**Question: What are the supported standard data types in Python?**

**Answer:**The supported standard data types in Python include the following.

Immutable

* Number
* String
* Tuple

1. List.
2. Dictionary.
3. sets.
4. -

------------------------------------------------------------------------------------------------------------------------------------------

Q. What are the key features of Python-

A. 1, Python is an interpreted language – So it does need to be compiled before execution. Unlike C.

2. Python is dynamically typed. There is no need to declare a variable with the data type. Python interprets the data type based on a value.

3. Python is OOP language. It has classes, inheritance, access specifiers (public and private keywords)

4. Cross platform – Python language written on a windows system will also run on Linux platform with little or no modification.

5. General purpose language- It can be used in Web development, Automation, Data science, Machine learning.

------------------------------------------------------------------------------------------------------------------------------------------

How is memory managed in Python?

1. Memory in python is managed by something called Python privte heap. Python memory manager takes care of allocation python private heap space to different python objects.

------------------------------------------------------------------------------------------------------------------------------------------

s

Q. What are python built in Modules?

A. Block of executable codes that can either be classes or function or variables that saves time for the programmers by calling these predefined functions.

Ex. Os,

Sys

Date time

Math

Random

JSON

------------------------------------------------------------------------------------------------------------------------------------------

Q Difference between a Pass and a cosntinue statement

Pass- This statement literally does nothing, it will not show or move the control anywhere. Programmers use this mainly to keep a place holder for there statements and finctions

Q. What is a PYTHONPATH?

The PYTHONPATH variable holds a **string with** the name of various directories that need to be added to the sys. path directory list by Python. The primary use of this variable is to allow users to import modules that are not made installable yet.

In short, we can say that it is an environment variable that you set before running the Python interpreter.

Q. What is Python namespace?

Pthon namespace ensures that object names are unique within a programme and can be used without any conflict.

Local Namespace: A variable used inside an Function, created when the function is called and deleted when the function returnss

Global namespace: A variable defined and called upon inside an module or program is triggere and is deleted when the script is finished running.

Builtin Name space: name of some builtin function in Python core. Or are defined in python exceptions

What is a thread and what are the different stages of the Life Cycle of as Thread.

Thread-

A Thread is a separate flow of execution. Which means your program we will have 2 or more things happening at once. But most of python 3 implementation thread do not actual execute at the same, it appears to.

1. Stage 1 : Creating a class where we can override the run method of the Thread Class.

Create a thread and tel it to start.

“”””

import logging

2import threading

3import time

4

5def thread\_function(name):

6 logging.info("Thread %s: starting", name)

7 time.sleep(2)

8 logging.info("Thread %s: finishing", name)

9

10if \_\_name\_\_ == "\_\_main\_\_":

11 format = "%(asctime)s: %(message)s"

12 logging.basicConfig(format=format, level=logging.INFO,

13 datefmt="%H:%M:%S")

14

15 logging.info("Main : before creating thread")

16 x = threading.Thread(target=thread\_function, args=(1,))

17 logging.info("Main : before running thread")

18 x.start()

19 logging.info("Main : wait for the thread to finish")

20 # x.join()

21 logging.info("Main : all done")

‘’’’’’

When we create a thread, we pass a function and a list that contains the arguments of the function. Here, we tell thread to run thread\_function and a list that contains the arguments to the function. Her it is args=(1,))

Output

19:42:52: Main : before creating thread

19:42:52: Main : before running thread

19:42:52: Thread 1: starting

19:42:52: Main : wait for the thread to finish

19:42:52: Main : all done

19:42:54: Thread 1: finishing

1. We make a call .start() on the new thread.

Q. What is inheritance in python?

A. As in all OOP, python provide the ability to inherit properties from another class. The class from where the properties is being derived is called the super class or the parent class and the class who is inheriting the properties is called the sub class or the child class.

Q. What are the types of inheritance and give example.

A. a. Single inheritance- when a function inherit from one parent class

Example-

*class Person:*

*def \_\_init\_\_(self,fname, lname):*

*self.fname = fname*

*self.lname = lname*

*def intro(self):*

*return "The Person name is {} {}".format(self.fname.capitalize(), self.lname.capitalize())*

*class Child(Person):*

*def \_\_init\_\_(self, fname, lname, id\_, add):*

*self.id\_ = id\_*

*self.add = add*

*Person.\_\_init\_\_(self, fname,lname)*

*def info(self):*

*return "The Person's id and addresss is {} and {}".format(self.id\_,self.add)*

b. Multi inheritance- When a class inherits from more than one parent class.

