Day # 03 - Control Flow and Loops

Aim:

- Learn about conditional statements and loops in Python.

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=-1):
    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 program that checks if a given number is positive, negative or zero.

In [ ]:

```
e1_input = get_input(int, "Enter a number: ","Enter a valid number...")

print("input: ",e1_input)

if e1_input == 0:
    print("Input number is Zero.")
elif e1_input > 0:
    print("Input number is Positive.")
else:
    print("Input number is Negative.")
```

```
input:  4
Input number is Positive.
```

Q2. Create a loop that prints the first 10 even numbers.

In [ ]:

```
even_numbers = []
i = 0
while len(even_numbers) != 11:

    if i%2 == 0:
        even_numbers.append(i)

    i += 1

print("First 10 even numbers: ",even_numbers)
```

```
First 10 even numbers:  [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Q3. Implement a program that finds the largest number in a list.

In [ ]:

```
e3_list = []

# taking input
while True:
    e3_input = get_input(int, "Enter a number in list: ", "Enter a valid number")

    if e3_input == -1:
        break

    e3_list.append(e3_input)

# checking the largest number
```

```
larget_number = e3_list[0]
for i in e3_list:
    if i > larget_number:
        larget_number = i

print("Entered list: ", e3_list)
print("Largest number: ",larget_number)
```

```
Entered list:  [-50, -100, -7, -3]
Largest number:  -3
```

Practice Questions...

Q1. Create a program that takes a year as input and checks if it is a leap year or not.

Leap year rules: How to calculate leap years.

- The year must be evenly divisible by 4.
- if the can also be evenly divided by 100, it is not a leap year.

  unless...
- The year is evenly divisible by 400.

Click for the source of these rules...

In [ ]:

```
q1_input = get_input(int, "Enter a year to check if it is leap year: ", "Enter a valid year...")
leap_year = False

if q1_input % 4 == 0:
    leap_year = True
    if q1_input % 100 == 0 and q1_input % 400 != 0:
        leap_year = False

print("Input year: ", q1_input)
if leap_year == True:
    print("This is a leap year.")

else:
    print("This is not a leap year.")
```

```
Input year:  2020
This is a leap year.
```

Q2. Given a list of integers, find all the even numbers and store them in a new list.

In [ ]:

```
q2_list = get_list(int, "Enter a number in list")
prime_list = []

for i in q2_list:
    if i % 2 == 0 and i not in prime_list:
        prime_list.append(i)

# prime_list = sorted(prime_list)
print("Input List: ", q2_list)
print("List of prime numbers: ", sorted(prime_list))
```

```
Input List:  [8, 9, 2, 10, 4, 7, 11, 15]
List of prime numbers:  [2, 4, 8, 10]
```

Q3. Write a Python program to check if a given number is a prime number.

In [ ]:

```
q3_input = get_input(int, "Enter a number: ", "Enter a valid number...")
prime = True

for i in range(2, q3_input):
    if q3_input % i == 0:
        prime = False
        break

if prime:
    print(f"{q3_input} is a prime number")

else:
    print(f"{q3_input} is not a prime number")
```

```
11 is a prime number
```

Q4. Create a program that generates the Fibonacci sequence up to a given number of terms.

Fibonacci Sequence: In mathematics, the Fibonacci sequence is a sequence in which each number is the sum of the two preceding ones. Numbers that are part of the Fibonacci sequence are known as Fibonacci numbers, commonly denoted Fn.

$$F_n = F_{n-1} + F_{n-2}$$

```
In [ ]: q4_input = get_input(int, "How many terms of Fibonacci sequence do you want: ", "Enter a valid number")

        Fibonacci = [0, 1]
        while len(Fibonacci) != q4_input:
            Fibonacci.append(Fibonacci[-1] + Fibonacci[-2])


        print(f"Fiboacci sequence upto {q4_input} terms: {Fibonacci}")
```

```
Fiboacci sequence upto 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Q5. Given a list of names, print all names starting with the letter 'A'.

```
In [ ]: name_list = ["Arslan", "Saad", "Umar", "Hamza", "Ahmed", "Fahad", "Husnain", "Asad"]
        output_list = []

        for name in name_list:
            if name.upper()[0] == 'A':
                output_list.append(name)

        print("Name list:                       ", name_list)
        print("List of names starting with 'A': ", output_list)
```

```
Name list:                        ['Arslan', 'Saad', 'Umar', 'Hamza', 'Ahmed', 'Fahad', 'Husnain', 'Asad']
List of names starting with 'A':  ['Arslan', 'Ahmed', 'Asad']
```

Q6. Implement a program that prints the multiplication table of a given number.

```
In [ ]: q6_input = get_input(int, "Enter a number for multiplication table: ", "Enter a valid number...")

        for i in range(11):
            print(f"{q6_input} x {i}    = {q6_input*i}")
```

```
5 x 0    = 0
5 x 1    = 5
5 x 2    = 10
5 x 3    = 15
5 x 4    = 20
5 x 5    = 25
5 x 6    = 30
5 x 7    = 35
5 x 8    = 40
5 x 9    = 45
5 x 10    = 50
```

Q7. Write a program that calculates the factorial of a given number.

In [ ]:

```python
def calculate_factorial(number):
    output = 1
    for i in range(2, number+1):
        output *= i

    return output
```

In [ ]:

```python
q7_input = get_input(int, "Enter a number to find factorial: ", "Enter a valid number")

print(f"Factorial of number {q7_input} = {calculate_factorial(q7_input)}")
```

```
Factorial of number 5 = 120
```

Q8. Create a loop that prints all prime numbers between 1 and 50.

In [ ]:

```python
prime_numbers = []

for i in range(2, 50):
    prime = True
    for j in range(2, i):
        if i%j == 0:
            prime = False

    if prime == True:
        prime_numbers.append(i)

print(f"Prime numbers in range (1,50): {prime_numbers}")
```

```
Prime numbers in range (1,50): [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Q9. Given a list of words, count the number of words with more than five characters.

In [ ]:

```
q9_list = get_list("mix", "Enter a word to add in list: ", stopping_character='stop')
out_list = []

for word in q9_list:
    if len(word) > 5:
        out_list.append(word)

print(f"Input list: {q9_list}")
print(f"Words with more than 5 characters: {len(out_list)} ==> {out_list}")
```

```
Input list: ['Hello', 'my', 'name', 'is', 'Arslan', 'Khalid', 'and', 'i am doing', 'Python Language']
Words with more than 5 characters: 4 ==> ['Arslan', 'Khalid', 'i am doing', 'Python Language']
```

Q10. Calculate the sum of digits of a given number.

In [ ]:

```
q10_input = str(get_input(int, "Enter a number to get sum of it's digits: ", "Enter a valid number"))
result = 0

for i in range(len(q10_input)):
    result += int(q10_input[i])

print(f"Input number: {q10_input}")
print(f"Sum of it's digits: {result}")
```

```
Input number: 12345
Sum of it's digits: 15
```