4MM013 Python Wrokshop-2
Feb-28, 2024

In this workshop you will practice packaging your code into functions, passing values in arguments and returning results.

Rewrite the following examples on Jupyter Notebook (preferred). Test the code and ensure it runs correctly, and explain what these code do through the comments.

Please write your name and University Id in the first cell of your workshop file.

Functions:

Example 1: The Python math module contains trig functions but they only work in radians. Write a function called deg_*sin* that prompts the user for an input in degree, converts it into radian and calculates the sin value of the given input.

Approach 1:

```
## -- these are the components we need from the math library
from math import pi, sin
    ## -- a function that prompts user for an input converts it in radian and
    calculates sin value
def deg_sin():
        x = float(input("Please input a number "))
        answer = sin(pi*x/180)
        print("result = {}".format(answer))
deg_sin()                    ## -- call the function
```

Approach 2:

```
from math import pi, sin
def deg_sin(x):
        answer = sin(pi*x/180)
        print("result = {}".format(answer))
inp = float(input("Please input a number "))
deg_sin(inp)
```

Approach 3:

```
from math import pi, sin

def deg_sin(x):

        return sin(pi*x/180)

inp = float(input("Please input a number "))

out = deg_sin(inp)

print("result = {}".format(out))
```

Question 1: Which of the above approach would you prefer to write a function and why?

Exercise:

1. Write a Python function that takes a list and returns a new list with distinct elements from the first list.
   Sample List : [1,2,3,3,3,3,4,5]
   Unique List : [1, 2, 3, 4, 5]
2. Write a Python function that takes a number as a parameter and checks whether the number is prime or not.
3. Write a program to print twin primes less than 1000. If two consecutive odd numbers are both prime then they are known as twin primes.
4. A positive integer is called a perfect number if it is equal to the sum of all of its divisors, including 1 but excluding the number itself. For example, 6 = 1 + 2 + 3. Write a function sumOfDivisors that takes a positive integer n and returns the sum of all its divisors (excluding n).
5. The smallest perfect number is 6. Write a program that finds and prints out the next perfect number. Hint: You should be able your sumOfDiviors function from your solution to previous question.
6. Write the python program to find the mean median from the following list:
   Marks = [20, 30, 35, 55, 70, 80, 90] Do not use the library function if there is one.
7. Write a function that converts a decimal number to binary number.
8. Write a function prodDigits() that inputs a number and returns the product of digits of that number.

Example 2: Lets Create a Simple function to add 2 numbers

```
def add(a,b):

   return a+b

add(2,3)
```

```
#instead of two I want to add, let's say 10 numbers

def mul(a,b,c,d,e,f,g,h,i,j):

    return a+b+c+d+e+f+g+h+i+j

mul(1,2,3,4,5,6,7,8,9,10)

# is this the efficient way to add 10 numbers? What if there are 100s of numbers?
```

Args(Variable Argument)

- ✓ Allows us to pass a variable number of non-keyword arguments to a function.
- ✓ This is used when you want to create a function but you yourself are unsure how many inputs the user will give, to handle that situation args is used.

Example 3:

```
def add(*args):

"""1. Function where args is used to add all the numbers given by the user

2. Asterik(*) here means number of variable, how it works is that asterik(*) creates a tuple and adds all the inputs in the tuple,

 which is accessed using loop"""

    sum = 0

    for i in args:

        sum += i

          print(args) # This is to check what args contains, as explained above it will be a   tuple where all the items will be listed

    return sum

print(add.__doc__) ## Access the documentation of the function

result = add(1,2,3,4,5,6,7,8,9,10) # No matter the number of argument the logic to add all the number still works

print(f"The addition is {result}")
```

Kwargs(Keyword Argument)

1. Allows us to pass any number of keyword arguments
2. Keyword arguments mean that they contain a key-value pair, like a python dictionary

```
# Basically its the same as args but we keep 2 asterik(**) which converts to dictionary


# Printing country with its capital
```

```
def capitals(**kwargs):

    print(kwargs)

    print()

    for (key, value) in kwargs.items():

        print(key, "->", value)

capitals(Nepal="Ktm", India="Delhi", China="Beijing", France="Paris")

# Write the output of the given function?
```

Exercise:

9. Write a function that will take in an unknown number of arguments and multiply all of them together and run the function for these 2 sets of numbers:
    a. 1,2,3,4,5
    b. 12,13,14

Set

A Set in Python programming is an unordered, *unchangeable**, and *unindexed* collection data type that is inerrable, mutable and has no duplicate elements.

*Note: Set items are unchangeable, but you can remove items and add new items.

Example 3:

Perform the following operations on given sets.

```
set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

set3={10,20,30}
```

a. Find the length of set1
b. Find the sum of elements in set 2
c. Union
d. Intersection
e. set1-set2
f. set1-set2
g. Check whether set3 is subset of the given sets
h. If U={1,2,3,…,50} then find complement of set1
i. Check if 15 is present in set1
j. Add 60 to the first set
k. Remove 60 from the second set

l. Remove all elements from set3
m. Print the maximum and minimum value in set 2