By: Rajashekar Vuppala

31 JUL 2018

Top 50 Python Interview Questions with Answers for Freshers

Introduction

I have been interviewing people on Python for the last 4 years. I have written this blog covering all the basic building blocks of Python. These 50 questions cover all important topics at different levels, get the best from this blog and ace your interview.

Good Luck with your preparation! ☺

Python Basic Level Interview Questions and Answers

91
Shares

1) A = 10, 20, 30

In the above assignment operation, what is the data type of ‘A’ that Python appreciates as?

Unlike other languages, Python appreciates ‘A’ as a tuple. When you print ‘A’, the output is (10,20,30). This type of assignment is called “**Tuple Packing**”.

2) >>> A = 1,2,3,4

>>> a,b,c,d = A

In the above assignment operations, what is the value assigned to the variable ‘d’?

4 is the value assigned to d. This type of assignment is called ‘**Tuple Unpacking**’.

3) >>> a = 10

>>> b = 20

Swap these two Variables without using the third temporary variable?

a, b = b, a

This kind of assignment is called a parallel assignment.

4) What is a Variable in Python?

When we say Name = ‘john’ in Python, the name is not storing the value ‘john’. But, ‘Name’ acts like a tag to refer to the object ‘john’. The object has types in Python but variables do not, all variables are just tags. All identifiers are variables in Python. Variables never store any data in Python.

5) >>> a = 10

```
>>> b = a
>>> a = 20
>>> print b
```

What is the output?

In Python, a variable always points to an object and not the variable. So here ‘b’ points to the object 10, so the output is 10.

6) How do you find the type and identification number of an object in Python?

type(<variableName>) gives the type of the object that variable is pointing to, and

id(<variablename>) give the unique identification number of the object that

7) >>> a = 0101

>>> b = 2

>>> c = a+b

What is the Value of c?

In Python, any number with leading 0 is interpreted as an octal number. So, the variable c will be pointing to the value 67.

8) How do you represent binary and hexadecimal numbers?

If ‘0b’ is leading then the Python interprets it as a binary number.

‘0x’ as hexadecimal.

9) What are the Arithmetic Operators that Python supports?

‘+’ : Addition

‘-’ : Subtraction

‘*’ : Multiplication

‘/’ : Division

‘%’ : Modulo division

‘**’ : Power Of

‘//’ : floor div

hand Python supports “Augmented Assignment Operators”; i.e.,

A += 10   Means A = A+10

B -= 10   Means B = B - 10

Other Operators supported by Python are & , |, ~ , ^ which are and, or, bitwise compliment and bitwise xor operations respectively.

10) What are the basic Data Types Supported by Python?

Numeric Data types: int, long, float, NoneType

String: str

Boolean: (True, False)

NoneType: None

11) How do you check whether the two variables are pointing to the same object in Python?

In Python, we have an operation called ‘is’ operator, which returns true if the two variables are pointing to the same object.

Example:

```
>>> a = "Hello world"
>>> c = a
>>> a is c
True
>>> id(a)
```

46117472

12) What is for-else and while-else in Python?

Python provides an interesting way of handling loops by providing a function to write else block in case the loop is not satisfying the condition.

Example :

```
    a = []

    for i in a:

        print "in for loop"

   else:

         print "in else block"
```

output:

in else block

The same is true with while-else too.

13) How do you programmatically know the version of Python you are using?

The version property under sys module will give the version of Python that we are using.

```
>>> import sys

>>> sys.version

'2.7.12 (v2.7.12:d33e0cf91556, June 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)]'

>>>
```

14) How do you find the number of references pointing to a particular object?

The getrefcount() function in the sys module gives the number of references pointing to a particular object including its own reference.

```
  >>> a = "JohnShekar"

>>> b = a

>>> sys.getrefcount(a)

3
```

Here, the object ‘JohnShekar’ is referred by a, b and getrefcount() function itself. So the output is 3.

15) How do you dispose a variable in Python?

