PYTHON INTERVIEW QUESTIONS

Most asked Interview Questions

BY : CODE OF GEEKS

CODE OF GEEKS

Python Interview Questions

This e-book contains **most frequently asked questions** on **Python – Programming language**.

Prepared by **CODE OF GEEKS**

In case of bug-reporting, please mail us at

support@codeofgeeks.com

SET 1 – Q1 – Q10

Q.What do you know about Python ?

A. Python is an interpreted, high level and object oriented programming language. Python is easy to understand and develop.

Q. Who developed Python ?

A. Python was developed by Guido Van Rossum.

Q. Why Python was named as “Python” ?

A. G.V Rossum picked the name “Python” from the famous TV show, **Monty Python’s Flying Circus** and rest is history.

Q. Python is “dynamically typed”. Explain.

A. In Python, unlike C/C++, we do not need to declare anything. Assignment statements binds a name to an object, and object can be of any type. If a name is assigned to an object of one type, then later, it may be assigned to an object of another type. It means that Python is **dynamically typed**.

Q. Is Python a scripting language ?

A. A scripting language is a programming language that does not use a compiler for executing the source code. Rather, it uses an interpreter to translate source code to machine code on the go. Python is **capable** of scripting but, is considered to be a **general purpose language**. General purpose programming language has wide range of application domains.

Q. What is Byte code ?

A. When a python code is compiled, a file (set of instructions) is generated. this is referred to as Byte code. It is Byte code only, which makes python **platform independent**. Further, this byte code is converted to machine code using Interpreter.

Q. What are flavors of Python?

A. Python flavors refers to the different types of Python compilers. Some flavors are cpython, jython, ironpython, PyPy.

Q. What is CPython ?

A. This is the standard python compiler implemented in C language.

Q. Do you what is Jython ?

A. This is the implementation of Python programming language which is designed to run on Java platform.

Q. Highlight the difference between C & Python ?

A.

C	PYTHON
C is procedural-oriented programming language.	Python is object-oriented programming language.
It is compulsory to declare the datatype of variables.	Type declaration is not required in Python.
C has for, while, do-while loops.	Python has for, while loops only.
C has switch statements.	Python does not have switch statements.
C does not allows Exception Handling.	Python allows Exception Handling.

SET 2 – Q11 – Q20

Q. What are builtin datatype in Python ?

A. None, int, float, complex, bool, str, bytes, list, tuple, range etc.

Q. You are given a mathematical expression in the form of string, write a python code to solve it.

A. Do it with the help of **eval()** function.

Q. Give a brief explanation about list and tuple ?

A. Lists are mutable(can be changed) whereas Tuples are immutable(can't be changed). Tuples are faster than lists.

General format for tuple : (4,5,6,9)

General format for list : [4,5,6,9]

Q. Why tuples are faster than lists ?

A. Tuples are stored in single block of memory whereas Lists are stored in two blocks, first block for object related information and second block for the value. Hence, tuples are faster than list.

Q. Differentiate between list and tuple ?

A. Tuple uses less memory whereas lists uses more memory. We can't modify tuples but lists can be modified.Tuples are faster than lists.

Q. What is PEP 8 ?

A. It is Python Enhancement Proposal. It refers to the formatting of python code for maximum readability and understandability.

Q. Explain memory management in Python ?

A. Memory Management in Python is managed by Python Private heap space. All python objects and data structures are stored in heap area. Programmers do not have any direct access to private heap area, interpreter takes care of it.

Q. Explain Garbage Collection in Python ?

A. Python has the concept of inbuilt garbage collector. It recycles all the unused memory so that they get available for future use.

Q. What is PYTHON PATH ?

A. It is an environment variable which we can use to set the additional directories, where modules and packages are looked for.

For MAC :

1. Open Terminal.app;

2. Open the file ~/.bash_profile in your text editor – e.g. atom ~/.bash_profile;
3. Add the following line :
export PYTHONPATH=”/Users/my/code” and save it.

For LINUX :

1. Open your terminal;

2. Open the file ~/.bashrc in your text editor – e.g. atom ~/.bashrc;
3. Add the following line to the end:

export PYTHONPATH=/home/my/code and save it.

SET 3 – Q21 – Q30

Q. What are python modules ?

A. Python modules are **.py files** containing python executable code. Some builtin modules are math, random, sys, json, itertools.

