# Python Interview Questions

1. How is memory management done in python
2. Series from dictionary

import pandas as pd

# create a dictionary

dictionary = {'A' : 10, 'B' : 20, 'C' : 30}

# create a series

series = pd.Series(dictionary)

print(series)

1. Tuple of list, is list mutable

Yes, list in a tuple of list is mutable, because the tuple variable just stores the reference of the list. There is no way of knowing for the tuple is the list got modified as it only has the reference.

1. Convert date to utc

import datetime

timezone\_aware\_dt = datetime.datetime.now(datetime.timezone.utc)

1. Loc vs iloc

Both are used in slicing of data from the Pandas DataFrame. They help in the convenient selection of data from the DataFrame. They are used in filtering the data according to some conditions.

Loc:

loc() is label based data selecting method which means that we have to pass the name of the row or column which we want to select. This method includes the last element of the range passed in it, unlike iloc(). loc() can accept the boolean data unlike iloc() .

Selecting data according to some conditions :

# selecting cars with brand 'Maruti' and Mileage > 25

display(data.loc[(data.Brand == 'Maruti') & (data.Mileage > 25)])

Selecting a range of rows from the DataFrame :

# selecting range of rows from 2 to 5

display(data.loc[2 : 5])

Updating the value of any column :

# updating values of Mileage if Year < 2015

data.loc[(data.Year < 2015), ['Mileage']] = 22

display(data)

iloc:

iloc() is a indexed based selecting method which means that we have to pass integer index in the method to select specific row/column. This method does not include the last element of the range passed in it unlike loc(). iloc() does not accept the boolean data unlike loc().

Selecting rows using integer indices:

# selecting 0th, 2th, 4th, and 7th index rows

display(data.iloc[[0, 2, 4, 7]])

Selecting a range of columns and rows simultaneously:

# selecting rows from 1 to 4 and columns from 2 to 4

display(data.iloc[1 : 5, 2 : 5])

1. Difference between lambda and normal function

The primary difference between a lambda and a regular function is that the lambda function evaluates only a single expression and yields a function object. ... A lambda function doesn't contain a return statement because it will have only a single expression which is always returned by default

1. Elasticsearch
2. Is \_\_init\_\_ method present in all classes?

It is not compulsory, because instance is created with \_\_new\_\_ method, \_\_init\_\_ is used for initialization, so a class without init will still work fine.

1. How to access private variables and methods in python?

Using object.\_Class\_\_privateMethod()

Or object.\_Class\_\_privateVariable

1. What are python dunder methods?

These are special methods in python used for operator overloading, for example, using \_\_add\_\_ can help you add 2 object instances with + operator.

1. Difference between remove and delete

In python del is a keyword and remove(), pop() are in-built methods. The purpose of these three are same but the behavior is different remove() method delete values or object from the list using value and del and pop() deletes values or object from the list using an index.

1. Bubble sort

def bubbleSort(arr):

n = len(arr)

# Traverse through all array elements

for i in range(n):

# Last i elements are already in place

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

# traverse the array from 0 to n-i-1

# Swap if the element found is greater

# than the next element

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

arr[j], arr[j+1] = arr[j+1], arr[j]

# Driver code to test above

arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("Sorted array is:")

for i in range(len(arr)):

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

1. Calculate recursive sum of digits of a number such that the last answer is a single digit.

def acc(num):

accum=0

while num!=0:

digit=num%10

num//=10

accum+=digit

if accum > 9:

return acc(accum)

return accum

1. How to remove spaces in python
   1. Maketrans
   2. Translate
   3. Replace
   4. Re:

pattern = re.compile(r'\s+')

re.sub(pattern, '', string)

* 1. “”.join(string.split())

1. Swapcase
2. Dogpile python
3. Why does python not release memory on exit?

Objects referenced from the global namespaces of Python modules are not always deallocated when Python exits. This may happen if there are circular references. There are also certain bits of memory that are allocated by the C library that are impossible to free (e.g. a tool like Purify will complain about these). Python is, however, aggressive about cleaning up memory on exit and does try to destroy every single object.

If you want to force Python to delete certain things on deallocation use the atexit module to run a function that will force those deletions.

1. Zip function in python
2. Difference between isdecimal, isdigit and isnumeric functions

As you can see, the main difference between the three functions is:

isdecimal() method supports only Decimal Numbers.

isdigit() method supports Decimals, Subscripts, Superscripts.

isnumeric() method supports Digits, Vulgar Fractions, Subscripts, Superscripts, Roman Numerals, Currency Numerators.

