

Python For Loop

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). When we know how many times we wanted to run a loop, then we use count-controlled loops such as for loops. It is also known as definite iteration.

For Loop Trough List

```
# Example 01:
    # Iterating through the list of words using for loop

# Assume the list of numbers
numbers = ["One", "Two", "Three", "Four", "Five"]

# Iterate the loop through each word in the list
for number in numbers:
    # print the numbers
    print(number)

One
Two
Three
Four
Five

# Example 02:
    # Calculating the average of the list number

# Assume the list of numbers
numbers_list = [1, 5, 7, 17, 24, 4, 12]

# Initialize variable for sum and list size
total_sum = 0
list_size = len(numbers_list)

# Iterate through each number of the list
for num in numbers_list :
    # Calculate the sum of the list
    total_sum += num

# print the sum of the number in the list
print(f"The sum of the numbers in the list : {total_sum}")

# calculate the average dividing the sum of the number in the list by
number of items in the list
average = total_sum / list_size
```

```
# print the calculated average
print(f"The average of the numbers in the list is : {average}")
```

The sum of the numbers in the list : 70
The average of the numbers in the list is : 10.0

Example 03 :

program to separates a list of numbers into even and odd lists

Assume a list of numbers

```
list_of_numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Initialize empty lists to store even and odd numbers

```
even_list = []
```

```
odd_list = []
```

Iterate through each number in the list

```
for current_number in list_of_numbers:
```

Check if the current number is even

```
if current_number % 2 == 0:
```

If condition is True, append it to the even_list

```
even_list.append(current_number)
```

If the is False, execute the else block

```
else:
```

append number to odd list

```
odd_list.append(current_number)
```

print the list of even and odd numbers

```
print(f"Even Number List : {even_list}")
```

```
print(f"odd Number List : {odd_list}")
```

Even Number List : [2, 4, 6, 8, 10]

odd Number List : [1, 3, 5, 7, 9]

Example 04 :

Program to iterate over a list of students and print a message indicating that each student will be off tomorrow using for loop

Assume a list of students

```
students = ['Arif', 'Ahmad', 'Imran', 'Babar', 'Rizwan', 'Haris',  
'Aman', 'Asad', 'Hassan', 'saim']
```

Iterate through each student in the list

```
for student in students:
```

```
# Print a message indicating that the Tomorrow is off  
print(f"{student} Tomorrow is off")
```

```
Arif Tomorrow is off  
Ahmad Tomorrow is off  
Imran Tomorrow is off  
Babar Tomorrow is off  
Rizwan Tomorrow is off  
Haris Tomorrow is off  
Aman Tomorrow is off  
Asad Tomorrow is off  
Hassan Tomorrow is off  
saim Tomorrow is off
```

```
# Example 05 :
```

```
# program iterates over a list of student names and prints a  
message indicating whether each student has a day off or must attend  
class tomorrow
```

```
# Assume a list of the student
```

```
students = ['Arif', 'Ahmad', 'Imran', 'Babar', 'Rizwan', 'Haris',  
            'Aman', 'Asad', 'Hassan', 'saim']
```

```
# Iterate through each student in the list
```

```
for student in students :
```

```
# Check if the student's name start with 'A'
```

```
if student.startswith('A') :
```

```
# If the condition is True, print a message indicating that  
the student has day off tomorrow
```

```
print(f"{student} Tomorrow is off")
```

```
# If the condition is False, execute the else block
```

```
else :
```

```
# print a message indicating tomorrow is on
```

```
print(f"{student} Tomorrow is on")
```

```
Arif Tomorrow is off  
Ahmad Tomorrow is off  
Imran Tomorrow is on  
Babar Tomorrow is on  
Rizwan Tomorrow is on  
Haris Tomorrow is on  
Aman Tomorrow is off  
Asad Tomorrow is off  
Hassan Tomorrow is on  
saim Tomorrow is on
```

```

# Example 06 :
    # Program iterates over a list of student names and appends
the string "Khan" to each student's name, creating a new list

# Assume a list of the student
students = ['Arif', 'Ahmad', 'Imran', 'Babar', 'Rizwan', 'Haris',
'Aman', 'Asad', 'Hassan', 'saim']

# Initialize an empty list to store the modified student names
students_new_list = []

# Iterate through each student in the list
for student in students :
    # Append the student name with the 'Khan' in students_new_list
    students_new_list.append(student + " " + "Khan")

# print the new list of the student
print(students_new_list)

['Arif Khan', 'Ahmad Khan', 'Imran Khan', 'Babar Khan', 'Rizwan Khan',
'Haris Khan', 'Aman Khan', 'Asad Khan', 'Hassan Khan', 'saim Khan']

