.
- Hence, while invoking function, in whatever order you pass parameters. Python is also considering in same order.

- In above example x is at first position, then in function definition it is mapped with salary. Similarly, y will be mapped with hike. And Z would be mapped with bonus.

Question-109: What is a “Keyword based Argument Passing”?

Answer:

Keyword based argument passing

- If you have remember print() function has Keyword based argument passing.
- Hence, Python is more flexible, you can even use keyword-based argument-or position-based parameter passing in your own function as well.
- Keyword based argument passing means, meaning of the argument is derived by its name and not by its position.

  totalSalary(bonus=z, salary=x, hike=y)

- When you are using keyword-based argument, you have to provide which argument should be mapped which parameter of the function.
- Certainly, you need to type more in this case.
- Once, you become expert in Python. Most of the time , you will be using position based argument passing.

Question-110: Can you mix position based and keyword-based arguments?

Answer: Yes, you can do that.

Mixing positional and keyword-based arguments

- You can mix both positional and keyword-based arguments.
- **Rule**: You must always put positional arguments, before keyword-based arguments.
- You must not pass same parameter with argument based and position based in single invocation. This would give error as below.

Question-111: Can you have default value for a parameter in a function?

Answer: Yes, you can have.

def totalSalary(salary,hike, bonus=10000):
In this case, bonus is having a default value. While invoking function, if you don’t pass the bonus value. Then it would consider a default value.

- If you pass value, then this will override the default value.
- If you don’t pass value then it will use default value defined, in function definition.
- In case of argument-based parameter passing, order of parameter does not matter.
- However, if you are mixing argument based and position-based parameter passing then you have to use positional parameters before that.

Question-112: Can a function return a value?

Answer: Yes, a function can return a value.

- There are function exists which can return or not return values.
- If you see build in function
    - print(): this does not return any value. Just print on console.
    - input(): this function return a value which user entered.

Question-113: It is true, that in Python every function returns a value?

Answer: Yes, if you see some function does not have return statement. It means it is implicitly returning None.

Question-114: Can you tell me some important point about return statement?

Answer:

- You cannot have more than 1 return statement in a function definition.
- However, as soon as first return statement is reached. Remaining would be ignored.
- As soon as first return statement is reached function would stop that line and return and nothing would be executed from that line onwards.
- Lines after return statement would never be reached.

Question-115: What is None type in Python?

Answer:

- None is not a value.
- None means, nothing known about it.
- You should not use None as part of any expression.
- Below line would not work and give error.
- However, you can assign None as a value to a variable.
- Your function can return a None value.
- If function does not return a value, then by default it would consider that function is returning a None value.

Question-116: What is the meaning, when a function returns a None?

Answer: If a function returns a None value. Either it means return value is not expected from that function or there is some mistake in that function.

Question-117: Can a function return a list?

Answer:

- Your function can return a list.
- Your function can return list, string, char, int, float, Boolean, None all.
- However, function can return only one this at a time.
- Your function can not return more than one value.
- If you want to return more than one value then pack them in list and then return.

Question-118: What is the scope of a variable defined in a function?

Answer:

- Variable created in a function are accessible only inside function.
- You cannot access those variables outside of function.
- Variable defined in a function is not accessible outside that function.

Question-119: Can you access a variable, which is defined in a Program but not in function?

Answer: Yes,

- Variable defined outside a function and part of program body is accessible inside function.

Question-120: What would happen when you have same variable name in function and program body?

Answer:

- If you are having same variable name in function and program body.
- Then variable which is insider function would shadow the variable defined in program body.
- When you create a variable inside a function, it means it create a new variable and scope remain for that variable in the function only.
- A variable declared outside function and part of program body. Is accessible inside the function. However, new copy of that variable is created if variable with same name is created inside a function.

Question-121: What is the use of global keyword?

Answer:

global keyword

- If you want to access variable created inside function, outside function (from program body) it has to be a global variable.
- In below program, you can access variable called "name" outside function as well and it holds the value which is overwritten in function.
- Try below program with global and without global keyword.

Question-122: What is the difference between scalar and list with regards to its value?

Answer:

- Name of the variable in case of scalar represent value itself.
- Name of the variable in case of list represent memory location of the list.

Question-123: What would happen when you pass scalar value to a function?