1. Difference between rfind and rindex

Rfind returns -1 if substring is not found, rindex returns valueError

1. Difference between partition and rpartition

Rpartition does it from reverse and if substring is not found, it returns null value followed by the string itself.

1. Function/constructor overloading in python
2. Is Python Argument Passing model a “Pass by Value” or “Pass by Reference”?

Pass by Object Reference

The variable scope plays an important role in how the variables are treated within the function. Any variable passed to the function is passed as an object reference, if there is any change to the variable, that is visible to the caller function as well.

1. How can you split a list and pass the elements as parameters in python function?

Use tuple(list) or \*list

1. Method overloading in python

Overloading in python is not possible as the latest method only is visible so we need to use dispatch from multipledispatch.

1. How would you convert an integer variable into a list of digits without the use of loops?

list(map(int,str(var)))

1. Write below code in more pythonic way.

dict = {}

for i in range(a, b):

if i % c == d:

dict[i] = i/c

dict\_ = {i: i/c for i in range(a,b) if i % c == d}

1. How would you invert a dictionary key value combination?

mydict = {'a': 1, 'c': 2, 'b': 1, 'e': 2, 'd': 3, 'h': 1, 'j': 3}

from collections import defaultdict

reversed\_dict = defaultdict(list)

for key, value in mydict.items():

reversed\_dict[value].append(key)

For a tuple??

s = [('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]

d = defaultdict(list)

for k, v in s:

d[k].append(v)

Another option is:

d = {}

for k, v in s:

d.setdefault(k, []).append(v)

But the first option is faster

1. Count the frequency of each character in a string.

s = 'mississippi'

d = defaultdict(int)

for k in s:

d[k] += 1

sorted(d.items())

Option 2:

from collections import Counter

test\_str = "GeeksforGeeks"

res = Counter(test\_str)

print ("Count of all characters in GeeksforGeeks is :\n "+ str(res))

Option 3:

res = {}

for keys in test\_str:

res[keys] = res.get(keys, 0) + 1

Option 4:

res = {i : test\_str.count(i) for i in set(test\_str)}

1. What are literals in python?

The variable names that we use for creating variables in python are called literals.

Like x=2 , then x is a numeric literal

1. Explain the below difference

B=256

B1=256

B is B1

---True

B=1000

B1=1000

B is B1

---False

It's because Python caches small (between -5 and 256) int objects internally, so the objects used in the comparison were taken from the cache and were the same.

1. Print occurrences of all 3 digit numbers in a given string.

inpStr = '123412345123456'

# O(1) array creation.

freq = [0] \* 1000

# O(n) string processing.

for val in [int(inpStr[pos:pos+3]) for pos in range(len(inpStr) - 2)]:

freq[val] += 1

# O(1) output of relevant array values.

print ([(num, freq[num]) for num in range(1000) if freq[num] > 1])

1. What's the bad magic number error?

The magic number comes from UNIX-type systems where the first few bytes of a file held a marker indicating the file type.

Python puts a similar marker into its pyc files when it creates them.

Then the python interpreter makes sure this number is correct when loading it.

Anything that damages this magic number will cause you problem. This includes editing the pyc file or trying to run a pyc from a different version of python (usually later) than your interpreter.

If they are your pyc files, just delete them and let the interpreter re-compile the py files. On UNIX type systems, that could be something as simple as:

rm \*.pyc

or:

find . -name '\*.pyc' -delete

If they are not yours, you'll have to either get the py files for re-compilation, or an interpreter that can run the pyc files with that particular magic value.

1. How to add N number of seconds to a date?

import datetime

a = datetime.datetime(100,1,1,11,34,59)

b = a + datetime.timedelta(0,3) # days, seconds, then other fields.

OR

b = a + datetime.timedelta(seconds=3)

print(a.time())

print(b.time())

1. What is the use of enumerate?

Enumerate is used to iterate over iterators and have an index association with each element. We can also specify from which index we need to start with.

1. How will you handle memory leaks in python
2. Difference between numpy and normal array

While creating numpy array, we can give the type of data we want to store in the array and it will try to upcast or downcast the input value to fit the datatype, which is not the case with normal arrays.

1. What are annotations.

The basic idea of them is to hint to the readers of the code about what to expect as values of arguments in functions. This can be used in addition to docstring to add more details to the functions.

1. When to use boto3 resource vs client

Resource gives you high-level object-oriented API access and client gives low-level AWS service access.

All AWS service operations are supported by clients but not all are covered by resource.

