1. **What is Python? List some popular applications of Python in the world of technology?**

- Python is a widely used general-purpose, high-level programming language.
- Python uses most of the Object-Oriented programming concepts. But we can also do functional programming in Python.
- Python is a multi-paradigm programming language. We can do functional, procedural and object-oriented programming with the help of Python.
- Python is a interpreted language. From the source code, the Python program runs directly. It turns the source code into intermediate language code, which is then again translated into machine language that needs to be run.
- Python does not require compilation before running, unlike Java and C.
- Python used for some popular applications:
  - System Scripting
  - Web Development
  - Game Development
  - Software Development
  - Complex Mathematics

2. What are the main benefits of using Python?

- **Easy to learn**: Python is simple language. It is easy to learn for a new programmer.
- **Large library**: There is a large library for utilities in Python that can be used for different kinds of applications.
- **Readability**: Python has a variety of statements and expressions that are quite readable and very explicit in their use. It increases the readability of overall code.
- **Memory management**: Python is very efficient in memory management. For a large data set like Big Data, it is much easier to program in Python.
- **Complex built-in Data types**: Python has built-in Complex data types like list, set, dict etc. These data types give very good performance as well as save time in coding new features.
- **Faster**: Though Python code is interpreted, still Python has very fast performance.
- **Wide usage**: Python is widely used among different organizations for different projects. Due to this wide usage, there are thousands of add-ons available for use with Python.
- Dynamically Typed Language
- Presence of Third-Party Modules
- Open Source and Community Development
- Portable across Operating Systems and Interactive

3. Is python case sensitive?

Yes, python is case-sensitive it cares about case- lowercase or uppercase. it distinguishes between identifiers like myname and Myname.

4. **What is the use of PEP 8 in Python?**

   The Python Enhancement Proposal, also known as PEP 8, is a document that provides instructions on how to write Python code. Coding conventions are about indentation, formatting, tabs, maximum line length, imports, line spacing etc.

   We use PEP 8 to bring consistency in our code. it is easier for other developers to read the code.

5. **What is namespace in Python?**

   It is a naming system used to make sure that all names are unique to avoid naming conflicts. A Namespace in Python is a mapping between a name and an object. It is currently implemented as Python dictionary.

6. **How will you improve the performance of a program in Python?**

   There are many ways to improve the performance of program. Some of these are as follows:

   - **Data Structure**: We must select the right data structure for our purpose in a Python program.
   - **Standard Library**: Wherever possible, we should use methods from standard library. Methods implemented in standard library have much better performance than user implementation.
   - **Abstraction**: a lot of abstraction and indirection can cause slow performance of a program. We should remove the redundant abstraction in code.
   - **Algorithm**: Use of right algorithm can make a big difference in a program. We must find and select the suitable algorithm to solve our problem with high performance.

7. **Describe the Python Functions?**

   A function is a piece of code that is only written once and can be executed whenever the program calls for it. To define a Python function, the **def** keyword is used.

8. **Types of functions?**

   - **Built-In Functions:** count (), duplicate (), len (), list (), range (), reversed () ,input(),Map(),reduce(),filter() etc.
   - **User-defined Functions:** User-defined functions are functions that are defined by a user.
   - **Anonymous functions:** Because they are not declared using the standard def keyword, these functions are also referred to as lambda functions.

9. **What is a lambda function? Why lambda forms in python does not have statements?**

   An anonymous function is known as a lambda function. We must use lambda keyword for creating a lambda expression.

Wherever we need a function, we use a lambda expression. This function can have any number of parameters but, can have just one statement.

A lambda function does not have statements as it is used to make new function and then return them at runtime.

Example:
```
a = lambda x,y : x+y
print(a(5, 6))
```
Output:
11

10. What are Comments in python?

single line comment - #
multiline comment - """ """

11. **What is variable in python?**

Variables: can store data of different types.

Example:
```
course_name = "BigData"
course_fee = 30000
course_rating = 4.9
is_starting = True
total_revenue = None
```

12. **What are local variables and global variables in Python? How can you share global variables across modules?**

Global Variables:
Variables declared outside a function or in global space and are not specified to any function are called global variables. These variables can be accessed by any function in the program.

Local Variables:
Any variable declared inside a function and are unique to that function is known as a local variable. It cannot be accessed outside of the function.

