Python for Loops

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages and works more like an iterator method as found in other object-orientated programming languages.

With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

Example

Print each fruit in a fruit list:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
      print(x)

apple   #output
banana
cherry
```

The for loop does not require an indexing variable to set beforehand.

Looping Through a String

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

Example

Loop through the letters in the word "banana":

```
for x in "banana":
  print(x)
b
a
n
a
n
a
```

The break Statement

With the break statement we can stop the loop before it has looped through all the items:

Example

Exit the loop when x is "banana":

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
     print(x)
     if x == "banana":
          break
```

```
apple
banana
```

Example

Exit the loop when x is "banana", but this time the break comes before the print:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
          break
    print(x)
```

```
apple
```

The continue Statement

With the continue statement we can stop the current iteration of the loop, and continue with the next:

Example

Do not print banana:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
          continue
   print(x)
```

apple
cherry

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.

Example

Using the range() function:

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

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

Note that range(6) is not the values of 0 to 6, but the values 0 to 5.

The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(2, 6), which means values from 2 to 6 (but not including 6):

Example

Using the start parameter:

```
for x in range(2, 6):
  print(x)
```

```
2
3
4
5
```

The range() function defaults to increment the sequence by 1, however it is possible to specify the increment value by adding a third parameter: range(2, 30, **3**):

Example

Increment the sequence with 3 (default is 1):

```
for x in range(2, 30, 3):
  print(x)
```

Else in For Loop

The else keyword in a for loop specifies a block of code to be executed when the loop is finished:

Example

Print all numbers from 0 to 5, and print a message when the loop has ended:

```
for x in range(6):
  print(x)
else:
  print("End of loop")
```

```
0
1
2
3
4
5
End of loop
```

Note: The else block will NOT be executed if the loop is stopped by a break statement.

Example

Break the loop when x is 3, and see what happens with the else block:

```
for x in range(6):
  if x == 3:
          break
  print(x)
else:
  print("End of loop ")
0
```

1
2

Here loop is not completed. It has stopped by break. So else is not working.

```
#  find sum of all natural numbers between 1 to n using while loop.
n=int(input('Enter a natural number: '))
s=0
i=0
while i<=n:
   s=s+i
   i+=1
print('The sum of first ',n,' natural numbers is ',s)
```

```
# find sum of all natural numbers between 1 to n using for loop.
n=int(input("Enter a number:"))
s=0
for i in range(n+1):
   s=s+i
print('The sum of first ',n,' natural numbers is ',s)
```

Try This Using for loop

1.Print all natural numbers from 1 to n.
2. Print all natural numbers in reverse (from n to 1).
3. Print all even numbers between 1 to n.
4. Print all odd number between 1 to n.
5. Find sum of all natural numbers between 1 to n.
6. Find sum of all even numbers between 1 to n.
7. Find sum of all odd numbers between 1 to n.
8.Print multiplication table of any number.
9. Calculate factorial of a number.
10. Enter a word and print reverse of it.
11. Enter a number and print its reverse.
12. Write a program to print all multiple of 5 with in n.
13. Input a number and check whether it is prime or not.

Method 1

```
# Check input number is a prime or not
n=int(input("Enter a number greater than 1:"))
x=True
for i in range(2,n):
    if n%i==0:
        x=False
        break
if x==True:
    print('prime')
else:
    print('Not prime')
```

Method 2

```
# Check input number is a prime or not
import math
n=int(input("Enter a number greater than 1:"))
x=True
for i in range(2,int(math.sqrt(n))+1):
    if n%i==0:
        x=False
        break
if x==True:
    print('prime')
else:
    print('Not prime')
```

Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

Example

Print each adjective for every fruit:

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
```

```
        for y in fruits:
                print(x, y)
```

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

1. Write a program to print the following number pattern using a loop.

   ```
   1
   1 2
   1 2 3
   1 2 3 4
   ```

   ```
   r=int(input("Enter number of rows"))
   for i in range(1,r+1,1):
       for j in range(1,i+1):
           print(j,end='')
       print('')
   ```

2. Write a program to print the following pattern using a loop.

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

   Try this

1. Write a program to print the following pattern using a loop.

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

2. Find all prime number between 1 to n.

```
# Check input number is a prime or not
import math
```

```
n=int(input("Enter a number greater than 1:"))
x=True
for i in range(2,int(math.sqrt(n))+1):
   if n%i==0:
      x=False
      break
if x==True:
   print('prime')
else:
   print('Not prime')


#Display all prime numbers between 1 to n
import math
n=int(input("Enter a number greater than 1:"))
for j in range(2,n+1):
   x=True
   for i in range(2,int(math.sqrt(j))+1):
      if j%i==0:
         x=False
         break
   if  x==True:
      print(j,' is prime')
```