‘del’ is the keyword statement used in Python to delete a reference variable to an object.

```
  >>> a = 'Hello world!!'

>>> b = a

>>> sys.getrefcount(a)

3

>>> del a

>>> sys.getrefcount(a)

Traceback (most recent call last):

    File "<pyshell#23>", line 1, in <module>

      sys.getrefcount(a)
```

2

You may also like: ***Top 45 HTML5 Basic Interview Questions and Answers You Must Prepare in 2018***

16) Write a program to check whether a given number is a prime number?

Synopsys:

```
----------------------

Python checkprime.py

Enter Number:23

23 is a prime number

Python checkprime.py

Enter Number:33

33 is a not prime number
```

Solution:

```
A = input(“Enter Number:”)

for j in xrange(2, int(A**0.5)+1):

    if a%j == 0:

        print A, ‘Not a prime number’

        break

else:

    print A, ‘This is a prime number’
```

17) What is the difference between range() and xrange() functions in Python?

range() is a function that returns a list of numbers, which will be an overhead if the number is too large. Whereas, xrange() is a generator function that returns an iterator which returns a single generated value whenever it is called.

18) What are the generator functions in Python?

Any function that contains at least one yield statement is called a generator function instead of a return statement. The difference between return and yield is, return statement terminates the function, and yield statement saving all its states pauses and later continues from there on successive calls.

19) Write a generator function to generate a Fibonacci series?

Solution:

```
def fibo(Num):
    a, b = 0, 1
        for I in xrange(Num):
        yield a,
        a, b = b, a+b
    Num = input(‘Enter a number:’)
    for i in fibo(Num):
        print i
```

20) What are the ideal naming conventions in Python?

Examples: is_prime(), test_var = 10 etc

Constants are all either uppercase or camel case.

Example: MAX_VAL = 50, PI = 3.14

None, True, False are predefined constants follow camel case, etc.

Class names are also treated as constants and follow camel case.

Example: UserNames

21) What happens in the background when you run a Python file?

When we run a .py file, it undergoes two phases. In the first phase it checks the syntax and in the second phase it compiles to bytecode (.pyc file is generated) using Python virtual machine, loads the bytecode into memory and runs.

22) What is a module in Python?

A module is a .py file in Python in which variables, functions, and classes can be defined. It can also have a runnable code.

Related reading: ***8 Up-and-Coming Programming Languages Developers Should Know***

23) How do you include a module in your Python file?

The keyword “import” is used to import a module into the current file.

Example: import sys #here sys is a predefined Python module.

24) How do you reload a Python module?

argument and reloads the module.

25) What is List in Python?

The List is one of the built-in data structures in Python. Lists are used to store an ordered collection of items, which can be of different type.

Elements in a list are separated by a comma and enclosed in square brackets.

Examples of List are:

A = [1,2,3,4]

B = [‘a’,’b’,’c’]

C = [1,’a’,’2’,’b’]

List in Python is sequence type as it stores ordered collection of objects/items. In Python String and tuple are also sequence types.

91
Shares

26) When do you choose a list over a tuple?

When there is an immutable ordered list of elements we choose tuple. Because we cannot add/remove an element from the tuple. On the other hand, we can add elements to a list using append () or extend() or insert(), etc., and delete elements from a list using remove() or pop().

Simple tuples are immutable, and lists are not. Based on these properties one can decide what to choose in their programming context.

27) How do you get the last value in a list or a tuple?

When we pass -1 to the index operator of the list or tuple, it returns the last value. If -2 is passed, it returns the last but one value.

Example:

```
>>> a = [1,2,3,4]

>>> a[-1]

4

>>> a[-2]

3

>>>

>>> b = (1,2,3,4)

>>> b[-1]

4

>>> b[-2]

3
```

28) What is Index Out Of Range Error?

When the value passed to the index operator is greater than the actual size of the tuple or list, Index Out of Range error is thrown by Python.

```
>>> a = [1,2,3,4]

>>> a[3]

4

>>> a[4]

Traceback (most recent call last):

  File "<pyshell#20>", line 1, in <module>

    a[4]

IndexError: list index out of range

>>> b = (1,2,3,4)

>>> b[4]

Traceback (most recent call last):

  File "<pyshell#22>", line 1, in <module>

    b[4]

IndexError: tuple index out of range

>>>
```

29) What is slice notation in Python to access elements in an iterator?