13. How will you share variables across modules in Python?

We can create a common module with variables that we want to share. This common module

can be imported in all the other modules. In this way, all the shared variables will be in one module and available for all other modules as well.

14. Why are local variable names beginning with an underscore discouraged?

They are used to indicate a private variable of a class.
As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

15. What is the different between data type vs data structure in python?

1.Data type: Specifies the kind of data**.**

- string: It can represent a sequence of characters.
- int: It is used for Integers
- long: It is used for very large integers of non-limited length.
- float: It is used for decimal numbers.
- complex: This one is for representing complex numbers
- Boolean: true or false

2. Data structures: collection of items

- List : ordered collection of items of mutable items
- Tuple: ordered collection of items of immutable items
- Dictionary: unordered collection of items of key value pairs
- Set: Unordered collection of unique items

16. What are the different built-in data types available in Python?

Python has the following built-in data types by default

- **Text Type:** strings
- **Numeric type**: Int,Long,Float,complex
- **Sequence type:** List,tuple,String
- **Mapping Type:** dictionary, it is a collection of key and value pairs.
- **Set Types:** These are unordered collections.
  - set: This is a collection of unique objects.
  - frozen set: This is a collection of unique immutable objects.
- **Boolean Type**: bool
- **Binary Type**: bytes,byte array
  - bytes: This is a sequence of integers in the range of 0-255.
  - byte array: like bytes, but mutable (see below); only available in Python 3.x
- **None Type**: None

17. What is the difference between a sequential Data structure and unordered Data structure?

1.sequential Data structure: can't change ordering

1.list = [1,2,3,4,5]
2.tuple = (1,2,3,4,5)
3.string

2.Non-sequence/unordered Data structure: ordering not important

1. dictionary ={"animal":"cat","fruits":"apple","model":15}
2. set = {1,2,3,4,5}

18. What is the difference between a Mutable Data structure and an Immutable Data structure?

- **Mutable Data structure** can be edited i.e., they can change at runtime.
  Example: List, Dictionary, etc.

- **Immutable Data structure** cannot be edited i.e., they cannot change at runtime.
  Example: String, Tuple, etc.

19. What is type() In python?

type(): Gives you datatype of given variable

Example:

print datatype of given variable?

```
course_name = "BigData"
course_fee = 30000
course_rating = 4.9
is_starting = True
total_revenue = None

print(type(course_name))
print(type(course_fee))
print(type(course_rating))
print(type(is_starting))
print(type(total_revenue ))
```

20. What is type conversion /type casting in Python?

Type conversion refers to the conversion of one data type into another.

- int() – converts any data type into integer type
- float() – converts any data type into float type
- hex() – converts integers to hexadecimal
- oct() – converts integer to octal
- tuple() – This function is used to convert to a tuple.
- set() – This function returns the type after converting to set.
- list() – This function is used to convert any data type to a list type.
- dict() – This function is used to convert a tuple of order (key,value) into a dictionary.
- str() – Used to convert integer into a string.
- complex(real,imag) – This function converts real numbers to complex(real,imag) number.
- ord() – returns an integer value that is equivalent to the Unicode value of the provided character.

Example1:

print Int value by taking string?

```
income = "1000"

New_income = int(income)+(int(income)*.1)

print(New_income)
```

Example2:

```
UpperCase_Char =ord( 'A')

lowercase_char = ord('a')

Special_Char = ord('$')

num = ord('9')

print('The Unicodes are:', UpperCase_Char, lowercase_char, Special_Char, num)
```

Output:

```
The Unicodes are: 65 97 36 57
```

21. What is the use of frozen set in Python?

This is a collection of unique immutable objects. Once we have set the values in a frozen set, we cannot change and cannot use and update methods from set.

we can use the objects in frozen set as keys in a Dictionary.

22. What is List in Python?

1.collection of different datatypes inside square brackets [].
2.ordered collection
3.mutable

23. What is Tuple in Python?

1.collection of different datatypes inside round brackets () .
2.ordered collection
3.immutable
4.Tuple takes much lesser space than a List in Python.
5.Tuple is faster than a List in Python. So it gives us good performance.
6.Since Tuple is immutable, we can use it in cases like Dictionary creation.

24. What is tuple unpacking?

First, let's discuss tuple packing. It is a way to pack a set of values into a tuple.

```
>>> mytuple=3,4,5
>>> mytuple
(3, 4, 5)
```

This packs 3, 4, and 5 into mytuple.

