

# Python Programming - Lab - 8

March 11, 2025

Python Programming - 2301CS404

Lab - 8

OM BHUT | 23010101033 | 122

## 1 User Defined Function

**1.0.1 01) Write a function to calculate BMI given mass and height. (BMI = mass/h\*\*2)**

```
[ ]: def calculateBmi(mass,height):  
      return mass/height**2
```

**1.0.2 02) Write a function that add first n numbers.**

```
[3]: def addFirstN(n : int):  
      return n*(n+1)/2  
addFirstN(50)
```

[3]: 1275.0

**1.0.3 03) Write a function that returns 1 if the given number is Prime or 0 otherwise.**

```
[6]: def checkPrime(n):  
      if(n==2 or n==1 or n==0 or n==4):  
          return 0  
      for i in range(2,n//2):  
          if n%i == 0:  
              return 0  
      return 1  
  
print(checkPrime(8))
```

0

1.0.4 04) Write a function that returns the list of Prime numbers between given two numbers.

```
[7]: def primeNumbersList(a,b):  
    ans = []  
    for i in range(a,b+1):  
        if (checkPrime(i) == 1):  
            ans.append(i)  
    return ans  
print(primeNumbersList(2,50))
```

[3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

1.0.5 05) Write a function that returns True if the given string is Palindrome or False otherwise.

```
[10]: def checkPalindrome(s1):  
    return s1 == s1[::-1]  
print(checkPalindrome("abab"))
```

False

1.0.6 06) Write a function that returns the sum of all the elements of the list.

```
[ ]: def sumOfAllElements(l1):  
    return sum(l1)
```

1.0.7 07) Write a function to calculate the sum of the first element of each tuples inside the list.

```
[1]: def sumOfFirstElementsOfTuple(l1):  
    sum = 0  
    for t in l1:  
        sum += t[0]  
    return sum  
list1 = [(1,2,3),(3,4,1),(4,8,2)]  
print(sumOfFirstElementsOfTuple(list1))
```

8

1.0.8 08) Write a recursive function to find nth term of Fibonacci Series.

```
[8]: def fibonacciSeries(n):  
    first = 0  
    second = 1  
    for i in range(n):  
        print(first,end=" ")  
        nextVal = first+second  
        first = second
```

```

        second = nextVal

def fibonacciSeriesUsingRecursion(n):
    if n==0 or n==1:
        return n
    else:
        return fibonacciSeriesUsingRecursion(n - 1) + \
        fibonacciSeriesUsingRecursion(n - 2)

# fibonacciSeries(30)
fibonacciSeriesUsingRecursion(6)

```

[8]: 8

1.0.9 09) 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

```

[12]: def giveNameFromRollNumber(dict1 : dict,rollNo):
        return dict1[rollNo]

dict1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'}
print(giveNameFromRollNumber(dict1,102))

```

Rahul

1.0.10 10) 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

```

[14]: def sumOfEndingScores(l1):
        sum=0
        for i in l1:
            sum += i%10
        return sum
scores = [200, 456, 300, 100, 234, 678]
print(sumOfEndingScores(scores))

```

18

1.0.11 11) 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'}

```
[4]: def invertDictionary(d1 : dict):
    invertedDictionary = {value:key for key,value in d1.items()}
    return invertedDictionary

d1 = {
    0:"om",
    1:"yash",
    2:"raj"
}
print(invertDictionary(d1))
```

```
{'om': 0, 'yash': 1, 'raj': 2}
```

1.0.12 12) Write a function to check whether the given string is Pangram or not.  
hint: Pangram is a string containing all the characters a-z atleast once.

“the quick brown fox jumps over the lazy dog” is a Pangram string.

```
[12]: def checkPanagram(s : str):
    set1 = {chr(i) for i in range(97,123)}
    set2 = set(s)
    if " " in set2:
        set2.remove(' ')
    return set1 == set2

s = "the quick brown fox jumps over the lazy dog"
# s="hello"
print(checkPanagram(s))
```

True

1.0.13 13) Write a function that returns the number of uppercase and lowercase letters in the given string.

example : Input : s1 = AbcDEfgh ,Ouputput : no\_upper = 3, no\_lower = 5

```
[14]: def countUpperLower(s : str):
    noOfUpper = 0
    noOfLower = 0
    for i in s:
        if i.isupper():
            noOfUpper+=1
        else:
            noOfLower+=1
    return {
        "upper":noOfUpper,
        "lower":noOfLower
    }
```

```
print(countUpperLower("AbcDEfgh"))
```

```
{'upper': 3, 'lower': 5}
```

**1.0.14 14) Write a lambda function to get smallest number from the given two numbers.**

```
[15]: x = lambda a,b : min(a,b)
      print(x(5,6))
```

```
5
```

**1.0.15 15) For the given list of names of students, extract the names having more than 7 characters. Use filter().**

```
[18]: l1 = ["jayesh","rajnikant","raftaar"]
      l2 = filter(lambda x: len(x)>=7,l1)
      print(list(l2))
```

```
['rajnikant', 'raftaar']
```

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

```
[22]: l1 = ["jayesh","rajnikant","raftaar","Ok","hello"]
      l2 = map(lambda x : x[0].upper() + x[1::] , l1)
      print(list(l2))
```

```
['Jayesh', 'Rajnikant', 'Raftaar', 'Ok', 'Hello']
```

**1.0.17 17) Write udfs to call the functions with following types of arguments:**

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

```
[3]: def positionalArguments(a,b):
      return a+b
      def defaultArguments(a=1,b=1):
          return a+b
      def variableLengthPositional(a,*args):
          print(a,args)
      def keywordArguments(a,**kwargs):
          print(a,kwars)
      def keywordOnly(*,a,b):
          return a+b
```

```
def positionalArgumentsOnly(a,b,/):  
    return a+b  
positionalArguments(1,2)  
positionalArguments(b=2,a=1)  
defaultArguments()  
variableLengthPositional(10,20,30,40)  
keywordArguments(10,b=20,c=30)  
keywordOnly(a=10,b=20)  
positionalArgumentsOnly(10,20)
```

[3]: 30

[ ]: