

# Day # 04 - Functions

Aim:

- Understand functions and how to define and call them.

```
In [ ]: #function to get dtype(int) input from user...

def get_input(data_type, input_label, error_label):
    while True:
        try:
            return data_type(input(input_label))
        except ValueError:
            print(error_label)
```

```
In [ ]: # function to get input list from user...

def get_list(data_type, input_label, error_label="Enter a valid entry..", stopping_character="q"):
    func_list = []

    while True:
        if data_type == "mix":
            i = input(input_label)
        else:
            i = get_input(data_type, input_label, error_label)

        if i != stopping_character:
            func_list.append(i)
        else:
            break

    return func_list
```

## Example Questions

Q1. Write a function to calculate the area of the circle given its radius.

```
In [ ]: import math

def cal_circle_area(radius):
    return math.pi * radius**2
```

```
In [ ]: e1_input = get_input(int, "Enter the radius of the circle", "Enter a valid number")
e1_ans = cal_circle_area(e1_input)

print(f"area of the circle is: {e1_ans}")

area of the circle is: 28.274333882308138
```

## Q2. Create a function to check if a number is prime.

```
In [ ]: def prime_number(number):  
        prime = True  
  
        for i in range(2, number):  
            if number % i == 0:  
                prime = False  
  
        return prime
```

```
In [ ]: e2_input = get_input(int, "Enter a number to check if it is prime", "Enter a va  
prime = prime_number(e2_input)  
  
print(f"{e2_input} is a prime number.") if prime else print(f"{e2_input} is not  
7 is a prime number.
```

## Q3. Implement a function that reverses a given string.

```
In [ ]: def reverse_string(string):  
        return string[::-1]
```

```
In [ ]: e3_input = input("Enter a string: ")  
  
print("Input string      : ", e3_input)  
print("reverse string    : ", reverse_string(e3_input))  
  
Input string      : HelloWorld!  
reverse string    : !dlroWolleH
```

## Practice Questions...

### Q1. given a list of numbers, creat function to find the sum of all positive numbers.

```
In [ ]: def find_positive_sum(data):  
        sum = 0  
  
        for i in data:  
            if i >= 0:  
                sum += i  
  
        return sum
```

```
In [ ]: q1_input = get_list(int, "Enter a numeric list, -1 to stop it.: ")  
q1_ans = find_positive_sum(q1_input)  
  
print(f"Input list: {q1_input}")  
print(f"Sum of all the positive numbers: {q1_ans}")  
  
Input list: [3, -2, 5, 2, -7, -9, 10]  
Sum of all the positive numbers: 20
```

Q2. Write a function to check if the given function if palindrome.

```
In [ ]: def palindrome(string):  
        reverse = string[::-1]  
  
        return True if reverse == string else False
```

```
In [ ]: q2_input = input("Enter a string").lower()  
q2_ans = palindrome(q2_input)  
print("Input stirng: ", q2_input)  
  
print("This is a palindrome stirng") if q2_ans else print("This is not a palind")  
  
Input stirng:  saas  
This is a palindrome stirng
```

Q3. Implement a function that returns the factorial of a given number using recursion.

```
In [ ]: def factorial(number):  
  
        if number == 0:  
            return 1  
  
        else:  
            return number * factorial(number - 1)
```

```
In [ ]: q3_input = get_input(int, "Enter a number: ", "Enter a valid number...")  
q3_ans = factorial(q3_input)  
  
print(f"Factorial of the number {q3_input}:  {q3_ans}")  
  
Factorial of the number 5:  120
```

Q4. Create a function to find the square of each element in a given list.

```
In [ ]: def squared_list(list):  
        return [x**2 for x in list]
```

```
In [ ]: q4_input = get_list(int, "Enter a number: ")  
q4_ans = squared_list(q4_input)  
  
print(f"Input list:      {q4_input}")  
print(f"Squared list:    {q4_ans}")  
  
Input list:      [2, 4, 6, 8, 10]  
Squared list:    [4, 16, 36, 64, 100]
```

Q5. Write a function to check if a number is even or odd and return "Even" or "Odd" accordingly.

```
In [ ]: def check_if_even(number):  
        return "Even" if number%2 == 0 else "Odd"
```

```
In [ ]: q5_input = get_input(int, "Enter a number: ", "Enter a valid number")  
  
print(f"{q5_input} is a/an {check_if_even(q5_input)} number.")
```

6 is a/an Even number.