Q. What do you mean by JSON ?

A. Python inbuilt module JSON, is basically used to work with text, written with JavaScript Object Notation.

Q. Explain the role of loads() & dumps() methods of JSON module.

A. loads() -> for parsing **json text/string**.

dumps() -> for converting **python code** to **json string**.

Q. What is the role of ord() method ?

A. It is used to change a character to integer.

Q. What the basic difference between list & array ?

A. Arrays have similar type of elements whereas list have dynamic type elements.

Q. Explain the role of __init()__ method ?

A. It is predefined method in python classes. This method is called when any object is created from a class, it basically initializes the attributes of a class.

Q. What are anonymous functions ?

A. **Lambda functions** are popularly called as **anonymous functions**. Such methods generally have multiple parameters but single statement.

For example :

a = lambda x,y : x+y

print(a(6,8)) will result in 14.

Q. What is self ?

A. It is an object or an instance of a class,which is explicitly included as first argument.

Q. What is **pass keyword** ?

A. Pass is a non operational statement. It actually does nothing.

Q. What is **continue keyword** ?

A. The continue statement skips all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop.

SET 4 – Q31 – Q40

Q. How we can generate randomize elements of lists ?

A. We can do this with the help of **shuffle()** method, present in **random** module.

Q. Differentiate between range and xrange ?

A.

range ()	xrange ()
It returns lists containing values.	returns generator objects.
takes more memory.	takes less memory.
supports slicing	does not supports slicing.

Q. How pickling works ?

A. Pick an object, convert it to string and then dump it into file, this defines the process of pickling.

Q. What are Generator functions ?

A. It is just like a normal function which generates a value using **yield keyword.**

Example :

```
def simgen():

    yield 1    yield 4    yield 9x=simgen()

print(x.__next__())

output : 1
```

Generator object is a python iterator.

Q. What are decorators ?

A. Decorators provide a simple and easy syntax for calling higher-order functions. In other words, a decorator is a function that takes another function and extends the behavior of the latter function without explicitly modifying it. It is referred to as meta programming.

Q. Tell me the role of ‘is’, ‘not’, ‘in’ keyword ?

A. **is :** return **true** if both operands are equal else **false**.

not : returns inversion of boolean value.

in : checks for the presence of an element in a list or any other datatype.

Q. Analyze the role of help() ?

A. It is used to display the documentation of modules, functions, classes, keywords etc.

Q. Analyze the role of dir() ?

A. It returns the list of all valid attributes.

Q. Give general syntax for ternary operator ?

A. [on true] if [expression] else [on false]

Example : “odd” if n%2==1 else “even”

Q. What are *args & **kwargs ?

A. ***args ->*** It is used when we need to pass unspecified number of paramenters within a method.

****kargs ->*** It is used when we need to pass unspecified number of paramenters within a method but usually used in dictionaries.

SET 5 – Q41 – Q50

Q. How we can delete files in Python ?

A.

```
import os
```

```
os.remove(filename)
```

Q. Explain docstring ?

A. A String written in triple quotes, which generally gives a complete introduction of class.

Q. What are different types of variables in Python ?

A. Instance variable, Static variable, Class variable.

Q. Explain Instance variable ?

A. Variables whose seperate copy is created in every instances, are instance variable. Ex. If z is an instance variable, and we create 2 instances, then there will 2 different copies of z in these 2 instances. Modification of one variable will not effect other copies.

Q. How we can access instance variable ?

A. With the help of dot (.) operator.

Q. What are instance methods ?

A. Methods that acts on the instances of a class are known as instance methods. They generally use self as their first parameter, which refers to the location of that instance in memory.

Q. Explain Static variable ?

A. They are also referred to as Class variables. In these type of variables, only single copy is available.

Q. What is namespace ?

A. A memory block, where names are mapped to objects.

Q. Explain me the role of Constructor ?

A. It is defined as the special method for initializing instance variable. Constructor is called at the time of creating objects.

Q. Explain Static methods ?

A. They are simple methods with no **self** argument. They work on class attributes not on instance attributes. They can be called through both class and instance.

SET 6 – Q51 – Q60

Q. Explain Abstract methods ?

A. Abstract methods are the methods, that does not have any body, and are further refined in the subclasses. Abstract methods are generally written with **decorator @abstractmethod** above them.