Answer:

- When you pass a scalar variable to a function.
- It makes a copy of that variable and pass it to function and work on that variable.
- As you can see in below program, value of variable "i" is not getting updated.

Question-124: What would happened when you pass non-scalar value to a function as an argument for example list?

Answer:

- If you use non-scalar value.
- Then not a copy of that variable is passed.
- Rather location of that variable would be passed to function.
- And any operation you do using memory location, then variable itself updated.
- When you are passing list to a function. It is actually passing its address or you can say reference.
- Hence, any update you do on the list inside the function. That would result actual data getting updated in memory.

Question-125: What is a recursion?

Answer: A function when call itself. It is called recursion. While using recursion, you must have termination condition.

Question-126: What is important thing you should remember about recursion function?

Answer: In case of recursion, you have to have termination condition. Else it would go in infinite loop.

Termination condition: If your code does not have termination condition then recursion will never end. It can go in infinite loop.

Question-127: Is recursion more memory efficient?

Answer: No, recursion is not memory efficient.

Question-128: What is Mutable means?

Answer:

Mutable:

- This can be changed (updated, altered etc.)
- List is a mutable data type.
- We can update list itself by using its on method i.e. list.append(n)

Question-129: What is immutable means?

Answer:

Immutable

  - This cannot be modified.
  - It means you cannot directly update/delete or alter that data type.
  - If you want to modify then you have to make a new modified copy of that data type.
  - New copy, it means it’s a new data and not same.
  - In Python one data type which is immutable is a **tuple**.

Question-130: What is sequence type in Python?

Answer:

Sequence type

- It is a data type, which can store 0 or more values.
- This must be Iterable.
- List is one example of such sequence.
- Another example of Sequence is Tuple.

Question-131: What is Tuple in Python?

Answer:

Tuple

- Tuple is immutable, sequence data type.
- Tuple can behave like list. But you cannot modify it directly.
- To create a list, you use square bracket []. While to create a Tuple, you use parenthesis [].
- Or if you simply assign comma separated values to a variable then also, it would create a Tuple.
- Similar to list, Tuple can also hold mix data type values.
- A Tuple can have another Tuple in it.
- A Tuple can have List in it.
- A Tuple can be empty.
- A Tuple can have single value.
- To create a single value Tuple, you must end it with “,” comma. While creation.

Question-132: Can you modify a Tuple?

Answer: No, you cannot modify a Tuple.

Question-133: How can you join two tuple?

Answer:

Joining Tuples

- To join two or more tuples, you can use “+” operator.

```
tupleA= 'The', 'Hadoop'
tupleB= 'Exam' , 'Learning' , 'Resources'
```

```
tuple=tupleA + tupleB

print(tuple)
```

Question-134: How can you swap tuple?

Answer:

You can swap tuple same way as you swap variable

```
tupleInt= 1,2,3,4,5

tupleChar= 'A', 'B', 'C', 'D'

tupleString= 'The' , 'HadoopExam'

tupleFloat= 1.0,2.0,3.0

tupleInt , tupleChar , tupleString, tupleFloat = tupleString, tupleFloat ,tupleInt , tupleChar
```

Question-135: Can you please tell something about Dictionary in Python?

Answer:

- Dictionary is a Mutable datatype.
- Dictionary is not a sequence type.
- Dictionary is a collection of two elements (or pair) in it.
- One element is called key and other one is called value.
- That's why it is known as key-value pair.
- Some examples you can think of key-value pair as below.
  - A->Apple
  - B->Book
  - C-> Cat
- In dictionary you can have as many key-value pair as you want.
- You can have unique key in a dictionary.
- Dictionary object itself is mutable, but whatever you use for key, has to be immutable.
  - So possible types for key could be string, float, int etc.
  - You cannot have list as key for dictionary. Because list is a mutable data type.
- Dictionary is not a list.
- len() function on a dictionary return, number of key-value pairs in a dictionary.
- You can search element is a dictionary using key only and not based on value. Hence, dictionary is considered as One-way tool.

Question-136: Dictionary key is mutable or immutable?

Answer: Keys, in dictionary are always immutable. But dictionary itself is mutable data structure.

Question-137: Can you convert a Tuple to a dictionary?

Answer: Yes, you can. For that you have to use dict() method.

Question-138: Does dictionary preserve order?

Answer: Yes, after Python 3.6 . Dictionary preserve order. Before that it was not.

