

**Q26. What is a string? How can we declare string in Python?**

Strings are sequences of character data, Strings can be created by enclosing characters inside a single quote or double-quotes. Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings

**Q27. How can we access the string using its index?**

Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets ( `[]` ).

**Q28. Write a code to get the desired output of the following**

```
string = "Big Data iNeuron"
desired_output = "iNeuron"
print(string)
print(desired_output)
```

**Q29. Write a code to get the desired output of the following**

```
string = "Big Data iNeuron"

desired_output = "norueNi"

print(string)

print(desired_output)
```

**Q30. Reverse the string given in the above question.**

```
string = "Big Data iNeuron"[::-1]
desired_output = "norueNi"[::-1]
print(string)
print(desired_output)
```

**Q31. How can you delete entire string at once?**

```
string1 = "Big Data iNeuron"
print(string1)
del(string1)
desired_output = "norueNi"
del(desired_output)
```

**Q32. What is escape sequence?**

An escape sequence is a sequence of characters that, when used inside a character or string, does not represent itself but is converted into another character or series of characters

**Q33. How can you print the below string?**

```
str1='iNeuron's Big Data Course'  
print(str1)
```

**Q34. What is a list in Python?**

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements.

**Q35. How can you create a list in Python?**

A list is created by placing elements inside square brackets [], separated by commas.

**Q36. How can we access the elements in a list?**

List literals are written within square brackets []. Lists work similarly to strings -- use the len() function and square brackets [] to access data, with the first element at index 0.

**Q37. Write a code to access the word "iNeuron" from the given list.**

```
lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]  
lst1=lst[4]  
print(lst1[2])
```

**Q38. Take a list as an input from the user and find the length of the list.**

```
list1 = [1,2,3 ]  
print(len(list1))
```

**Q39. Add the word "Big" in the 3rd index of the given list.**

```
lst = ["Welcome", "to", "Data", "course"]  
lst.insert(2,"big")  
print(lst)
```

**Q40. What is a tuple? How is it different from list?**

The primary difference between tuples and lists is that tuples are immutable as opposed to lists which are mutable. Therefore, it is possible to change a list but not a tuple. The contents of a tuple cannot change once they have been created in Python due to the immutability of tuples.

**Q41. How can you create a tuple in Python?**

A tuple is created by placing all the items (elements) inside parentheses () , separated by commas.Example:AND,OR,NOT.

**Q42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.**

We can't add our name in the tuple because it is immutable.

**Q43. Can two tuple be appended. If yes, write a code for it. If not, why?**

Tuple is immutable, although you can use the + operator to concatenate several tuples.

```
s=(2,5,8)

s_append = s + (8, 16, 67)

print(s_append)

print(s)
```

**Q44. Take a tuple as an input and print the count of elements in it.**

```
s=(2,5,8)

print(len(s))
```

**Q45. What are sets in Python?**

A set is an unordered collection of items

**Q46. How can you create a set?**

A set is created by placing all the items (elements) inside curly braces {}, separated by comma, or by using the built-in set() function

**Q47. Create a set and add "iNeuron" in your set.**

```
course = {"you", "welcome"}

course.add("ineuron")

print(course)
```

**Q48. Try to add multiple values using add() function.**

```
course = {"you", "welcome"}
```

PRASHANTH U

```
course.add("ineuron")
```

```
course.add("big")
```

```
course.add("data")
```

```
print(course)
```

**Q49. How is update() different from add()?**

add method directly adds elements to the set while the update method converts first argument into set then it adds the list is hashable therefore we cannot add a hashable list to unhashable set.

**Q50. What is clear() in sets?**

The clear() method removes all elements in a set.

**Q51. What is frozen set?**

The frozenset() function returns an immutable frozenset object initialized with elements from the given iterable.

**Q52. How is frozen set different from set?**

Frozen set is just an immutable version of a Python set object. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation. Due to this, frozen sets can be used as keys in Dictionary or as elements of another set.

**Q53. What is union() in sets? Explain via code.**

The union() method returns a new set with elements from the set and all other sets (passed as an argument).