Q. Abstract class contains only abstract methods. True ?

A. No, it can also contain concrete methods.

Q. In Python, is it possible to write abstract method with body also ?

A. Yes.

Q. Can we create instances of Abstract class ?

A. No.

Q. Can we have nested class in Python ?

A. Yes, we can have nested class in Python.

Q. What do know about MRO ?

A. It is Method Resolution Order.It is the order in which Python looks for a method in a hierarchy of classes. Starting from current class, searching in parent class in depth first, left to right fashion, without searching same class twice.

Q. What is method overloading ?

A. It refers to the two or more methods having same method name but different parameters as well as body.

Q. What is method overriding w.r.t classes ?

A. Whenever a method in a subclass is defined with same definition as that of a method in base class, then this method is said to be overrided. This explains overriding mechanism.

Q. What is an Interface ?

A. Interface can be defined as the abstract class containing no concrete methods.

Q. What is Duck Duck Typing ?

A. In this, object type is not checked while invoking method or object. Python uses duck duck typing.

SET 7 – Q61 – Q70

Q. What are Exceptions ?

A. Exceptoins are run time errors that can be handled by a programmer. All these exceptions are represented as **classes**.

Q. Can you name some Exceptions that may occur during execution of a program ?

A. Arithmetic Exception, Type Error, Syntax Error, Assertion Error, EOF Error.

Q. How can we handle exceptions in Python ?

A. Using **try, except, and finally**.

Q. How a code behaves inside a finally block ?

A. **Finally block** is executed every time.

Q. Tell me the situation when finally block will not be executed ?

A. When we use **sys.exit()**.

Q. What do you mean by monkey patching ?

A. Whenever a class is dynamically modified, then, it is known as monkey patching.

Q. What do you know about Numpy Package ?

A. It is a general purpose array processing package. It provides a high performance multidimensional array object and tools for working with these arrays. It is a fundamental package for scientific computing with python. It is used for fast searching, statistics, algebra, histogram etc.

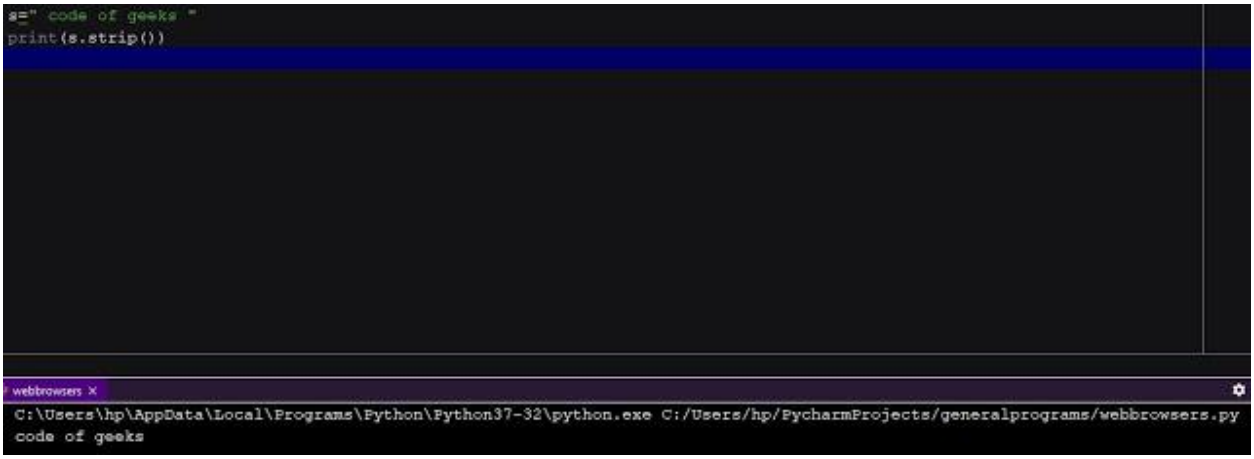
Q. What are axes ?

A. In Numpy, dimensions are called axes. The number of axes, are referred to as **rank**.

Q. What is the role of strip() ?

A. strip() is an inbuilt function in Python programming language that returns a copy of the string with both leading and trailing characters removed.

Example :



Q. What is the role of split() ?

A. It splits a string into a list where each word is a list item.

Example :

```
s=" code of geeks "
print(s.split())
```

```
webbrowsers ×
C:\Users\hp\AppData\Local\Programs\Python\Python37-32\python.exe C:/Users/hp/PycharmProjects/generalprograms/webbrowsers.py
['code', 'of', 'geeks']
```

SET 8 – Q71 – Q80

Q. What is the concept of dictionary in python ?

A. Dictionary is an unordered collection of data items, it is used to store data values like a map.

It usually stores the data in the form of key-value pairs.