In Python, to access more than one element from a list or a tuple we can use ‘:’ operator. Here is the syntax. Say ‘a’ is list

Example:

```
   >>> a = [1,2,3,4,5,6,7,8]
>>> a[3:]     # Prints the values from index 3 till the end
[4, 5, 6, 7, 8]
>>> a[3:6]   #Prints the values from index 3 to index 6.
[4, 5, 6]
>>> a[2::2]   #Prints the values from index 2 till the end of the list with step count 2.
[3, 5, 7]
>>>
```

The above operations are valid for a tuple too.

30) How do you convert a list of integers to a comma separated string?

List elements can be turned into a string using join function.

Example:

```
   >>> a = [1,2,3,4,5,6,7,8]
>>> numbers = ','.join(str(i) for i in a)
>>> print numbers
1,2,3,4,5,6,7,8
>>>
```

You might enjoy reading: ***Top 55 Java Interview Questions and Answers for***

31) What is the difference between Python append () and extend () functions?

The extend() function takes an iterable (list or tuple or set) and adds each element of the iterable to the list. Whereas append takes a value and adds to the list as a single object.

Example:

```
 >>> b = [6,7,8]
>>> a = [1,2,3,4,5]
>>> b = [6,7,8]
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6, 7, 8]
>>> c = ['a','b']
>>> a.append(c)
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, ['a', 'b']]
>>>
```

32) What is List Comprehension? Give an Example.

List comprehension is a way of elegantly constructing a list. It’s a simplified way of constructing lists and yet powerful.

Example:

```
>>> a = [x*2 for x in xrange(10)]
```

In the above example, a list is created by multiplying every value of x by 2. Here is another example-

```
    >>> [x**2 for x in xrange(10) if x%2==0]
[0, 4, 16, 36, 64]
>>>
```

Here the numbers ranging from 0-9 and divisible by 2 are raised by power 2 and constructed into a list.

33) Tell me about a few string operations in Python?

Here are the most commonly used text processing methods.

```
>>> a = 'hello world'
>>> a.upper()    #Converts to Uppercase
'HELLO WORLD'
>>> a.lower()    #Converts to Lowercase
'hello world'
>>> a.capitalize()    #First Letter is capitalized.
'Hello world'
>>> a.title()    #First character of the every word is capitalized.
'Hello World'
>>> a.split()    #Splits based on space and returns a list. Split takes an optional parameter to split. By #default it considers space based on which it splits
>>> record.split(':')

['name', 'age', 'id', 'salary']

>>>

>>> a = '\n\t    hello world \t\n'

>>> print a

        hello world

>>> a.strip()    #strips leading and trailing white spaces and newlines.

'hello world'

>>> a.lstrip()    #left strip

'hello world \t\n'

>>> a.rstrip()    #right strip

'\n\t\thello world'

>>>

>>> a = 'hello'

>>> a.isalpha()    #returns true only if the string contains alphabets.

True

>>> a = 'hello world'    #As there is a space, it returned false.

>>> a.isalpha()

False

>>> a = '1234'
>>> a = '1234a'

>>> a.isdigit()    #returned false as the string contains a alphabet

False

>>> a.isalnum()       # returns true if the string contains alphabets and digits.

True

>>>

>>> a = 'Name: {name}, Age: {age}'

>>> a.format(name='john', age='18')

'Name: john, Age: 18'

>>>
```

34) How do you create a list which is a reverse version on another list in Python?

Python provides a function called reversed(), which will return a reversed iterator. Then, one can use a list constructor over it to get a list.

Example:

```
>>> a.format(name='john', age='18')

'Name: john, age: 18'

>>> a =[10,20,30,40,50]

>>> b = list(reversed(a))

>>> b

[50, 40, 30, 20, 10]
```

```
[10, 20, 30, 40, 50]
```

35) What is a dictionary in Python?

