

1. What is a Class? How do you create it in Python ?

Answer:

A class is a blue print/ template of code /collection of objects that has same set of attributes and behaviour. To create a class use the keyword class followed by class name beginning with an uppercase letter. For example, a person belongs to class called Person class and can have the attributes (say first-name and last-name) and behaviours / methods (say showFullName()). A Person class can be defined as:

```
class Person():  
    #method  
    def inputName(self,fname,lname): self.fname=fname self.lastname=lastname  
    #method  
    def showFullName() (self):  
print(self.fname+" "+self.lname)person1 = Person() #object instantiation  
person1.inputName("Ratan","Tata") #calling a method inputName person1.  
showFullName() #calling a method showFullName()
```

Note: whenever you define a method inside a class, the first argument to the method must be self (where self – is a pointer to the class instance). self must be passed as an argument to the method, though the method does not take any arguments.

Top 48 Python Django Interview Questions And Answers

2. What happens if you put an else statement after a block ?

Answer:

The code in the else block is executed after the for loop completes, unless a break is encountered in the for loop execution. in which case the else block is not executed.

3. Explain the use of break and continue in Python looping ?

Answer:

The break statement stops execution of the current loop. and transfers control to the next

block. The continue statement ends the current block's execution and jumps to the next iteration of the loop.

4. What are Exception Handling? How do you achieve it in Python ?

Answer:

Exception Handling prevents the codes and scripts from breaking on receipt of an error at run-time might be at the time doing I/O, due to syntax errors, data types doesn't match.

Generally it can be used for handling user inputs.

The keywords that are used to handle exceptions in Python are:

try – it will try to execute the code that belongs to it. May be it used anywhere that keyboard input is required.

except – catches all errors or can catch a specific error. It is used after the try block.
`x = 10 + 'Python' #TypeError: unsupported operand type(s) try:`

```
x = 10 + 'Python'
```

```
except:
```

```
print("incompatible operand types to perform sum")
```

raise – force an error to occur

```
o raise TypeError("dissimilar data types")
```

finally – it is an optional clause and in this block cleanup code is written here following "try" and "except".

5. Explain Inheritance in Python with an example ?

Answer:

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. They are different types of inheritance supported by Python. They are:

single, multi-level, hierarchical and multiple inheritance. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

Single Inheritance – where a derived class acquires the members of a single super class.

multi-level inheritance – a derived class d1 in inherited from base class base1, and d2 is

inherited from base2.

hierarchical inheritance – from one base class you can inherit any number of child classes

multiple inheritance – a derived class is inherited from more than one base class.

ex:

```
class ParentClass:
```

```
    v1 = "from ParentClass – v1"
```

```
    v2 = "from ParentClass – v2"
class ChildClass(ParentClass):
```

```
    pass
c = ChildClass()
print(c.v1)
print(c.v2)
```

6. What Do You Think Is The Output Of The Following Code Fragment? Is There Any Error In The Code ?

Answer:

```
list = ['a', 'b', 'c', 'd', 'e']
```

```
print (list[10:])
```

The result of the above lines of code is []. There won't be any error like an IndexError.

You should know that trying to fetch a member from the list using an index that exceeds the member count (for example, attempting to access list[10] as given in the question) would yield an IndexError. By the way, retrieving only a slice at an opening index that surpasses the no. of items in the list won't result in an IndexError. It will just return an empty list.

7. What Is A Built-In Function That Python Uses To Iterate Over A Number Sequence ?

Answer:

range() generates a list of numbers, which is used to iterate over for loops.

```
for i in range(5):
```

```
    print(i)
```

The range() function accompanies two sets of parameters.

```
range(stop)
```

stop: It is the no. of integers to generate and starts from zero. eg. range(3) == [0, 1, 2].

`range([start], stop[, step])`

start: It is the starting no. of the sequence.

stop: It specifies the upper limit of the sequence.

step: It is the incrementing factor for generating the sequence.

Points to note:

Only integer arguments are allowed.

Parameters can be positive or negative.

8. What is the difference between tuples and lists in Python ?

Answer:

The main differences between lists and tuples are – Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as read-only lists.

Top 48 Python Django Interview Questions And Answers

9. How will you get all the values from the dictionary ?

Answer:

Using `dictionary.values()` function, we can get all the values from the dictionary object.

`print dict.values()` # Prints all the values.

10. What is the purpose pass statement in python ?

Answer:

pass statement – The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

11. How will you set the starting value in generating random numbers ?

Answer:

`seed([x])` – Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None.