Q. How we can access key,values in python dictionary ?

A.

```
mydic={1:"code",2:" of ",3:" geeks "}
for key,value in mydic.items():
    print("KEY = ",key)
    print("VALUE = ", value)
```

```
webbrowsers ×
C:\Users\hp\AppData\Local\Programs\Python\Python37-32\python.exe C:/Users/hp/PycharmProjects/generalprograms/webbrow
KEY =  1
VALUE =  code
KEY =  2
VALUE =   of
KEY =  3
VALUE =   geeks
```

Q. How to run a python program on Linux ?

A. Python 2 : python <filename>.py

Python 3 : python3 <filename>.py

Q. Explain the role of setTrace() method ?

A. It is present in **sys module,** responsible for monitoring the code flow of a program.

Q. Explain the concept of enumerate() ?

A. It is used to iterate through a sequence, it retrieves index as well as value at same time.

Q. Write python code for reading a file ?

A.

```
f=open("file.txt","r")
str=f.read()
print(str)
f.close()
```

Q. Why do we use with statement in file handling ?

A. It will auto close, a file it opens.

Q. Explain the purpose of tell() & seek() ?

A. tell() : It is knowing the position of file pointer.

seek() : It sets the new position of pointer.

Q. What is the role of encode() ?

A. It is to convert a string to **byte**.

Q. Explain regex ?

A. Python has a built-in package called **re** , which can be used to work with Regular Expressions.

SET 9 – Q81 – Q90

Q. Write a python code for retrieving all words starting with ‘a’ from a string ” mystring “.

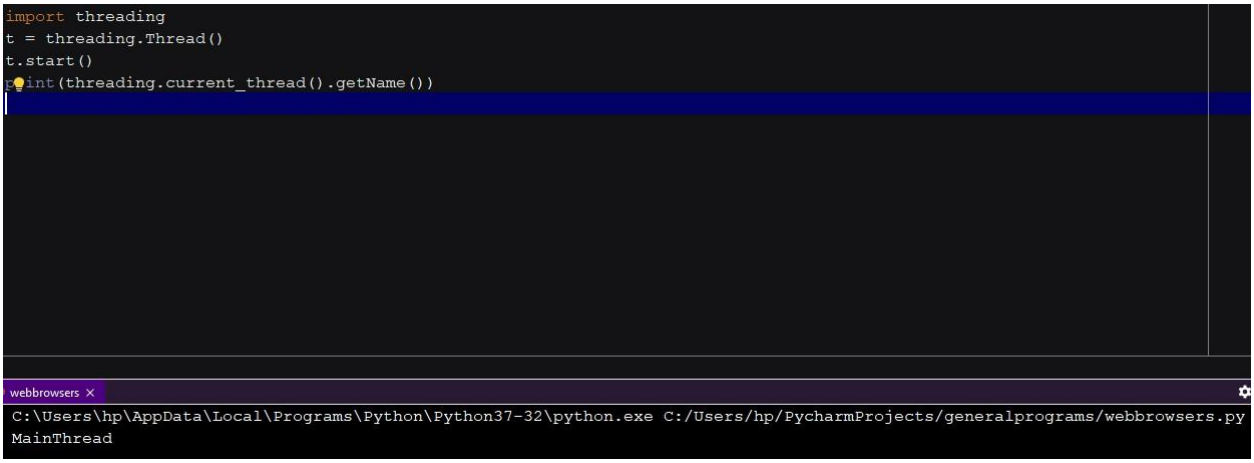
A. res = re.findall(r ‘ a[\w]* ‘, mystring)

Q. What is a Thread ?

A. Thread is a smallest executable unit.

Q. How we can create threads in Python ?

A.



Q. What is GIL ?

A. It is Global Interpreter Lock. It does not allow more than one thread to execute at a time.

Q. What are Daemon Threads ?

A. Daemon Threads are the threads that run continously. They are used for big processing system.

Q. How can we send email through Python code ?

A.