Now, we will unpack the values from the tuple into variables x, y, and z.

```
>>> x,y,z=mytuple
>>> x+y+z
12
```

25. What is set in python?

1.set is unordered collection
2.each element is unique and is of immutable type(it won't accept list datatypes)
3.it is mutable datatype so we can add and remove elements

26. What is Dictionary in Python?

1.collection of key value pairs in {}.
2.values are mutable : add or remove value of particular key
3.keys are immutable: like it will not accept lists as keys
4.unordered collection

27. How can we create a dictionary with ordered set of keys in Python?

In a normal dictionary in Python, there is no order maintained between keys. To solve this problem, we can use OrderDict class in Python.

This class is available for use since version 2.7.It is similar to a dictionary in Python, but it maintains the insertion order of keys in the dictionary collection.

28. What is the use of zip() function in Python?

In Python, we have a built-in function zip() that can be used to aggregate all the Iterable objects of an Iterator. We can use it to aggregate Iterable objects from two iterators as well.

By using zip() function we can divide our input data from different sources into fixed number of sets.

Example:

```
list_1 = ['a', 'b', 'c']
list_2 = ['1', '2', '3']
for a, b in zip(list_1, list_2):
    print a, b
```

Output:

```
a1

b2

c3
```

29. What are the different types of operators in Python?

1. **Arithmetic Operators:** perform basic arithmetic operations. For example, "+" is used to add.
2. **Relational operators/comparison operators** are used to comparing values. These operators test the conditions and then return a Boolean value.
3. **Assignment Operators:** are used to assigning values to the variables.
4. **Logical Operators:** are used to perform logical operations like And, Or, and Not.
5. **Membership Operators:** are used to check if a sequence is presented in an object. In and not in operators.
6. **Identity Operators:** they are used to check two variables that are in the same memory area. is and is not operators.
7. **Bitwise Operators:** are used to performing operations over the bits. The binary operators (&, |, OR) work on bits.

30. What are Arithmetic Operators?

1. addition (+)
2. multiplication (*)
3. subtraction (*)
4. division (/)
5. modulo (%) *gives reminder
6. floor (//) *gives interger
7. power (**)

Example:

```
a =10
b =10

print('addition:',a+b)
print('subtraction:',a-b)
print('multiplication:',a*b)
print('division:',a/b)
print('floor division:',a//b)
print('module:',a%b)
print('exponent',a**b)
```

31. What are Relational operators/comparison operators?

1.==

2.>

3.<

4.>=

5.<=

6.!=

Example:

```
a ='sandeep'
b ='sandeep is big data Engineer'
```

```
print('a is equal to b:',a==b)
print('a is greaterthen b:',a>b)
print('a is lessthen b:',a<b)
print('a is greaterthen or equal b:',a>=b)
print('a is lessthen or equal b:',a<=b)
print('a is not equal b:',a!=b)
```

32. What are the Assignment Operators?

1. =	Eg: x = 5	same as x = 5
2. +=	Eg: x += 3	same as x = x + 3
3. -=	Eg: x -= 3	same as x = x - 3
4. *=	Eg: x *= 3	same as x = x * 3
5. /=	Eg: x /= 3	same as x = x / 3
6. %=	Eg: x %= 3	same as x = x % 3
7. //=	Eg: x //= 3	same as x = x // 3
8. **=	Eg: x **= 3	same as x = x ** 3

33. What are the logical Operaters?

1. and
2. or
3. not

Example:

```
print(True and False)
print(True and True)
print(True or False)
print(False or True)
print(not True)
print(not False)
```

34. What are membership operators?

in and not in operators

Example:

```
Name = "Sandeep Nookala"
print(sandeep in Name)
print(sandeep not in Name)
```

35. What are identity operators?

is and is not operators

Example:

```
a ='sandeep'
b ='sandeep is big data Engineer'

print( a is b)
print(a is not b)
```

36. Bitwise Operators?

The binary operators (&, |,**~,^,>>,<<**) work on bits.

Example:

```
a = 10
b = 4

print("a & b =", a & b)   # Print bitwise AND operation

print("a | b =", a | b)    # Print bitwise OR operation

print("a ^ b =", a ^ b)   # print bitwise XOR operation

print("~a =", ~a)  # Print bitwise NOT operation

print("a >> 1 =", a >> 1)  # print bitwise right shift operator

print("b >> 1 =", b >> 1)  # print bitwise right shift operator

print("a << 1 =", a << 1) # print bitwise left shift operator

print("b << 1 =", b << 1) # print bitwise left shift operator
```

37. What is the use of // operator in Python?

Python provides // operator to perform floor division of a number by another. The result of // operator is a whole number (without decimal part).