Q6. Calculate the area of a triangle given its base and height using a function.

```
In [ ]: def triangle_area(base, height):  
        return 1/2* base * height
```

```
In [ ]: q6_input_1 = get_input(int, "Enter the base of the triangle: ", "Enter a valid  
q6_input_2 = get_input(int, "Enter the height of the triangle: ", "Enter a valid  
  
print(f"Base of the triangle: {q6_input_1}")  
print(f"Height of the triangle: {q6_input_2}")  
print(f"Area of the triangle: {triangle_area(q6_input_1, q6_input_2)}")
```

Base of the triangle: 10  
Height of the triangle: 20  
Area of the triangle: 100.0

Q7. Create a function that takes a list of strings and returns the list sorted alphabetically.

```
In [ ]: #using built-in function:  
def sort_abc(abc):  
    return sorted(abc)
```

```
In [ ]: def manual_soritng(abc):  
        result = list(abc)  
        n = len(abc)  
  
        # Bubble sort  
        for i in range(n):  
            for j in range(n-i-1):  
                if result[j] > result[j+1]:  
                    result[j], result[j+1] = result[j+1], result[j]  
  
        return result
```

```
In [ ]: q7_input = get_list("mix", "Enter an alphabet: ", stopping_character='-1')  
  
print(f"Input list {q7_input}")  
print(f"built-in sorting function: {sort_abc(q7_input)}")  
print(f"manual sortingfunction: {manual_soritng(q7_input)}")
```

Input list ['a', 'd', 'f', 'b', 'e', 'g', 'c']  
built-in sorting function: ['a', 'b', 'c', 'd', 'e', 'f', 'g']  
manual sortingfunction: ['a', 'b', 'c', 'd', 'e', 'f', 'g']

Q8. Write a function that takes two lists and returns their intersection (common elements).

```
In [ ]: def find_common_elements(list_1, list_2):
        common_elements = []

        for i in list_1:
            if i in list_2:
                common_elements.append(i)
                list_2.remove(i)

        return common_elements
```

```
In [ ]: q8_input_1 = get_list("mix", "Enter an element for list 1, -1 to stop. ", stopp
q8_input_2 = get_list("mix", "Enter an element for list 2, -1 to stop. ", stopp

print(f"Input list 1:      {q8_input_1}")
print(f"Input list 2:      {q8_input_2}")
print(f"Common Elements:   {find_common_elements(q8_input_1, q8_input_2)}")

Input list 1:      ['h', 'e', 'l', 'l', 'o', 'w', 'o', 'r', 'l', 'd']
Input list 2:      ['h', 'e', 'l', 'l', 'o', 'A', 'r', 's', 'l', 'a', 'n']
Common Elements:   ['h', 'e', 'l', 'l', 'o', 'r', 'l']
```

Q9. Write a function to check if a given year is a leap year or not.

```
In [ ]: def leap_year(year):

        if year % 4 == 0:
            if year % 100 == 0 and year % 400 != 0:
                return False
            return True
        return False
```

```
In [ ]: q9_input = get_input(int, "Enter a year to check if it is leap year: ", "Enter
print(f"YES!! {q9_input} is a leap year.") if leap_year(q9_input) else print(f'
YES!! 2024 is a leap year.
```

Q10. Create a function that takes a number as input and prints its multiplication table.

```
In [ ]: def multiplication_table(number):

        print(f"Multiplication Table of {number}")
        for i in range(1, 10+1):
            print(f"{number} x {i} = {number*i}")
```

```
In [ ]: q10_input = get_input(int, "Enter a number: ", "Enter a valid number")
multiplication_table(q10_input)
```

### Multiplication Table of 13

$$13 \times 1 = 13$$

$$13 \times 2 = 26$$

$$13 \times 3 = 39$$

$$13 \times 4 = 52$$

$$13 \times 5 = 65$$

$$13 \times 6 = 78$$

$$13 \times 7 = 91$$

$$13 \times 8 = 104$$

$$13 \times 9 = 117$$

$$13 \times 10 = 130$$

---