Example

*class Person\_first:*

*def \_\_init\_\_(self,fname, lname):*

*self.fname = fname*

*self.lname = lname*

*def intro(self):*

*return "The Person name is {} {}".format(self.fname.capitalize(), self.lname.capitalize())*

*class Person\_second:*

*def \_\_init\_\_(self,color, music):*

*self.color = color*

*self.music = music*

*def extra\_cc(self):*

*return "The person's favourite color and music is {} and {} respectively".format(self.color,self.music)*

*class Child(Person\_first,Person\_second):*

*def \_\_init\_\_(self, fname, lname, id\_, add,color,music):*

*self.id\_ = id\_*

*self.add = add*

*Person\_first.\_\_init\_\_(self, fname,lname)*

*Person\_second.\_\_init\_\_(self, color,music)*

*def info(self):*

*return "The Person's id and addresss is {} and {}".format(self.id\_,self.add)*

c. Multi level inheritance- When a class derives property from another class that already derives property from another. That mean if we have a grandparent, parent and child condition.

Example:

*class Grandparents():*

*def \_\_init\_\_(self,fname, lname):*

*self.fname = fname*

*self.lname = lname*

*def intro(self):*

*return "The name is {} {}".format(self.fname.capitalize(), self.lname.capitalize())*

*class Parents(Grandparents):*

*def \_\_init\_\_(self,fname, lname,age):*

*self.age = age*

*Grandparents.\_\_init\_\_(self,fname,lname)*

*def info(self):*

*return "The age is {}".format(self.age)*

*class Child(Parents):*

*def \_\_init\_\_(self,fname,lname, age,address):*

*self.address = address*

*Parents.\_\_init\_\_(self, fname,lname, age)*

*def getaddress(self):*

*return "The address is {}".format(self.address.capitalize())*

*person = Child("asmeeta","dewari",29,"assam")*

*print(person.intro())*

*print(person.info())*

*print(person.getaddress())*

d. Hierarchical inheritance- when more than one class is derived from a single parent. A parent have more than 1 child

example

#Heirarchical inheritance

*class Parent():*

*def \_\_init\_\_(self,Name):*

*self.Name = Name*

*def getParentName(self):*

*return "This is Parent with Name {}".format(self.Name)*

*class Child1(Parent):*

*def \_\_init\_\_(self,cName,Name):*

*self.cName = cName*

*Parent.\_\_init\_\_(self,Name)*

*def getchildName(self):*

*return "This is Child 1 with Name {}".format(self.cName)*

*class Child2(Parent):*

*def \_\_init\_\_(self,cName,Name):*

*self.cName = cName*

*Parent.\_\_init\_\_(self,Name)*

*def getchildName(self):*

*return "This is Child 2 with Name {}".format(self.cName)*

*print(child1.getParentName())*

*print(child1.getchildName())*

*print(child2.getParentName())*

*print(child2.getchildName())*

e. Hybrid Inheritance- When there is more than one inheritance involved in a single scenario.

Like Multilevel with Hierarchical inheritance

Example-s

Q How can you reverse a List without using any module?

List[::-1]

-----------------------------------------------------------------------------------------------------------------------------------------

#### ****Question: Does Python support an intrinsic do-while loop?****

**Answer:**No Python does not support an intrinsic do-while loop.

---------------------------------------------------------------------------------------------------------------------------------------

#### Question: What is the map() function used for in Python?

**Answer:** The map() function applies a given function to each item of an iterable. It then returns a list of the results. The value returned from the map() function can then be passed on to functions to the likes of the list() and set().

Typically, the given function is the first argument and the iterable is available as the second argument to a map() function. Several tables are given if the function takes in more than one arguments.

Filter() - -

----------------------------------------------------------------------------------------------------------------------------------------

@differece between an list and a numpy array?

the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the arrays along each dimension.

The Python core library provided [**Lists**](https://webcourses.ucf.edu/courses/1249560/pages/lists). A list is the Python equivalent of an array, but is resizeable and can contain elements of different types.

--------------------------------------------------------------------------------------------------------------------------------------

#### Question: What is the lambda function?

**Answer:**Lambda functions are small anonymous functions in Python. It has single expressions and accepts multiples arguments.

-----------------------------------------------------------------------------------------------------------------------------------------

#### What is the \_\_init\_\_ method?

The \_\_init\_\_ is the constructor method similar to the constructors in other OOP languages. It executes immediately when we create an object for the class. It’s used to initialize the initial data for the instance.

