
Assignment -1

Q.1 Why do we call Python as a general purpose and high-level programming language?

Ans. General purpose programming language because it is designed to be used for writing software in the widest variety of application domains. A general purpose language has this status because it does not include language constructs to be used within a specific application domain.

High level programming language because the amount of abstraction, it is very abstract and uses natural processing elements, which make its easier to use and understand. It makes whole process simpler and more automated than low level language.

Q.2 Why is Python called a dynamically typed language?

Ans. Python is a dynamically typed language because it doesn't know about the type of the variable until the code is run. It means it store the value at some memory location and then binds that variable name to that memory container and makes the contents of the container accessible through that variable name. So the data type doesn't matter as it will get to know the type of the value at the run-time.

Q.3 List some pros and cons of Python programming language?

Pros of Python:

Simple and Easy

Extensive libraries

Portable – Python is portable, which means it can be run on any other platform. Here you need to write code only once and you can run it anywhere.

Highly scalable

Large community

Flexible and extensible

Cons of Python:

Issues with design

Slower than compiled languages

Python's memory consumption and garbage connection – Python language comes with high memory usage and high memory consumption. It uses reference counting in its garbage collection which often leads to potential memory losses.

Python is dynamically typed

Multithreading in Python – The multithreading in python is not true multithreading due to its global interpreter lock (GIL). The multithreading model in python doesn't have threads that run at same time. Here, one thread can hold the GIL at one time, which clearly implies that this is not exactly multithreading.

Q.4 In what all domains can we use Python?

Ans. Python is used in many application domains

Web Development

Data Analysis and Visualization

Machine Learning and AI

Scientific Computation – Packages likes pandas and numpy used for this purpose

Desktop GUI Applications

Software Deleopment

Q.5 What are variable and how can we declare them?

Ans. Variable is the name given to the specific memory location. Python has no command for declaring a variable. Declaring in python is very simple.

Just name the variable

Assign the required value to it

The data type of the variable will be automatically determined from the value assigned, we don't have need to define it explicitly.

Ex- Declare int variable

a=5

Q.6 How can we take an input from the user in Python?

Ans. In python we can take input from user by using input() function.

Q.7 What is the default datatype of the value that has been taken as an input using input() function?

Ans. String datatype is the default datatype for the value that has taken by using input() function.

Q.8 What is type casting?

Ans. Type casting is a method to convert the variable data type into certain data type in order to the operation required to be performed by users. There are two type of type casting in python-

1) Implicit type casting- In this method python converts data type into another data type automatically.

Example- Python automatic converts b to float

b = 8.5 (Implicit type casting)

2) Explicit type casting- In this method, python needs user involvement to convert the variable data type into certain data type to the operation required.

Example- Type casting int to float

a=5(int variable)

b = float(a)(explicit type casting)

Q.9 Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Ans. Yes we can take more than one input from the user using single input() function in python by using two methods – 1) using split() 2)Using list comprehension

1) Using split() method – This function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator. Generally users use a split() method to split python string but one can use it in taking multiple inputs.

Example- Taking multiple inputs at a time

a,b,c = input("Enter values ").split()

taking multiple inputs at a time and type casting using list() function

x = list(map(int,input("Enter multiple values").split()))

2) Using list comprehension –List comprehension is an elegant way to define and create list in Python. It also used in getting multiple inputs from users.

Example-

x = [int(x) for x in input("Enter value").split()]

Q.10 What are keywords?

Ans. Python has a set of keywords that are reserved words that cannot be used as variable names, function names or any other identifiers.

Example:

And, or, elif, except, lambda, except, return, global, yield, while, try etc.

Q.11 Can we use keywords as a variable? Support your answer with reason.

Ans. We cannot use a keyboard as a variable name, function name or any other identifier. They are used to define the syntax and the structure of the python language. In python , keywords are the case sensitive.

Q.12 What is indentation? What's the use of indentaion in Python?

Ans. Indentation refers to the spaces at the beginning of a code line. Python uses indentation to indicate the block of code.

Python indentation is the way to telling the interpreter that the group of statements belongs to a particular block of code. A block is the combination of all these statements. Most programming language like C, C++, JAVA uses braces {} and Python uses indentation to define the block of code.

Q.13 How can we throw some output in Python?

Ans. Using print()

Q.14 What are operators in Python?

Ans. Operators are used to perform operation on variables and values. Python divides the operators in the following groups-

-Airthmetic Operators

-Assignment Operators

-Comparasion Operators

-Logical Operators

-Identity Operator

-Membership Operators

-Bitwise Operators

Q.15 What is difference between / and // operators?