Example:

```
10// 4 = 2
-10//4 = -3
```

38. What are Conditional Statements?

1.If: used to execute a block of code only when a specific condition is met.
2.elif: used to check multiple conditions in sequence and execute different code blocks depending on which condition is true.
3.else: used to execute a different block of code if the if condition is False.
4.nested If: if statement inside another if statement.

39. What are loops?

There are two types of loops in Python, for and while.
1.for: For loops iterate over a given sequence.
2.while: While loops repeat as long as a certain boolean condition is met.

40. What is a break, continue, and pass in Python?

- **break** statement is used to terminate the loop. when some condition is met and the control is transferred to the next statement.

- **Continue** Allows skipping some part of a loop when some specific condition is met and the control is transferred to the beginning of the loop.

- **Pass** is to do nothing. It is just a placeholder for a statement that is required for syntax purpose. It does not execute any code or command. This is basically a null operation.

41. Can we Pass a function as an argument in Python?

Functions can be passed as parameters to other functions because they are objects. Higher-order functions are functions that can take other functions as arguments.

42. What is swapcase() function in Python?

is to change all uppercase characters into lowercase ones and vice versa.

43. **What is shuffle () method?**

It rearranges components each time when it is called and creates different result. Used to randomize the items of a list.

Example:
```
from random import shuffle
x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
```

```
shuffle(x)
print(x)
```
The output of the following code is as below.
['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']

44. What are *args and *kwargs?

To pass a variable number of arguments to a function we use the special syntax *args and **kwargs in the function specification.

45. What are special characters?

1. \: escape character
2. \n: newline
3. \t: tab space

46. What is module in python?

Module:is a single file(.py) containing python code. it can include functions classes and variables that can reuse in other programs.

Example:

1.mymodule.py

```
person = {'name': 'sandeep','age':30}
def say_hello(name):
   return print(f"hello,{name}!,How are you?")

def say_bye(name):
   return print(f"hello,{name},take care!")
```

2.practice.py

```
import mymodule
mymodule.say_hello('sandeep')
mymodule.say_bye('padma')

from mymodule import person
print(person['age'])
```

output:

hello,sandeep!,How are you?

hello,padma,take care!
30

47. What is package in python?

Packages: collection of modules organized in directory with an __init__.py file.it allows to structure your python projects logically.

Example:

```
my_pacakage/
  __init__.py
    math_utils.py
    mymodule.py
    string_utils.py
```

from my_pacakage import math_utils,mymodule,string_utils

48. What is library in python?

library:A library is a collection of modules and packages that provide pre-written functionality for your program.libraries are typically larger and more feature rich than packages or modules.

Example:

```
import pandas as pd
import seaborn as sb
```

49. What is pip?

pip: pip stands for "pip installs packages".it is tha package manager for python that allows you to install, update and manage python libraries/packages from python package index(PyPi)

Example:

```
pip install pandas
pip install seaborn
```

50. **What is the difference between Python Arrays and lists?**

Arrays and lists,have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements.

Example:

```
import array as arr
My_Array=arr.array('i',[1,2,3,4])
My_list=[1,'abc',1.20]
print(My_Array)
print(My_list)
```

Output:

array(‘i’, [1, 2, 3, 4]) [1, ‘abc’, 1.2]

51. How to remove values to a python array?

Array elements can be removed using pop() or remove() methods. pop() returns deleted value. remove() return remaining values.

Example:

```
a=arr.array('d', [1.1, 2.2, 3.8, 3.1, 3.7, 1.2, 4.6])
print(a.pop())
print(a.pop(3))
a.remove(1.1)
print(a)
```

Output:

4.6
3.1
array(‘d’, [2.2, 3.8, 3.7, 1.2])

52. Is del the same as remove()? What are they?

del and remove() methods are you to remove elements. delete used to remove an element at a certain index, remove() used to remove an element by its value.

Example:

```
list1=[3,4,5,6,7]
list2=[3,4,5,6,7]

del list1[3]
list2.remove(5)
print(list1)
print(list2)
```

Output:

[3, 4, 5, 7]
[3, 4, 7]

53. What is Iterable in Python?

An Iterable is an object (lists,tuples,strings,set,dictionary,files and Generators). can be iterated by an Iterator.