Resource exposes subresources and collections of AWS resources with lazy evaluation which are loaded only when required further reducing API calls.

Code differences:

To list all objects in S3 bucket using resource:

import boto3

s3 = boto3.resource('s3')

bucket = s3.Bucket('mybucket')

for obj in bucket.objects.all():

print(obj.key, obj.last\_modified)

Same with client code but without pagination(this will only list 1000 objects, need to include pagination to list all objects)

import boto3

client = boto3.client('s3')

response = client.list\_objects\_v2(Bucket='mybucket')

for content in response['Contents']:

obj\_dict = client.get\_object(Bucket='mybucket', Key=content['Key'])

print(content['Key'], obj\_dict['LastModified'])

1. Write a list comprehension function to transpose a matrix

[[row[i] for row in matrix] for i in range(len(matrix))]

1. Collections module

Counter

OrderedDict

Defaultdict

Namedtuple

Deque

ChainMap

1. Json load vs loads and dump vs dumps
2. Itertools module
3. Asyncio module
4. Functools module
5. Random module
6. Dis module
7. Base64 module
8. Queue deque module
9. Regex module
10. Python packages
11. CommonWealth exceptions
12. Exec and eval
13. Descriptor
14. List destructing
15. Mixins
16. Basic curses with python
17. Chapter 195 - <file:///C:/Learnings/Big%20Data/books/PythonNotesForProfessionals_ezuwbf.pdf>
18. Difference between list and arrays

Lists are heterogeneous arrays are homogeneous

1. Reverse of a list

List[::-1] or reverse

1. Difference between series and dataframe

Series is an array of items and can be used to populate a column in a dataframe

1. What is an rdd

RDD in Apache Spark is an immutable collection of objects

1. How to create python package

First, we create a directory and give it a package name, preferably related to its operation.

Then we put the classes and the required functions in it.

Finally, we create an \_\_init\_\_.py file inside the directory, to let Python know that the directory is a package.

1. What is a namedtuple?

It is a subclass of tuple, which is used to create tuple-like objects with named fields and fixed lengths.

1. Two ways of creating namedtuples?

Using typing and collections.

From typing import NamedTuple

Or

From collections import namedtuple

1. What is a pyc file in python

Python automatically compiles your script to compiled code, so called byte code, before running it. When a module is imported for the first time, or when the source is more recent than the current compiled file, a .pyc file containing the compiled code will usually be created in the same directory as the .py file.

Simply running a python file will not create a pyc file.

1. list, tuple diff

Tuples can have values with heterogeneous datatypes, list is also a heterogenous collection but is most commonly used as a homogeneous collection and when the elements are to accessed by some kind of order.

Tuples are immutable, lists are mutable.

Because tuples are immutable, copying tuple to another object is not possible, it will be the same object. Whereas list will create a copy of the object but store at a different reference.

Lists occupy more space in memory than tuples as python allocates larger blocks with low overhead to tuples. Lists also tend to over allocate space to handle append operations faster.

Tuples are stored as struct and need no information regarding pointers whereas lists are stored with a layer of indirecton.

1. List sorting without using generic function:

Get familiar with mergesort algorithm and other algorithms in general

**import** **numpy** **as** **np**

**def** selection\_sort(x):

**for** i **in** range(len(x)):

swap = i + np.argmin(x[i:])

(x[i], x[swap]) = (x[swap], x[i])

**return** x

1. Can we use list as key in dict?

No, because lists are not hashable. An object is hashable if it has a hash value which never changes during its lifetime, while list does not have a \_\_hash\_\_ method and it is mutable. Simply speaking, if the list changes, the key in dictionary can no longer be referenced.

All immutable objects can be used as keys in dictionary and set members.

1. Dict sorting both key and value sorting

Sorted(dict) sorts on key, to sort on value, use sorted(x.items(), key=operator.itemgetter(1)) or

dict(sorted(x.items(), key=lambda item: item[1]))

1. What is List comprehension and where you use it?

It is a syntactic construct that helps to create a list from an existing list with some conditions on the go.

1. What is difference between iterator and iterable?

Iterator is an object that can be iterated upon. An iterable is an object that has an \_\_iter\_\_ method which returns an iterator, or which defines a \_\_getitem\_\_ method that can take sequential indexes starting from zero (and raises an IndexError when the indexes are no longer valid).

An iterator is an object with a next (Python 2) or \_\_next\_\_ (Python 3) method. An ITERATOR is an object that is self-iterable, meaning that it has an \_\_iter\_\_ method that returns self.