Ans. /--→ Floating point division

// ---→Floor division sometimes also called int division

Example: 4/2----- 2.0(ans)

5//3 ---- 1(ans)

Q.16 Write a code that gives following as an output.

```

iNeuroniNeuroniNeuroniNeuron

```

Ans. print("```")

print("iNeuroniNeuroniNeuroniNeuron")

print("```")

Q.17 Write a code to take a number as an input from the user and check if the number is odd or even.

Ans.

```
def check(n):
    if n % 2 == 0:
        print("Even")
    else:
        print("Odd")


n = int(input("Enter the number"))
check(n)
```

Q.18 What are boolean operator?

Ans. Boolean operators are those that result in Boolean values of True and False. These include and, or and not. While and & or required two operands, not is a unary operator. Boolean operators are most commonly used in arithmetic computations and logical comparasion.

Q.19 What will the output of the following?

```

1 or 0

0 and 0
```

True and False and True

1 or 0 or 0

```

Ans. 1 or 0 => 1

0 and 0 => 0

True and False and True => False

1 or 0 or 0 => 1

Q.20 What are conditional statements in Python?

Ans. Conditional statements are useful when we execute a specific code based on a condition. It also helps in decision making in python, preferably when we prefer to execute a piece of code only if certain conditionals are met.

Q.21 What is use of 'if', 'elif' and 'else' keywords?

Ans. IF – It executes the conditional block only when the statement is true.

If(Something isTrue)

// executes the block

ELIF – Useful when we try to put up one more check in case the “if” conditional statement fails.

ELSE – The else keyword catches anything which isn’t caught by the proceding conditions.

Q.22 Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

```
def check(age):
    if age >= 18:
        print("I can vote")
    else:
        print("I can't vote")
age = int(input("Enter your age : "))
check(age)
```
```

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

```
def check(a, b, c):
    if a > b and a >c:
        print(a)
    elif b > a and b > c:
        print(b)
    else:
        print(c)


a, b, c = map(int,input("Enter three numbers :
").split())
check(a,b,c)
```

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```numbers = [12, 75, 150, 180, 145, 525, 50]```

```
def check(numbers):
    for i in numbers:
        if i%5 == 0 and i <= 150:
            print(i)
        elif i > 500:
            break
        else:
            continue
numbers = [12, 75, 150, 180, 145, 525, 50]
check(numbers)
```
```

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

Ans. A string is a single character or a collection of characters. String in python are surrounded by either single quotation marks or double quotation marks.

'hello' is same as "hello"

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

Ans. Syntax: string_name[index]

Example: s = "hello"

Print(s[0])-----→o/p- h

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

```

string = "Big Data iNeuron"

desired_output = "iNeuron"

```

```
Ans. s = "Big Data iNeuron"
print(s[len(s)-7:])
```

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

```

string = "Big Data iNeuron"

desired_output = "norueNi"

```

```
Ans. s = "Big Data iNeuron"
print(s[len(s):len(s)-8:-1])
```

Q30. Reverse the string given in the above question.

```
Ans s = "Big Data iNeuron"
print(s[::-1])
```

Q31. How can you delete entire string at once?

Ans. Using del keyboard

Ex. del s --→ here s is a string name

Q32. What is escape sequence?

Ans. An escape sequence is a special character used in the form of backslash(\) followed by a character that is required. These characters are used to represent whitespace. Whitespace gives characters like space, tab, etc.

Example. \'→ this represent single quote \n→ this represent new line \t→ this represents tab

Q33. How can you print the below string?

```
'iNeuron's Big Data Course'
```

Ans. Using backslash we can print

```
print("'iNeuron\'s Big Data Course'")
```

Q34. What is a list in Python?

Ans. List are used to store multiple items in a single variable.

Q35. How can you create a list in Python?

Ans. List1 = [] -→ it will create empty list

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

Ans. List items are indexed and we can access them by referring to the index number

```
Example. thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

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

```lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]```

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

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

```
Ans. lst = list(map(int,input("Enter list elements : ").split()))
print(len(lst))
```

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

```lst = ["Welcome", "to", "Data", "course"]```

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

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

Ans. Tuple are used to store multiple items in a single variable. A tuple is a collection which is ordered and unchangeable. Tuple processing is faster than the list.

Difference between tuple and list-

1.Tuple are immutable and list are mutuable.

2. Tuple iteration is faster than list iteration.

3. Tuple consumes less memory then list.

Q41. How can you create a tuple in Python?