In Python, dictionaries are kind of hash or maps in another language. Dictionary consists of a key and a value. Keys are unique, and values are accessed using keys. Here are a few examples of creating and accessing dictionaries.

Examples:

```
>>> a = dict()

>>> a['key1'] = 2

>>> a['key2'] = 3

>>> a

{'key2': 3, 'key1': 2}

>>> a.keys()       # keys() returns a list of keys in the dictionary

['key2', 'key1']

>>> a.values()       # values() returns a list of values in the dictionary

[3, 2]

>>> for i in a:       # Shows one way to iterate over a dictionary items.

    print i, a[i]

key2 3

key1 2

>>> print 'key1' in a    # Checking if a key exists

True
```

```
{'key2': 3}
```

36) How do you merge one dictionary with the other?

Python provides an update() method which can be used to merge one dictionary on another.

Example:

```
>>> a = {'a':1}

>>> b = {'b':2}

>>> a.update(b)

>>> a

{'a': 1, 'b': 2}
```

37) How to walk through a list in a sorted order without sorting the actual list?

In Python we have function called sorted(), which returns a sorted list without modifying the original list. Here is the code:

```
>>> a

[3, 6, 2, 1, 0, 8, 7, 4, 5, 9]

>>> for i in sorted(a):

   print i,

0 1 2 3 4 5 6 7 8 9

>>> a
```

```
>>>
```

38) names = [‘john’, ‘fan’, ‘sam’, ‘megha’, ‘popoye’, ’tom’, ‘jane’, ‘james’,’tony’]

Write one line of code to get a list of names that start with character ‘j’?

Solution:

```
>>> names = ['john', 'fan', 'sam', 'megha', 'popoye', 'tom', 'jane', 'james', 'tony']

>>> jnames=[name for name in names if name[0] == 'j']     #One line code to filter names that start with ‘j’

>>> jnames

['john', 'jane', 'james']

>>>
```

39) a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Write a generator expression to get the numbers that are divisible by 2?

Solution:

```
>>> a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

>>> b = (i for i in a if i%2 == 0)    #Generator expression.

>>> for i in b: print i,

0 2 4 6 8
```

40) What is a set?

A Set is an unordered collection of unique objects.

41) a = “this is a sample string with many characters”

Write a Python code to find how many different characters are present in this string?

Solution:

```
>>> a = "this is a sample string with many characters"

>>> len(set(a))

16
```

42) a = “I am Singh and I live in singh malay and I like Singh is King movie and here I am”

Write a program that prints the output as follows:

I :4

am : 2

Singh : 3

and so on.. i.e <word> : <number of its occurrence> in the string ‘a’ .

```
>>> for item in set(a.split()):
    k[item] = a.split().count(item)
>>> for item in k: print item, k[item]
and 3
king 1
like 1
I 4
movie 1
is 1
am 2
malay 1
here 1
live 1
in 1
singh 3
```

43) What is *args and **kwargs?

*args is used when the programmer is not sure about how many arguments are going to be passed to a function, or if the programmer is expecting a list or a tuple as argument to the function.

**kwargs is used when a dictionary (keyword arguments) is expected as an argument to the function.

44) >>> def welcome(name='guest',city): print 'Hello', name, 'Welcome to', city

What happens with the following function definition?

Here the issue is with function definition, it is a syntax error and the code will not run. The default argument is following the non-default argument, which is not

91
Shares

is the right way of defining:

def welcome(city, name='guest'): print 'Hello', name, 'Welcome to', city

The order of passing values to a function is, first one has to pass non-default arguments, default arguments, variable arguments, and keyword arguments.

45) Name some standard Python errors you know?

TypeError: Occurs when the expected type doesn’t match with the given type of a variable.

ValueError: When an expected value is not given- if you are expecting 4 elements in a list and you gave 2.

NameError: When trying to access a variable or a function that is not defined.

IOError: When trying to access a file that does not exist.

IndexError: Accessing an invalid index of a sequence will throw an IndexError.

KeyError: When an invalid key is used to access a value in the dictionary.

We can use **dir(__builtin__)** will list all the errors in Python.