An iterator is consumable, iterable objects can be reiterated. List is an iterable object, hence it can be re-iterated.

1. What is a generator?

Generators are iterators, a kind of iterable you can only iterate over once. Generators do not store all the values in memory, they generate the values on the fly.

Python generators are a simple way of creating iterators. If a function contains at least one yield statement (it may contain other yield or return statements), it becomes a generator function. Generator is basically an iterator that creates an iterable object.

1. Give examples of built-in iterables

Tuples, lists, dictionaries, sets, frozen sets, strings, byte strings, byte arrays, ranges and memoryviews

1. What is the use of the yield keyword in Python? What does it do?

Yield is a keyword that is used to return values from a function and create a generator.

1. How to get Traceback?
2. How to write class

class A:

class variables

\_\_init\_\_ method

\_\_repr\_\_ method

Instance methods

1. What is self?

It represents the instance of the class. The object is implicitly passed to every method available on it, but we have to get it explicitly in every method while writing methods.

It can be named anything but it should always be the first argument to each function.

1. What is constructor?

A constructor is a special kind of method that Python calls when it instantiates an object using the definitions found in your class. Python relies on the constructor to perform tasks such as initializing (assigning values to) any instance variables that the object will need when it starts.

1. What is \_\_init\_\_?

It is the initializer method in python. When an object is created the init method is called.

Instances of a class usually store some sort of state information or data and the methods of the class offer a way to manipulate or do something with that state information or data. \_\_init\_\_ allows us to initialize this state information or data while creating an instance of the class.

1. What is \_\_init\_\_ and \_\_new\_\_

\_\_new\_\_ is the constructor method and \_\_init\_\_ is initializer method.

Python implicitly provides a default and typical implementation of \_\_new\_\_() which will invoke the superclass’s \_\_new\_\_() method to create a new instance object and then return it.

The order of calling would be \_\_new\_\_ and \_\_init\_\_

1. Why is \_\_init\_\_ always called after \_\_new\_\_?

\_\_new\_\_ is the first step of instance creation. It's called first, and is responsible for returning a new instance of your class.

1. Which is the destructor method in python

\_\_del\_\_

1. What is the \_\_doc\_\_ method in a class?

It outputs the string representation used in the class declaration.

1. What is decorator? How it works? How to write custom decorator?

Decorators are used to modify the working of a function or a class.

def my\_decorator(func):

def wrapper():

print("Something is happening before the function is called.")

func()

print("Something is happening after the function is called.")

return wrapper

@my\_decorator

def say\_whee():

print("Whee!")

1. What is the \_\_mro\_\_ method

It defines the method resolution order. In Python 3, it is defined at Class level, i.e. you cannot call it with your object instance.

The mro() method returns a python list while \_\_mro\_\_ attribute returns a tuple.

During multiple inheritance, sibling classes are inherited in the order they are defined in the subclass definition.

1. Difference between repr and str methods

Str method returns readable format of the object and repr method returns the official representation of an object. It is important to have repr method in every class, but str method is optional.

\_\_repr\_\_ goal is to be unambiguous

\_\_str\_\_ goal is to be readable

1. Why there is no Default implementation of \_\_repr\_\_ method in python?

Having a default for \_\_repr\_\_ would act like:

return "%s(%r)" % (self.\_\_class\_\_, self.\_\_dict\_\_)

It would have been too dangerous (for example, too easy to get into infinite recursion if objects reference each other). So Python cops out. Note that there is one default which is true: if \_\_repr\_\_ is defined, and \_\_str\_\_ is not, the object will behave as though \_\_str\_\_=\_\_repr\_\_.

1. What is the difference between class and instance variables?

Class variable is shared across all instances while instance variable is specific to an instance. Hence mutable objects like lists are not declared as class variables, rather included as instance variables.

1. What is super function in python and where you will use

Super function is used to inherit the parent class attributes and methods.

1. Single linked list?
2. Which version you are using?
3. Deprecation in python latest version or version which you are working
4. Memory management in Python?

Done by python memory manager.

1. What all the Advanced features added or removed from the latest version of python?
2. Built in data types in Python? Give some example?
3. Is it Python Call by reference or Call by value?

In python, object references are passed by value.

1. The difference between range and xrange?

Xrange is deprecated in python 3

The benefit of using xrange over range in python 2 was that the evaluation of the range or iterator was done lazily in case of xrange and not stored as a list in memory.

