PYTHON MEGA ASSIGNMENT

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

Ans. In python string is a series of characters defined by either single or double quotation marks. Python does not have a character data type, a single character is simply a string with a length of 1.

s= ‘welcome to python’

print(s)

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

Ans. Individual characters of a String can be accessed by Indexing. Indexing of strings starts from 0 till n-1, where n is the size of string. Negative indexing helps to access characters from the back of the String,

s= ‘welcome to python’

print(s[0])

print(s[-1])

output

w

n

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

string = "Big Data iNeuron"
desired_output = "iNeuron"
Ans.

```
string1 = "Big Data iNeuron"
s=string1[9:16]
print(s)
```

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

string = "Big Data iNeuron"
desired_output = "norueNi"
Ans.

```
string1 = "Big Data iNeuron"
s = string1[-1:-8:-1]
print(s)
```

Q30. Reverse the string given in the above question.

Ans.

```
string1 = "Big Data iNeuron"
s = string1[::-1]
print(s)
```

Q31. How can you delete entire string at once?

Ans.

```
string1 = "Big Data iNeuron"
s = string1.replace(string1, '')
print(s)
```

Q32. What is escape sequence?

Ans. Escape sequences helps to include special characters in strings. To do this, simply add a backslash (\) before the character you want to escape.

Q33. How can you print the below string?

'iNeuron's Big Data Course'

Ans.

```
# method-1
string1 = 'iNeuron\'s Big Data Course'
print(string1)

# method-2
string1 = "iNeuron's Big Data Course"
print(string1)
```

Q34. What is a list in Python?

Ans. Lists are one among the built-in data types in python. They are sequential data type used to store multiple items.

Q35. How can you create a list in Python?

Ans. A list is created by placing elements inside square brackets [].

list1 = [1, “welcome”, 45.5, 2]

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

Ans. Elements in a list can be accessed by indexing.

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"]
p = lst[4][-1]
print(p)
```

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

Ans.

```
# method-1
lst1 = ["Welcome", "to", "Data", "course"]
print(len(lst1))

# method-2
count = 0
for i in lst1:
    count += 1
print(count)
```

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

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

Ans.

```
lst1 = ["Welcome", "to", "Data", "course"]
lst1.insert(3, "Big")
print(lst1)
```

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

Ans. Tuple is a collection of items separated by commas. Tuple is similar to list but it contains immutable objects. Lists have several built-in methods, whereas tuple does not have many built-in methods.

Q41. How can you create a tuple in Python?

Ans. A tuple is created by placing all the items (elements) inside parentheses (), separated by commas.

The empty tuple is written as two empty parentheses

tup1 = ()

Another tuple is:

Tup2 = ('physics', 'chemistry', 26, 45)

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. Tuples are immutable data types. One cannot add or remove elements from tuples. There's no append() or extend() methods for tuples. So you can do it by tuple concatenation.

```
# Adding name to tuple
t1 = (1, 2, 'car', 'hai')
t1_append  = t1+('Anu', )
print(t1_append)
```

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

Ans. There is no append() method for tuples, instead you can append tuples by concatenation.

```
# concatenating two tuples
t1 = (1, 2, 'car', 'hai')
t1_append  = t1+(2, 45, 'welcome')
print(t1_append)
```

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

Ans.

```
t1 = tuple(input("Enter elements separated by space: ").split())
print(type(t1))
count = 0
for i in t1:
    count += 1
print(count)
```

Q45. What are sets in Python?

Ans. Sets are used to store multiple items in a single variable. They are iterable, unordered and has no duplicate elements. There is no index attached to any element in a python set. Hence, they do not support any indexing or slicing operation.

Q46. How can you create a set?

Ans. Sets are written with curly brackets. set() function is used for type casting in Python

set1={"apple", "banana", "cherry"}
set2 = {1, 5, 7, 9, 3}

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

Ans.

```
set1 = {"apple", "banana", "cherry"}
set1.add('iNeuron')
print(set1)
```

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

Ans.

```
set1 = {"apple", "banana", "cherry"}
set2 = {1, 5, 7, 9, 3}
for items in set2:
    set1.add(items)
print("The set after adding multiple items: ", set1)
```

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

Ans. add() method used to add single value to a set, whereas update() method to add sequence values to a set. update() function, accepts only iterable sequences while add function accepts single argument.

Q50. What is clear() in sets?

Ans. Clear() method empties the whole set inplace.

Q51. What is frozen set?

Ans. Frozen sets in Python are immutable objects. It can be done with *frozenset().*The elements of the frozen set remain the same after creation.

Q52. How is frozen set different from set?

Ans. Frozen set is just an immutable version of a Python set. 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.

Ans. The union of two given sets A and B is a set which consists of all the elements of A and all the elements of B such that no element is repeated.

```
set1 = {"apple", "banana", "cherry", 5, 3, 25}
set2 = {1, 5, 7, 9, 3}
result = set1.union(set2)
print("set1 U set2: ",result)

#output is
set1 U set2:  {'apple', 1, 3, 5, 7, 9, 'banana', 'cherry', 25}
```

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

Ans. The intersection() method returns a set that contains the elements that are common between two or more sets.

