PYTHON BASIC

(DAY 6)

LOOPS

Loops in Python are used to execute a block of code repeatedly as long as a specified condition is met. Python supports two types of loops:

**for Loop**

**while Loop**

### **for Loop**

The for loop in Python is used to iterate over a sequence (such as a list, tuple, string, or range).

#### ***Syntax:-***

for variable in sequence:  
 # block of code

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LOOPS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# want to execute a certain specific code line repeativily

# There are two types of loops in python "for loop" and "while loop"

# \_\_\_\_\_\_\_\_ for \_\_\_\_\_\_\_\_\_\_\_

for i in range(10) : # range(10) means that this "for" loop execute code 10 times

# "i" is iterate variable which denotes how many time code already executed and it start from 0 to n(range-1)

print("Hello world!", i)

for i in range(4,9) : # start iteration from 4 to 9-1=8

print("Again Hello world!", i)

for i in range(4,9,2) : # start iteration from 4 to 9-1=8 with 2 jump

print("And Again Hello world!", i)

# so from above now we easily understood that

# syntax of range is : - range(start, end, jump)

### **while Loop**

The while loop in Python is used to repeatedly execute a block of code as long as a condition is true.

#### ***Syntax:-***

while condition:  
 # block of code

# \_\_\_\_\_\_\_\_\_\_\_\_\_ WHILE \_\_\_\_\_\_\_\_\_\_\_\_

i =1

while(i<=10) : # "while" loop execute code till condition is true

print("While Hello world!",i)

i+=1

# What is difference b/w for and while loop? --> If you already know that how many time iterations need then use "for" and if you know only condition not exact iterations then use "while" loop.

condition = True

while condition :

user\_input = input("Do you want to execute code y/n : ")

if user\_input == "n" :

condition = False

break # "break" statement exit while loop wherease "continue" statement skip that iteration

print("Welcome to the while loop")

print("You are outside while loop ")

In Python, break and continue statements are used to control the flow of loops. They allow you to exit or skip iterations within a loop based on certain conditions.

### **break Statement**

The break statement is used to terminate the loop prematurely. When a break statement is encountered inside a loop, the loop stops executing, and the control flows to the next statement after the loop.

### **continue Statement**

The continue statement is used to skip the current iteration and proceed to the next iteration of the loop. When a continue statement is encountered inside a loop, the rest of the code inside the loop for that iteration is skipped, and the loop continues with the next iteration.

# break and continue

# break

count =0

for i in range(10) :

if i%2==0 :

continue # skip the iteration when i%2 == 0 means i is even

count+=1

print(count)

# continue

count=0

for i in range(10) :

if i%5 == 0 :

break # it will stop "for" loop if "i" is divisible by 5

count+=1

print(count)

# let's consider an example

ls = [52,41,63,96,85,7,45,86,6,9,12,36,72,11,22,33]

# if 85 present in ls, then return it's index

count=0

for item in ls :

if item == 85 :

print(count)

break # it will stop "for" loop after condition item == 85 is true

count+=1

ls = [52,41,63,96,85,7,45,86,6,9,12,85,36,72,11,22,33]

# if 85 present more than one time than return index of last one

index =0

count =0

if 85 in ls :

for item in ls :

if item == 85 :

index =count

count+=1

print("index is : ",index)

else :

print("85 is not present in list")

# find min, max from list with using loop

ls = [52,41,63,96,85,7,45,86,6,9,12,36,72,11,22,33]

max =None

min = None

for x in range(len(ls)) :

if ls[x]>max:

max = ls[x]

elif ls[x]<min :

min = ls[x]

print("max in list : ",max)

print("min in list : ",min)