Python Interview Questions

Python Programming:

1. What type of language is python? Programming or scripting?
Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.

2. Is python case sensitive?
Yes, python is a case sensitive language.

3. What is a lambda function in python?
An anonymous function is known as a lambda function. This function can have any number of parameters but can have just one statement.

4. What is the difference between deep and shallow copy?
Shallow copy is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values.
Deep copy is used to store the values that are already copied. Deep copy doesn't copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored.

5. What are the generators in Python?
Generators are a way of implementing iterators. A generator function is a normal function except that it contains yield expression in the function definition making it a generator function.

6. What is slicing in Python?
Slicing in Python is a mechanism to select a range of items from Sequence types like strings, list, tuple, etc.

7. Why is the "pass" keyword used in Python?
The "pass" keyword is a no-operation statement in Python. It signals that no action is required. It works as a placeholder in compound statements which are intentionally left blank.

8. What are decorators in Python?
Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented by the @decorator_name in Python and are called in bottom-up fashion

9. What is the key difference between lists and tuples in python?
The key difference between the two is that while lists are mutable, tuples on the other hand are immutable objects.

10. How to display multiple images in a jupyter notebook?

```
for ima in images:
 plt.figure()
 plt.imshow(ima)
```

11. What is zip() function in Python?

Python zip() function returns a zip object, which maps a similar index of multiple containers. It takes an iterable, convert into iterator and aggregates the elements based on iterables passed.

12. Write a program to find all the prime number between 1 and 200.

```
lower = 1
upper = 200
print("Prime numbers between", lower, "and", upper, "are:")
for num in range(lower, upper + 1):
     all prime numbers are greater than 1
  if num > 1:
     for i in range(2, num):
        if (num % i) == 0:
           break
     else:
        print(num)
```

13. What is split() function used for?

Split fuction is used to split a string into shorter string using defined seperatos. letters = (” A, B,

C”)

n = text.split(“,”)

print(n)

o/p: [‘A’, ‘B’, ‘C’]

14. What are Keywords in Python?

Keywords in Python are reserved words which are used as identifiers, function name or variable name. They help define the structure and syntax of the language.

15. What is function?

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing. Python gives you many built-in functions like print(),

16. Why do we need break and continue in Python?
Both break and continue are statements that control flow in *Python loops*. break stops the current loop from executing further and transfers the control to the next block. continue jumps to the next iteration of the loop without exhausting it.

17. What is recursion?
When a function makes a call to itself, it is termed *recursion*. But then, in order for it to avoid forming an infinite loop, we must have a base condition.

18. How do you calculate the length of a string?
This is simple. We call the function len() on the string we want to calculate the length of.

19. What is PEP 8?
PEP 8 is a coding convention that lets us write more readable code. In other words, it is a set of recommendations.

20. Explain join() and split() in Python.

join() lets us join characters from a string together by a character we specify.

```
','.join('12345')
‘1,2,3,4,5’
```

split() lets us split a string around the character we specify.

```
  '1,2,3,4,5'.split(',')
[‘1’, ‘2’, ‘3’, ‘4’, ‘5’]
```

21. What is Monkey Patching?
Monkey patching refers to modifying a class or module at runtime (dynamic modification). It extends Python code at runtime.
Example:

```
from pkg.module import MyClass
def sayhello(self):
      print("Hello")

MyClass.sayhello=sayhello
```

22. What do you mean by *args and **kwargs?
In cases when we don’t know how many arguments will be passed to a function, like when we want to pass a list or a tuple of values, we use *args.
Example:

```
  def func(*args):
   for i in args:
      print(i)
  func(3,2,1,4,7)
```

**kwargs takes keyword arguments when we don’t know how many there will be.

```
  def func(**kwargs):
```

```
  for i in kwargs:
      print(i,kwargs[i])
func(a=1,b=2,c=7)
```

22. What is PYTHONPATH in Python?

PYTHONPATH is an environment variable which you can set to add additional directories where Python will look for modules and packages. This is especially useful in maintaining Python libraries that you do not wish to install in the global default location.