Q. Write python code for connecting to database ?

A.

```
import mysql.connector
con=mysql.connector.connect(host='localhost',database='db',user='root',password='pass')

' we use cursor for executing SQL commands '
cursor=con.cursor()
cursor.execute("select * from mytable")

' fetching data from table '
row=cursor.fetchall()

for rows in row:
    print(rows)
```

Q. Write python program for finding unique words in a string.

A.

```
s = " lorem ipsum lorem ipsum lorem ipsum lorem ipsum "
s1=set()
l=s.split()
l1=[]
for i in l:
    if i not in l1:
        l1.append(i)
print(l1)
```

```
webbrowsers ×
C:\Users\hp\AppData\Local\Programs\Python\Python37-32\python.exe C:/Users/hp/PycharmProjects/generalprograms/webbrowsers
['lorem', 'ipsum']
```

Some questions on Object Oriented Programming

Q. Explain OOPS methodology ?

A. The entire OOPS methodology has been derived from a single root called object. In OOPS, all programs involve creation of classes and object. Five important features of OOPS are :

1. Classes & Objects

2. Encapsulation

3. Abstraction

4. Inheritance

5. Polymorphism

Q. What is Procedural oriented programming ?

A. It is the methodology where programming is done using procedures & functions.

SET 10 – Q91 – Q101

Q. Is Python an object oriented or object based programming language ?

A. Object Based programming language does not support all features of OOPs like Polymorphism and Inheritance. Python is an **object oriented programming language**. Some of **object based programming languages** are JavaScript, Visual Basics.

Q. Explain Encapsulation with example ?

A. Encapsulation is a mechanism where the data and the code that act on the data will bind together.

Example : Class is an example of encapsulation. As class is the collection of data members and member functions, hence class binds them together.

Q. Explain Abstraction with example ?

A. Abstraction refers to the act of hiding the background details and providing only necessary details to the user.

Example : Let us the example of a car. While driving car, driver(user) only knows basic operations like gearing system, brakes, but is totally unaware about the internal happenings and processes. This is abstraction.

Q. Explain Polymorphism with example ?

A. Polymorphism represents the ability to assume several different forms of a program.

Example : Overloading of methods is an example of Polymorphism. Other examples include, abstract classes and interfaces.

Q. Explain Inheritance with example ?

A. Inheritance refers to the process in which child class inherits the properties of its parent class.

Example : Suppose we are having two classes A and B, so

class B(A) -> Child class B, is inheriting the property of its parent class A.

Some questions on Advanced Python

Q. What is Flask ?

A. Flask is a light weighted web micro framework, It is used to build web applications with Python.

Q. What is Django ?

A. It is an open source framework, to ease up the creation of complex database oriented websites.

Django contains prewritten code, unlike flask.

Q. What is Pyramid ?

A. It is a web framework for larger applications, they are heavily configurable.

Q. Explain SciPy ?

A. All numerical code resides inside SciPy. It is used for large and complex algebraic functions.

Q. What is use of MatplotLib ?

A. It provides basic 3D plotting.

Q. What is the role of PyChecker ?

A. PyChecker is the tool that help us find bugs or perform static analysis of code.

So, these are the 100 most asked interview questions on Python Programming Language.

We, at **CODE OF GEEKS**, wish you all the best for your upcoming future.

LET'S PYTHON

PYTHON CODEBOOK

Your way to Competitive Programming.

AUTHOR - CODE OF GEEKS

CODE OF GEEKS

PYTHON

CODEBOOK

In this e-book, we will look at different Python Hacks. This e-book is useful for anyone who wants to brush up Python concepts and that too in very less time. This will prove to be a great reference if you want to start competitive programming with Python.

PYTHON CODEBOOK by CODE OF GEEKS

www.codeofgeeks.com

For permissions contact:

support@codeofgeeks.com

Taking inputs :

s = input() // taking string as input

n = int(input()) // taking int as input

b = bool(input()) // taking boolean value as input

l = list(input().split(‘,’)) // taking list as a input where elements are seperated by comma

s = tuple(input().split(‘,’)) // taking tuple as a input where elements are seperated by comma

Taking Multiple input in one line :

a,b = input().split(separator, maxsplit)

Taking a list of ‘n’ integers as input :

list(map(int,input().strip().split()))[:n]

Printing a formatted output :

1. Let us assume that we have to print values of two variables – a=10 b=20 as 10—-20, We can do this by : print(a,b,sep=’—‘)
2. The output displayed by the print() method can be formatted as you like. ‘%’ operator is used for this purpose. It joins a string with a variable or value. Example :
   print(“string” % (variable-list))

Useful Basic Methods – Python

1. Converting a number from octal, binary and hexadecimal system to decimal number system.

n1 = 0o17 # representation of octal numbers

n3 = 0x1c2 # representation of hexadecimal number

We can do this with the help of int() method like int(n1), int(n2), int(n3).

