

PYTHON

Interview Guide

simplilearn



Ace Your Python Interview

Today, the world of Big Data and Analytics has gained enormous popularity and consequently, Python has become the preferred programming language among tech professionals. Scalability, simpler coding, and its collection of libraries and frameworks are just some of the features that make Python suitable for companies engaged in data or machine learning-based initiatives.

Are you applying for a job that involves knowledge of Python? If yes, we have your back. We have clubbed a list of 50 most popular questions you can expect in an interview. So prepare ahead of time, and crack your python-related interview in the first go.

Topics Covered

- Python Operators
- Lists, Tuples and Dictionaries
- Python Arrays
- Output questions
- Django
- Python in-built functions
- Numpy and Pandas
- Matplotlib
- Scikit Learn

Q: What Is the Difference Between a Shallow Copy and Deep Copy?

A: Deepcopy creates a different object and populates it with the child objects of the original object. Therefore, changes in the original object are not reflected in the copy.

`copy.deepcopy()` creates a Deep Copy.

Shallow copy creates a different object and populates it with the references of the child objects within the original object. Therefore, changes in the original object are reflected in the copy.

`copy.copy` creates a Shallow Copy.

Q: How Is Multithreading Achieved in Python?

A: Multithreading usually implies that multiple threads are executed concurrently. The Python Global Interpreter Lock doesn't allow more than one thread to hold the Python interpreter at that particular point of time. So multithreading in python is achieved through context switching. It is quite different from multiprocessing which actually opens up multiple processes across multiple threads.

Q: Discuss Django Architecture.

A: Django is a web service used to build your web pages. Its architecture is as shown:

- ✓ Template: the front end of the web page
- ✓ Model: the back end where the data is stored
- ✓ View: It interacts with the model and template and maps it to the URL
- ✓ Django: serves the page to the user

Q: What Advantage Does the Numpy Array Have over a Nested List?

A: Numpy is written in C so that all its complexities are backed into a simple to use a module. Lists, on the other hand, are dynamically typed. Therefore, Python must check the data type of each element every time it uses it. This makes Numpy arrays much faster than lists.

Numpy has a lot of additional functionality that list doesn't offer; for instance, a lot of things can be automated in Numpy.

Q: What are Pickling and Unpickling?

Pickling

- ✓ Converting a Python object hierarchy to a byte stream is called pickling
- ✓ Pickling is also referred to as serialization

Unpickling

- ✓ Converting a byte stream to a Python object hierarchy is called unpickling
- ✓ Unpickling is also referred to as deserialization

If you just created a neural network model, you can save that model to your hard drive, pickle it, and then unpickle to bring it back into another software program or to use it at a later time.

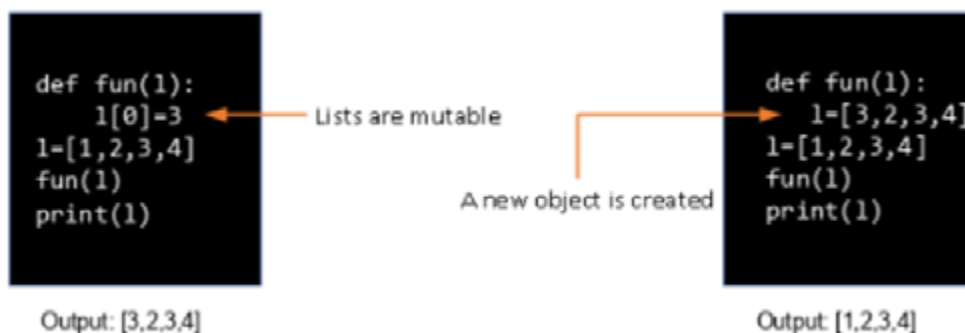
Q: How is Memory managed in Python?

A: Python has a private heap space that stores all the objects. The Python memory manager regulates various aspects of this heap, such as sharing, caching, segmentation, and allocation. The user has no control over the heap; only the Python interpreter has access.

Q: Are Arguments in Python Passed by Value or by Reference?

