# Python

Python Interview Questions and Answers


# Python Interview Questions and Answers

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?

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 variable is pointing to.

Ex:

```
print(type(b)) #<class 'int'>
print(id(b)) #1452987584
```



7) a = 0101

b = 2

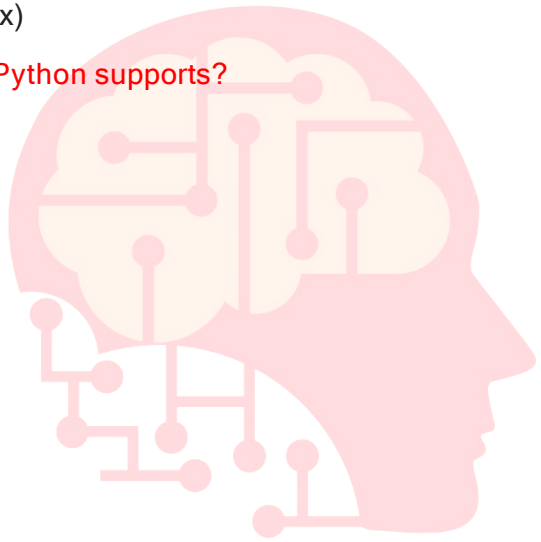
c = a+b

What is the Value of c?

In Python2, any number with leading 0 is interpreted as an octal number. So, variable a points to 65(Equalent in Decimal) then the variable c will be pointing to the value 67 i.e 65+2.

In Python3, a=0101 (Doesn't support syntax)

8) What are the Arithmetic Operators that Python supports?



| Operator | Description | Example |
|---|---|---|
| + Addition | Adds values on either side of the operator. | a + b = 30 |
| - Subtraction | Subtracts right hand operand from left hand operand. | a – b = -10 |
| * Multiplication | Multiplies values on either side of the operator | a * b = 200 |
| / Division | Divides left hand operand by right hand operand | b / a = 2 |
| % Modulus | Divides left hand operand by right hand operand and returns remainder | b % a = 0 |
| ** Exponent | Performs exponential (power) calculation on operators | a**b =10 to the power 20 |
| // | Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) – | print(9/4) *#2.25*<br>print(9//4) *#2*<br><br>print(-9/4) *#-2.25*<br>print(-9//4) *#-3* |

## 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


print(a is c) #Returns true if the two variables are pointing to the same object
print(id(a)) #64450416
print(id(c)) #64450416
```



## 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 = [10,20,30]
for i in a:
    print("in for loop")
else:
    print("in else block")

#Output
in for loop
in for loop
in for loop
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
print(sys.version)
```

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.

```
import sys
x = "JohnShekar"
y = x
print(sys.getrefcount(x))
```

Here, the object 'JohnShekar' is referred by x, y 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.

```
import sys
x = "JohnShekar"
y = x
print(sys.getrefcount(x))

del x
print(sys.getrefcount(x)) #NameError: name 'x' is not defined
```

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

range() and xrange() are two functions that could be used to iterate a certain number of times in for loops in Python.

In Python 3, there is no xrange , but the range function behaves like xrange in Python 2.

If you want to write code that will run on both Python 2 and Python 3, you should use range().

Example:

```
# initializing a with range()
a = range(1, 10000)


# initializing a with xrange()
x = xrange(1, 10000)


print("The return type of range() is : ")
print(type(a))


# testing the type of x
print("The return type of xrange() is : ")
print(type(x))
```



17) What are the ideal naming conventions in Python?

All variables and functions follow lowercase and underscore naming convention.

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

### 18) 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.

### 19) 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.

### 20) 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.



### 21) How do you reload a Python module?

There is a function called reload() in Python, which takes module name as an argument and reloads the module.