46) How Python supports encapsulation with respect to functions?

Python supports inner functions. A function defined inside a function is called an inner function, whose behavior is not hidden. This is how Python supports encapsulation with respect to functions.

47) What is a decorator?

In simple terms, decorators are wrapper functions that takes a callable as an argument and extends its behavior and returns a callable. Decorators are one kind

48) What is PEP8?

PEP 8 is a coding convention about how to write Python code for more readability.

Read more: ***Top 30 Interview Questions and Answers on Angular 5***

49) How do you open an already existing file and add content to it?

In Python, open(<filename>,<mode>) is used to open a file in different modes. The open function returns a handle to the file, using which one can perform read, write and modify operations.

Example:

```
F = open(“simplefile.txt”,”a+”) #Opens the file in append mode

F.write(“some content”)    #Appends content to the file.

F.close()    # closes the file.
```

50) What mode is used for both writing and reading in binary format in file.?

“wb+” is used to open a binary file in both read and write format. It overwrites if the file exists. If the file does not exist it creates a new file for reading and writing.

Conclusion

So, these are the important questions that will help you in your interview preparation. Got any other queries, or you want any question to be answered? Do comment here and will be happy to answer!

experienced professionals, that will include in-depth Python Scripting Interview Questions, data structure questions, etc. Come back soon!

Get Started | Kickstart your Python Basics Training here!

Comments **Community** 1 **Login**

♡ **Recommend** Tweet Share **Sort by Best**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

ALSO ON **OPENCAMPUS**

Top 55 Java Interview Questions and Answers

2 comments • a year ago

Avatar **sakshi singh** — Thank you for sharing,this blog are very easy to

Top Quality Management Tips to

1 comment • 10 months ago

Avatar **Jack L Mash** — We are passing through a new phase. A phase of

91
Shares

Avatar **Jayla Juliette** — My all the efforts were well

Avatar **malaysiaexcelr078** — nice article . thank you

POPULAR POSTS

RACI Matrix: How does it help Project Managers?

FEBRUARY 26, 2019

RACI Matrix: How does it help Project Managers?

Top 10 reasons you should know to get an AWS Certification

FEBRUARY 19, 2019

Why AWS? The Top 10 reasons to get an AWS Certification

RAID Log: How is it useful for a Project Manager?

NOVEMBER 19, 2018

RAID Log: How is it useful for a Project Manager?

IT Service Manager: The Hero of 2020

OCTOBER 29, 2018

IT Service Manager: The Hero of 2020

An Ultimate Guide to Control Charts in Six Sigma

OCTOBER 26, 2018

91
Shares

Top 10 Popular Linux Distros for Penetration Testing and Ethical Hacking

OCTOBER 23, 2018

Top 10 Popular Linux Distros for Penetration Testing and Ethical Hacking

Popular Courses

- Project Management Professional® Training
- Lean Six Sigma Green Belt
- Lean Six Sigma Black Belt
- PMI-ACP
- CSM
- ITIL-Foundation
- PRINCE2 Foundation & Practitioner

Join us!

- Become our trainer
- Become our affiliate
- Create a course
- Careers

Contact & Support

91
Shares

☎ USA +1 855 542 2678

☎ USA +1 347 994 9941

☎ IND +91 741 666 4433

☎ HKG +852 5808 4350

Support

✉ support@greycampus.com

Company

About us

Customer Speak

Contact us

For Enterprise

Sitemap

Privacy policy

Cookie policy

Terms of use

FAQ

Cancellation / Rescheduling policy

Disclaimer

"PMI®", "PMBOK®", "PMP®" and "PMI-ACP®" are registered marks of the Project Management Institute, Inc.

The Swirl logo™ is a trade mark of AXELOS Limited.

ITIL is a registered trade mark of AXELOS Limited. Used under licence of AXELOS Limited. All rights reserved.

PRINCE2 is a registered trade mark of AXELOS Limited. Used under licence of AXELOS Limited. All rights reserved.

IASSC® is a registered trade mark of International Association for Six Sigma Certification

91
Shares