



ACCA4020

Python – Loops

Dr. Seena Joseph

Loops

- Loops in Python allow you to execute a block of code multiple times.
- Example:
 - Write a program to print “Hello World” five times.
 - Write a program to print numbers from 1 to 10.

Without loop

```
#Without Loop  
  
print('Hello world')  
print('Hello world')  
print('Hello world')  
print('Hello world')  
print('Hello world')
```

```
#Without Loop  
  
print('1')  
print('2')  
print('3')  
print('4')  
print('5')  
print('6')  
print('7')  
print('8')  
print('9')  
print('10')
```



Is it a good approach ?

Loops

Is it a good approach ?

Repeating code without loop is inefficient and not scalable for large numbers.

For example: Write a program to print numbers from 1 to 1000.

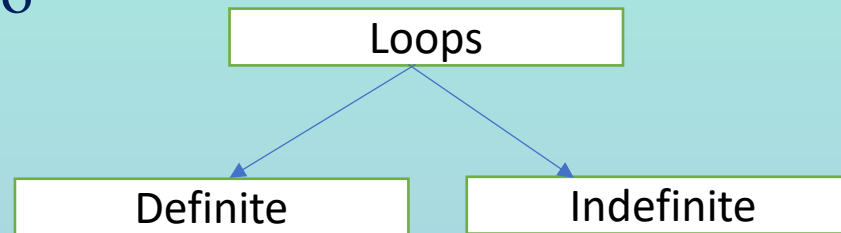
Loops

- Example:
 - Write a program to print “Hello World” five times.
 - Write a program to print numbers from 1 to 10.

With Loop

Loops allow you to **execute a block of code multiple times** without writing repetitive statements.

In python loops as classified into



- Definite Loop (**for loop**) – runs a fixed number of times
- Indefinite Loop (**while loop**) – runs until a condition a met

Loops

- **for loop:** it is used when the programmer knows how many times a task/or a set of tasks need to be repeated.

for *varname* in range(*maxCount*):

Your code block goes here

If you know how many times you need to loop at the start then use a counting loop

for index in range (5):

print('Hello world')

You want print 'Hello World' **Five Times**

Which is efficient ?

```
#with Loop
for i in range(5):
    print('Hello World')
```

```
#Without Loop

print('Hello world')
print('Hello world')
print('Hello world')
print('Hello world')
print('Hello world')
```

Loops

- **for loop (Definite Loop):**

```
#with Loop
for index in range(10):
    print(index)
```

Index starts at 0

```
0
1
2
3
4
5
6
7
8
9
```

Notice: since the index starts at 0 we add 1 because we want to print this is the 1st time around the loop instead of this is the 0th time around the loop

```
#with Loop
for index in range(10):
    print(index+1)
```

```
1
2
3
4
5
6
7
8
9
10
```

Loops

- for loop:

```
maxLoops = int(input("Enter the number of loops: "))  
  
print("Index values are: ", end=" ")  
for index in range(maxLoops):  
    print(index, end=" ")  
  
print() ←  
print("end of loop")
```

*What happens if
you leave this
out??*

*Notice we can dynamically
decide the number of times we
should loop*

The variable name *index* can be any name you choose, “*index*” or just “*i*” are quite common names for this variable

Loops

- for loop:

```
maxLoops = int(input("Enter the number of loops: "))

print("Index values are: ", end=" ")
for index in range(maxLoops):
    print(index, end=" ")
print()
print("end of loop")
```

Try without print()

Sample output

```
Enter the number of loops: 7
Index values are:  0 1 2 3 4 5 6
end of loop
```

Loops

- **for loop**
- **Example** :Generate 5 random numbers

```
import random
```

```
random.seed()
```

```
for i in range(5):  
    print(random.randint(1, 100))
```

```
import random
```

```
random.seed()
```

```
for i in range(5):  
    print(random.randint(a: 1, b: 100))
```

```
14
```

```
63
```

```
57
```

```
75
```

```
22
```

Loops

For Loops:

Multiplication Table of 3

```
# Multiplication table pf 3  
for i in range(11):  
    print(f'{i} x {3} = {i*3}')
```

Multiplication Table fof 3

```
0 x 3 = 0  
1 x 3 = 3  
2 x 3 = 6  
3 x 3 = 9  
4 x 3 = 12  
5 x 3 = 15  
6 x 3 = 18  
7 x 3 = 21  
8 x 3 = 24  
9 x 3 = 27  
10 x 3 = 30
```

Exercise 1

1. What will be output of the following code ?

```
for i in range(3):  
    print('Hello')
```

2. Write a python program that prompts the user to enter a number and prints its multiplication table.

Sample output

```
Enter the number : 8  
Multiplication Table for 8  
1 x 8 = 8  
2 x 8 = 16  
3 x 8 = 24  
4 x 8 = 32  
5 x 8 = 40  
6 x 8 = 48  
7 x 8 = 56  
8 x 8 = 64  
9 x 8 = 72  
10 x 8 = 80
```

3. Write a python program to print the following pattern using loop.

```
*  
**  
***  
****  
*****
```

Loops

range(): The range function in python is used to generate a sequence of numbers. It is commonly used in loops, especially for loops to iterate over a sequence of numbers. Syntax:

range(start, stop, step)

Start: the starting value of the sequence(inclusive). If not provided it defaults to '0'

Stop: The end value (exclusive)

Step: The difference between each number in the sequence. If not provided, it defaults to 1.