2. Converting a decimal integer to octal, binary and hexadecimal system.

bin() : For decimal to binary conversion.

oct() : For decimal to octal conversion.

hex() : For decimal to hexadecimal conversion.

3. Mathematical methods

ceil(x) : It raises x value to the next higher integer value. For example, ceil(4.5) gives 5.

floor(x) : It decreases x value to previous integer value. For example, floor(4.5) gives 4.

degrees(x) : It converts angle value x from radians to degrees.

radians(x) : It converts x value from degree to radians.

sin(x) : It gives a sine value of x.

cos(x) : It gives a cosine value of x.

tan(x) : It gives a tan value of x.

exp(x) : It gives exponential of x.

fabs(x) : It gives absolute value of x. Like fabs(-4.53) gives 4.53.

factorial(x) : It gives the factorial of x.

fmod(x,y) : It gives remainder of division of x & y. Like, fmod(13.5,3) gives 1.5.

fsum(val) : It gives the accurate sum of floating point values.

log10(x) : It gives base-10 logarithm of x.

sqrt(x) : It gives the square-root of number x.

pow(x,y) : It raises x value to the power y.

pow(x,y,z) : It raises x value to the power y mod z

gcd(x,y) : It is used to find the greatest common divisor of x & y.

trunc(x) : It returns real value of x is truncated to integer value. Like trunc(43.545) returns 43.

isnan(x) : It returns True if x is not a number.

eval(expression) : It returns evaluated arithmetic expression. Like, eval(3*7) gives 21.

Strings Tip & Methods – Python

Finding the length of String : len(string_name)

Indexing in Strings :

0	1	2	3	4	5	6
p	Y	t	h	o	n	n
-7	-6	-5	-4	-3	-2	-1

Reversing a String :

```
i=1
n=len(s)
while i<=n:
    print(s[-i],end=' ')
    i+=1
```

Slicing a String :

string-name[start : stop : stepsize]

If given string is “pythonn” so s[0:7:2] gives pton as output.

s[::2] : access entire string in steps of 2.

s[2::] : access string s[2] to ending.

Repeating a String :

```
s = ‘pythons’
print(s*2)
```

gives “pythonspythons” as output.

Concatenation of Strings :

Strings can be concatenated with one another using ‘+‘ operator.

```
s1 = "string 1"
s2 = "string 2"
s3 = s1 + s2
print(s3)
```

Output : string 1string 2

Removing Spaces from String :

We can remove extra spaces with the help of lstrip(), rstrip(), strip() methods.

```
s = "     Python      "

print(s.lstrip())
print(s.rstrip())
print(s.strip())
```

Output :

Python

Python

Python

String Methods :

1. s.find(substring, beginning, ending) : It is used to find the first occurrence of given substring in a string. It returns -1 if given substring is not available.

2. s.count(substring, beginning, ending) : It is used to find the total number of occurrences of a substring in a main string.

3. s.replace(old, new) : It is used to replace a substring with another substring.

4. sep.join(str) : When a group of strings are given, it is possible to join them all and make a single string. Syntax : seperator.join(string)

```
l = ["one","two","three"]
s = "-".join(l)
print(s)
```

Output : one-two-three

5. s.upper() : It converts to upper-case string.

6. s.lower() : It converts to lower-case string.

7. s.swapcase() : It converts all lowercase letters to uppercase letters and vice versa.

8. s.title() : It converts a string in such way that first letter of a word in string is a uppercase letter.

```
s = "pYthon"

print(s.upper()) // PYTHON

print(s.lower()) // python

print(s.swapcase()) // PyTHON
```

print(s.title()) // Python

9. s.startswith() : It is used to know whether a string is starting with a substring or not. Like,

s.startswith('P') is used check whether a substring starts with 'P'.

10. s.endswith() : It is used to know whether a string is ending with a substring or not. Like,

s.endswith('P') is used check whether a substring ends with 'P'.

11. s.alnum() : It returns True if all characters in the string are alpha

numeric (A-Z, a-z, 0-9).

12. s.alpha() : It returns True if string has atleast one character and all other characters are alphabetic (A-Z, a-z).

13. s.isdigit() : It returns True, if the string contains only numeric digits (0-9).

14. s.islower() : It returns True, if at least one or more letters are in lowercase.