A: Arguments are passed in python by a reference. This means that any changes made within a function are reflected in the original object.

Consider two sets of code shown below:



In the first example, we only assigned a value to one element of 'l', so the output is `[3, 2, 3, 4]`.

In the second example, we have created a whole new object for 'l'. But, the values `[3, 2, 3, 4]` doesn't show up in the output as it is outside the definition of the function.

Q: How Would You Generate Random Numbers in Python?

A: To generate random numbers in Python, you must first import the random module.

The random() function generates a random float value between 0 & 1.

```
> random.random()
```

The randrange() function generates a random number within a given range.

Syntax: randrange(beginning, end, step)

Example - > random.
randrange(1,10,2)

Q: What Does the // Operator Do?

A: In Python, the / operator performs division and returns the quotient in the float.

For example: 5 / 2 returns 2.5

The // operator, on the other hand, returns the quotient in integer.

For example: 5 // 2 returns 2

Q: What Does the 'is' Operator Do?

A: The 'is' operator compares the id of the two objects.

```
list1=[1,2,3]
```

```
list2=[1,2,3]
```

```
list3=list1
```

```
list1 == list2 True
```

```
list1 is list2 False
```

```
list1 is list3 True
```

Q: What Is the Purpose of the Pass Statement?

A: The pass statement is used when there's a syntactic but not an operational requirement. For

example - The program below prints a string ignoring the spaces.

```
var="Si mplilea rn"
```

```
for i in var:
```

```
if i==" ":
```

```
    pass
```

```
else:
```

```
    print(i,end="")
```

Here, the pass statement refers to 'no action required.'

Q: How Will You Check If All the Characters in a String Are Alphanumeric?

A: Python has an inbuilt method isalnum() which returns true if all characters in the string are alphanumeric.

Example -

```
>> "abcd123".isalnum()
```

Output: True

```
>>"abcd@123#".isalnum()
```

Output: False

Another way is to use regex as shown.

```
>>import re
```

```
>>bool(re.match('[A-Za-z0-9]+$','abcd123'))
```

Output: True

```
>> bool(re.match('[A-Za-z0-9]+$','abcd@123'))
```

Output: False

Q: How Will You Merge Elements in a Sequence?

A: There are three types of sequences in Python:

- ✓ Lists
- ✓ Tuples
- ✓ Strings

Example of Lists -

```
>>l1=[1,2,3]
```

```
>>l2=[4,5,6]
```

```
>>l1+l2
```

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

Example of Tuples -

```
>>t1=(1,2,3)
```

```
>>t2=(4,5,6)
```

```
>>t1+t2
```

Output: (1,2,3,4,5,6)

Example of String -

```
>>s1="Simpli"
```

```
>>s2="learn"
```

```
>>s1+s2
```

Output: 'Simplilearn'

Q: How Would You Remove All Leading Whitespace in a String?

A: Python provides the inbuilt function `lstrip()` to remove all leading spaces from a string.

```
>>" Python".lstrip
```

Output: Python

Q: How Would You Replace All Occurrences of a Substring with a New String?

A: The `replace()` function can be used with strings for replacing a substring with a given string.
Syntax:

```
str.replace(old, new, count)
```

`replace()` returns a new string without modifying the original string.

Example -

```
>>"Hey John. How are you, John?".replace("john","John",1)
```

Output: "Hey John. How are you,

John?"

Q: What Is the Difference Between `Del` and `Remove()` on Lists?

A: **Del**

- ✓ Del removes all elements of a list within a given range
- ✓ Syntax: `del list[start:end]`

Remove()

- ✓ removes the first occurrence of a particular character
- ✓ Syntax: `list.remove(element)`

Here is an example to understand the two statements -

```
>>lis=['a', 'b', 'c', 'd']
```

```
>>del lis[1:3]
```

```
>>lis
```

Output: ["a","d"]

```
>>lis=['a', 'b', 'b', 'd']
```

```
>>lis.remove('b')
```

```
>>lis
```

Output: ['a', 'b', 'd']

Note that in the range 1:3, the elements are counted up to 2 and not 3.