If  $A = \{ 23, 453, 43, 66, 21 \}$  and  $B = \{ 43, 785, 12, 35, 23, 545 \}$

Then  $A \cup B = \{ 23, 453, 43, 66, 21, 43, 785, 12, 35, 545 \}$

**Q55. What is dictionary in Python?**

Dictionaries are Python's implementation of a data structure that is more generally known as an associative array.

**Q56. How is dictionary different from all other data structures.**

A dictionary is a general-purpose data structure for storing a group of objects. A dictionary has a set of keys and each key has a single associated value. When presented with a key the dictionary will • A dictionary has a set of keys and each key has a single associated value.

**Q57. How can we declare a dictionary in Python?**

A Dictionary in python is declared by enclosing a comma-separated list of key-value pairs using curly braces({})

**Q58. What will the output of the following?**

```
var = {}  
  
print(type(var))  
  
<class 'dict'>
```

**Q59. How can we add an element in a dictionary?**

Add an item to a dictionary by inserting a new index key into the dictionary, then assigning it a particular value.

**Q60. Create a dictionary and access all the values in that dictionary.**

```
thisdict = {  
  
    "brand": "Ford",  
  
    "model": "Mustang",  
  
    "year": 1964  
  
}  
  
x = thisdict["model"]
```

**OutPut-Mustang**

**Q61. Create a nested dictionary and access all the element in the inner dictionary.**

```
myfamily = {  
  
    "child1" : {
```

```
"name" : "Emil",  
  
"year" : 2004  
  
,  
  
"child2" : {  
  
    "name" : "Tobias",  
  
    "year" : 2007  
  
},  
  
"child3" : {  
  
    "name" : "Linus",  
  
    "year" : 2011  
  
}  
  
}
```

Output-{'child1': {'name': 'Emil', 'year': 2004}, 'child2': {'name': 'Tobias', 'year': 2007}, 'child3': {'name': 'Linus', 'year': 2011}}

**Q62. What is the use of get() function?**

The get() method returns the value of the item with the specified key.

**Q63. What is the use of items() function?**

items() method is used to return the list with all dictionary keys with values

**Q64. What is the use of pop() function?**

List pop in Python is a pre-defined, in-built function that removes an item at the specified index from the list.

**Q65. What is the use of popitem() function?**

Python dictionary popitem() method is used to return and remove the last inserted element of the dictionary (dict).

**Q66. What is the use of keys () function?**

Python dictionary keys() function is used to return a new view object that contains a list of all the keys in the dictionary. The Python dictionary keys() method returns an object that contains all the keys in a dictionary.

To use the Python dictionary keys() function, we must follow the below syntax:

```
dict_name.keys()
```

**Q67. What is the use of values() function?**

The VALUE Function[1] is categorized under Excel Text functions. It will convert a text string that represents a number into a number. Thus, the function will convert text that appears in a recognized format (a number, date, or time format) into a numeric value.

**Q68. What are loops in Python?**

Looping means repeating something over and over until a particular condition is satisfied. A for loop in Python is a control flow statement that is used to repeatedly execute a group of statements as long as the condition is satisfied. Such a type of statement is also known as an iterative statement

**Q69. How many type of loop are there in Python?**

There are two types of loops in Python, for and while.

**Q70. What is the difference between for and while loops?**

Both for loop and while loop is used to execute the statements repeatedly while the program runs. The major difference between for loop and the while loop is that for loop is used when the number of iterations is known, whereas execution is done in the while loop until the statement in the program is proved wrong.

**Q71. What is the use of continue statement?**

A continue statement ends the current iteration of a loop. Program control is passed from the continue statement to the end of the loop body. A continue statement can only appear within the body of an iterative statement, such as do , for , or while .

**Q72. What is the use of break statement?**

The break statement is frequently used to terminate the processing of a particular case within a switch statement. Lack of an enclosing iterative or switch statement generates an error.

**Q73. What is the use of pass statement?**

The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements.