Example:
data = [1,2,3,4,5,7,8,9,10]

54. What are iterators in Python?

used to iterate over iterable objects. iterator uses iter() and next() functions.

Example:
1.print first four values in list.

```
data = [1,2,3,4,5,7,8,9,10]
result = iter(data)
print(next(result))
print(next(result))
print(next(result))
print(next(result))
```

2.print all values in list.

```
data = [1,2,3,4,5,7,8,9,10]
result = iter(data)

for i in result:
   print(i)
```

55. What is a generator in Python?

It also a regular function which contain yield keyword. the yield keyword pauses the current execution and allows it to be resumed whenever necessary.

It is a very simple implementation of an Iterator.Generator function is memory efficient then iterater .A Generator is more compact than an Iterator.

iter() and next() functions are automatically created in Generator.

Example:

```
def f1():
    for i in range(10):
        return i
print(f1())
```

problem: it will retun only first value in range(10).

print all values in given range(10).

```
def f1():
    for i in range(10):
        yield i
result = f1()
for i in result:
    print(i)
```

print first four values in given range(10).

```
def f1():
    for i in range(10):
        yield i
result = f1()
print(next(result))
print(next(result))
print(next(result))
print(next(result))
```

56. **What is the Python decorator?**

decorator is a function that modifies other functions. We call decorator function in existing function by using @ symbol.

Example:

```
def login_required(page):
    def inner(user_name,login_status):
        if login_status == False:
            print("Kindly Login")
            return
```

```
        return page(user_name,login_status)
    return inner

@login_required
def profile(user_name,login_status):
    print("welcome to your profile,Edit your Details")

def about():
    print("weclome to our company")

@login_required
def orders(user_name,login_status):
    print("weclome to orders page")

profile('sandeep',True)
about()
orders('padma',False)
```

57. What is the usage of enumerate () function in Python?

The enumerate() function is used to iterate iterable object and retrieve the index position and its corresponding value at the same time.

58. What is the improvement in enumerate() function of Python?

In Python, enumerate() function is an improvement over regular iteration.The enumerate() function returns index and corresponding value that gives (0, item[0]).

Example:

```
thelist=['a','b']
for i,j in enumerate(thelist):
     print i,j
```

Output:

```
0 a
1 b
```

59. **What is the difference between a shallow copy and a deep copy?**

A deep copy copies an object into another. This means that if you make a change to a copy of an object, it won't affect the original object. In Python, we use the function deepcopy() for this, and we import the module copy. We use it like:

```
import copy
b=copy.deepcopy(a)
```

A shallow copy, copies one object's reference to another. So, if we make a change in the copy, it will affect the original object. For this, we have the function copy(). We use it like:

```
b=copy.copy(a)
```

A shallow copy has faster program execution whereas a deep coy makes it slow.

60. What is a negative index in Python and why are they used?

They are two types of index in python.

- Positive index used to access elements from beginning,'0' means first element and '1' means second element.

- Negative index used to access elements from ending,'-1' means last element and '-2' means last second element.

61. **What is monkey patching in Python?**

Dynamic modifications of a class or module at run-time.

62. How is memory management done in Python?

Python uses its private heap space to manage the memory. Basically, all the objects and data structures are stored in the private heap space. Even the programmer can't access this private space as the interpreter takes care of this space. Python also has an inbuilt garbage collector, which recycles all the unused memory and makes it available to the heap space.

63. What is Pickling and Unpickling in Python?

Pickling: Pickling is a process by which a Python object can be converted into a byte stream over network. also known as serialization/ Marshalling.

Unpickling: unpickling is a process by which constructs Python object from a byte stream over network. also known as de-serialization/ de-Marshalling.

Python has a module named pickle. This module has the implementation of a powerful algorithm for serialization and de-serialization of Python object. We can write it to a file or a database.

64. What is slicing in Python?

We can use Slicing in Python to get a substring from a String. also, we can extract some part of a list. Need to specify where to start the slicing, where to end, and specify the step. It returns a new object from the existing object.

65. **What are modules in Python? Name a few regularly utilized modules in Python?**

A Python module is a.py file. It also holds runnable Python code. It is a script written in Python with import statements, classes, functions and variables etc. Jar/zip files and DLL files can be modules too. We can use a module in another Python script by importing it or by giving the complete namespace.
With Modules, we can divide the functionality of our application in smaller chunks that can be easily managed.