Q: How Do You Display the Contents of a Text File in Reverse Order?

A: You can display the contents of a text file in reverse order using the following steps:

- ✓ Open the file using the `open()` function
- ✓ Store the contents of the file into a list
- ✓ Reverse the contents of the list

- ✓ Run a for loop to iterate through the list

Q: Differentiate Between `append()` and `extend()`.

A: **`append()`**

- ✓ `append()` adds an element to the end of the list
- ✓ Example -

```
>>lst=[1,2,3]
>>lst.append(4)
>>lst
Output:[1,2,3,4]
```

`extend()`

- ✓ `extend()` adds elements from an iterable to the end of the list
- ✓ Example -

```
>>lst=[1,2,3]
>>lst.extend([4,5,6])
>>lst
Output:[1,2,3,4,5,6]
```

Q: What Is the Output of the below Code? Justify Your Answer.

```
>>def addToList(val, list=[]):
>> list.append(val)
>> return list
>>list1 = addToList(1)
>>list2 = addToList(123,[])
>>list3 = addToList('a')
>>print ("list1 = %s" % list1)
>>print ("list2 = %s" % list2)
>>print ("list3 = %s" % list3)
```

A: Output:

```
list1 = [1,'a']
list2 = [123]
list3 = [1,'a']
```

Note that `list1` and `list3` are equal. When we passed the information to the `addToList`, we did it without a second value. If we don't have an empty list as the second value, it will start off with an empty list, which we then append. For `list2`, we appended the value to an empty list, so its value becomes `[123]`.

For `list3`, we're adding 'a' to the list. Because we didn't designate the list, it is a shared value. It means the list doesn't reset and we get its value as `[1, 'a']`.

Remember that a default list is created only once during the function and not during its call number.

Q: What Is the Difference Between a List and a Tuple?

A: Lists are mutable while tuples are immutable.

Example:

- ✓ **List**

```
>>lst = [1,2,3]
>>lst[2] = 4
>>lst
Output:[1,2,4]
```

- ✓ **Tuple**

```
>>tpl = (1,2,3)
>>tpl[2] = 4
>>tpl
```

Output:TypeError: 'tuple'

the object does not support item assignment

There is an error because you can't

change the tuple 1 2 3 into 1 2 4.
You have to completely reassign
tuple to a new value.

Q: What Is Docstring in Python?

A: Docstrings are used in providing documentation to various Python modules, classes, functions, and methods.

Example -

```
def add(a,b):
    """This function adds two
    numbers."""
    sum=a+b
    return sum
```

```
sum=add(10,20)
print("Accessing docstring
method 1:",add.__doc__)
```

```
print("Accessing docstring
method 2:",end="")
```

```
help(add)
```

Output -

Accessing docstring method 1: This
function adds two numbers.

Accessing docstring method 2:
Help on function add-in module
__main__:

```
add(a, b)
```

This function adds two numbers.

Q: What Is Docstring in Python?

A: Docstrings are used in providing documentation to various Python modules, classes, functions, and methods.

Example -

```
def add(a,b):
    """This function adds two
    numbers."""
    sum=a+b
    return sum
```

```
sum=add(10,20)
```

```
print("Accessing docstring
method 1:",add.__doc__)
```

```
print("Accessing docstring
method 2:",end="")
```

```
help(add)
```

Output -

Accessing docstring method 1: This
function adds two numbers.

Accessing docstring method 2:
Help on function add-in module
__main__:

```
add(a, b)
```

This function adds two numbers.

Q: How Do You Use Print() Without the Newline?

A: The solution to this depends on the Python version you are using.

Python v2

```
>>print("Hi. "),
```

```
>>print("How are you?")
```

Output: Hi. How are you?

Python v3

```
>>print("Hi",end=" ")
```

```
>>print("How are you?")
```

Output: Hi. How are you?

Q: How Do You Use the Split() Function in Python?

A: The split() function splits a string into a number of strings based on a specific delimiter.