Question-139: How can you extract all the keys from a dictionary?

Answer: You can use keys() method for that.

Question-140: How can you extract all values from a dictionary?

Answer: For that you can use values() method.

Question-141: How can you extract all key-value from a dictionary?

Answer: You can use items() method for that.

Question-142: What popitem() method do on a dictionary?

Answer: It removes the last element from dictionary and also return the same.

Question-143: Which are the collection data type is immutable in Python?

Answer: Tuple is immutable. However, list and dictionary are mutable.

Question-144: Which function you can use to convert a tuple to a list and vice versa?

Answer: You can use tuple() function to convert a list into tuple. And similarly, you can use list() method to convert a list into tuple.

Question-145: How can you remove all elements from a dictionary?

Answer: You can use clear() method for that.

Question-146: How can you copy a dictionary to another dictionary?

Answer: You can use copy method for that.

Question-147: How can you unpack a Tuple?

Answer:

- You can unpack a Tuple in variables.
- However, while unpacking number of variables in left hand side in same number as they are in tuple.

#Creating a Tuple

```
tuple=('Amit', 'John', 'Mike', 'Simon', 'Venkat')

#You can unpack a Tuple in variables as below
a,b,c,d, e = tuple
```

Question-148: Is Tuple mutable?

Answer: No, Tuple is not Mutable.

- Apache Spark Professional Training
- Apache OOZie (Hadoop workflow) Professional Training
- Beginner AWS Training Course- **(HETRNAWS101)**
- Core Java 1z0-808 Exam training
- JAX-WS (Java WebService HandsOn Training)
- Scala Programming Training
- Python Programming Training
- Hortonworks Administration Professional Trainings

MAPR HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- **MapR Hadoop Developer Certification**
- **MapR HBase NoSQL Certification**
- **MapR Spark Developer Certification (In Progress)**

HORTONWORKS HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- HDPCD : NO Java (Hortonworks Developer Certification)
- HDPCD : Spark (Spark Developer Certifications)
- HDPCA : Hortonworks Administration Certification
- Hortonworks Administration Professional Trainings

CLOUDERA HADOOP AND SPARK CERTIFICATION (AVAILABLE)

- CCA131 : Hadoop Administrator
- CCA-175 Cloudera® (Hadoop and Spark Developer)
- CCP:DE575 : Cloudera® Data Engineer Certification
- CCA159 : Cloudera Data Analyst Certifications

DATABRICKSA OREILLY SPARK CERTIFICATION (AVAILABLE)

- Apache Spark Developer

AWS: AMAZON WEBSERVICE CERTIFICATION (AVAILABLE)

- AWS Solution Architect : Associate
- AWS Solution Architect: Professional
- AWS Developer : Associate
- AWS Sysops Admin : Associate

MICROSOFT AZURE CERTIFICATION (AVAILABLE)

- Azure 70-532
- Azure 70-533

DATA SCIENCE CERTIFICATION (AVAILABLE)

- EMC E20-007

EMC CERTIFICATIONS (AVAILABLE)

- EMC E20-007

SAS ANALYTICS CERTIFICATION (AVAILABLE)

- SAS Base A00-211
- SAS Advanced A00-212
- SAS Analytics : A00-240
- SAS Administrator : A00-250

ORACLE JAVA CERTIFICATION (AVAILABLE)

- Java 1z0-808
- Java 1z0-809
- Java 1z0-897 (Java WebService Certification)

ORACLE DATABASE CLOUD CERTIFICATION (AVAILABLE)

- 1z0-060 (Oracle 12c)
- 1z0-061 (Oracle 12c)

Subscribe Here for Regular Updates: Like New Training Module launched

Become Author and Trainer: We are looking for Author (Writing Technical Books) and Trainer (Creating Training Material): **No Compromise on Quality.**

Benefit: You will get very good revenue sharing. Please drop us an email to hadoopexam@gmail.com (For the skills, you feel you are master)

We are sure, you are good at least one technology. Don't limit your potential, contact us immediately with your skill. Our expert team will contact you with more detail. You training and Books will reach to all our existing network and with our expert marketing team we will help you to reach as much as technical professional, with our Smart Advertising network. Contact us with sending an email hadoopexam@gmail.com

Opportunity to share your knowledge with all learners who are in need. We are helping 1000's of learners since last 4 years and established ourselves with Quality low cost material.