# Example 07 :
    # Using the For loop to print the Items of list and
their index

# Assume a list of the numbers
my_list = [14, 8, 42, 25, 36, 15, 2, 45, 56, 29]

# Iterate through each number in the list
for num in range(len(my_list)):
    # print the items of the list and their corresponding indexes
    print(num, my_list[num])

0 14
1 8
2 42
3 25
4 36
5 15
6 2
7 45
8 56
9 29

# Example 08 :
    # Program to prompts the user to enter their orders
interactively, allowing them to input multiple orders until they
decide to quit by entering 'q'. and then stores the entered orders in
a list

```

```

# Initialize an empty list
order = []

# Iterate 10 times to give the user the option to enter up to 10 orders
for a in range(10) :
    # Prompt the user to enter their order
    given_order = input("Please enter your order or press q to quit : ")

    # Check if the input is q, break the loop
    if given_order == 'q' :
        # If the condition is True break the loop
        break

    # If the condition is False execute the else block
    else :
        order.append(given_order)

# Print the list of orders when the user finishesthe ordering or chooses to quit
print(order)

```

```

Please enter your order or press q to quit : usb
Please enter your order or press q to quit : laptop
Please enter your order or press q to quit : charger
Please enter your order or press q to quit : cable
Please enter your order or press q to quit : q
['usb', 'laptop', 'charger', 'cable']

```

For Loop Through String

Even strings are iterable objects, they contain a sequence of characters

```

# Example 02:
# Iterating through a string using for loop

# Iterate the loop through "Muhammad Ali"
for char in "Muhammad Ali" :
    # print the each character
    print(char)

```

```

M
u
h
a
m

```

m
a
d

A
l
i

The range() function

To loop through a set of code a specified number of times, we can use the range() function. The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number, but the last number are not included.

```
# Example 01:  
    # Print the String specified number of times using for loop
```

```
# Iterating the loop specified times  
for a in range(10):  
    print(f"{a} Babar Azam")
```

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

```
# Example :  
    # Increment the sequence with 3 (default is 1)
```

```
# Iterate the loop specified times increment the sequence with 3  
for x in range(1, 20, 3):  
    print(f"{x} Babar Azam")
```

```
1 Babar Azam  
4 Babar Azam  
7 Babar Azam  
10 Babar Azam  
13 Babar Azam  
16 Babar Azam  
19 Babar Azam
```

```
# Example :  
    # Decrement the sequence with -1
```

```
# Iterate the loop specified times decrement the sequence with -1
for y in range(10, 0, -1):
    print(f"{y} Babar Azam")
```

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

Example :

program demonstrating the use of a continue statement in a for loop

Iterate the loop through a specified times

```
for i in range(10):
    # Check if the current number is 5
    if i == 5 :
        # If the condition is True, skip printing it and move to the next iteration
        continue
```

If the condition is False execute the else block

```
else:
    # print the numbers with string
    print(f"{i} Babar Azam")
```

```
0 Babar Azam
1 Babar Azam
2 Babar Azam
3 Babar Azam
4 Babar Azam
6 Babar Azam
7 Babar Azam
8 Babar Azam
9 Babar Azam
```

Example 07 :

program to demonstrate the use of a break statement in a for loop

Iterate the loop through a specified times

```
for a in range(10):
    # Check if the current number is 7
    if a == 7 :
```

```

        # If the condition is True, exit the loop using break
statement
        break

    # If the condition is False, execute the else block
else:
    # print the numbers with string
    print(f"{a} Babar Azam")
0 Babar Azam
1 Babar Azam
2 Babar Azam
3 Babar Azam
4 Babar Azam
5 Babar Azam
6 Babar Azam

```

For Loop Through a Dictionary

```

# example 01 :
    # Program to demonstrates how to iterate for loop through the
values of a dictionary and print each value individually

# Assume a dictionary
my_dict = {"brand": "Ford", "model": "Mustang", "year": 2012}

# Iterate the loop through the values of the dictionary
for value in my_dict.values() :
    # print each value
    print(value)

Ford
Mustang
2012

# example 02 :
    # Program to demonstrates how to iterate for loop through the
keys of a dictionary and print each value individually

# Assume a dictionary
my_dict = {"brand": "Ford", "model": "Mustang", "year": 2012}

# Iterate the loop through the keys of the dictionary
for keys in my_dict :
    # If we iterate the for loop through dictionary by default it iterate
through keys
    # print each keys
    print(keys)

```


brand	model	year
-------	-------	------