1. What can go in the keys of a dictionary?

All immutable datatypes can be used as a key because their hash value will never change.

s

1. How to delete a key in a dictionary?

my\_dict.pop('key', None)

1. What are static methods in python?

Static methods in Python are extremely similar to python class level methods, the difference being that a static method is bound to a class rather than the objects for that class.

This means that a static method can be called without an object for that class. This also means that static methods cannot modify the state of an object as they are not bound to it.

You can create a static method with “@staticmethod”.

1. How does python script work?

With Python, it uses an interpreter rather than a compiler. An interpreter works in exactly the same way as a compiler, with one difference: instead of code generation, it loads the output in-memory and executes it directly on your system

1. What is the constructor syntax?
2. Is there a numeric data type in python?

n Python, number data types are used to store numeric values. There are four different numerical types in Python:

int (plain integers): this one is pretty standard -- plain integers are just positive or negative whole numbers.

long (long integers): long integers are integers of infinite size. They look just like plain integers except they're followed by the letter "L" (ex: 150L).

float (floating point real values): floats represent real numbers, but are written with decimal points (or scientific notaion) to divide the whole number into fractional parts.

complex (complex numbers): represented by the formula a + bJ, where a and b are floats, and J is the square root of -1 (the result of which is an imaginary number). Complex numbers are used sparingly in Python.

1. What is namespace in python?

In Python, every name introduced has a place where resides and can be found.

This space is known as a namespace. It is an address location where a variable name is mapped to the object placed. Whenever the variable is searched out, this address location will be searched, to get the corresponding object.

1. Explain the difference between local and global namespaces.

Local namespaces are created within a function when that function is called. Global namespaces are created when the program starts.

1. 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 dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

1. How memory is managed in Python?

* Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
* The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
* Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

1. What are the tools that help to find bugs or perform static analysis?

PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

1. How can you copy an object in Python?

To copy an object in Python, you can try copy.copy () or copy.deepcopy() for the general case. You cannot copy all objects but most of them.

1. Copy vs deepcopy

Copy will create a collection object with references shared with the original object, hence any changes to the new object will reflect in the original object, whereas deepcopy will create a new bag with copies of original object elements

1. How to add new columns to pandas dataframe?
   1. By declaring a new list as column.
   2. By using Dataframe.insert()
   3. By using dataframe.assign()
   4. By using a dictionary
2. How to convert string column to datetime in pandas?
   1. By using pd.to\_datetime()
   2. By using Dataframe.astype(‘datetime64[ns]’)
3. What is the purpose of abstract class in python?

It is used to create blueprint of methods to be implemented by child classes, so when you want your classes to conform to some standard methods and implement them, you can have all these classes be child of abstract class.

1. What is an iterator in python?

In Python, an iterator is an object which implements the iterator protocol. The iterator protocol consists of two methods. The \_\_iter\_\_() method, which must return the iterator object, and the next() method, which returns the next element from a sequence.

1. What is Monkey Patching in Python?

Monkey patching is reopening the existing classes or methods in class at runtime and changing the behavior, which should be used cautiously, or you should use it only when you really need to.

1. Is there a benefit to using one over the other? In Python 2, they both seem to return the same results:

>>> 6/3

2

>>> 6//3

2

In Python 3.0, 5 / 2 will return 2.5 and 5 // 2 will return 2. The former is floating point division, and the latter is floor division, sometimes also called integer division.

In Python 2.2 or later in the 2.x line, there is no difference for integers unless you perform a from \_\_future\_\_ import division, which causes Python 2.x to adopt the behavior of 3.0

Regardless of the future import, 5.0 // 2 will return 2.0 since that’s the floor division result of the operation.

1. What is the use of \_\_name\_\_==”\_\_main\_\_” condition check?

When you want to implement a part of a code when the file is run as a program or independently, then you declare that code in this condition block. It will not be executed when it is imported as a module in another py file.

1. What is the importance of wholetextfiles in pyspark while processing files?

Wholetextfile will read each file as a single rdd element and return an rdd whose first element is filename and second is the content of the file. This serves well when you have a large list of small files.

Another way is to use spark.read.text(filepath, wholetext=True).select(input\_file\_name(), “value”).rdd

1. Why to convert a df to rdd?

Pyspark uses rdd since 1.0 was launched. Dataframe is new in 1.3, hence many transformations are not available in df as in rdd. So we convert dataframes to rdds.

1. Why use rdd.foreachpartition instead of foreach

Foreachpartition is used when there are heavy initializations, like database connections, or aws transfers which are to be initialized once per partition whereas foreach is used to apply on each element of a dataframe/dataset/rdd.