23. What is the difference between .py and .pyc files?

.py files contain the source code of a program. Whereas, .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

24. What is pickling and unpickling?

Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using the dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

25. How is Python interpreted?

Python language is an interpreted language. The Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

26. What is the difference between lists and tuples?

Lists	Tuples
Lists are mutable, i.e., they can be edited	Tuples are immutable (they are lists that cannot be edited)
Lists are usually slower than tuples	Tuples are faster than lists
Lists consume a lot of memory	Tuples consume less memory when compared to lists
Lists are less reliable in terms of errors as unexpected changes are more likely to occur	Tuples are more reliable as it is hard for any unexpected change to occur
Lists consist of many built-in functions	Tuples do not have any built-in functions
Syntax: list_1 = [10, ‘Intellipaat’, 20]	Syntax: tup_1 = (10, ‘Intellipaat’ , 20)

27. **What is a dictionary in Python?**

Python dictionary is one of the supported data types in Python. It is an unordered collection of elements. The elements in dictionaries are stored as key–value pairs. Dictionaries are indexed by keys.
dict={'Country':'India','Capital':'New Delhi', }

28. What are the common built-in data types in Python?

Python supports the below-mentioned built-in data types:

Immutable data types:

- Number
- String
- Tuple

Mutable data types:

- List
- Dictionary
- set

29. What is type conversion in Python?

Python provides you with the much-needed functionality of converting one form of data type into another needed one, and this is known as type conversion.

Type conversion in Python is classified into two types:

1. Implicit type conversion: In this form of type conversion, Python Interpreter helps in automatically converting the data type into another data type without any user involvement.
2. Explicit type conversion: In this form of type conversion, the data type has to be changed into the required type by the user.

Various functions of explicit conversion are shown below:
int() – Converts any data type into an integer
float() – Converts any data type into float
ord() – Converts characters into an integer
hex() – Converts integers to hexadecimal strings
oct() – Converts integers to octal strings
tuple() – Converts the data type to a tuple
set() – Returns the type after converting to a set
list() – Converts any data type to a list
dict() – Used to convert a tuple of order (key,value) into a dictionary
str() – Used to convert integers into a string complex(real,imag) – Used to convert real numbers to complex (real,imag) numbers

30. What does [::-1] do?

This [::-1] is an example of slice notation and helps reverse the sequence with the help of indexing.

[Start,stop,step count]
How would you convert a string into lowercase?
We use the lower() method for this.

```
AyuShi'.lower()
```

'ayushi'

31. How do you get a list of all the keys in a dictionary?

```
mydict={'a':1,'b':2,'c':3,'e':5}
mydict.keys()
```

32. How do you reverse a list?
Using the reverse() method.

```
a.reverse()
```

33. How will you convert a list into a string?

```
nums=['one','two','three','four','five','six','seven']
s=' '.join(nums)
```

34. How will you remove a duplicate element from a list?

```
list=[1,2,1,3,4,2]
set(list)
```

35. Explain Python List Comprehension.
The *list comprehension in python* is a way to declare a list in one line of code. Let's take a look at one such example.

```
[i for i in range(1,11,2)]
[1, 3, 5, 7, 9]
```

36. Create a new list to convert the following list of number strings to a list of numbers.
nums=['22','68','110','89','31','12']
We will use the int() function with a list comprehension to convert these strings into integers and put them in a list.

```
[int(i) for i in nums]
[22, 68, 110, 89, 31, 12]
```

3**7. What is the difference between append() and extend() methods?**
Both append() and extend() methods are methods used to add elements at the end of a list.

- append(element): Adds the given element at the end of the list that called this append() method
- extend(another-list): Adds the elements of another list at the end of the list that called this extend() method

38. Write a sorting algorithm for a numerical dataset in Python.
The code to sort a list in Python can be written as follows:

```
my_list = ["8", "4", "3", "6", "2"]
my_list = [int(i) for i in list]
my_list.sort()
print (my_list)
```

39. **Explain split(), sub(), and subn() methods of 're' module in Python.**
These methods belong to the Python RegEx 're' module and are used to modify strings.

- split(): This method is used to split a given string into a list.
- sub(): This method is used to find a substring where a regex pattern matches, and then it replaces the matched substring with a different string.
- subn(): This method is similar to the sub() method, but it returns the new string, along with the number of replacements.

40. Write a Python program to check whether a given string is a palindrome or not, without using an iterative method.

A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam, nurses run, etc.

Consider the below code:

```
def fun(string):
s1 = string
s = string[::-1]
if(s1 == s):
return true
else:
return false
print(fun("madam"))
```

41. How will you capitalize the first letter of a string?
Simply using the method capitalize().

```
'ayushi'.capitalize()
'Ayushi'
```

42. Write code to print everything in the string except the spaces.

```
for i in s:
if i==' ': continue
print(i,end='')
```

Numpy

43. Why is python numpy better than lists?

Python numpy arrays should be considered instead of a list because they are fast, consume less memory and convenient with lots of functionality.