Some of the commonly used built-in modules are:

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

66. Math module and functions?

Math is a built-in module, below functions are available within the math module.
1.ceil: higer value
2.floor: lower value
3.fabs(Absolute value): removes + and -
4.factorial:
5.sqrt:
6.pow:
7.sin:
8.cos:
9.tan:

10.log:

```
import math
total_sales = -20.56
print(math.ceil(total_sales))
print(math.floor(total_sales))
print(math.fabs(total_sales))
```

67. **Python Global Interpreter Lock (GIL)?**

Python Global Interpreter Lock (GIL) is a type of process lock , used by Python whenever it deals with processes. Generally, Python only uses only one thread to execute the set of written statements. The performance of the single-threaded process and the multi-threaded process will be the same in Python and this is because of GIL in Python.

68. What is the usage of help() and dir() function in Python?

help(): The help() function displays modules, keywords, and attributes-related.

Dir(): dir() function displays defined symbols

69. How will you perform Static Analysis on a Python Script?

We can use Static Analysis tool called PyChecker for this purpose.PyChecker can detect errors in Python code.PyChecker also gives warnings for any style issues.Some other tools to find bugs in Python code are pylint and pyflakes.

70. How are arguments passed in a Python method? By value or by reference?

Python's argument-passing model is neither “Pass by Value” nor “Pass by Reference” but it is “Pass by Object Reference”.

Depending on the type of object you pass in the function, the function behaves differently. Immutable objects show “pass by value” whereas mutable objects show “pass by reference”.

71. What is the Docstring in Python?

The docstrings are declared using ”’triple single quotes”’ or “”””triple double quotes””” just below the class, method, or function declaration. A Docstring is used for adding comments or summarizing a piece of code in Python.All functions should have a docstring and docstrings can be accessed using the help function.

72. What is __init__?

__init__ is a method or constructor in Python. This method is automatically called to allocate memory when a new object is created. All classes have the __init__ method.

73. What is the difference between append() and extend() functions of a list in Python?

We can call standard functions like append() and extend() on a list.
We call append() method to add an item to the end of a list.We call extend() method to add another list to the end of a list.
In append() we have to add items one by one. But in extend() multiple items from another list can be added at the same time.

74. What is the difference between split() and slicing in Python?

Both split() function and slicing work on a String object.

By using split() function, we can get the list of words from a String.
Example:
'a b c '.split()
Output:
[‘a’, ‘b’, ‘c’]
Slicing is a way of getting substring from a String. It returns another String.
Example:
'a b c'[2:3]
Output:
b

75. How do you profile a Python script?

Python includes a profiler called cProfile. It not only gives the total running time, but also times each function separately, and tells you how many times each function was called, making it easy to determine where you should make optimizations.

76. What is None in Python?

Python, None keyword is an object, is used to define a null variable or an object. it is None Type data type. We can assign None to any variable, but you cannot create other NoneType objects. During comparison we must use “is” operator for None instead of “==” operator.

77. What is the difference between ‘is’ and ‘==’ in Python?
We use ‘is’ to check an object against its identity.
We use ‘==’ to check equality of two objects.

Example:

```
lst = [10,20, 20]
lst == lst[:]
True

lst is lst[:]
False
```

78. How can we do Functional programming in Python?

In Functional Programming, we decompose a program into functions. These functions take input and after processing give an output. The function does not maintain any state.

Python provides built-in functions that can be used for Functional programming. Some of these functions are:
I. Map()
II. reduce()
III. filter()

Event iterators and generators can be used for Functional programming in Python.

79. Can you explain the filter(), map(), and reduce() functions?

filter()- This function filters elements that satisfy some conditional logic.

Example:

```
set(filter(lambda x:x>4, range(7)))
```

output:

```
{5, 6}
```

map()-applies a function to each element in the iterable.

Example:

```
set(map(lambda x:x**3, range(7)))
```

output:

```
{0, 1, 64, 8, 216, 27, 125}
```

reduce()- This function reduces all values to single value.

Example:

```
reduce(lambda x,y:y-x, [1,2,3,4,5])
```

output:

3

Let's understand this:
2-1=1 3-1=2 4-2=2 5-2=3
Hence, 3.