Ans. tup=() --→by using paratheses

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

Ans. We cannot add element in tuple because tuple is a collection which is unchangeable (immutable).

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

Ans. We can use the + operator to concatenate the value onto our tuple.

```
a_tuple = (1,2,3)
b_tuple = (4,5,6)
a_tuple = a_tuple+b_tuple
print(a_tuple)
```

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

```
tup = tuple(map(int,input("Enter elements:
").split()))
print(len(tup))
```

Q45. What are sets in Python?

Ans. A set is an unordered and mutual collection of unique elements. Sets are used to store multiple items in a single variable.

Q46. How can you create a set?

Ans. Sets have the elements in the curly brackets with each of them separated by comma. It can contain values of different data types.

S = {1,2,3,4}

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

```
Ans. s = set()
s.add("iNeuron")
print(s)
```

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

Ans. We can not add multiple values using add() function. Add() function take only one argument.

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

Ans. To add a single element use the add() function. To add multiple elements to a set use the update function. Add() is quicker than update(). Add() takes only one arguments but update() can take multiple sequences.

Q50. What is clear() in sets?

Ans. The clear method empties the set.

Q51. What is frozen set?

Ans. Frozen set are just the immutable version of the python set object.

Q52. How is frozen set different from set?

Ans. Frozen seta are immutable version of the python set object. While elements of a set can be modified at any time, elements of a set remain same after creation.

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

Ans. Union() will return a set that contains all items from both the sets, duplicates are excluded.

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.union(y)

print(z)
```

Q54. What is intersection() in sets? Explain via code.

And. Intersection() will return a set that contains the items that exist in both the sets.

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.intersection(y)
```

```
print(z)
```

Q55. What is dictionary in Python?

Ans. Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is ordered , changeable and do not allow duplicates.

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

Ans. The main difference is dictionaries stores data values in key values pair and in other data structures values stored at a particular index. The items in dictionaries are accessed by using key and in other data structure values is accessed through index number.

Q57. How can we delare a dictionary in Python?

```
dict = {}
print(type(dict))
```

Q58. What will the output of the following?

```var = {}

print(type(var))```

Ans. <class 'dict'>

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

Ans. Adding an item to the dictionary is done by using new index key and assigning the value to it.

```
dict = {}
dict["color"] = "red"
print(dict)
```

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

```
dict = {}
dict["color"] = "red"
```
```

```
dict["name"] = "shivam"
for x in dict.values():
    print(x)
```

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
  }
}
for x in myfamily.keys():
    print(myfamily.get(x))
```

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

Ans. get() function is used to access value of a specified key. It also used to access an inner dictionary.

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

Ans. The items() function will return each item in a dictionary.

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

Ans. Removes the element with a specified key.

Q65. What is the use of popitems() function?

Ans. Remove the last item from the dictionary.

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
car.popitem()
print(car)
```

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

Ans. Returns a list containing the dictionary's key.

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

Ans. Returns a list containing the dictionary's values.

Q68. What are loops in Python?

Ans The loops are the variable in python to fulfill the looping needs. We can run single statement or set of statements repeatedly using a loop command.

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

Ans. There are three types of loops in python –

1. FOR loop

2. WHILE loop

3. NESTED loop

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

Ans. FOR loop is used only when the number of iterations is known but WHILE loop is used when the number of iteration is unknown.

If the condition is not mentioned in the FOR loop, then the loop iterates infinite number of times but in case of WHILE loop it will results in compilation error.

Q71. What is the use of continue statement?

Ans. With the continue statement we can stop the current iteration of the loop and continue with the next.

Q72. What is the use of break statement?

Ans. With the break statement we can stop the loop before it has looped through all items.

Q73. What is the use of pass statement?

Ans. Loops, functions cannot be empty but if you for some reason have a loop or function with no content then we can put pass statement to avoid getting an error.

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

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

Q75. How can you loop over a dictionary?

Ans. By using keys(), values() and items() we can loop over dictionary.