### 22) 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.

## 23) 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.



## 24) 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] #List
print(a[-1])#4
```

```
print(a[-2])#3


b = (1,2,3,4) #Tuple
print(b[-1])#4
print(b[-2])#3
```

## 25) 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] #List
print(a[5])#IndexError: list index out of range
```

## 26) 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

```
a[startindex:EndIndex:Step]
```

Example:

```
a = [100,200,300,400,500,600,700,800]


print(a[3:]) # Prints the values from index 3 till the end [400, 500, 600, 700, 800]
```

```
print(a[3:6])#Prints the values from index 3 to index 6. [400, 500, 600]
print(a[2::2])#Prints the values from index 2 till the end of the list with step count 2. [300, 500, 700]
```

The above operations are valid for a tuple too.

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

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

```
a = [1,2,3,4,5,6,7,8]
print(a)

numbers = ','.join(str(i) for i in a)
print(numbers)
```

28) 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:

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

```
print(a)#[1, 2, 3, 4, 5, 6, 7, 8]


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

**********

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

Here are the most commonly used text processing methods.

```
#Creating strings
name = "John" #  a string
mychar = 'S' # a character
print(name)
print(mychar)
```



```
#you can also use the following syntax to create strings.
name1 = str() # this will create empty string object
name2 = str("newstring") # string object containing 'newstring'
print(name1)
print(name2)
```

```
#========Strings are immutable=========================
str1="welcome"
str2="welcome"

print(id(str1),id(str2))  #57660416 ,57660416

str2=str2+"to python"
print(id(str1),id(str2))  #57660416 ,59955200(changed means immutable)

#==========  + and * with string=======================
str="welcome"
print(str+" to Python programming") # welcome to Python programming
print(str *3) #welcomewelcomewelcome
#============Slicing=============
str="welcome"
print(str[1:3]) #el
print(str[:6])#welcom
print(str[4:])#ome
```

```
print(str[1:-1]) #elcom #elimate 1 char from end
print(str[1:-2]) #elco  #eleminate 2 chars from end


#============String Functions in Python==========
print(len("hello")) #5
print(max("abc")) #c
print(min("abc")) #a


#==================in  and not in  operators=====
s1  =  "Welcome"
print("come" in s1)# True
print("come" not in s1) #False


#==============Strings comparison===============
print("tim" == "tie") #False
print("free" != "freedom") #True
print ("arrow" > "aron") #True
print ("right" >= "left") #True
print ("teeth" < "tee") #False
```

```
print ("yellow" <= "fellow") #False
print ("abc" > "") #True


#==============Testing strings==================
s = "welcome to python"
print(s.isalnum()) #False
print("Welcome".isalpha()) #True
print("2012".isdigit()) #True
print("first Number".isidentifier())#False
print(s.islower()) #True
print("WELCOME".isupper()) #True
print(" ".isspace()) #True


#===========Searching for Substrings================
s = "welcome to python"
print(s.endswith("thon")) #True
print(s.startswith("good")) #False
print(s.find("come")) #3
print(s.find("become")) #-1
```

```
print(s.count("o")) #3


#==============Converting Strings===============
s = "String in PYTHON"
s1 =  s.capitalize()
print(s1) #String in python

s2 = s.title()
print(s2)#String In Python


s3 = s.lower()
print(s3) #string in python


s4 = s.upper()
print(s4) #STRING IN PYTHON

s5 = s.swapcase()
print(s5) #sTRING IN python
```

```
s6 = s.replace("in", "on")
print(s6) #String on PYTHON


print(s) #String in PYTHON
```

30) 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 =[10,20,30,40,50]
print(a)
b = list(reversed(a))#[10, 20, 30, 40, 50]
print(b) #[50, 40, 30, 20, 10]
```



31) 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:

```
########Retrieving, modifying and adding elements in the dictionary###############
friends = {'tom' : '111-222-333','jerry' : '666-33-111'}
print(friends) #{'tom': '111-222-333', 'jerry': '666-33-111'}

#Retrieving elements from the dictionary
print(friends['tom']) # 111-222-333

#Adding elements into the dictionary
friends['bob'] = '888-999-666'
print(friends) #{'tom': '111-222-333', 'jerry': '666-33-111', 'bob': '888-999-666'}

#Modify elements into the dictionary
friends['bob'] = '888-999-777'
print(friends) #{'tom': '111-222-333', 'jerry': '666-33-111', 'bob': '888-999-777'}

#Delete element from the dictionary
del friends['bob']
print(friends) #{'tom': '111-222-333', 'jerry': '666-33-111'}
```

32) 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)
print(a) #{'a': 1, 'b': 2}
```

33) 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=[500,300,400,200,100]
print(a)#[500, 300, 400, 200, 100]
print(sorted(a)) #[100, 200, 300, 400, 500]
```



34) 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'
print(jnames)
```

35) What is a set?

A Set is an unordered collection of unique objects.

36) 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"
print(len(set(a))) #16
```

37) 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.