44. Describe the map function in Python?

map function executes the function given as the first argument on all the elements of the iterable given as the second argument.

45. Generate array of '100' random numbers sampled from a standard normal distribution using Numpy

np.random.rand(100) will create 100 random numbers generated from standard normal distribution with mean 0 and standard deviation 1.

46. How to count the occurrence of each value in a numpy array?

Use numpy.bincount()

```
arr = numpy.array([0, 5, 5, 0, 2, 4, 3, 0, 0, 5, 4, 1, 9, 9])
numpy.bincount(arr)
```

The argument to bincount() must consist of booleans or positive integers. Negative integers are invalid.

47. Does Numpy Support Nan?

nan, short for "not a number", is a special floating point value defined by the IEEE-754 specification. Python numpy supports nan but the definition of nan is more system dependent and some systems don't have an all round support for it like older cray and vax computers.

48. What does ravel() function in numpy do?

It combines multiple numpy arrays into a single array

49. What is the meaning of axis=0 and axis=1?

Axis = 0 is meant for reading rows, Axis = 1 is meant for reading columns Follow Steve Nouri for more AI and Data science posts: https://lnkd.in/gZu463X

50. What is numpy and describe its use cases?

Numpy is a package library for Python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions. In simple words, Numpy is an optimized version of Python lists like Financial functions, Linear Algebra, Statistics, Polynomials, Sorting and Searching etc.

51. How to remove from one array those items that exist in another?

```
a = np.array([5, 4, 3, 2, 1])
b = np.array([4, 8, 9, 10, 1])
From 'a' remove all of 'b'
np.setdiff1d(a,b)
Output:
array([5, 3, 2])
```

51. How to sort a numpy array by a specific column in a 2D array?

```
Choose column 2 as an example
import numpy as np
arr = np.array([[1, 2, 3], [4, 5, 6], [0,0,1]])
arr[arr[:,1].argsort()]
Output
array([[0, 0, 1], [1, 2, 3], [4, 5, 6]])
```

52. How to reverse a numpy array in the most efficient way?

```
import numpy as np
arr = np.array([9, 10, 1, 2, 0])
reverse_arr = arr[::-1]
```

53. How to calculate percentiles when using numpy?

```
import numpy as np
arr = np.array([11, 22, 33, 44 ,55 ,66, 77])
perc = np.percentile(arr, 40)   Returns the 40th percentile
print(perc)
```

54. What Is The Difference Between Numpy And Scipy?

NumPy would contain nothing but the array data type and the most basic operations: indexing, sorting, reshaping, basic element wise functions, et cetera. All numerical code would reside in **SciPy.** SciPy contains more fully-featured versions of the linear algebra modules, as well as many other numerical algorithms. Follow Steve Nouri for more AI and Data science posts: https://lnkd.in/gZu463X

55. What Is The Preferred Way To Check For An Empty (zero Element) Array?

For a numpy array, use the size attribute. The size attribute is helpful for determining the length of numpy array:

```
arr = numpy.zeros((1,0))
arr.size
```

56. What Is The Difference Between Matrices And Arrays?

Matrices can only be two-dimensional, whereas arrays can have any number of dimensions

57. How can you find the indices of an array where a condition is true?
Given an array a, the condition arr > 3 returns a boolean array and since False is interpreted as 0 in Python and NumPy.

```
import numpy as np
arr = np.array([[9,8,7],[6,5,4],[3,2,1]])
arr > 3
array([[True, True, True],
[ True, True, True],
[False, False, False]], dtype=bool)
```

58. How to find the maximum and minimum value of a given flattened array?

```
import numpy as np
a = np.arange(4).reshape((2,2))
max_val = np.amax(a)
min_val = np.amin(a)
```

59. Write a NumPy program to calculate the difference between the maximum and the minimum values of a given array along the second axis.

```
import numpy as np
arr = np.arange(16).reshape((4, 7))
res = np.ptp(arr, 1)
```

60. Find median of a numpy flattened array

```
import numpy as np
arr = np.arange(16).reshape((4, 5))
res = np.median(arr)
```

61. Write a NumPy program to compute the mean, standard deviation, and variance of a given array along the second axis

```
import numpy as np
import numpy as np
x = np.arange(16)
mean = np.mean(x)
std = np.std(x)
var= np.var(x)
```