**Q74. What is the use of range() function?**

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

**Q75. How can you loop over a dictionary?**

You can loop through a dictionary by using a for loop. When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

**### Coding problems**

**Q76. Write a Python program to find the factorial of a given number.**

```
num = 7

#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero

if num < 0:

    print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

    print("The factorial of 0 is 1")

else:

    for i in range(1,num + 1):

        factorial = factorial*i

    print("The factorial of",num,"is",factorial)
```

**Q77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is  $SI = (P \cdot R \cdot T) / 100$**

P = 1000

T = 5

R = 4



$$SI = (P * T * R)/100$$

```
print('The Simple Interest is', si)
```

**Q78. Write a Python program to calculate the compound interest. Formula of compound interest is  $A = P(1 + R/100)^t$ .**

```
def compound_interest(principle, rate, time):
```

```
    # Calculates compound interest
```

```
    Amount = principle * (pow((1 + R / 100), t))
```

```
    CI = Amount - principle
```

```
    print("Compound interest is", CI)
```

```
compound_interest(10000, 10.25, 5)
```

**Q79. Write a Python program to check if a number is prime or not.**

```
n = 5
```

```
# Check if the number is greater than 1
```

```
if n > 1:
```

```
    for i in range(2, int(n/2)+1):
```

```
        if (n % i) == 0:
```

```
            print(num, "is not a prime number")
```

```
            break
```

```
    else:
```

```
        print(n, "is a prime number")
```

```
# If the number is less than 1, its also not a prime number.
```

```
else:
```

```
    print(n, "is not a prime number")
```

**Q80. Write a Python program to check Armstrong Number.**

```
num = int(input("Enter a number: "))

# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10

# display the result
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

**Q81. Write a Python program to find the n-th Fibonacci Number.**

```
def Fibonacci(n):
    if n <= 0:
        print("Incorrect input")
    # First Fibonacci number is 0
    elif n == 1:
        return 0
    # Second Fibonacci number is 1
    elif n == 2:
        return 1
    else:
```

```
        return Fibonacci(n-1)+Fibonacci(n-2)

print(Fibonacci(10))
```

**Q82. Write a Python program to interchange the first and last element in a list.**

```
def swapList(newList):
    size = len(newList)

    # Swapping
    temp = newList[0]
    newList[0] = newList[size - 1]
    newList[size - 1] = temp

    return newList
```

```
newList = [12, 35, 9, 56, 24]
```

```
print(swapList(newList))
```

**Q83. Write a Python program to swap two elements in a list.**

```
def swapList(sl,pos1,pos2):
    sl[pos1], sl[pos2] = sl[pos2], sl[pos1]
    return sl
```

```
List = [9, 11, 5, 3, 6, 27, 4]
```

```
pos1, pos2= 3,5
```

```
print(List)
```

```
print("Swapped List: ",swapList(List,pos1-1,pos2-1))
```

**Q84. Write a Python program to find N largest element from a list.**

#N largest elements

#function

```
def N_max_elements(list, N):
```

```
    result_list = []
```

```
    for i in range(0, N):
```

```
        maximum = 0
```

```
        for j in range(len(list)):
```

```
            if list[j] > maximum:
```

```
                maximum = list[j]
```

```
        list.remove(maximum)
```

```
        result_list.append(maximum)
```

```
    return result_list
```

#test

```
list1 = [2, 6, 41, 85, 0, 3, 7, 6, 10]
```

```
N = 2
```

```
print(N, "max elements in ",list1)
```

# Calling and printing the function

```
print(N_max_elements(list1, N))
```

**Q85. Write a Python program to find cumulative sum of a list.**

```
def Cumulative_sum(lists):  
    cum_list = []  
    lenlength = len(lists)  
    cum_list = [sum(lists[0:x:1]) for x in range(0, length+1)]  
    return cum_list[1:]
```

```
lists = [10, 15, 20, 25, 30]  
print (Cumulative_sum(lists))
```

**Q86. Write a Python program to check if a string is palindrome or not.**

# function which return reverse of a string

```
def isPalindrome(s):  
    return s == s[::-1]
```

```
# Driver code  
s = "malayalam"  
ans = isPalindrome(s)
```

```
if ans:  
    print("Yes")  
else:  
    print("No")
```

**Q87. Write a Python program to remove i'th element from a string.**

```
def remove(string, i):  
  
    a = string[ : i]
```

```
        b = string[i + 1: ]

    return a + b

# Driver Code
if __name__ == '__main__':

    string = "geeksFORgeeks"

    # Remove nth index element
    i = 5

    # Print the new string
    print(remove(string, i))
```

**Q88. Write a Python program to check if a substring is present in a given string.**