12. How will you check in a string that all characters are decimal ?

Answer:

`isdecimal()` – Returns true if a unicode string contains only decimal characters and false otherwise.

13. What are the commands that are used to copy an object in Python ?

Answer:

The command that is used to copy an object in python includes:

- `copy.copy()` function: This makes a copy of the file from source to destination. It returns a shallow copy of the parameter that is passed.

- `copy.deepcopy()`: This also creates a copy of the object from source to destination. It returns a deep copy of the parameter that is passed to the function.

The dictionary consists of all the objects and the `copy()` method which is used as:

```
newdict = olddict.copy()
```

The assignment statement doesn't copy any object but it creates a binding between the target and the object that is used for the mutable items. Copy is required to keep a copy of it using the modules that is provided to give generic and shallow operations.

14. How can the ternary operators be used in python ?

Answer:

The ternary operator is the operator that is used to show the conditional statements. This consists of the true or false values with a statement that has to be evaluated for it. The operator will be given as:

```
[on_true] if [expression] else [on_false]
```

```
x, y = 25, 50
```

```
big = x if x < y else y
```

This is the lowest priority operator that is used in making a decision that is based on the values of true or false. The expression gets evaluated like if x Yes, this is valid. The function

will then return a None object. The end of a function is defined by the block of code being executed (i.e., the indenting) not by any explicit keyword.

15. What is the lambda operator ?

Answer:

The lambda operator is used to create anonymous functions. It is mostly used in cases where one wishes to pass functions as parameters. or assign them to variable names.

16. Explain different ways to trigger / raise exceptions in your python script ?

Answer:

The following are the two possible ways by which you can trigger an exception in your Python script. They are:

raise — it is used to manually raise an exception general-form:

raise exception-name ("message to be conveyed")

Eg: >>> voting_age = 15

>>> if voting_age < 18: raise ValueError("voting age should be atleast 18 and above")

output: ValueError: voting age should be atleast 18 and above 2. assert statement assert

statements are used to tell your program to test that condition attached to assert

keyword, and trigger an exception whenever the condition becomes false. Eg: >>> a = -10

>>> assert a > 0 #to raise an exception whenever a is a negative number output:

AssertionError

Another way of raising an exception can be done by making a programming mistake, but that's not

usually a good way of triggering an exception.

Top 48 Python Django Interview Questions And Answers

17. How Do You Count The Occurrences Of Each Item Present In The List Without Explicitly Mentioning Them ?

Answer:

Unlike sets, lists can have items with same values. In Python, the list has a `<count()>` function which returns the occurrences of a particular item.

Count The Occurrences Of An Individual Item.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']  
print(weekdays.count('mon'))
```

#output: 3

Count The Occurrences Of Each Item In The List.

We'll use the list comprehension along with the `<count()>` method. It'll print the frequency of each of the items.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']  
print([[x,weekdays.count(x)] for x in set(weekdays)])
```

18. How do I make a Python script executable on UNIX ?

Answer:

You need to do two things:

The script file's mode must be executable and the first line must begin with `"#!"` followed by the path of the Python interpreter. The first is done by executing `chmod +x scriptfile` or perhaps `chmod 755 'script' file`.

The second can be done in a number of ways.

The most straightforward way is to write:

```
#!/usr/local/bin/python
```

As the very first line of your file, using the pathname for where the Python interpreter is installed on your platform. If you would like the script to be independent of where the Python interpreter lives, you can use the `"env"` program. Almost all UNIX variants support the following, assuming the python interpreter is in a directory on the users `$PATH`:

```
#!/usr/bin/env python
```

Don't do this for CGI scripts. The `$PATH` variable for CGI scripts is often minimal, so you need to use the actual absolute pathname of the interpreter. Occasionally, a user's environment is so full that the `/usr/bin/env` program fails; or there's no `env` program at all.

In that case, you can try the following hack (due to Alex Rezinsky):

```
#!/bin/sh
"""."""
exec python $0 ${1+"$@"}
```

The minor disadvantage is that this defines the script's `__doc__` string. However, you can fix that by adding:

```
__doc__ = """...Whatever..."""
```

19. Why don't my signal handlers work ?

Answer:

The most common problem is that the signal handler is declared with the wrong argument list. It is called as:

[handler](#) (signum, frame)

So it should be declared with two arguments:

```
def handler(signum, frame):
```

20. How do I test a Python program or component ?

Answer:

[Python](#) comes with two testing frameworks:

The `documentation test module` finds examples in the documentation strings for a module and runs them, comparing the output with the expected output given in the documentation string.