62. Calculate covariance matrix between two numpy arrays

```
import numpy as np
x = np.array([2, 1, 0])
y = np.array([2, 3, 3])
cov_arr = np.cov(x, y)
```

62. Compute Compute pearson product-moment correlation coefficients of two given numpy arrays

```
import numpy as np
x = np.array([0, 1, 3])
y = np.array([2, 4, 5])
cross_corr = np.corrcoef(x, y)
```

64. Develop a numpy program to compute the histogram of nums against the bins

```
import numpy as np
nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])
bins = np.array([0, 1, 2, 3])
np.histogram(nums, bins)
```

65. Get the powers of an array values element-wise

```
import numpy as np
x = np.arange(7)
np.power(x, 3)
```

66. Write a NumPy program to get true division of the element-wise array inputs

```
import numpy as np
x = np.arange(10)
np.true_divide(x, 3)
```

Pandas

67. What is a series in pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types. The row labels of the series are called the index. By using a 'series' method, we can easily convert the list, tuple, and dictionary into series. A Series cannot contain multiple columns.

68. What features make Pandas such a reliable option to store tabular data?

Memory Efficient, Data Alignment, Reshaping, Merge and join and Time Series.

69. What is reindexing in pandas?

Reindexing is used to conform DataFrame to a new index with optional filling logic. It places NA/NaN in that location where the values are not present in the previous index. It returns a new object unless the new index is produced as equivalent to the current one, and the value of copy becomes False. It is used to change the index of the rows and columns of the DataFrame.

70. How will you create a series from dict in Pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types.

```
import pandas as pd
info = {'x' : 0., 'y' : 1., 'z' : 2.}
a = pd.Series(info)
```

71. How can we create a copy of the series in Pandas?

Use pandas.Series.copy method

```
import pandas as pd
pd.Series.copy(deep=True)
```

72. What is groupby in Pandas?

GroupBy is used to split the data into groups. It groups the data based on some criteria. Grouping also provides a mapping of labels to the group names. It has a lot of variations that can be defined with the parameters and makes the task of splitting the data quick and easy.

73. What is vectorization in Pandas?

Vectorization is the process of running operations on the entire array. This is done to reduce the amount of iteration performed by the functions. Pandas have a number of vectorized functions like aggregations, and string functions that are optimized to operate specifically on series and DataFrames. So it is preferred to use the vectorized pandas functions to execute the operations quickly.

74. Mention the different types of Data Structures in Pandas

Pandas provide two data structures, which are supported by the pandas library, Series, and DataFrames. Both of these data structures are built on top of the NumPy.

75. What Is Time Series In pandas

A time series is an ordered sequence of data which basically represents how some quantity changes over time. pandas contains extensive capabilities and features for working with time series data for all domains.

76. How to convert pandas dataframe to numpy array?

The function to_numpy() is used to convert the DataFrame to a NumPy array.
DataFrame.to_numpy(self, dtype=None, copy=False)
The dtype parameter defines the data type to pass to the array and the copy ensures the returned value is not a view on another array.

77. Write a Pandas program to get the first 5 rows of a given DataFrame

```
import pandas as pd
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
df.iloc[:5]
```

78. Develop a Pandas program to create and display a one-dimensional array-like object containing an array of data.

```
import pandas as pd
pd.Series([2, 4, 6, 8, 10])
```

79. Write a Python program to convert a Panda module Series to Python list and it's type.

```
import pandas as pd
ds = pd.Series([2, 4, 6, 8, 10])
type(ds)
ds.tolist()
type(ds.tolist())
```

80. Develop a Pandas program to add, subtract, multiple and divide two Pandas Series.

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 9])
```

```
sum = ds1 + ds2
sub = ds1 - ds2
mul = ds1 * ds2
div = ds1 / ds2
```

81. Develop a Pandas program to compare the elements of the two Pandas Series.

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])
ds1 == ds2
ds1 > ds2
ds1 < ds2
```

82. Develop a Pandas program to change the data type of given a column or a Series.

```
import pandas as pd
s1 = pd.Series(['100', '200', 'python', '300.12', '400'])
s2 = pd.to_numeric(s1, errors='coerce')
```

83. Write a Pandas program to convert Series of lists to one Series

```
import pandas as pd
s = pd.Series([ ['Red', 'Black'], ['Red', 'Green', 'White'] , ['Yellow']])
s = s.apply(pd.Series).stack().reset_index(drop=True)
```

