Darshan UNIVERSITY
योग: कर्मसु कौशलम्

# Python Programming - 2301CS404

# Lab - 7 (Part-2)

## User Defined Function

### 12. Write a function to calculate the sum of the first element of each tuples inside the list.

In [1]:
```
def sumFirstElements(tuplesList):
    total = 0
    for t in tuplesList:
        total += t[0]
    return total

sampleTuplesList = [(4, 2), (7, 8), (1, 3), (9, 5)]
print(sumFirstElements(sampleTuplesList))
```

```
21
```

### 13. Write a function to get the name of the student based on the given rollno.

Example: Given dict1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'} find name of student whose rollno = 103

In [1]:
```
def getStudentName(rollno, dict1):
    for key in dict1:
        if key == rollno:
            return dict1[key]

dict1 = {101: 'Dhara', 102: 'Manu', 103: 'Hetvi', 104: 'Piyu'}
print(getStudentName(103, dict1))
```

```
Hetvi
```

## 14. Write a function to get the sum of the scores ending with zero.

Example : scores = [200, 456, 300, 100, 234, 678]

Ans = 200 + 300 + 100 = 600

In [3]:
```
def sumScoresEndingWithZero(scores):
    total = 0
    for score in scores:
        if score % 10 == 0:
            total += score
    return total

scores = [200, 456, 300, 100, 234, 678]
print(sumScoresEndingWithZero(scores))
```

```
600
```

## 15. Write a function to invert a given Dictionary.

hint: keys to values & values to keys

Before : {'a': 10, 'b':20, 'c':30, 'd':40}

After : {10:'a', 20:'b', 30:'c', 40:'d'}

In [4]:
```
def invertDictionary(originalDict):
    invertedDict = {}
    for key in originalDict:
        value = originalDict[key]
        invertedDict[value] = key
    return invertedDict

originalDict = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
print(invertDictionary(originalDict))
```

```
{10: 'a', 20: 'b', 30: 'c', 40: 'd'}
```

## 16. Write a function that returns the number of uppercase and lowercase letters in the given string.

example : Input : s1 = AbcDEfgh ,Ouptput : no_upper = 3, no_lower = 5

In [2]:
```
def countCase(s1):
    noUpper = 0
    noLower = 0
    for ch in s1:
        if ch >= 'A' and ch <= 'Z':
```

```
            noUpper += 1
        elif ch >= 'a' and ch <= 'z':
            noLower += 1
    return noUpper, noLower

s1 = "DharaMaruLMAO"
upperCount, lowerCount = countCase(s1)
print("no_upper =", upperCount)
print("no_lower =", lowerCount)
```

```
no_upper = 6
no_lower = 7
```

## 17.Write a lambda function to get smallest number from the given two numbers.

In [6]:
```
getSmallest = lambda a, b: a if a < b else b
print(getSmallest(10, 25))
```

```
10
```

## 18. For the given list of names of students, extract the names having more that 7 characters. Use filter().

In [4]:
```
names = ['Dhara', 'ZenishaPanchasara', 'Priya', 'Sonu', 'Bhavika', 'Khushi']
longNames = list(filter(lambda name: len(name) > 7, names))
print(longNames)
```

```
['ZenishaPanchasara']
```

## 19. For the given list of names of students, convert the first letter of all the names into uppercase. use map().

In [5]:
```
capitalizedNames = list(map(lambda name: name[0].upper() + name[1:], names))
print(capitalizedNames)
```

```
['Dhara', 'ZenishaPanchasara', 'Priya', 'Sonu', 'Bhavika', 'Khushi']
```

## 20. Write udfs to call the functions with following types of arguments:

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable Legngth Positional(*args) & variable length Keyword Arguments (**kwargs)
5. Keyword-Only & Positional Only Arguments

In [9]:
```
#Positional Arguments
```

```python
def showDetailspos(name, age):
    print("Positional Arguments Name:", name)
    print("Positional Arguments Age:", age)

showDetailspos("Dharaa", 20)

#Keyword Arguments
def showDetailskey(name, age):
    print("Keyword Arguments Name:", name)
    print("Keyword Arguments Age:", age)

showDetailskey(age=20, name="Dharaa")

#Default Arguments
def showDetailsdef(name, age=18):
    print("Name:", name)
    print("Age:", age)

showDetailsdef("Dharaa")

#Variable Length Positional
def totalMarksvlp(*marks):
    total = 0
    for m in marks:
        total += m
    print("Variable Length Positional Total Marks:", total)

totalMarksvlp(70, 80, 90, 85)

#Variable length keywords
def studentInfovlk(**kwargs):
    for key in kwargs:
        print("Variable length keywords" + key + ":", kwargs[key])

studentInfovlk(name="Dharaa", age=20, grade="A")

#Keyword only
def showResultko(name, *, grade, marks):
    print("Keyword only Name:", name)
    print("Keyword only Grade:", grade)
    print("Keyword only Marks:", marks)

showResultko("Dharaa", grade="A", marks=92)

#Positional Only
def showStudentpo(name, age, /):
    print("Positional Only Name:", name)
    print("Positional Only Age:", age)

showStudentpo("Dharaa", 20)
```

```
Positional Arguments Name: Dharaa
Positional Arguments Age: 20
Keyword Arguments Name: Dharaa
Keyword Arguments Age: 20
Name: Dharaa
Age: 18
Variable Length Positional Total Marks: 325
Variable length keywordsname: Dharaa
Variable length keywordsage: 20
Variable length keywordsgrade: A
Keyword only Name: Dharaa
Keyword only Grade: A
Keyword only Marks: 92
Positional Only Name: Dharaa
Positional Only Age: 20
```