Darshan UNIVERSITY

योग: कर्मसु कौशलम्

# Python Programming - 2301CS404

## Lab - 8

Dhol Namra

Enroll:23010101407

11-01-2025

## User Defined Function

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

In [1]:
```
def BMI (mass,h):
        return mass/(h**2)

BMI(50,5)
```

Out[1]: 2.0

### 02) Write a function that add first n numbers.

In [3]:
```
def sum(n):

    return (n*(n+1)/2)
sum(3)
```

Out[3]: 6.0

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

In [39]:
```
number = int(input("Enter a number to check: "))
def is_prime(number):
  if number >1:
    for i in range(2,number):
      if (number % i) == 0:
        return False
      else:
        return True
```

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

In [ ]:
```
number = int(input("Enter a number to check: "))
def is_prime(number):
  if number >1:
    for i in range(2,number):
      if (number % i) == 0:
        return False
      else:
        return True
```

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

In [7]:
```
number = input("Enter a number to check: ")
def is_palindrom(number):
    if (number == number[::-1]):
        print("The string is a palindrome.")
    else:
        print("The string is not a palindrome.")
is_palindrom(number)
```

```
The string is not a palindrome.
```

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

In [20]:
```
def sum_of_list(numbers):
    return sum(numbers)

numbers = [1, 2, 3, 4, 5]
print(f"The sum of the list is: {sum_of_list(numbers)}")
```

```
The sum of the list is: 15
```

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

In [26]:
```
def sum_of_first_elements(tuple_list):
    return sum(t[0] for t in tuple_list)
```

```
tuples = [(1, 2), (3, 4), (10, 6)]
print(f"The sum of the first elements is: {sum_of_first_elements(tuples)}")
```

```
The sum of the first elements is: 14
```

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

In [30]:
```
def fibonacci(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

fibonacci(4)
```

Out[30]: 3

## 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

In [34]:
```
def get_student_name(rollno, student_dict):
    return student_dict.get(rollno, "Roll number not found")

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

```
Jay
```

## 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

In [41]:
```
def sum_scores_ending_with_zero(scores):
    total = 0
    for score in scores:
        if score % 10 == 0:
            total += score
    return total
print(sum_scores_ending_with_zero([200, 456, 300, 100, 234, 678]))
```

```
600
```

## 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'}

In [43]:
```
def invert_dictionary(d):
    """
    Invert keys and values in a dictionary.

    :param d: Dictionary to invert
    :return: Inverted dictionary
    """
    return {v: k for k, v in d.items()}
invert_dictionary({'a':10,'b':20})
```

Out[43]:
```
{10: 'a', 20: 'b'}
```

## 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 atlest once.

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

In [ ]:
```
def is_pangram(s):

    return set("abcdefghijklmnopqrstuvwxyz").issubset(s.lower())
is_pangram('the quick brown fox jumps over the lazy dog')
```

## 13) 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 [ ]:
```
s1  = AbcDEfgh
def
```

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

In [ ]:
```
x=2
y=5
smallest = lambda x, y: x if x < y else y

print(smallest(x,y))
```

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

In [ ]:
```
def filter_long_names(names):

    return list(filter(lambda name: len(name) > 7, names))
filter_long_names(['harry','sejal','marko jansen'])
```

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

In [1]:
```
students = ["alice", "bob", "charlie", "dave", "eve"]

capitalized_students = list(map(lambda name: name.capitalize(), students))

print(capitalized_students)
```

```
['Alice', 'Bob', 'Charlie', 'Dave', 'Eve']
```

## 17) 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 [3]:
```
def pos_args(name, age):
    print(name, age)

pos_args("Alice", 25)


def kw_args(name, age):
    print(name, age)

kw_args(age=30, name="Bob")


def default_args(name, age=20):
    print(name, age)

default_args("Charlie")
default_args("David", 35)


def var_pos_args(*args):
    print(args)

var_pos_args(10, 20, 30)


def var_kw_args(**kwargs):
    print(kwargs)

var_kw_args(name="Eve", age=22)


def kw_only(*, name, age):
    print(name, age)

kw_only(name="Frank", age=28)
```

```
def pos_only(name, age, /):
    print(name, age)

pos_only("Grace", 24)
```

```
Alice 25
Bob 30
Charlie 20
David 35
(10, 20, 30)
{'name': 'Eve', 'age': 22}
Frank 28
Grace 24
```

In [ ]: