bcs0fnfpw

March 12, 2025

Python Programming - 2301CS404

Khushi Patel| 23010101202 | 17-1-25

Lab - 8

1 User Defined Function

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

[44]:

```
mass=float(input("enter mass"))
height=float(input("enter height"))

def bmi(mass,height):
    print(mass/height**2)
bmi(mass,height)
```

```
enter mass 51
enter height 1.66
```

```
18.507766003774133
```

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

[17]:

```
def addn(n):
    sumn=0;
    for i in range(0,n+1):
        sumn=sumn+i
    print(sumn)
addn(10)
```

```
55
```

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

[1]:

```
def isPrime(n):
    flag=True;
    for i in range(2,n-1):
        if(n%i==0):
            flag=False;
    if(flag==True):
        return 1;
    else:
        return 0;

isPrime(7)
# isPrime(6)
```

[1]:

```
1
```

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

[6]:

```
def PrimeBetween(n1,n2):
    l1=[]
    for i in range(n1,n2+1):
        flag=True
        for j in range(2,i):
            if(i%j==0):
                flag=False;
                break;
        if(flag==True):
            l1.append(i)
    print(l1)
PrimeBetween(1,10)
```

```
[1, 2, 3, 5, 7]
```

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

[9]:

```
def isPalindrome(s):
    if(s==s[::-1]):
        print('Palindrome');
    else:
        print('not palindrome')
isPalindrome('khuhk')
```

```
Palindrome
```

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

[16]:

```
def sumlist(a):
    suma=0;
    for i in a:
        suma=suma+i;
    print(suma)
sumlist([1,5,6,2,3,8,2])
```

```
27
```

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

[32]:

```
def sumtuple(a):
    suma=0;
    for i in a:
            suma=suma+i[0];
    print(suma)

sumtuple([(1,5),(6,2),(3,8),(2,5)])
```

```
12
```

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

[10]:

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

num = int(input("Enter number:"))
print(f"Fibonacci sequence till {num}th term is ", end="")
for i in range(num):
    print(fibonacci(i),end=",")
```

```
Enter number: 20
```

```
Fibonacci sequence till 20th term is
0,0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,
```

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

[69]:

```
d={101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'}
def findkey(index,dictonary):
    return dictonary.get(index)
findkey(103,d)
```

[69]: 'Jay'

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

[36]:
```
scores = [200, 456, 300, 100, 234, 678]
def addzero(scores):
    sumn=0;
    for i in scores:
        if(i%10==0):
            sumn=sumn+i
    print(sumn)
addzero(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’}

[16]:
```
dict={'a': 10, 'b':20, 'c':30, 'd':40}
def alt_key(dict):
 return {v: k for k, v in dict.items()}

alt_key(dict)
```

[16]: {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.

[7]:
```
def isPanagram(string):
    string=string.lower()
    for i in range(97,123):
        if(chr(i) not in string):
            print("not panagram");
            return
    print("panagram")

isPanagram("the quick brown fox jumps over the lazy dog")
```

```
panagram
```

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

[10]:
```
def countcase(string):
    no_lower=0;
    no_upper=0;
    for i in string:
        if i.isupper()==True:
            no_upper+=1
        else:
            no_lower+=1
    print(no_upper);
    print(no_lower);
countcase('AbcDEfgh')
```

```
3
5
```

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

[19]:
```
print((lambda x,y:x if (x<y) else y)(2,3))
```

```
2
```

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

[32]:
```
a = ["Matthew", "John", "Samuel", "Jessica", "Alice", "Charlie", "David",␣
 ↪"Emily", "Andrew", "Sophia"]
def length_7(list):
    return len(list)>=7

b=filter(length_7,a)
print(list(b))
```

```
['Matthew', 'Jessica', 'Charlie']
```

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

[64]:
```
a=['matthew', 'john', 'samuel', 'jessica', 'alice', 'charlie', 'david',␣
 ↪'emily', 'andrew', 'sophia']
def capital(list):
    return list.capitalize()
```

```
b=map(capital,a)
print(list(b))
```

```
['Matthew', 'John', 'Samuel', 'Jessica', 'Alice', 'Charlie', 'David', 'Emily',
'Andrew', 'Sophia']
```

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

[57]:

```
# keyword argument
def si(p,r,t):
    return (p*r*t)/100
si(p=1000,r=10,t=2)

# positional argument
def si_key(p,r,t,/):
    return (p*r*t)/100
si_key(1000,2,2)

# default
def si_def(p,t,r=10):
    return(p*r*t)/100
si_def(1000,2)

# variable length positional
def si_key(*args):
    print(args)
si_key(1000,2,3)

# variable length keyword
def si_key_arg(**kwargs):
    print(kwargs)
si_key_arg(a=10)
```

```
(1000, 2, 3)
{'a': 10}
```