Coding problems

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

```
def fact(n):
    if n==0 or n==1:
        return 1
    return n*fact(n-1)
print(fact(5))
```

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

```
def check(p,r,t):
    si = (p*r*t)/100
    print(si)
p,r,t = map(int,input("Enter values: ").split())
check(p,r,t)
```

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

```
def check(p,r,t):
    amount = p*((1+r/100)**t)
    ci = amount - p
    print(ci)
p,r,t = map(int,input("Enter values: ").split())
check(p,r,t)
```

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

```
def check(n):
    if (n == 1):
        return 1
    flag = 1
    for i in range(2, n):
        if n % i == 0:
            flag = 0
    if flag ==1:
        return 1
    else:
        return 0



n = int(input("Enter a number: "))
if check(n) == 1:
    print("Prime number")
else:
    print("Not a prime number")
```

Q80. Write a Python program to check Armstrong Number.

```
def check(n):
    sum = 0
    n1 = n
```

```
    while n > 0:
        d = n % 10
        sum += d**3
        n = n//10
    if sum == n1:
        print("Armstrong number")
    else:
        print("Not an armstrong number")


n = int(input("Enter the number : "))
check(n)
```

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

```
def check(n):
    if n == 0 or n==1:
        return n
    else:
        return check(n-1)+check(n-2)


n = int(input("Enter the number: "))
print(check(n))
```

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

```
lst = list(map(int,input("Enter the numbers:
").split()))

lst[0],lst[len(lst)-1] = lst[len(lst)-1],lst[0]
print(lst)
```

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

```
lst = list(map(int,input("Enter the numbers:
").split()))

lst[0],lst[len(lst)-1] = lst[len(lst)-1],lst[0]
print(lst)
```

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

```
lst = list(map(int,input("Enter the numbers:
").split()))
n = int(input("Nth largest element--Enter n value:
"))
lst.sort()
print(lst[-n:])
```

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

```
lst = list(map(int,input("Enter the numbers:
").split()))
new_list = []
s = 0
for i in range(0,len(lst)):
    s += lst[i]
    new_list.append(s)
print(new_list)
```

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

```
s = input("Enter string: ")
if s == s[::-1]:
    print("Palindrome")
```

```
else:
    print("Not a Palindrome")
```

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

```
s = input("Enter string: ")
i = int(input("Enter the index number which u want
to remove: "))
x = s.replace(s[i],'')
print(x)
```

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

```
txt = "The best things in life are free!"
print("free" in txt)
```

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

```
txt = "The best things in life are free!"
k = int(input("Put the length which you want: "))
a = list(txt.split(" "))
new_list = []
for x in a:
    if len(x) > k:
        new_list.append(x)

print(new_list)
```

Q90. Write a Python program to extract unique dictionary values.

```
my_dict = {
    'hi':[5,3,8,0],
```

```
    'there':[22,51,63,77],
    'how':[7,0,22]
}
my_result = list(sorted({elem for val in
my_dict.values() for elem in val}))

print(my_result)
```

Q91. Write a Python program to merge two dictionary.

Ans. Using update()

```
a_dict = {"name":"shivam","gender":"male"}
b_dict = {"address":"India","Region":"North"}
a_dict.update(b_dict)
print(a_dict)
```

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}

```

```
lst = [('Sachin', 10), ('MSD', 7), ('Kohli', 18),
('Rohit', 45)]
d = dict(lst)
print(d)
```

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)]

```

```

```
lst = [9,5,6]
l1 = []
for x in lst:
    l1.append((x,x**3))
print(l1)
```

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)]

```

```

```
test_tuple1 = (7, 2)
test_tuple2 = (7, 8)
lst = []
for i in test_tuple1:
    for j in test_tuple2:
        tup = [i,j]
        lst.append(tuple(tup))

for i in test_tuple2:
    for j in test_tuple1:
        tup = [i,j]
        lst.append(tuple(tup))

print(lst)
```

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)]

```

```
lst = [('for', 24), ('Geeks', 8), ('Geeks', 30)]
sort_by_second = sorted(lst,key = lambda item:
item[1])
print(sort_by_second)
```

Q96. Write a python program to print below pattern.

```

*

* *

* * *

* * * *

* * * * *

```

```
for i in range(5):
    for j in range(i+1):
        print("* ",end=" ")
    print("")
```

Q97. Write a python program to print below pattern.

```

  *

 **

***
```

```
 ****

*****
```

```
for i in range(5):
    for j in range(1,5-i):
        print(" ",end=" ")
    for k in range(0,i+1):
        print("*",end=" ")
    print()
```

Q98. Write a python program to print below pattern.

```
  *

 **

 ***

****

*****
```

```
def pattern(n):
    k = 2 * n - 2
    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")
```

```
pattern(5)
```

Q99. Write a python program to print below pattern.

```
1

1 2

1 2 3

1 2 3 4

1 2 3 4 5
```

```
for i in range(1,6):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

Q100. Write a python program to print below pattern.

```
A

B B

C C C

D D D D

E E E E E
```

```
alpha = 65
for i in range(0,5):
    for j in range(0,i+1):
        print(chr(alpha+i),end=" ")
    print()
```