38) 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.

39) 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.
```

39) What are the built-in type does python provides?

There are mutable and Immutable types of Pythons built in types.

**Mutable built-in types**

List

Sets

Dictionaries

**Immutable built-in types**

Strings

Tuples

Numbers

40) What is module and package in Python?

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

The folder of Python program is a package of modules. A package can have modules or subfolders.

41) Explain how can you generate random numbers in Python?

To generate random numbers in Python, you need to import command as

```
import random
print(random.random())
```

This returns a random floating point number in the range [0,1)

42)How to connect to the Oracle Database using python script?

Using cx_Oracle module.

Example:

```
import os
import cx_Oracle


# Set folder in which Instant Client is installed in system path
os.environ['PATH'] = 'E:\\app\\OracleHomeUser1\\instantclient_18_3'
```

```
# Connect to hr account in Oracle Database 11g Express Edition
con = cx_Oracle.connect("hr", "hr", "localhost:1521/pdborcl")
cur = con.cursor()

query="select * from employees"

cur.execute(query)

for cols in cur:
    print(cols[0],"  ",cols[1],"  ",cols[2])

print("Completed!!!")

cur.close()
con.close()
```



43) How to connect to the Microsoft Excel and read write data in to excel using python script?

**Reading data from Excel:**

```python
# import openpyxl module
import openpyxl

# Give the location of the file
path = "C:\SeleniumPractice\data3.xlsx"

workbook = openpyxl.load_workbook(path)
sheet = workbook["Sheet1"]

rows=sheet.max_row
cols=sheet.max_column

print(rows) # print the total number of rows
print(cols) # ptint total number of column

for r in range(1,rows+1):
    for c in range(1,cols+1):
        print(sheet.cell(row=r, column=c).value,end='      ')
    print()
```

**Writing data into Excel:**

```
# import openpyxl module
import openpyxl


# Give the location of the file
path = "C:\SeleniumPractice\Test2.xlsx"

workbook = openpyxl.load_workbook(path)
sheet= workbook.active


for r in range(1,5):
    for c in range(1,3):
        sheet.cell(row=r, column=c).value = "abcdef" #(or)sheet.cell(row=r, column=c, value='xyz')


workbook.save(path)
```



44) What is the difference between list and tuples?

| *LIST vs TUPLES* | |
|---|---|
| **LIST** | **TUPLES** |
| Lists are mutable i.e they can be edited. | Tuples are immutable (tuples are lists which can't be edited). |
| Lists are slower than tuples. | Tuples are faster than list. |
| Syntax: list_1 = [10, 'Chelsea', 20] | Syntax: tup_1 = (10, 'Chelsea' , 20) |

45) Explain Inheritance in Python with an example.

Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

They are different types of inheritance supported by Python:

1. Single Inheritance - where a derived class acquires the members of a single super class.
2. Multi-level inheritance - a derived class d1 in inherited from base class base1, and d2 are inherited from base2.
3. Hierarchical inheritance - from one base class you can inherit any number of child classes
4. Multiple inheritance - a derived class is inherited from more than one base class. 46).

How can you randomize the items of a list in place in Python?

Consider the example shown below:

```
from random import shuffle
x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
shuffle(x)
print(x)
```

47). Write a sorting algorithm for a numerical dataset in Python.

The following code can be used to sort a list in Python:

```
list = ["1", "4", "0", "6", "9"]
list = [int(i) for i in list]
list.sort()
print (list)
```

48) How to print current date & time?

Time module is available.

Example:

```
import time;

localtime = time.asctime( time.localtime(time.time()) )
print ("Local current time :", localtime)
```



********** Thank You **********