This method allocates memory to the object as soon as it is executed.

-----------------------------------------------------------------------------------------------------------------------------------------

#### Question: What is the lambda function?

**Answer:**An anonymous function is known as a lambda function. This function can have only one statement but can have any number of parameters.

--------------------------------------------------------------------------s----------------------------------------------------------------

Question: How are files deleted in Python?

**Answer:**To delete a file in Python:

1. Import OS module
2. Use os.remove() function

#### Explain some methods of the list.

#### -----------------------------------------------------------------------------------------------------------------------------------------

#### 12. What’s the negative indexing in lists?

But, negative indexing allows us to access the index from the end of the list. The start of the negative indexing is -1. And it keeps on increasing like -2, -3, -4, etc.., till the length of the list.

----------------------------------------------------------------------------------------------------------------------------------------

#### #39. What is the difference between is and ==?

The == operator is used to check whether two objects have the same value or not. The is operator is used to check whether two objects are referring to the same memory location or not.

----------------------------------------------------------------------------------------------------------------------------------------

### **71. What advantages do NumPy arrays offer over (nested) Python lists?**

**Nested Lists**:

* Python lists are efficient general-purpose containers that support efficient operations like insertion,appending,deletion and concatenation.
* The limitations of lists are that they don’t support “vectorized” operations like element wise addition and multiplication, and the fact that they can contain objects of differing types mean that Python must store type information for every element, and must execute type dispatching code when operating on each element.
* Can consist of elements of different data type.
* Can have nested elements of different size
* Flexible to modify
* Consume more memory
* Can be printed entirely without using Loop

**Numpy**:

* NumPy is more efficient and more convenient as you get a lot of vector and matrix operations for free, which helps to avoid unnecessary work and complexity of the code.Numpy is also efficiently implemented when compared to nested
* NumPy array is faster and contains a lot of built-in functions which will help in FFTs, convolutions, fast searching, linear algebra,basic statistics, histograms,etc.
* Can only consist of elements of same data type
* Can only consist of nested element of the same size
* Not very flexible and ready for modification
* Consume less memory and faster
* Cannot be printed entirely without using a loop.

------------------------------------------------------------------------------------------------------------------------------------------

### **Write a Python program to check whether a given string is a palindrome or not, without using an iterative method?**

def fun(string):

s1 = string

s = string[::-1]

if(s1 == s):

return true

else:

return false

print(fun(“madam”))

------------------------------------------------------------------------------------------------------------------------------------------

### **95. Write a program in Python to execute the Bubble sort algorithm?**

def bubbleSort(x):

n = len(x)

# Traverse through all array elements

for i in range(n-1):

for j in range(0, n-i-1):

if x[j] > x[j+1] :

x[j], x[j+1] = x[j+1], x[j]

# Driver code to test above

arr = [25, 34,47, 21, 22, 11,37]

bubbleSort(arr)

print ("Sorted array is:")

for i in range(len(arr)):

print ("%d" %arr[i]),

**Output:**

11,21,22,25,34,37,47

-----------------------------------------------------------------------------------------------------------------------------------------

### **96. Write a program in Python to produce Star triangle?**

def Star\_triangle(n):

for x in range(n):

print(' '\*(n-x-1)+'\*'\*(2\*x+1))

Star\_triangle(9)

Output:

\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------------------------------------------------------

### **97. Write a program to produce Fibonacci series in Python?**

Fibonacci series refers to the series where the element is the sum of two elements prior to it.

n = int(input("number of terms? "))

n1, n2 = 0, 1

count = 0

if n <= 0:

print("Please enter a positive integer")

elif n == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < n:

print(n1)

nth = n1 + n2

n1 = n2

n2 = nth

count += 1

------------------------------------------------------------------------------------------------------------------------------------------

### **98. Write a program in Python to check if a number is prime?**

num = 13

if num > 1:

for i in range(2, int(num/2)+1):

if (num % i) == 0:

print(num, "is not a prime number")

break

else:

print(num, "is a prime number")

else:

print(num, "is not a prime number")

**Output:**

13 is a prime number

### **99. Write a sorting algorithm for a numerical dataset in Python?**

code to sort a list in Python:

my\_list = ["8", "4", "3", "6", "2"]

my\_list = [int(i) for i in list]

my\_list.sort()

print (my\_list)

**Output:**

2,3,4,6,8

### **100. Write a Program to print ASCII Value of a character in python?**

x= 'a'

# print the ASCII value of assigned character stored in x

print(" ASCII value of '" + x + "' is", ord(x))

**Output:** 65

-----------------------------------------------------------------------------------------------------------------------------------------

### **6) What are the significant features of the pandas Library?**

The key features of the panda's library are as follows:

* Memory Efficient
* Data Alignment
* Reshaping
* Merge and join
* Time Series

---------------------------------------------------------------------------------------------------------------------------------------

Difference between pandas series nd dataframe? How would you create Panda Dataframe an d Panda Seres?