The `unit test module` is a fancier testing framework modeled on Java and Smalltalk testing frameworks.

For testing, it helps to write the program so that it may be easily tested by using good modular design. Your program should have almost all functionality encapsulated in either functions or class methods. And this sometimes has the surprising and delightful effect of making the program run faster because local variable accesses are faster than global accesses.

Furthermore the program should avoid depending on mutating global variables, since this makes testing much more difficult to do.

The “global main logic” of your program may be as simple as:

```
if __name__=="__main__":  
    main_logic()
```

at the bottom of the main module of your program.

Once your program is organized as a tractable collection of functions and class behaviors, you should write test functions that exercise the behaviors.

A test suite can be associated with each module which automates a sequence of tests.

You can make coding much more pleasant by writing your test functions in parallel with the “production code”, since this makes it easy to find bugs and even design flaws earlier.

“Support modules” that are not intended to be the main module of a program may include a self-test of the module.

```
if __name__ == "__main__":  
    self_test()
```

Even programs that interact with complex external interfaces may be tested when the external interfaces are unavailable by using “fake” interfaces implemented in Python.

21. How the string does get converted to a number ?

Answer:

- To convert the string into a number the built-in functions are used like int() constructor.

It is a data type that is used like int('1')==1.

- float() is also used to show the number in the format as float('1')==1.

- The number by default are interpreted as decimal and if it is represented by int('0x1') then it gives an error as ValueError. In this the int(string,base) function takes the parameter to convert string to number in this the process will be like int('0x1',16)==16. If the base parameter is defined as 0 then it is indicated by an octal and 0x indicates it as hexadecimal number.

- There is function eval() that can be used to convert string into number but it is a bit

slower and present many security risks like `__import__('os').system("rm -rf$HOME")` – use of this will delete the home directory of the system.

22. What is the difference between deep and shallow copy ?

Answer:

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. These references point to the original objects and the changes made in any member of the class will also affect the original copy of it. Shallow copy allows faster execution of the program and it depends on the size of the data that is used.

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. The changes made in the original copy won't affect any other copy that uses the object. Deep copy makes execution of the program slower due to making certain copies for each object that is been called.

23. How is Multithreading achieved in Python ?

Answer:

Python has a multi-threading package but if you want to multi-thread to speed your code up.

Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your 'threads' can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.

This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.

All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn't a good idea.

24. How does global value mutation used for thread-safety ?

Answer:

The global interpreter lock is used to allow the running of the thread one at a time. This is internal to the program only and used to distribute the functionality along all the virtual machines that are used. Python allows the switching between the threads to be performed by using the byte code instructions that are used to provide platform-independence. The `sys.setcheckinterval()` method is used that allow the switching to occur during the implementation of the program and the instruction. This provides the understanding in the field of accounting to use the byte code implementation that makes it portable to use. The atomicity can be provided such that the shared variables can be given as built-in data types.

Top 48 Python Django Interview Questions And Answers

25. Write a program to read and write the binary data using python ?

Answer:

The module that is used to write and read the binary data is known as struct. This module allows the functionality and with it many functionalities to be used that consists of the string class. This class contains the binary data that is in the form of numbers that gets converted in python objects for use and vice versa. The program can read or write the binary data is:

```
import struct
```

```
f = open(file-name, "rb")
```

```
# This Open() method allows the file to get opened in binary mode to make it portable for  
# use.
```

```
s = f.read(8)
```

```
x, y, z = struct.unpack(">hhl", s)
```

The '>' is used to show the format string that allows the string to be converted in big-endian data form. For homogenous list of data the array module can be used that will allow the data to be kept more organized fashion.

26. What is the difference between `del()` and `remove()` methods of list ?

Answer:

To remove a list element, you can use either the del statement if you know exactly which element(s) you are deleting or the remove() method if you do not know.

27. What are the key features of Python ?

Answer:

These are the few key features of Python:

Python is an interpreted language. That means that, unlike languages like C and its variants, Python does not need to be compiled before it is run. Other interpreted languages include PHP and Ruby.

Python is dynamically typed, this means that you don't need to state the types of variables when you declare them or anything like that. You can do things like `x=111` and then `x="I'm a string"` without error

Python is well suited to object orientated programming in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++'s public, private), the justification for this point is given as "we are all adults here" In Python, functions are first-class objects. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects

Writing Python code is quick but running it is often slower than compiled languages.

