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 [24]:
```
t1 = [(10, 456), (20, 100), (30, 678)]
def sumoffirstelement(t1):
    sum=0
    for i in t1:
        sum += i[0]
    return sum
sumoffirstelement(t1)
```

Out[24]: 60

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 [12]:
```
s1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'}
k = 103
def studentname(s1, k):
    return s1.get(k)
print(studentname(s1, k))
```

```
Jay
```

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 [22]:
```
l2 = [200, 456, 300, 100, 234, 678]
def sumscore(l2):
    total = 0
    for i in l2:
        if i%10==0:
            total += i
    return total
n = sumscore(l2)
print(n)
```

```
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 [33]:
```
s1 = {'a': 10, 'b':20, 'c':30, 'd':40,'a':10,'b':20}
def InvertDict(s1):
    s2 = {}
    for i,j in s1.items():
        s2[j]=i
    return s2
InvertDict(s1)

# def InvertDict():
#     i=9
#     j=10
#     temp = i
#     i = j
#     j = temp
#     return i,j
# i,j=InvertDict()
# print("i , j",i,j)
```

Out[33]:
```
{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 [35]:
```
s1="AbcDEfgh"
def ul():
    lower=0
    upper=0
    for i in s1:
        if i.islower():
            upper+=1
        else:
            lower+=1
    print("UpperCase:",upper)
```

```
    print("LowerCase:",lower)
ul()
```

```
UpperCase: 5
LowerCase: 3
```

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

In [40]:
```
s1 = lambda a,b : a if a < b else b
s1(32,200)
```

Out[40]: 32

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

In [42]:
```
l1 = ["abcdef","adhbaahd","qjwkdbkp","iuqwd"]
student = list(filter(lambda x: len(x) > 7, l1))
student
```

Out[42]: ['adhbaahd', 'qjwkdbkp']

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

In [48]:
```
l1 = ["abc","xyz","pqr","klm"]
student = list(map(lambda x: x[0].upper() + x[1:], l1))
student
```

Out[48]: ['Abc', 'Xyz', 'Pqr', 'Klm']

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 [1]:
```
#Positional Arguments
def demo(name, age):
    print(f"Hello {name}, you are {age} years old.")

demo("Nikhil", 22)
```

```
Hello Nikhil, you are 22 years old.
```

In [2]:
```
#Keyword Arguments
def demo2(name, age):
    print(f"Hello {name}, you are {age} years old.")
demo2(age=20, name="abc")
```

```
Hello abc, you are 20 years old.
```

In [3]:
```python
#Default Arguments
def demo3(name, age=18):
    print(f"Hello {name}, you are {age} years old.")
demo3("xyz")
demo3("abc", 21)
```

```
Hello xyz, you are 18 years old.
Hello abc, you are 21 years old.
```

In [4]:
```python
#Variable Legngth Positional(*args)
def add(*args):
    total = sum(args)
    print("Total:", total)
add(10, 20, 30)
add(5, 15)
```

```
Total: 60
Total: 20
```

In [6]:
```python
# variable length Keyword Arguments (**kwargs)
def show_details(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")
show_details(name="aabc", age=25, city="jamnagar")
```

```
name: aabc
age: 25
city: jamnagar
```

In [7]:
```python
#Keyword-Only & Positional Only Arguments
def details(id, /, name, *, age):
    print(f"ID: {id}, Name: {name}, Age: {age}")
details(101, "abc", age=25)
```

```
ID: 101, Name: abc, Age: 25
```

In [ ]:
```python

```