Syntax -

```
string.split(delimiter, max)
```

Where:

the delimiter is the character based on which the string is split. By default it is space.

max is the maximum number of

splits

Example -

```
>>var="Red,Blue,Green,Orange"
```

```
>>lst=var.split(",2)
```

```
>>print(lst)
```

Output:

```
['Red','Blue','Green, Orange']
```

Here, we have a variable var whose values are to be split with commas. Note that '2' indicates that only the first two values will be split.

Q: Is Python Object-oriented or Functional Programming?

A: Python is considered a multi-paradigm language.

Python follows the object-oriented paradigm

- ✓ Python allows the creation of objects and their manipulation through specific methods
- ✓ It supports most of the features of OOPS such as inheritance and polymorphism
- ✓ Python follows the functional programming paradigm
- ✓ Functions may be used as the first-class object
- ✓ Python supports Lambda functions which are characteristic of the functional paradigm

Q: Write a Function Prototype That Takes a Variable Number of Arguments.

A: The function prototype is as follows:

```
def function_name(*list)
```

```
>>def fun(*var):
```

```
>> for i in var:
```

```
print(i)
```

```
>>fun(1)
```

```
>>fun(1,25,6)
```

In the above code, * indicates that there are multiple arguments of a variable.

Q: What Are *args and *kwargs?

A: ***args**

- It is used in a function prototype to accept a varying number of arguments.
- It's an iterable object.
- Usage - def fun(*args)

***kwargs**

- It is used in a function prototype to accept the varying number of keyworded arguments.
- It's an iterable object
- Usage - def fun(**kwargs):
fun(colour="red".units=2)

Q: "in Python, Functions Are First-class Objects." What Do You Infer from This?

A: It means that a function can be treated just like an object. You can assign them to variables, or pass them as arguments to other functions. You can even return them from other functions.

Q: What Is the Output Of: Print(__name__)? Justify Your Answer.

A: `__name__` is a special variable that holds the name of the current module. Program execution starts from main or code with 0 indentations. Thus, `__name__` has a value `__main__` in the above case. If the file is imported from another module, `__name__` holds

the name of this module.

Q: What Is a Numpy Array?

A: A numpy array is a grid of values, all of the same type, and is indexed by a tuple of non-negative integers. The number of dimensions determines the rank of the array. The shape of an array is a tuple of integers giving the size of the array along each dimension.

Q: What Is the Difference Between Matrices and Arrays?

A: Matrices

- A matrix comes from linear algebra and is a two-dimensional representation of data
- It comes with a powerful set of mathematical operations that allow you to manipulate the data in interesting ways

Arrays

- An array is a sequence of objects of similar data type
- An array within another array forms a matrix

Q: How Do You Get Indices of N Maximum Values in a Numpy Array?

A:

```
>>import numpy as np
>>arr=np.array([1, 3, 2, 4, 5])
>>print(arr.argsort( ) [ -N: ][: -1])
```

Q: How Would You Obtain the Res_set from the Train_set and the Test_set from Below?

A:

```
>>train_set=np.array([1, 2, 3])
>>test_set=np.array([[0, 1, 2], [1, 2,
```

```
3]])
```

```
Res_set [[1, 2, 3], [0, 1, 2], [1, 2, 3]]
```

Choose the correct option:

1. `res_set = train_set.append(test_set)`
2. `res_set = np.concatenate([train_set, test_set])`
3. `resulting_set = np.vstack([train_set, test_set])`
4. None of these

Here, options a and b would both do horizontal stacking, but we want vertical stacking. So, option c is the right statement.

```
resulting_set = np.vstack([train_set, test_set])
```

Q: How Would You Import a Decision Tree Classifier in Sklearn? Choose the Correct Option.

1. `from sklearn.decision_tree import DecisionTreeClassifier`
2. `from sklearn.ensemble import DecisionTreeClassifier`
3. `from sklearn.tree import DecisionTreeClassifier`
4. None of these

A: 3. `from sklearn.tree import DecisionTreeClassifier`

Q: You Have Uploaded the Dataset in Csv Format on Google Spreadsheet and Shared It Publicly. How Can You Access This in Python?