Fortunately, Python allows the inclusion of C based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it's really quite quick because a lot of the number crunching it does isn't actually done by Python

Python finds use in many spheres – web applications, automation, scientific modelling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice.

28. How To Convert A List Into Other Data Types ?

Answer:

Sometimes, we don't use lists as is. Instead, we have to convert them to other types.

Turn A List Into A String.

We can use the `<"".join()>` method which combines all elements into one and returns as a string.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat']
listAsString = ' '.join(weekdays)
print(listAsString)
#output: sun mon tue wed thu fri sat
```

Turn A List Into A Tuple.

Call Python's `<tuple()>` function for converting a list into a tuple. This function takes the list as its argument. But remember, we can't change the list after turning it into a tuple because it becomes immutable.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat']
listAsTuple = tuple(weekdays)
print(listAsTuple)
#output: ('sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat')
```

Turn A List Into A Set.

Converting a list to a set poses two side-effects.

Set doesn't allow duplicate entries, so the conversion will remove any such item if found.

A set is an unordered collection, so the order of list items would also change.

However, we can use the `<set()>` function to convert a list to a set.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat','sun','tue']
listAsSet = set(weekdays)
print(listAsSet)
#output: set(['wed', 'sun', 'thu', 'tue', 'mon', 'fri', 'sat'])
```

Turn A List Into A Dictionary.

In a dictionary, each item represents a key-value pair. So converting a list isn't as straight forward as it were for other data types.

However, we can achieve the conversion by breaking the list into a set of pairs and then call the `<zip()>` function to return them as tuples.

Passing the tuples into the <dict()> function would finally turn them into a dictionary.

```
weekdays = ['sun','mon','tue','wed','thu','fri']  
listAsDict = dict(zip(weekdays[0::2], weekdays[1::2]))  
print(listAsDict)  
#output: {'sun': 'mon', 'thu': 'fri', 'tue': 'wed'}
```

29. What Is NumPy And How Is It Better Than A List In Python ?

Answer:

NumPy is a Python package for scientific computing which can deal with large data sizes. It includes a powerful N-dimensional array object and a set of advanced functions.

Also, the NumPy arrays are superior to the built-in lists. There are a no. of reasons for this.

NumPy arrays are more compact than lists.

Reading and writing items is faster with NumPy.

Using NumPy is more convenient than to the standard list.

NumPy arrays are more efficient as they augment the functionality of lists in Python.

What is the Python programming language? List out key benefits.

Python is an object oriented programming language with modules, threads, exceptions and automatic memory management.

Some of the benefits to using Python are as follows:

High level language.

Modular

Object Oriented

Built-in data structure

Open source

30. How is compile-time and run-time code checking done in Python ?

Answer:

Python has a unique way of performing compile-time and run-time code checking. A small portion checking is carried out during compile-time checking, but most of the checks such as type, name, etc are postponed until code execution. If the Python code references a user-defined function that does not exist, the code will compile successfully and the code will fail with an exception only when the code execution path references the function which does not exist.

31. What is the usage of help() and dir() function in Python ?

Answer:

Help() and dir() both functions are accessible from the Python interpreter and used for viewing a consolidated dump of built-in functions.

Help() function: The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, etc.

Dir() function: The dir() function is used to display the defined symbols.

32. What is the difference between range & xrange ?

Answer:

For the most part, xrange and range are the exact same in terms of functionality. They both provide a way to generate a list of integers for you to use, however you please. The only difference is that range returns a Python list object and xrange returns an xrange object.

This means that xrange doesn't actually generate a static list at run-time like range does. It creates the values as you need them with a special technique called yielding. This technique is used with a type of object known as generators. That means that if you have a really gigantic range you'd like to generate a list for, say one billion, xrange is the function to use.

This is especially true if you have a really memory sensitive system such as a cell phone that you are working with, as range will use as much memory as it can to create your array

of integers, which can result in a Memory Error and crash your program. It's a memory hungry beast.

Top 48 Python Django Interview Questions And Answers

33. Is python a case sensitive language ?

Answer:

Yes! Python is a case sensitive programming language.

34. What Are The Data Types Supports in Python Language ?

Answer:

1. Python Int
2. Python Float
3. Python Complex
4. Python Boolean
5. Python Dictionary
6. Python List
7. Python Tuple
8. Python Strings
9. Python Set

Every data type in python language is internally implemented as a class

Python language data types are categorized into two types.

They are::

1. Fundamental Types
2. Collection Types

35. How do you check the file existence and their types in Python ?

Answer:

`os.path.exists()` – use this method to check for the existence of a file. It returns True if the

file exists, false otherwise. Eg: `import os; os.path.exists('/etc/hosts')`

`os.path.isfile()` – this method is used to check whether the give path references a file or not. It returns True if the path references to a file, else it returns false. Eg: `import os; os.path.isfile('/etc/hosts')`

`os.path.isdir()` – this method is used to check whether the give path references a directory or not. It returns True if the path references to a directory, else it returns false. Eg: `import os; os.path.isfile('/etc/hosts')`

`os.path.getsize()` – returns the size of the given file

`os.path.getmtime()` – returns the timestamp of the given path.

36. How is “self” explicitly defined in a method ?

Answer:

“Self” is a reference variable and an instance attribute that is used instead of the local variable inside the class. The function or the variable of the self like `self.x` or `self.meth()` can be used in case the class is not known. There are no variables declared as local. It doesn't have any syntax and it allow the reference to be passed explicitly or call the method for the class that is in use. The use of `writebaseclass.methodname(self,)` shows that the method of `_init_()` can be extended to the base class methods. This also solves the problem that is syntactic by using the assignment and the local variables. This tells a way to the interpreter the values that are to be used for the instance variables and local variables. The use of explicit `self.var` solves the problem mentioned above.

37. What is the use of `join()` for a string rather than list or tuple method ?

Answer:

The functions and the methods that are used for the functionality uses the string module.

This string module is represented as by using the `join` function in it:

`“, “.join(['1', '2', '4', '8', '16'])` that results in `“1, 2, 4, 8, 16”`

The string variable that is used provide a fixed string literal to allow the names that are used to be bounded to the strings. `join()` is a string method that is used to provide a separator string to use the function over the sequence of the string and insert the function

to an adjacent elements. The method uses any number of arguments that follow some rules that has to be put up for the sequence objects that the class defines for itself. The join is used for the string module that is used to join the string characters together as it is given in the program. The example is given as:

```
string.join(['1', '2', '4', '8', '16'], ", ")
```

38. How is memory managed in Python ?

Answer:

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.

39. Explain what Flask is and its benefits ?

Answer:

Flask is a web micro framework for Python based on “Werkzeug, Jinja2 and good intentions” BSD license. Werkzeug and Jinja2 are two of its dependencies. This means it will have little to no dependencies on external libraries. It makes the framework light while there is little dependency to update and less security bugs.

A session basically allows you to remember information from one request to another. In a flask, a session uses a signed cookie so the user can look at the session contents and modify. The user can modify the session if only it has the secret key Flask.secret_key.

40. How will you check in a string that all characters are in uppercase ?

Answer:

`isupper()` – Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.

Top 48 Python Django Interview Questions And Answers

41. How will you merge elements in a sequence ?

Answer:

`join(seq)` – Merges (concatenates) the string representations of elements in sequence `seq` into a string, with separator string.

42. How will you check in a string that all characters are in lowercase ?

Answer:

Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.

43. How do you create a Python function ?

Answer:

Functions are defined using the `def` statement. An example might be `def foo(bar):`

44. How are modules used in a Python program ?

Answer:

Modules are brought in via the import statement.

45. What is the Python interpreter prompt ?

Answer:

Three greater-than signs: >>> Also, when the interpreter is waiting for more input the prompt changes to three periods.

46. How are instance variables different from class variables ?

Answer:

Instance variables are variables that are created locally within a class to refer to an object of the class. A class variable is one that is created globally in a class and is accessible within all instance of that class.

Class variables are declared with keyword static and Instance variables are declared without static keyword.

Class variables can be accessed anywhere within that class whereas an instance variable can only be accessed within the particular object of the class.

As class variables are common to all objects of a class, changes made to these variables through one object will reflect in another. As each object will have its own copy of instance variables, changes made to these variables through one object will not reflect in another object.

Class variables can be accessed using either class name or object reference. Instance variables can be accessed only through object reference.

47. What is the structure of a while loop ?

Answer:

while : ... The ellipsis represents a code block to be executed. until the condition becomes false. The condition is an expression that is considered true unless it evaluates to 0, null or false.

48. Under what circumstances would you use a while statement rather than for ?

Answer:

The while statement is used for simple repetitive looping and the for statement is used when one wishes to iterate through a list of items, such as database records, characters in a string, etc.