80. How is Error/Exceptional handling done in Python?

Interupting Normal Execution of a code. There are 3 main keywords i.e. try, except, and finally

- Try is the block of a code that is monitored for errors.
- Except block gets executed when an error occurs.
- finally block gets executed irrespective of whether an error occurred or not.

```
try:
  risky code
except:
  print('Error')
else:
  print('No Error')
finally:
  print('always prints message')
```

Example:

```
balance = 550.90

try:
   deposit = float(input('Enter amount:'))
except ValueError:
   print("please Enter valid amount")
else:
   final_Balance = balance+deposit
   print(final_Balance)
finally:
   print("Thank you")
```

81. How is File handling done in Python? What are the different file processing modes supported by Python?

File Handling: Reading, writing, deleting, creating a File
modes: There are different ways to open files in Python.

1. **read(r):** used to open a file in read-only mode. Read a file by opening it. It's the default setting.
2. **write(w):** open a file in write-only mode. If contains information, information would be lost. A brand-new file is also created
3. **append(a):** is used to open a file in append mode. append it to the end.
4. **Read write(r+/rw)**: used to open in both read-only and write-only modes.
5. **write Read(w+/wr):** used to open in both write-only and read-only modes.

Process:

- **open()**
- **read/write**
- **close()**

82. How to read a file in Python?

read file:

```
file1 = open("D:\Big_Data\DataSets\word1.txt",mode='r')
print(file1.read())
file1.close()
```

read firstline for file:

```
firstline = open("D:\Big_Data\DataSets\word.txt",mode='r')
print(firstline.readline())
firstline.close()
```

convert lines into list format:

```
list = open("D:\Big_Data\DataSets\word.txt",mode='r')
print(list.readlines())
list.close()
```

83. Write a file in python?

write to file

```
file1 = open("D:\Big_Data\DataSets\word1.txt",mode='w')
file1.write('Hi sandeep!')
file1.close()
```

84. How to Append a file inpython?

append to file

```
file1 = open("D:\Big_Data\DataSets\word1.txt",mode='a')
file1.write('How are you?')
file1.close()
```

85. How do I read and write at the same time?

read+write file

```
file1 = open("D:\Big_Data\DataSets\word1.txt",mode='r+')
print(file1.read())
file1.write('how about you?')
file1.close()
```

86. How do I write and read at the same time?

write+read file

```
file1 = open("D:\Big_Data\DataSets\word1.txt",mode='w+')
file1.write('i am a big data engineer')
file1.seek(0)
print(file1.read())
file1.close()
```

87. Explain tell and seek function?

tell() :tells current position of the pointer

```
file = open("D:\Big_Data\DataSets\word1.txt",mode='r')
print(file.read())
print(file.tell())        #gives cursor location
file.close()
```

seek(): moves cursor pointer to specified location

```
file = open("D:\Big_Data\DataSets\word1.txt",mode='r')
print(file.read())
file.seek(0)              #moves cursor location to first index
print(file.tell())
file.close()
```

88. How to Read and write file using with ?

with read statment:

```
with open("D:\Big_Data\DataSets\word1.txt",'r') as File:
   File=File.read()
   print(File)
```

with write statment:

```
with open("D:\Big_Data\DataSets\word1.txt",'w') as File:
   File.write('How are you?')
```

89. How will you execute a Python script in Unix?

To execute a Python script in Unix, we need to have Python executor in Unix environment.In addition to that we have to add following line as the first line in a Python script file.

```
#!/usr/local/bin/python
```

This will tell Unix to use Python interpreter to execute the script.

90. What functions or methods will you use to delete a file in Python?

To delete a file in Python, you need to import the OS Module. After that, you may use remove() or unlink().

using remove():

```
import os
os.chdir('C:\\Users\\lifei\\Desktop')
os.remove('try.py')
```

using unlink ():

```
os.unlink('try.py')
```

91. If you have data with name of customers and their location, which data type will you use to store it in Python?

In Python, we can use dict data type to store key value pairs. In this example, customer name can be the key and their location can be the value in a dict data type.

Dictionary is an efficient way to store data that can be looked up based on a key.

92. How can we retrieve data from a MySQL database in a Python script?

To retrieve data from a database we have to make use of the module available for that database. For MySQL database, we import MySQLdb module in our Python script.

We have to first connect to a specific database by passing URL, username,password and the name of database.