15. isupper() : It returns True, if at least one or more letters are in uppercase.

Formatting the Strings :

Formatting a string means presenting the string in a clearly understandable manner. The format() method is used to format strings.

```
id = 10

name = "Code of Geeks"

sal = 1345.345

s = '{},{},{}'.format(id,name,sal)

s1 = '{}-{}-{}'.format(id,name,sal)
```

Output :

10,Code of Geeks,1345.345

Sorting the String :

We can sort the string alphabetically using sort() method and sorted() method.

Creating Lists using range() function :

We can use range() function to generate a sequence of integers which can be stored in a list. The format of range() function is :

range(start, stop, stepsize)

If not mentioned, start is specified to be 0 and stepsize is taken 1.

Above code will result in a list of 10 elements - 0 to 9.

```
l1 = range(10)
for i in l1:
   print(i)
```

Accessing list elements :

```
l=[1,2,3,4,5]
i=0
while i<len(l):
   print(l[i])
   i+=1
```

Output : 1 2 3 4 5

Concatenating two lists :

```
l1 = [1,2,3,4]
l2 = [5,6,7]
```

```
print(l1+l2)
```

Output : [1,2,3,4,5,6,7]

Repetition of Lists :

```
l = [10,20]
print(l*2)
```

Output : [10,20,10,20]

Membership in Lists :

```
l = [10,20,30,40,50]
a = 30
print(a in l)
```

Output :

True

List Methods

1. list.index(x) : It returns the first occurance of x in list.
2. list.append(x) : It appends the element x at the end of list.
3. list.insert(i,x) : It inserts x in the i-th position of list.
4. list.copy() : It copies all elements of a list to a new list and returns it.
5. list.extend(list1) : It appends list1 to list.
6. list.count(x) : It returns the total occurrences of x in list.
7. list.remove(x) : It removes element ‘x’ from the list.
8. list.pop() : It removes the ending element from the list.
9. list.sort() : It is used to sort the element of lists.

10. list.reverse() : It is used to reverse a list.

11. list.clear() : It is used to delete all the elements of a list.

12. max() : It is used to find the maximum element in a list.

13. min() : It is used to find the minimum element in a list.

```
l=[2,4,6,23]
print(max(l))
print(min(l))
```

Output :

23

2

2D Lists :

Suppose we want to create a 3X3 matrix, so we can represent as list – of – lists. For example,

mat = [[3,4,5],[4,6,2],[4,7,2]]

Creation :

```
for r in mat:
   for c in r :
      print(c,end=' ')
   print()
```

Tuple Creation :

```
tup = tuple(range(4,9,2))
print(tup)
```

Output :

4,6,8

Note !! Many lists methods can be applied to tuples as well.

Sets – Python

A Set is an unordered collection data type that is iterable, mutable, and has no duplicate elements.

Basic set operations & methods :

1. Set.add() : If we want to add a single element to an existing set, we can use the .add() operation.

It adds the element to the set and returns 'None'.

```
set1 = set('codeofgeeks')
set1.add('z')
print(set1)
```

Output :

{'c','d','e','f','g','k','o','s','z'}

2. Set.remove() :

This operation removes element from the set. If element does not exist, it raises a KeyError.

```
s = set([1,2,3,4,5])
s.remove(4)
print(s)
```

Output :

{1,2,3,5}

3. Set.pop() : This operation removes and return an arbitrary element from the set. If there are no elements to remove, it raises a KeyError.

```
s = set([2,3,4,5])
print(s.pop())
```

Output :

2

4. Set.difference() : It defines the difference between the number of elements in two sets.

Set is immutable to the .difference() operation (or the – operation).

```
s = set("code of geeks")
print(s.difference("geeks "))
```

Output :

{'f','c','o','d'}

5. Set.union() : Union of two given sets is the smallest set which contains all the elements of both the sets.

```
s = set("code of geeks")
print(s.union("geeks "))
```

Output :

{'g', 'o', 'd', 'f', 'c', 'k', 'e', 's', ' '}

6. Set.intersection() : It is the largest set which contains all the elements that are common to both the sets.

```
s = set("code of geeks")
print(s.intersection("geeks "))
```

Output :

{'s', 'e', 'g', ' ', 'k'}

Dictionaries :

Dictionary represents a group of elements arranged in the form of key-pair pair. The Key and its value are seperated by a colon(:).

dict = {'Name' : 'Vikas', 'Id' : 20}

Dictionary Methods :

1. dict.clear() : It removes all key-value pairs from dictionary.

