Module 4: Data Structure

Assignment Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to work with the structure of the data. Tasks To Be Performed:

1. Create a list named 'myList' that has the following elements: 10, 20, 30, 'apple', True, 8.10: a. Now in the 'myList', append these values: 30, 40 b. After that, reverse the elements of the 'myList' and store that in 'reversedList'
2. Create a dictionary with key values as 1, 2, 3 and the values as 'data', 'information' and 'text': a. After that, eliminate the 'text' value from the dictionary b. Add 'features' in the dictionary c. Fetch the 'data' element from the dictionary and display it in the output
3. Create a tuple and add these elements 1, 2, 3, apple, mango in my_tuple.
4. Create another tuple named numeric_tuple consisting of only integer values 10, 20, 30, 40, 50: a. Find the minimum value from the numeric_tuple b. Concatenate my_tuple with numeric_tuple and store the result in r1 c. Duplicate the tuple named my_tuple 2 times and store that in 'newdupli'
5. Create 2 sets with the names set1 and set2, where set1 contains {1,2,3,4,5} and set2 contains {2,3,7,6,1} Perform the below operation: a. set1 union set2 b. set1 intersection set2 c. set1 difference set2

In [ ]:
```
import numpy as np
```

In [ ]:
```
##1,
x = np.arange(2, 11).reshape(3,3)
print(x)
```

In [ ]:
```
##2,
import numpy as np
a = [1, 2, 3, 4]
print("Original array")
print(a)
x = np.asfarray(a)
print("Array converted to a float type:")
print(x)
```

In [ ]:
```
##3,
import numpy as np
x = [10, 20, 30]
print("Original array:")
print(x)
x = np.append(x, [10, 20, 30, 40, 50, 60, 70, 80, 90])
print("After append values to the end of the array:")
print(x)
```

In [ ]:
```
##Create two NumPy arrays and add the elements of both the arrays and store the result in sumArray.

array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])

sumArray = np.add(array1, array2)

print(sumArray)
```

In [ ]:
```
##5.
array1 = np.arange(10,100,10).reshape(3,3)
print(array1)

first_row=array1[0]
print(first_row)

last_element=array1[-1][-1]
print(last_element)
```

Module 4: NumPy Assignment

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to use the NumPy package and do some tasks related to that. Tasks To Be Performed:

1. Create a 3x3 matrix array with values ranging from 2 to 10.
2. Create a NumPy array having user input values and convert the integer type to the float type of the elements of the array. For instance: Original array [1, 2, 3, 4] Array converted to a float type: [1. 2. 3. 4.]
3. Write a NumPy program to append values to the end of an array. For instance: Original array: [10, 20, 30] . After that, append values to the end of the array: [10 20 30 40 50 60 70 80 90]
4. Create two NumPy arrays and add the elements of both the arrays and store the result in sumArray.
5. Create a 3x3 array having values from 10-90 (interval of 10) and store that in array1 Perform the following tasks: a. Extract the 1st row from the array b. Extract the last element from the array

In [ ]:
```
import numpy as np
```

In [ ]:
```
##1,
x = np.arange(2, 11).reshape(3,3)
print(x)
```

In [ ]:
```
##2,
import numpy as np
a = [1, 2, 3, 4]
print("Original array")
print(a)
x = np.asfarray(a)
print("Array converted to a float type:")
print(x)
```

In [ ]:
```
##3.
import numpy as np
x = [10, 20, 30]
print("Original array:")
print(x)
x = np.append(x, [10, 20, 30, 40, 50, 60, 70, 80, 90])
print("After append values to the end of the array:")
print(x)
```

In [ ]:
```
##4,Create two NumPy arrays and add the elements of both the arrays and store the result in sumArray.

array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])

sumArray = np.add(array1, array2)

print(sumArray)
```

In [ ]:
```
##5.
array1 = np.arange(10,100,10).reshape(3,3)
print(array1)

first_row=array1[0]
print(first_row)

last_element=array1[-1][-1]
print(last_element)
```

Module 4: NumPy Case Study

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python Certification Course Problem Statement: You work in XYZ Company as a Python developer. The company officials want you to build a Python program. Tasks To Be Performed:

1. Create a function that takes dimensions as tuples e.g. (3, 3) and a numeric value and returns a NumPy array of the given dimension filled with the given value e.g.: solve((3, 3), 5) will return [[5, 5, 5], [5, 5, 5], [5, 5, 5]]
2. Create a method that takes n NumPy arrays of the same dimensions, sums them and returns the answer.
3. Given a 2 D Array of N X M Dimension, write a function that accepts this array as well as two numbers N and M. The method should return the top-left N X M sub matrix, e.g: [[1, 2, 3], [4, 5, 6], [7, 8, 9],] top_left_sub_matrix (matrix, 2, 2) -> should return: [[1, 2] [4, 5]] Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python Certification Course
4. Given a 2 D Array of N X M Dimension, write a function that accepts this array as well as two numbers N and M. The method should return the bottom-right N X M sub matrix, e.g: [[1, 2, 3], [4, 5, 6], [7, 8, 9],] sub_matrix(matrix, 1, 1) -> should return : (Keep in mind these arrays are zero indexed) [[5, 6] [8, 9]]
5. Given a 1 D NumPy Array. Write a function that accepts this array as parameters. The method should return a dictionary with 'mean' and 'std_dev' as key and array's mean and array's standard deviation as values: [1, 1, 1] solution(arr) -> should return : {'mean': 1.0, 'std_dev': 0.0}

In [ ]:
```
import numpy as np
##1,
def solve(dim_tuple,fillnumber):

    return np.full(dim_tuple,fillnumber)

x= solve((3,3),5)

print(x)
```

In [ ]:
```
##2,
a = np.array([1, 1, 1, 1, 1])
b = np.array([5, 6])
sumWithRoll(a, b, offset=1)
print(a)
```

In [ ]:
```
##3,
def top_left_sub_matrix(matrix, n, m):
    return [row[:m] for row in matrix[:n]]
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(top_left_sub_matrix(matrix, 2, 2))
```

In [ ]:
```
##4,
def sub_matrix(matrix, n, m):
    return [row[-m:] for row in matrix[-n:]]
sub_matrix(matrix, 2, 2)
```

In [ ]:
```
##5,
def solution(arr):
    return {'mean': np.mean(arr), 'std_dev': np.std(arr)}

arr = [1, 1, 1]
print(solution(arr))
```