--------------------------------------------------------------------------------------------------------------------------------------

How do you add a new column to the Dataframe?

--------------------------------------------------------------------------------------------------------------------------------------

What is the diffrenec between iloc and loc?

* The **loc** basically works for the labels of our index. It can be understood as if we insert in loc[4], which means we are looking for that values of DataFrame that have an index labeled 4.
* The **iloc** basically works for the positions in the index. It can be understood as if we insert in iloc[4], which means we are looking for the values of DataFrame that are present at index '4`.

--------------------------------------------------------------------------------------------------------------------------------------

### **19) How to get the items of series A not present in series B?**

We can remove items present in **p2** from **p1** using isin() method.

1. **import** pandas as pd
2. p1 = pd.Series([2, 4, 6, 8, 10])
3. p2 = pd.Series([8, 10, 12, 14, 16])
4. p1[~p1.isin(p2)]

--------------------------------------------------------------------------------------------------------------------------------------

### **20) How to get the items not common to both series A and series B?**

We get all the items of p1 and p2 not common to both using below example:

1. **import** pandas as pd
2. **import** numpy as np
3. p1 = pd.Series([2, 4, 6, 8, 10])
4. p2 = pd.Series([8, 10, 12, 14, 16])
5. p1[~p1.isin(p2)]
6. p\_u = pd.Series(np.union1d(p1, p2))  # union
7. p\_i = pd.Series(np.intersect1d(p1, p2))  # intersect
8. p\_u[~p\_u.isin(p\_i)]

--------------------------------------------------------------------------------------------------------------------------------------

### **21) How to get the minimum, 25th percentile, median, 75th, and max of a numeric series?**

We can compute the minimum, 25th percentile, median, 75th, and maximum of **p** as below example:

1. **import** pandas as pd
2. **import** numpy as np
3. p = pd.Series(np.random.normal(14, 6, 22))
4. state = np.random.RandomState(120)
5. p = pd.Series(state.normal(14, 6, 22))
6. np.percentile(p, q=[0, 25, 50, 75, 100])

--------------------------------------------------------------------------------------------------------------------------------------

### **22) How to get frequency counts of unique items of a series?**

We can calculate the frequency counts of each unique value p as below example:

1. **import** pandas as pd
2. **import** numpy as np
3. p= pd.Series(np.take(list('pqrstu'), np.random.randint(6, size=17)))
4. p = pd.Series(np.take(list('pqrstu'), np.random.randint(6, size=17)))
5. p.value\_counts()

--------------------------------------------------------------------------------------------------------------------------------------

### **23) How to convert a numpy array to a dataframe of given shape?**

We can reshape the series **p** into a dataframe with 6 rows and 2 columns as below example:

1. **import** pandas as pd
2. **import** numpy as np
3. p = pd.Series(np.random.randint(1, 7, 35))
4. # Input
5. p = pd.Series(np.random.randint(1, 7, 35))
6. info = pd.DataFrame(p.values.reshape(7,5))
7. print(info)

---------------------------------------------------------------------------------------------------------------------

### **24) How can we convert a Series to DataFrame?**

The Pandas **Series.to\_frame()** function is used to convert the series object to the DataFrame.

1. Series.to\_frame(name=None)

**name:** Refers to the object. Its Default value is None. If it has one value, the passed name will be substituted for the series name.

1. s = pd.Series(["a", "b", "c"],
2. name="vals")
3. s.to\_frame()

--------------------------------------------------------------------------------------------------------------------------------------

### **25) What is Pandas NumPy array?**

Numerical Python (Numpy) is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single-dimensional array elements. The calculations using Numpy arrays are faster than the normal Python array.

---------------------------------------------------------------------------------------------------------------------

### **40) Define GroupBy in Pandas?**

In Pandas, **groupby()** function allows us to rearrange the data by utilizing them on real-world data sets. Its primary task is to split the data into various groups. These groups are categorized based on some criteria. The objects can be divided from any of their axes.

**DataFrame.groupby(by=None, axis=0, level=None, as\_index=True, sort=True, group\_keys=True, squeeze=False, \*\*kwargs)**