```
set1 = {"apple", "banana", "cherry", 5, 3, 25}
set2 = {1, 5, 7, 9, 3}
result = set1.intersection(set2)
print("set1 & set2: ", result)

# output is
set1 & set2:  {3, 5}
```

Q55. What is dictionary in Python?

Ans. Python Dictionary is used to store the data in a key-value pair format. It is a collection which is ordered, changeable and do not allow duplicates. Dictionaries are written with curly brackets, and can be referred to by using the key name.

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

Ans. Dictionary in Python is a collection of key-value pairs, used to store data values like map, unlike other data types which hold only a single value as an element. The keys of dictionary can be of any data type. Keys are case sensitive, can't be repeated and must be immutable.

Q57. How can we declare a dictionary in Python?

Ans. Dictionary can also be created by the built-in function *dict()*. Each key is separated from its value by the colon (:).

Employee = {"Name": "manu", "Age": 27, "salary":25000,"Company":"xyz"}

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.

```
# method-1
dict1 = {"Name": "manu",
         "Age": 27,
         "salary":25000,
         "Company": "xyz"}
dict1["place"] = "Delhi"
# method-2
dict1.update(skills='python')
print(dict1)
```

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

Ans.

```
dict1 = {"Name": "manu",
         "Age": 27,
         "salary":25000,
         "Company": "xyz"}
value = dict1.values()
print(value)
```

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

Ans.

```
dict1={"Name":"manu",
         "Age":27,
         "salary":25000,
         "Company": "xyz"}

#creating nested dictionary (dict2 inside dict1)
dict1.update(dict2={'Name': 'John', 'Age': '27', 'Sex': 'Male'})
print(dict1)

# Accessing name from dict2
print(dict1['dict2']['Name'])

# Accessing all the values of nested dictionary dict2
value = dict1['dict2'].values()
print(value)

output
{'Name': 'manu', 'Age': 27, 'salary': 25000, 'Company': 'xyz', 'dict2': {'Name':
'John', 'Age': '27', 'Sex': 'Male'}}
John
dict_values(['John', '27', 'Male'])
```

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

Ans. The get() method returns the value for the specified key if the key is in the dictionary. If not, then it will return None (if get() is used with only one argument).

Syntax: *dictionary*.get(*keyname, default = none*)

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

Ans. Python dictionary method items() returns a list of dictionary's (key, value) tuple pairs.

Syntax: *dictionary.items()*

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

Ans. The pop() method removes the element at the specified index from the list and returns the removed item. If the index is not given, then the last element is popped out and removed.

Syntax: *list_name.pop(index)*

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

Ans. popitem() method removes the last inserted key-value pair from the dictionary and returns it as a tuple. It doesn't take any parameters.

Syntax: *dictionary.popitem()*

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

Ans. keys() method extracts the keys of the dictionary and returns the list of all the available keys as a view object.

Syntax: *dict.keys()*

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

Ans. Python dictionary method values() returns a list of all the values available in a given dictionary. This method doesn't take any parameters.

Syntax: *dictionary.values()*

Q68. What are loops in Python?

Ans. A loop is a sequence of instructions that is continously repeated until a certain condition is reached. So, we can run a 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 of loops are available in the Python programming language.

a) while loop
b) for loop
c) Nested loop

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

Ans.

while loop	**for loop**
• Repeats a statement or group of statements while a given condition is TRUE. • when the condition becomes false, the line immediately after the loop in the program is executed. • It tests the condition before executing the loop body. • ***Syntax:*** *while expression: statement(s)*	• A for loop is used for iterating over a sequence or other iterable objects. • This type of loop executes a code block multiple times and abbreviates the code that manages the loop variable. • ***Syntax:*** *for iterator_var in sequence: statements(s)*

Q71. What is the use of continue statement?

Ans. The continue keyword is used to end the current iteration in a loop and continues to the next iteration. i.e, when the continue statement is executed in a loop, the code inside the loop following the continue statement will be skipped for the current iteration and the next iteration of the loop will begin.

Q72. What is the use of break statement?

Ans. The break keyword is used to break out a loop. It is used to terminate the execution of the loop.

Q73. What is the use of pass statement?

Ans. The pass statement is a null statement which can be used as a placeholder for future code. This helps you to avoid getting an error when empty code is not allowed.

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

Ans. The Python range() function returns a sequence of numbers, in a given range starting from 0 by default, and increments by 1 (by default), and stops before a specified number. It can take maximum of three arguments.

Syntax: *range(start, stop, step)*

Q75. How can you loop over a dictionary?

Ans.

```
# Accessing keys using for loop
for keys in dict1:
    print(keys)

# Accessing values using for loop
for values in dict1.values():
    print(values)

# Accessing key-values as tuples using for loop
for kv in dict1.items():
    print(kv)

Output
Keys
Name
Age
salary
Company

values
manu
27
25000
Xyz

Key-value
('Name', 'manu')
('Age', 27)
('salary', 25000)
('Company', 'xyz')
```