A: We can use the following code:

```
>>link = https://docs.google.com/spreadsheets/d/...
>>source = StringIO.
```

```
StringIO(requests.get(link).
content))
>>data = pd.read_csv(source)
```

Q: What Is the Difference Between the Two Data Series given Below?

`df['Name']` and `df.loc[:, 'Name']`, where:

```
df = pd.DataFrame(['aa', 'bb',
'xx', 'uu'], [21, 16, 50, 33],
columns = ['Name', 'Age'])
```

Choose the correct option:

1. 1 is the view of original dataframe and 2 is a copy of original dataframe
2. 2 is the view of original dataframe and 1 is a copy of original dataframe
3. Both are copies of original dataframe
4. Both are views of original dataframe

A: 3. Both are copies of the original dataframe.

Q: You Get the Error “temp.Csv” While Trying to Read a File Using Pandas. Which of the Following Could Correct It?

Error:

```
Traceback (most recent
call last): File "<input>",
line 1, in<module>
UnicodeEncodeError:
```

'ascii' codec can't encode character.

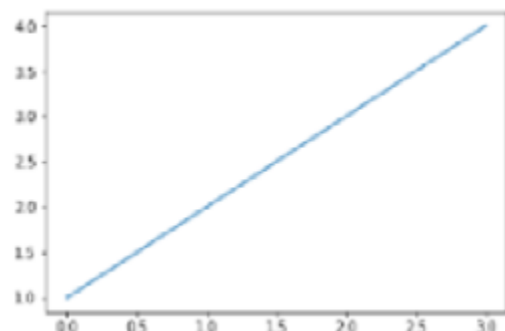
Choose the correct option:

1. `pd.read_csv("temp.csv", compression='gzip')`
2. `pd.read_csv("temp.csv", dialect='str')`
3. `pd.read_csv("temp.csv", encoding='utf-8')`
4. None of these

A: The error relates to the difference between utf-8 coding and a Unicode.

So option 3. `pd.read_csv("temp.csv", encoding='utf-8')` can correct it.

Q. How Do You Set a Line Width in the Plot given Below?



```
>>import matplotlib.pyplot as plt
```

```
>>plt.plot([1,2,3,4])
```

```
>>plt.show()
```

Choose the correct option:

1. In line two, write `plt.plot([1,2,3,4], width=3)`
2. In line two, write `plt.plot([1,2,3,4], line_width=3)`
3. In line two, write `plt.plot([1,2,3,4], lw=3)`
4. None of these

A: 3. In line two, write `plt.plot([1,2,3,4], lw=3)`

Q. How Would You Reset the Index of a Dataframe to a given List? Choose the Correct Option.

1. `df.reset_index(new_index,)`
2. `df.reindex(new_index,)`
3. `df.reindex_like(new_index,)`
4. None of these

A: 3. `df.reindex_like(new_index,)`

Q. How Can You Copy Objects in Python?

A: The function used to copy objects in Python are:

`copy.copy` for shallow copy and `copy.deepcopy()` for deep copy

Looking forward to a career as a Data Scientist? Check out the Data Science with Python Training Course and get certified today.

Q. What Is the Difference Between `range()` and `xrange()` Functions in Python?

A: **`range()`**

`range` returns a Python list object

`xrange()`

`xrange` returns an `xrange` object

Q. How Can You Check Whether a Pandas Dataframe Is Empty or Not?

A: The attribute `df.empty` is used to check whether a pandas data frame is empty or not.

```
>>import pandas as pd
>>df=pd.DataFrame({'A':[]})
>>df.empty
```

Output: True

Q. Write a Code to Sort an Array in

Numpy by the (N-1)Th Column.