2. dict.copy() : It copies content of one dictionary to another one.

3. dict.get() : It returns the value associated with key ' k '.

4. dict.items() : It returns an object that contains key-value pairs of dictionary.

5. dict.keys() : It returns a sequence of keys from the dictionary 'd'.

6. dict.values() : It returns a sequence of values from the dictionary 'd'.

7. dict.pop(k,v) : It removes the key 'k' and its value from 'd'.

8. dict.update(x) : It adds all elements from dictionary 'x' to 'd'.

9. zip() : converts two lists to a dictionary.

Some miscellaneous concepts

itertools – Iterator functions for efficient looping.

1. itertools.product : This tool computes the cartesian product of input iterables. It is equivalent to nested for-loops.

```
from itertools import product
print(list(product([1,2,3],repeat = 2)))
```

Output :

[(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)]

2. itertools.permutations(iterable[, r]) :

It returns successive r length permutations of elements in the iterable.

If r is not specified or is None, then r defaults to the length of the iterable and all possible full-length permutations are generated.

```
from itertools import permutations
l=['1','2','3']
print(list(permutations(l)))
```

Output :

[('1', '2', '3'), ('1', '3', '2'), ('2', '1', '3'), ('2', '3', '1'), ('3', '1', '2'), ('3', '2', '1')]

3. itertools.combinations(iterable[, r]) :

This tool returns the length subsequences of elements from the input iterable.

Combinations are emitted in lexicographic sorted order.

```
from itertools import combinations
l=['1','2','3']
print(list(combinations(l,2)))
```

Output :

[('1', '2'), ('1', '3'), ('2', '3')]

Bisect

This module provides support for maintaining a list in sorted order without having to sort the list after each insertion. The module is called bisect because it uses a basic bisection algorithm to do its work.

The following functions are provided:

bisect.bisect_left(list, item[, lo[, hi]])

Locate the proper insertion point for item in list to maintain sorted order. The parameters lo and hi may be used to specify a subset of the list which should be

considered; by default the entire list is used. If item is already present in list, the insertion point will be before (to the left of) any existing entries.

bisect.bisect_right(list, item[, lo[, hi]])

Similar to bisect_left(), but returns an insertion point which comes after any existing entries of item in list.

bisect.bisect(…)

Alias for bisect_right().

bisect.insort_left(list, item[, lo[, hi]])

Insert item in list in sorted order. This is equivalent to list.insert(bisect.bisect_left(list, item, lo, hi), item). This assumes that list is already sorted.

bisect.insort_right(list, item[, lo[, hi]])

Similar to insort_left(), but inserting item in list after any existing entries of item.

bisect.insort(…)

Alias for insort_right().

```
import bisect
l = []
print(" ENTER ANY 5 ELEMENTS ")
for i in range(0,5):
  c=int(input())
  bisect.insort(l,c)
print(l)
```

Output

ENTER ANY 5 ELEMENTS :

-> 5

-> 3

-> 8

-> 9

-> 2

[2, 3, 5, 8, 9]

Python Regex

ReGex, also termed as Regular Expression, is the efficient way of handling regular expressions in a Python Program. It generally involves a sequence of characters that forms a search pattern.
To work with regular expressions, we have to import **re** module.

Important Regex Functions :

re.findall() : It returns a list containing all matches.
Example : Below code will the find the existence of word “geek” in a string.

```
import re
s = ‘code of geeks is for programming geeks’
x = re.findall(‘geek’,s)
print(x)
```

Output
[‘geek’,’geek’]

re.search() : It searches the string for a match. In the case of more than 1 match only the first occurrence of the match will be returned.
Example : Below code will the find the existence of word ‘a’ in a string.

```
import re
s = ‘sdadfghja’
x = re.search(‘a’,s)
print(x.start())
```

Output
2

re.split() : It is used to split a string into list, as per the separator specified.
Example : Below code will split the string into a list taking ‘-‘ as a separator.

```
import re
txt = "code-of-geeks"
x = re.split("-", txt)
print(x)
```

Output
[‘code’, ‘of’, ‘geeks’]

re.sub() : It replaces the matches as per the text specified.
Example : Below code will replace “geeks” to “god”

```
import re
s = "code of geeks"
x = re.sub("geeks","god",s)
print(x)
```

Output
code of god

****