Python Assignment Questions

Basic Python Interview Questions

Q1. What is the difference between list and tuples in Python?

LIST vs 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)

Q2. What are the 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).

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 modeling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice. Learn more about Big Data and its applications from the Data Engineering Training.

Q3.What are Dict and List comprehensions?

Ans: Dictionary and list comprehensions are just another concise way to define dictionaries and lists.

Example of list comprehension is-

x=[i for i in range(5)]

The above code creates a list as below-

4

[0,1,2,3,4]

Example of dictionary comprehension is-

x=[i : i+2 for i in range(5)]

The above code creates a list as below-

[0: 2, 1: 3, 2: 4, 3: 5, 4: 6]

Q4.What are the common built-in data types in Python?

Ans: The common built-in data types in python are-

Numbers– They include integers, floating-point numbers, and complex numbers. eg. 1, 7.9,3+4i

List– An ordered sequence of items is called a list. The elements of a list may belong to different data types. Eg. [5,'market',2.4]

Tuple– It is also an ordered sequence of elements. Unlike lists , tuples are immutable, which means they can't be changed. Eg. (3,'tool',1)

String– A sequence of characters is called a string. They are declared within single or double-quotes. Eg. "Sana", 'She is going to the market', etc.

Set– Sets are a collection of unique items that are not in order. Eg. {7,6,8}

Dictionary– A dictionary stores values in key and value pairs where each value can be accessed through its key. The order of items is not important. Eg. {1:'apple',2:'mango}

Boolean– There are 2 boolean values- True and False.

Q5.What is slicing in Python?

Ans: Slicing is used to access parts of sequences like lists, tuples, and strings. The syntax of slicing is-[start:end:step]. The step can be omitted as well. When we write [start:end] this returns all the elements of the sequence from the start (inclusive) till the end-1 element. If the start or end element is negative i, it means the ith element from the end. The step indicates the jump or how many elements have to be skipped. Eg. if there is a list-[1,2,3,4,5,6,7,8]. Then [-1:2:2] will return elements starting from the last element till the third element by printing every second element.i.e. [8,6,4].

Q6.What are Keywords in Python?

Ans: Keywords in python are reserved words that have special meaning. They are generally used to define type of variables. Keywords cannot be used for variable or function names. There are following 33 keywords in python-

And

Or

Not

If

Elif

Else

For

While

Break

As

Def

Lambda

Pass

Return

True

False

Try

With

Assert

Class

Continue

Del

Except

Finally

From

Global

Import

In

Is

None

Nonlocal

Raise

Yield

Q7. What are python modules? Name some commonly used built-in modules in Python?

Ans: Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.

Some of the commonly used built-in modules are:

os

sys

math

random

data time

JSON

Q8.What are local variables and global variables in Python?

Global Variables:

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

Example:

```
a=2
def add():
b=3
c=a+b
print(c)
add()
```

Output: 5

Q9.What is type conversion in Python?

Ans: Type conversion refers to the conversion of one data type into another.

int() – converts any data type into integer type

float() – converts any data type into float type

ord() – converts characters into integer

hex() – converts integers to hexadecimal

oct() – converts integer to octal

tuple() – This function is used to convert to a tuple.

set() – This function returns the type after converting to set.

list() – This function is used to convert any data type to a list type.

dict() – This function is used to convert a tuple of order (key, value) into a dictionary.

str() – Used to convert integer into a string.

complex(real,imag) – This function converts real numbers to complex(real,imag) number.

Q10. What is the difference between Python Arrays and lists?

Ans: Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements.

Example:

```
import array as arr
My_Array=arr.array('i',[1,2,3,4])
My_list=[1,'abc',1.20]
print(My_Array)
print(My_list)
```

Output:

array(‘i’, [1, 2, 3, 4]) [1, ‘abc’, 1.2]

Q11. What are functions in Python?

Ans: A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used.

Example:

calling the function

Q12.What is __init__?

Ans: __init__ is a method or constructor in Python. This method is automatically called to allocate memory when a new object/ instance of a class is created. All classes have the __init__ method.

Here is an example of how to use it.

```
class Employee:
    def __init__(self, name, age,salary):
        self.name = name
        self.age = age
        self.salary = 20000
E1 = Employee("XYZ", 23, 20000)
print(E1.name)
print(E1.age)
print(E1.salary)
```

Q13.What is a lambda function?

Ans: An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.

Example:

```
a = lambda x,y : x+y
print(a(5, 6))
```

Q14. What does [::-1} do?

Ans: [::-1] is used to reverse the order of an array or a sequence.

For example:

```
import array as arr
My_Array=arr.array('i',[1,2,3,4,5])
My_Array[::-1]
```

Output: array('i', [5, 4, 3, 2, 1])

[::-1] reprints a reversed copy of ordered data structures such as an array or a list. the original array or list remains unchanged.

Q15. How can you randomize the items of a list in place in Python?

Ans: Consider the example shown below:

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

The output of the following code is as below.

['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']

Q16. What is a dictionary in Python?

Ans: The built-in datatypes in Python is called dictionary. It defines one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values. Dictionaries are indexed by keys.

Q17. What are negative indexes and why are they used?

Ans: The sequences in Python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive uses '0' that is uses as first index and '1' as the second index and the process goes on like that.

The index for the negative number starts from '-1' that represents the last index in the sequence and '-2' as the penultimate index and the sequence carries forward like the positive number.

The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as S[:-1]. The negative index is also used to show the index to represent the string in correct order.

Q18. How to remove values to a python array?

Ans: Array elements can be removed using pop() or remove() method. The difference between these two functions is that the former returns the deleted value whereas the latter does not.

Example:

```
a=arr.array('d', [1.1, 2.2, 3.8, 3.1, 3.7, 1.2, 4.6])
print(a.pop())
print(a.pop(3))
a.remove(1.1)
print(a)
```

Coding Questions

1. Write a programme to print where a number is even or odd.

```
number = input("Enter a number ")
x = int(number)%2
if x == 0:
   print(" The number is Even ")
else:
   print(" The number is odd ")
```

2. Python program to convert the temperature in degree centigrade to Fahrenheit

```
c = input(" Enter temperature in Centigrade: ")
f = (9*(int(c))/5)+32
print(" Temperature in Fahrenheit is: ", f)
```

3. Python program to find the area of a triangle whose sides are given

```
import math
a = float(input("Enter the length of side a: "))
b = float(input("Enter the length of side b: "))
c = float(input("Enter the length of side c: "))
s = (a+b+c)/2
area = math.sqrt(s*(s-a)*(s-b)*(s-c))
```

```
print(" Area of the triangle is: ", area)
```

4. Python program to find out the average of a set of integers

```
count = int(input("Enter the count of numbers: "))
i = 0
sum = 0
for i in range(count):
    x = int(input("Enter an integer: "))
    sum = sum + x
avg = sum/count
print(" The average is: ", avg)
```

5. Python program to find the product of a set of real numbers

```
i = 0
product = 1
count = int(input("Enter the number of real numbers: "))
for i in range(count):
    x = float(input("Enter a real number: "))
    product = product * x
print("The product of the numbers is: ", product)
```

6. Python program to find the circumference and area of a circle with a given radius

```
import math
r = float(input("Input the radius of the circle: "))
c = 2 * math.pi * r
area = math.pi * r * r
print("The circumference of the circle is: ", c)
print("The area of the circle is: ", area)
```

7. Python program to check whether the given integer is a multiple of 5

```
number = int(input("Enter an integer: "))
if(number%5==0):
    print(number, "is a multile of 5")
else:
    print(number, "is not a multiple of 5")
```

8. Python program to check whether the given integer is a multiple of both 5 and 7

```
number = int(input("Enter an integer: "))
if((number%5==0)and(number%7==0)):
    print(number, "is a multiple of both 5 and 7")
else:
    print(number, "is not a multiple of both 5 and 7")
```

9. Python program to find the average of 10 numbers using while loop

```
count = 0
sum = 0.0
while(count<10):
    number = float(input("Enter a real number: "))
    count=count+1
    sum = sum+number
avg = sum/10;
print("Average is :",avg)
```

10. Python program to display the given integer in reverse manner

```
number = int(input("Enter a positive integer: "))
rev = 0
while(number!=0):
    digit = number%10
```

```
    rev = (rev*10)+digit
    number = number//10
print(rev)
```

11. Python program to find the geometric mean of n numbers

```
c = 0
p = 1.0
count = int(input("Enter the number of values: "))
while(c<count):
    x = float(input("Enter a real number: "))
    c = c+1
    p = p * x
gm = pow(p,1.0/count)
print("The geometric mean is: ",gm)
```

12. Python program to find the sum of the digits of an integer using while loop

```
sum = 0
number = int(input("Enter an integer: "))
while(number!=0):
    digit = number%10
    sum = sum+digit
    number = number//10
print("Sum of digits is: ", sum)
```

13. Python program to display all the multiples of 3 within the range 10 to 50

```
for i in range(10,50):
    if (i%3==0):
        print(i)
```

14. Python program to display all integers within the range 100-200 whose sum of digits is an even number

```
for i in range(100,200):
    num = i
    sum = 0
    while(num!=0):
        digit = num%10
        sum = sum + digit
        num = num//10
    if(sum%2==0):
        print(i)
```

15. Python program to check whether the given integer is a prime number or not

```
num = int(input("Enter an integer greater than 1: "))
isprime = 1

for i in range(2,num//2):
    if (num%i==0):
        isprime = 0
        break
if(isprime==1):
    print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

16. Python program to generate the prime numbers from 1 to N

```
num =int(input("Enter the range: "))
for n in range(2,num):
    for i in range(2,n):
```

```
        if(n%i==0):
            break
    else:
        print(n)
```

17. Python program to implement linear search

```
numbers = [4,2,7,1,8,3,6]
f = 0

x = int(input("Enter the number to be found out: "))
for i in range(len(numbers)):
    if (x==numbers[i]):
        print(" Successful search, the element is found at position", i)
        f = 1
        break
if(f==0):
    print("Oops! Search unsuccessful")
```

18. Python program to implement binary search

```
def binarySearch(numbers, low, high, x):
    if (high >= low):
        mid = low + (high - low)//2
        if (numbers[mid] == x):
            return mid
        elif (numbers[mid] > x):
            return binarySearch(numbers, low, mid-1, x)
        else:
            return binarySearch(numbers, mid+1, high, x)
    else:
```

```
        return -1
numbers = [ 1,4,6,7,12,17,25 ]

x = 7
result = binarySearch(numbers, 0, len(numbers)-1, x)
if (result != -1):
   print("Search successful, element found at position ", result)
else:
   print("The given element is not present in the array")
```

19. Python program to find the odd numbers in an array

```
numbers = [8,3,1,6,2,4,5,9]
count = 0
for i in range(len(numbers)):
   if(numbers[i]%2!=0):
      count = count+1
print("The number of odd numbers in the list are: ", count)
```

20. Python program to insert a number to any position in a list

```
numbers = [3,4,1,9,6,2,8]
print(numbers)
x = int(input("Enter the number to be inserted: "))
y = int(input("Enter the position: "))
numbers.insert(y,x)
print(numbers)
```

21. Python program to delete an element from a list by index

```
numbers = [3,4,1,9,6,2,8]
print(numbers)
```

```
x = int(input("Enter the position of the element to be deleted: "))
numbers.pop(x)
print(numbers)
```

22. Python program to implement matrix addition

```
X = [[8,5,1],
[9 ,3,2],
[4 ,6,3]]
Y = [[8,5,3],
[9,5,7],
[9,4,1]]
result = [[0,0,0],
[0,0,0],
[0,0,0]]
for i in range(len(X)):
   for j in range(len(X[0])):
      result[i][j] = X[i][j] + Y[i][j]
for k in result:
   print(k)
```

23. Python program to implement matrix multiplication

```
X = [[8,5,1],
[9 ,3,2],
[4 ,6,3]]
Y = [[8,5,3],
[9,5,7],
[9,4,1]]
result = [[0,0,0,0],
[0,0,0,0],
[0,0,0,0]]
```

```
for i in range(len(X)):
   for j in range(len(Y[0])):
      for k in range(len(Y)):
         result[i][j] += X[i][k] * Y[k][j]
for x in result:
   print(x)
```

24. Python program to check leap year

```
year = int(input("Enter a year: "))
if (year % 4) == 0:
   if (year % 100) == 0:
      if (year % 400) == 0:
         print(year, " is a leap year")
      else:
         print(year, " is not a leap year")
   else:
      print(year, " is a leap year")
else:
   print(year, " is not a leap year")
```

25. Python program to find the Nth term in a Fibonacci series using recursion

```
def Fib(n):
   if n<0:
      print("The input is incorrect.")
   elif n==1:
      return 0
   elif n==2:
      return 1
   else:
      return Fib(n-1)+Fib(n-2)
```

```
print(Fib(7))
```

26. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

```
def dedupe_v1(x):
  y = []
  for i in x:
    if i not in y:
      y.append(i)
  return y
def dedupe_v2(x):
    return list(set(x))

a = [1,2,3,4,3,2,1]
print a
print dedupe_v1(a)
print dedupe_v2(a)
```

27. Create n*n matrix, take any number of rows and columns.

```
N = 4
print("The dimension : " + str(N))
res = [list(range(1 + N * i, 1 + N * (i + 1)))
                    for i in range(N)]
print("The created matrix of N * N: " + str(res))
```

28. Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5,
between 2000 and 3200 (both included).
The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints:

Consider use range(

end) method

Solution:

```
l=[]
for i in range(2000, 3201):
    if (i%7==0) and (i%5!=0):
        l.append(str(i))

print ','.join(l)
```

29. Write a program which can compute the factorial of a given numbers.

The results should be printed in a comma-separated sequence on a single line.

Suppose the following input is supplied to the program:

8

Then, the output should be:

40320

Solution:

```
def fact(x):
    if x == 0:
        return 1
    return x * fact(x - 1)

x=int(raw_input())
print fact(x)
```

30. Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence.

Example:

0100,0011,1010,1001

Then the output should be:

1010

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
value = []
items=[x for x in raw_input().split(',')]
for p in items:
    intp = int(p, 2)
    if not intp%5:
        value.append(p)
print ','.join(value)
```