

Flow Control (For loop)

Chapter 02c

While loop vs. For Loop

- While loop continues while its condition is True
- For loop executes the block of code a certain number of times.

for Loop: Definite and Indefinite

- **Iteration** – is a repeat of an action.
- **Definite Iteration**
 - Repeats an action a predefined number of times.
- **Indefinite Iteration**
 - Repeats an action until the program determines it needs to stop.

for Loop

The FOR statement will execute the block a certain number of times.

The FOR Statement consists of the following:

- The **for** keyword
- A **iteration variable** name
- The **in** keyword
- A call to the **range()** method, with up to three integers passed to it.
- A **colon**
- Starting on the next line, the for clause (**indented** block of code) will be executed.

for Loops and range method Syntax

```
for num in range(5):  
    print(num)
```

Iteration variable num starts at 0 up to not including 5.

A call to the **range()** method, can have up to three integers passed to it.

OUTPUT:

0

1

2

3

4

num

for Loops and range method Syntax

```
for num in range(5):
```

```
    print(num, end = " ")
```

num starts at 0 up to not including 5.

A call to the **range()** method, can have up to three integers passed to it.

#Keeps output on same line

OUTPUT:

0 1 2 3 4

for Loops Syntax

```
for num in 5:  
    print(num, end = " ")
```

**#Error 'int' object is not iterable.
#Iteration is a group of something
#(string or number) to go one after
#another.**

for Loops Syntax (**start** and **stop** values)

```
for num in range(12, 16):  
    print(num, end = " ")
```

#Num starts at 12, stops at 16 (not including stop value).

OUTPUT:

12 13 14 15

for Loops Syntax (**start**, **stop** and **step** values)

```
for num in range(0, 10, 2):  
    print(num, end = " ")
```

#Num starts at 0, stops at 10

#(not including stop value), steps up by 2.

OUTPUT:

0 2 4 6 8

num

for Loops Syntax (range **negative** values)

```
for num in range(5, -1, -1):  
    print(num, end = " ")
```

#Num starts at 5, stops at -1

#(not including stop value), steps down by -1.

OUTPUT:

5 4 3 2 1 0

num

for Loops and range values are exactly what you want.

- FOR Statement

- When range is exactly what you want,
- range keyword is not needed
- **[Insert values in brackets]**
- [Values can be numbers or strings]

for Loops and range values are exactly what you want.

#range keyword is not needed

```
for steps in [1, 2, 3, 4, 5]:  
    print(steps)
```

OUTPUT:

1

2

3

4

5

for Loops and range values are exactly what you want.

#range keyword is not needed

```
for character in ["Hi there!"]:  
    print(character, end = " ")
```

OUTPUT:

Hi there!

break and continue

- Can also be used in for Loops.