Loops

range():

Example 1:

```
for i in range(6):  
    print(i)
```

Sample output :

```
0  
1  
2  
3  
4  
5
```

Example 2 : start=5 ; end = 8 (exclusive so up to 7) step=1

Sample output :

```
5  
6  
7
```

```
for i in range(5, 8):  
    print(i)
```

Example 3 : start=3 ; end = 10 (exclusive so up to 9) step=2

Sample output :

```
for i in range(3,10,2):  
    print(i)
```

```
3  
5  
7  
9
```

Example 4 : start=10; end = 0 (exclusive , so up to 1) step= -1 i.e decrement

Sample output :

```
for i in range(10, 0, -1):  
    print(i)
```

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1
```

Exercise 2

1. Write a python program that calculates the sum of first 10 natural numbers.
2. Write a python program to print even numbers from 2 to 20.
3. Write a python program to print odd numbers from 11 to 35.
4. Write a python program to print numbers from 100 down to 85

Loops

while loop (Indefinite Loop)

while condition:

Your code goes here

1. Condition: an expression or condition that is checked, before each iteration of the loop. If the condition evaluates to True , the loop continues. If it evaluates to False the loop stops.

In a **while loop**, the loop control variable is checked to see if the condition is till **True**, before each iteration.

```
i=1
while i<5:
    print(i)
    i=i+1
```

Loops

while loop

Loop control variable:

Initialization: The loop control variable is usually initialized before the loop starts.

Update: After each iteration, the loop control variable is updated, which affects the continuation or termination of the loop.

Termination Condition: The loop control variable is compared against a condition to decide when to exit the loop.

Commonly Used in for and while Loops: The variable helps control the flow of iterations in both types of loops.

Loop

while loop

Write a Python program that continuously prompts the user to enter a number. The program will keep adding the entered number to the total. After each input, display the total so far. The program should stop once the total reaches or exceeds 100.

Initialisation :Start with a total value of 0

Termination condition

```
total = 0
```

```
while total < 100:
```

```
    number = int(input("Enter number:"))
```

```
    total = total + number
```

```
    print("Total so far is {}".format(total))
```

```
print("End of loop")
```

Update

Loop

while loop

Example 2: Write a Python program that repeatedly asks the user if they want to continue looping. If the user enters "y" or "yes", the loop continues, printing "Looping again". If the user enters anything else, the loop stops and the program prints "End of loop". The program should not be case-sensitive and should ignore any extra spaces in the user's input.

Initialisation :

Termination condition

Update

```
goAgain = "yes"

while goAgain == "y" or goAgain == "yes":

    print("Looping again")

    goAgain = input("Keep going ? yes[y]")
    goAgain = goAgain.strip().lower()

print("End of loop")
```

```
goAgain = "yes"
while goAgain == "y" or goAgain == "yes":
    print("Looping again")
    goAgain = input("Keep going ? yes[y]")
    goAgain = goAgain.strip().lower()

print("End of loop")
```

Loop

while loop

Example 3: Write a Python program that simulates throwing a dice. Keep throwing the dice until the user decides to stop. After each roll, display the number rolled and keep a running total of the score. When the user decides to stop, display the total score they achieved.

```
import random

random.seed()
totalScore = 0

goAgain = "yes"
while goAgain == "y" or goAgain == "yes":
    print("Throwing dice")
    dice = random.randint(a: 1, b: 6)
    totalScore = totalScore + dice
    print("The dice shows: {}".format(dice))

    goAgain = input("Throw again? yes[y]")
    goAgain = goAgain.strip().lower()

print("End of loop")
print("The total score is: {}".format(totalScore))
```

Sample Output

```
The dice shows: 2
Throw again? yes[y]y
Throwing dice
The dice shows: 4
Throw again? yes[y]y
Throwing dice
The dice shows: 2
Throw again? yes[y]n
End of loop
The total score is: 8
```

Exercise 3

1. A nuclear reactor allows the operator to enter the initial temperature and pressure
 - When the temperature exceeds 1234oC the program prints out a final warning and ends
 - The user continually entered new pressure values until the program ends
 - The temperature depends on the pressure and the change in temperature is calculated as follows:
 - $\text{temperatureChange} = (\text{newPressure} - \text{oldPressure}) * 5$

Sample output

```
Enter intial temperature :500
Enter initial pressure :200
Enter new pressure :250
Pressure change is 50
Temperature change is 250
Current Temperature is : 750
Enter new pressure :290
Pressure change is 40
Temperature change is 200
Current Temperature is : 950
Enter new pressure :500
Pressure change is 210
Temperature change is 1050
Current Temperature is : 2000
High Temp
```

2. Write a Python program that generates a random number between 1 and 100. The user must guess the number, and after each guess, the program should provide hints such as "Too high!" or "Too low!" to guide the user. The loop should continue until the user correctly guesses the number.

Exercise 4

1. Write a program in Python using loop that asks the teacher to enter the marks for 10 students. The program should then:
 - Count how many students passed (marks ≥ 50).
 - Count how many students failed (marks < 50).
 - Calculate and display the average mark of the class.
 2. Write a program in Python that asks the user to enter a positive integer. The program should then determine and display all the factors of that number. A factor is a number that divides the given number exactly without leaving a remainder.
 3. Write a program in Python that simulates a simple traffic monitoring system. The program should ask the user to enter the speed of vehicles (in km/h) recorded by a road sensor for 10 vehicles. The program should then:
 - Count how many vehicles exceeded the speed limit of 80 km/h.
 - Calculate the average speed of all vehicles.
 - Display the total number of vehicles that were speeding and the average speed.
- Use a loop to collect the speed of each vehicle from the user and perform the required calculations.