84. Write a Pandas program to create a subset of a given series based on value and condition

```
import pandas as pd
s = pd.Series([0, 1,2,3,4,5,6,7,8,9,10])
n = 6
```

```
new_s = s[s < n]
new_s
```

85. Develop a Pandas code to alter the order of index in a given series

```
import pandas as pd
s = pd.Series(data = [1,2,3,4,5], index = ['A', 'B', 'C','D','E'])
s.reindex(index = ['B','A','C','D','E'])
```

86. Write a Pandas code to get the items of a given series not present in another given series.

```
import pandas as pd
sr1 = pd.Series([1, 2, 3, 4, 5])
sr2 = pd.Series([2, 4, 6, 8, 10])
result = sr1[~sr1.isin(sr2)]
result
```

87. What is the difference between the two data series df[‘Name’] and df.loc[:, ‘Name’]?

First one is a view of the original dataframe and second one is a copy of the original dataframe.

88. Write a Pandas program to display the most frequent value in a given series and replace everything else as “replaced” in the series.

```
import pandas as pd
import numpy as np
np.random.RandomState(100)
num_series = pd.Series(np.random.randint(1, 5, [15]))
result = num_series[~num_series.isin(num_series.value_counts().index[:1])] = 'replaced'
```

89. Write a Pandas program to find the positions of numbers that are multiples of 5 of a given series.

```
import pandas as pd
import numpy as np
num_series = pd.Series(np.random.randint(1, 10, 9))
result = np.argwhere(num_series % 5==0)
```

90. How will you add a column to a pandas DataFrame?

```
importing the pandas library
import pandas as pd
info = {'one' : pd.Series([1, 2, 3, 4, 5], index=['a', 'b', 'c', 'd', 'e']),
two' : pd.Series([1, 2, 3, 4, 5, 6], index=['a', 'b', 'c', 'd', 'e', 'f'])}
info = pd.DataFrame(info)
Add a new column to an existing DataFrame object
info['three']=pd.Series([20,40,60],index=['a','b','c'])
```

91. How to iterate over a Pandas DataFrame?
You can iterate over the rows of the DataFrame by using for loop in combination with an iterrows() call on the DataFrame.

OOPS Python Interview Questions

92. Explain Inheritance in Python with an example.
Ans: Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.
They are different types of inheritance supported by Python:

1. Single Inheritance – where a derived class acquires the members of a single super class.
2. Multi-level inheritance – a derived class d1 in inherited from base class base1, and d2 are inherited from base2.
3. Hierarchical inheritance – from one base class you can inherit any number of child classes
4. Multiple inheritance – a derived class is inherited from more than one base class.

93.What makes Python object-oriented?
Again the frequently asked Python Interview Question
Python is object-oriented because it follows the Object-Oriented programming paradigm. This is a paradigm that revolves around classes and their instances (objects). With this kind of programming, we have the following features:

- Encapsulation
- Abstraction
- Inheritance
- Polymorphism
- Data hiding

. What is an object?
An object is a real-world entity which is the basic unit of OOPs for example chair, cat, dog, etc. Different objects have different states or attributes, and behaviors.

94. What is a class?
A class is a prototype that consists of objects in different states and with different behaviors. It has a number of methods that are common the objects present within that class.

95. Differentiate between data abstraction and encapsulation.

Data abstraction	Encapsulation
Solves the problem at the design level	Solves the problem at the implementation level
Allows showing important aspects while hiding implementation details	Binds code and data together into a single unit and hides it from the world

96. What is a constructor?
A constructor is a special type of method that has the same name as the class and is used to initialize objects of that class.

97. What is an exception?
An exception is a kind of notification that interrupts the normal execution of a program. Exceptions provide a pattern to the error and transfer the error to the exception handler to resolve it. The state of the program is saved as soon as an exception is raised.

98. What is exception handling?
Exception handling in Object-Oriented Programming is a very important concept that is used to manage errors. An exception handler allows errors to be thrown and caught and implements a centralized mechanism to resolve them.

99. What is the difference between an error and an exception?
Error - Errors are problems that should not be encountered by applications.
Exception - Conditions that an application might try to catch.

100. What is a try/ catch block?
A try/ catch block is used to handle exceptions. The try block defines a set of statements that may lead to an error. The catch block basically catches the exception.

101. What is a finally block?
A finally block consists of code that is used to execute important code such as closing a connection, etc. This block executes when the try block exits. It also makes sure that finally block executes even in case some unexpected exception is encountered.