Once we establish the connection, we can open a cursor with cursor() function. On an open cursor,we can run fetch() function to execute queries and retrieve data from the database tables.

93. What are Access Specifiers in Python?

Python uses the '_' symbol to determine the access control for a specific data member or a member function of a class. A Class in Python has three types of Python access modifiers:

- Public Access Modifier: The members of a class that are declared public are easily accessible from any part of the program. All data members and member functions of a class are public by default.
- Protected Access Modifier: The members of a class that are declared protected are only accessible to a class derived from it. All data members of a class are declared protected by adding a single underscore '_' symbol before the data members of that class.
- Private Access Modifier: The members of a class that are declared private are accessible within the class only, the private access modifier is the most secure access modifier. Data members of a class are declared private by adding a .

94. What is the difference between xrange and range in Python?

In Python, we use range(0,10) to create a list in memory for 10 numbers.
Python provides another function xrange() that is similar to range() but xrange() returns a sequence object instead of list object. In xrange() all the values are not stored simultaneously in memory. It is a lazy loading based function.

But as per Python documentation, the benefit of xrange() over range() is very minimal in regular scenarios.As of version 3.1, xrange is deprecated.

95. How will you specify source code encoding in a Python source file?

By default, every source code file in Python is in UTF-8 encoding. But we can also specify our own encoding for source files. This can be done by adding following line after #! line in the source file.

-*- coding: encoding -*-

In the above line we can replace encoding with the encoding that we want to use.

96. What is a metaclass in Python?

A metaclass in Python is also known as class of a class. A class defines the behavior of an instance. A metaclass defines the behavior of a class.
One of the most common metaclass in Python is type. We can subclass type to create our own metaclass.We can use metaclass as a class-factory to create different types of classes.

97. How would you work with numbers other than those in the decimal number system?

With Python, it is possible to type numbers in binary, octal, and hexadecimal.

- Binary numbers are made of 0 and 1. To type in binary, we use the prefix 0b or 0B.
  >>> int(0b1010)
  10

- To convert a number into its binary form, we use bin().
  >>> bin(0xf)
  '0b1111'

- Octal numbers may have digits from 0 to 7. We use the prefix 0o or 0O.
  >>> oct(8)
  '0o10'

- Hexadecimal numbers may have digits from 0 to 15. We use the prefix 0x or 0X.
  >>> hex(16)
  '0x10'

98. Which sorting technique is used by sort() and sorted() functions of python?

Python uses the Tim Sort algorithm for sorting. It's a stable sorting whose worst case is O(N log N). It's a hybrid sorting algorithm, derived from merge sort and insertion sort, designed to perform well on many kinds of real-world data.

99. What is the significance of functions that start and end with _ symbol in Python?

Python provides many built-in functions that are surrounded by _ symbol at the start and end of the function name. As per Python documentation, double _ symbol is used for reserved names of functions.

100. How do you perform unit testing for Python code?

We can use the unit testing modules **unittest** or **unittest2** to create and run unit tests for Python code.
We can even do automation of tests with these modules. Some of the main components of unit test are as follows:
I. **Test fixture**: We use test fixture to create preparation methods required to run a test. It can even perform post-test cleanup.
II. **Test case**: This is main unit test that we run on a piece of code. Wecan use **Testcase** base class to create new test cases.
III. **Test suite**: We can aggregate our unit test cases in a Test suite.
IV. **Test runner**: We use test runner to execute unit tests and produce reports of the test run.

101. How will you check in Python, if a class is subclass of another class?

Python provides a useful method issubclass(a,b) to check whether class a is a subclass of b.

Example:
int is not a subclass of long
issubclass(int,long)
False

Example:
bool is a subclass of int
issubclass(bool,int)
True

102. How will you debug a piece of code in Python?

In Python, we can use the debugger pdb for debugging the code. To start debugging we have to enter following lines on the top of a Python script.

```
import pdb
```

pdb.set_trace()

After adding these lines, our code runs in debug mode. Now we can use commands like breakpoint, step through, step into etc for debugging.

103. What are the popular Python libraries used in Data analysis?

Some of the popular libraries of Python used for Data analysis are:

I. **Pandas**: Powerful Python Data Analysis Toolkit

II. **SciKit**: This is a machine learning library in Python.

III. **Seaborn**: This is a statistical data visualization library in Python.

IV. **SciPy**: This is an open source system for science, mathematics and engineering implemented in Python.