python-programming-lab-8

March 13, 2025

Python Programming - 2301CS404

Gohel Neel

Enrollnment No. : 23010101089

Roll No. 340

Date: 13-01-2025

Lab - 8

1 User Defined Function

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

[2]:
```
mass = int(input("Enter Mass: "))
h = int(input("Enter height: "))

def BMI(mass , h):
    return mass/h**2
BMI(mass  ,  h )
```

```
Enter Mass:  50
Enter height:  123
```

[2]:
```
0.003304911097891467
```

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

[11]:
```
def add(n):
    ans = 0
    for i in range(0,n+1):
        ans = ans + i
    return ans

n = int(input("Enter n"))
add(n)
```

```
Enter n 2
```

[11]:
```
3
```

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

[18]:
```
def is_prime(n):
    for i in range(2,n):
        if n % i == 0:
            return 0
    return 1
n = int(input("Enter number: "))
is_prime(n)
```

```
Enter number:  5
```

[18]:
```
1
```

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

[20]:
```
def prime_in_range(start, end):
    l1 = []
    for n in range(start, end + 1):
        if n > 1:
            for i in range(2, n):
                if n % i == 0:
                    break
            else:
                l1.append(n)
    return l1

start = int(input("Enter the starting number: "))
end = int(input("Enter the ending number: "))
print(prime_in_range(start, end))
```

```
Enter the starting number:  1
Enter the ending number:  4
```

```
[2, 3]
```

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

[22]:
```
def is_palindrome(s):
    return s == s[::-1]

s = input("Enter a string: ")
```

```
print(is_palindrome(s))
```

```
Enter a string:  121
```

```
True
```

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

[23]:
```
def sum_of_list(lst):
    return sum(lst)

lst = list(map(int, input("Enter numbers separated by space: ").split()))
print(sum_of_list(lst))
```

```
Enter numbers separated by space:  1 2 3 4 5
```

```
15
```

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

[ ]:
```
def sum_first_elements(lst):
    return sum(t[0] for t in lst)

lst = [(1, 2), (3, 4), (5, 6)]
print(sum_first_elements(lst))
```

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

[25]:
```
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

n = int(input("Enter the term number: "))
print(fibonacci(n))
```

```
Enter the term number:  4
```

```
3
```

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

[26]:
```
def get_student_name(rollno):
    students = {
        101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'
    }
    return students.get(rollno, "Student not found")

rollno = int(input("Enter roll number: "))
print(get_student_name(rollno))
```

```
Enter roll number:  101
```

```
Ajay
```

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

[27]:
```python
def sum_scores_ending_with_zero(scores):
    return sum(score for score in scores if score % 10 == 0)

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

```
600
```

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’}

[38]:
```python
def invert_dict(d):
    return {v: k for k, v in d.items()}

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

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

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 atlest once.

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

[6]:
```python
def is_pangram(s):
    s = s.lower()
    alphabet_set = set('abcdefghijklmnopqrstuvwxyz')
    string_set = set(char for char in s if char.isalpha())
    return string_set == alphabet_set
```

```
s = input("Enter a string: ")
is_pangram(s)
```

```
Enter a string:  the quick brown fox jumps over the lazy dog
```

```
[6]: 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 ,Ouptput : no_upper = 3, no_lower = 5

```
[9]: def count_case(s):
         upper = sum(1 for char in s if char.isupper())
         lower = sum(1 for char in s if char.islower())
         return upper, lower

     s = "AbcDEfgh"
     upper, lower = count_case(s)
     print(f"no_upper = {no_upper}, no_lower = {no_lower}")
```

```
no_upper = 3, no_lower = 5
```

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

```
[37]: smallest = lambda a, b: a if a < b else b

      n1 = int(input("Enter first number: "))
      n2 = int(input("Enter second number: "))
      print(smallest(n1, n2))
```

```
Enter first number:  1
Enter second number:  5

1
```

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

```
[1]: name = ["abc" , "abcdefghi" , "abcfdretjdbsnkwlsngdl" , "nenk"]
     ans = list(filter(lambda i:len(i)>7 , name))
     print(ans)
```

```
['abcdefghi', 'abcfdretjdbsnkwlsngdl']
```

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

[2]:
```python
def nameUpper(s):
    return s.title()
name = ["abc" , "abcdefghi" , "abcfdretjdbsnkwlsngdl" , "nenk"]
ans = list(map(nameUpper , name))
print(ans)
```

```
['Abc', 'Abcdefghi', 'Abcfdretjdbsnkwlsngdl', 'Nenk']
```

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 Legngth Positional(*args) & variable length Keyword Arguments (**kwargs)
5. Keyword-Only & Positional Only Arguments

[3]:
```python
# Positional
def add(n1,n2):
    return n1+n2
add(5,3)

# Keyword
def add(n1,n2):
    return n1+n2
add(n2 = 5 , n1 = 10)

# Default
def add(n1,n2 = 2):
    return n1+n2
add(5)

# Variable Legngth Positional
def add(n1 , *n):
    sum = n1
    for i in n:
        sum = sum + i
    print(sum)
add(1,2,3,4,5)
```

```
15
```

[ ]:
```python

```