```
def check(string, sub_str):
    if (string.find(sub_str) == -1):
        print("NO")
    else:
        print("YES")
```

```
string = "hello how do you do"
sub_str = "I am good"
check(string, sub_str)
```

**Q89. Write a Python program to find words which are greater than given length k.**

```
def word_k(k, s):
    # split the string where space comes
    word = s.split(" ")
```

```
# iterate the loop for every word
for x in word:
    # if length of current word
    if len(x)>k:
        # greater than k then
        print(x)
k = 3
s = "Python is good"
word_k(k, s)
```

**Q90. Write a Python program to extract unquire dictionary values.**

```
test_dict = {'gfg' : [5, 6, 7, 8],
             'is' : [10, 11, 7, 5],
             'best' : [6, 12, 10, 8],
             'for' : [1, 2, 5]}

print("The original dictionary is : " + str(test_dict))
x=[]
for i in test_dict.keys():
    x.extend(test_dict[i])
x=list(set(x))
x.sort()
# printing result
print("The unique values list is : " + str(x))
```

**Q91. Write a Python program to merge two dictionary.**

```
dict_1 = {1: 'a', 2: 'b'}
dict_2 = {2: 'c', 4: 'd'}
```

```
print(dict_1 | dict_2)
```

**Q92. Write a Python program to convert a list of tuples into dictionary.**

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

```
def Convert(tup, di):  
    for a, b in tup:  
        di.setdefault(a, []).append(b)  
    return di
```

# Driver Code

```
tups=[('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)],  
      {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}  
dictionary = { }  
print (Convert(tups, dictionary))
```

**Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.**

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

```
# creating a list  
list1 = [9, 5, 6]  
res = [(val, pow(val, 3)) for val in list1]  
print(res)
```

**Q94. Write a Python program to get all combinations of 2 tuples.**

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

```
# original tuples  
print("test_tuple1 : " + str(test_tuple1))  
print("test_tuple2: " + str(test_tuple2))  
result = [(x, y) for x in test_tuple1 for y in test_tuple2]
```



```
result = result + [(x, y) for x in test_tuple1 for y in test_tuple2]
print("The resultant tuple : " + str(result))
```

**Q95. Write a Python program to sort a list of tuples by second item.**

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]

```
def Sort_Tuple(tup):
    return(sorted(tup, key = lambda x: x[1]))

tup = [('for', 24), ('Geeks', 8), ('Geeks', 30)]
print(Sort_Tuple(tup))
```

**Q96. Write a python program to print below pattern.**

```
*
* *
* * *
* * * *
* * * * *
```

```
def pypart(n):
    myList = []
    for i in range(1,n+1):
        myList.append("*"*i)
    print("\n".join(myList))
```

# Driver Code

n = 5

pypart(n)

**Q97. Write a python program to print below pattern.**

PRASHANTH U

```
*  
**  
***  
****  
*****
```

```
n=5;i=0  
while(i<=n):  
print(" " * (n - i) + "*" * i)  
i+=1
```

**Q98. Write a python program to print below pattern.**

```
*  
* *  
* * *  
* * * *  
* * * * *
```

```
k = n - 1  
for i in range(0, n):  
    for j in range(0, k):  
        print(end=" ")  
    k = k - 1  
    for j in range(0, i+1):  
        print("* ", end="")  
    print("\r")
```

# Driver Code

```
n = 5  
triangle(n)
```

**Q99. Write a python program to print below pattern.**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

# Python 3.x code to demonstrate star pattern

# Function to demonstrate printing pattern of numbers

```
def numpat(n):
    num = 1
    for i in range(0, n):
        num = 1
        for j in range(0, i+1):
            print(num, end=" ")
            num = num + 1
        print("\r")
```

# Driver code

```
n = 5
numpat(n)
```

**Q100. Write a python program to print below pattern.**

```
A
B B
C C C
D D D D
```

PRASHANTH U

E E E E E

```
def alphapat(n):  
    num = 65  
    for i in range(0, n):  
        for j in range(0, i+1):  
            ch = chr(num)  
            print(ch, end=" ")  
        num = num + 1  
        print("\r")  
  
n = 5  
alphapat(n)
```