A: This can be achieved by using `argsort()` function. Let us take an array X; the code to sort the (n-1)th column will be `x[x[:,n-2].argsort()]`

The code is as shown below:

```
>>import numpy as np
>>X=np.
array([[1,2,3],[0,5,2],[2,3,4]])
>>X[X[:,1].argsort()]
```

Output:

```
array([[1,2,3],[0,5,2],[2,3,4]])
```

Q. How Do You Create a Series from a List, Numpy Array, and Dictionary?

A: The code is as shown:

```
>> #Input
>>import numpy as np
>>import pandas as pd
>>mylist =
list('abcedfghijklmnopqrstuvwxyz')
>>myarr = np.arange(26)
>>mydict = dict(zip(mylist, myarr))
>> #Solution
>>ser1 = pd.Series(mylist)
>>ser2 = pd.Series(myarr)
>>ser3 = pd.Series(mydict)
>>print(ser3.head())
```

Q. How Do You Get the Items Not Common to Both Series a and Series B?

```
A: >> #Input
>>import pandas as pd
>>ser1 = pd.Series([1, 2, 3, 4, 5])
>>ser2 = pd.Series([4, 5, 6, 7, 8])
>> #Solution
>>ser_u = pd.Series(np.
union1d(ser1, ser2)) # union
```

```
>>ser_i = pd.Series(np.
intersect1d(ser1, ser2)) # intersect
>>ser_u[~ser_u.isin(ser_i)]
```

Q. How Do You Keep Only the Top Two Most Frequent Values as It Is and Replace Everything Else as 'other' in a Series?

A: >> #Input

```
>>import pandas as pd
>>np.random.RandomState(100)
>>ser = pd.Series(np.random.
randint(1, 5, [12]))
>> #Solution
>>print("Top 2 Freq:", ser.value_
counts())
>>ser[~ser.isin(ser.value_counts().
index[:2])] = 'Other'
>>ser
```

Q. How Do You Find the Positions of Numbers That Are Multiples of Three from a Series?

A: >> #Input

```
>>import pandas as pd
>>ser = pd.Series(np.random.
randint(1, 10, 7))
>>ser
>> #Solution
>>print(ser)
>>np.argwhere(ser % 3==0)
```

Q. How Do You Compute the Euclidean Distance Between Two Series?

A: The code is as shown:

```
>> #Input
>>p = pd.Series([1, 2, 3, 4, 5, 6, 7, 8,
9, 10])
>>q = pd.Series([10, 9, 8, 7, 6, 5, 4,
3, 2, 1])
```

```
>> #Solution
>>sum((p - q)**2)**.5
>> #Solution using func
>>np.linalg.norm(p-q)
```

You can see that the Euclidean distance can be calculated using two ways.

Q. How Do You Reverse the Rows of a Data Frame?

```
>> #Input
>>df = pd.DataFrame(np.
arange(25).reshape(5, -1))
>> #Solution
>>df.iloc[::-1, :]
```

Q. If You Split Your Data into Train/Test Splits, Is It Possible to over Fit Your Model?

A: Yes. One common beginner mistake is re-tuning a model or training new models with different parameters after seeing its performance on the test set.

Q. Which Python Library Is Built on Top of Matplotlib and Pandas to Ease Data Plotting?

A: Seaborn is a Python library built on top of matplotlib and pandas to ease data plotting. It is a data visualization library in Python that provides a high-level interface for drawing statistical informative graphs.

BREAKING INTO PYTHON

Cracking a job interview requires careful preparation apart from the right blend of skills and knowledge. There are a number of emerging job opportunities that demand proficiency in Python. As recruiters hunt for professionals with relevant skills, you need to ensure that you have a thorough knowledge of Python fundamentals. You can enroll in Simplilearn's [Data Science with Python course](#) to gain expertise in this language and become a potential candidate for your next job interview.



simplilearn

INDIA

Simplilearn Solutions Pvt Ltd.
53/1 C, Manoj Arcade, 24th Main,
Harlkunte
2nd Sector, HSR Layout
Bangalore: 560102
Call us at: 1800-212-7688

USA

Simplilearn Americas, Inc.
201 Spear Street, Suite 1100,
San Francisco, CA 94105
United States
Phone No: +1-844-532-7688