**Loops**

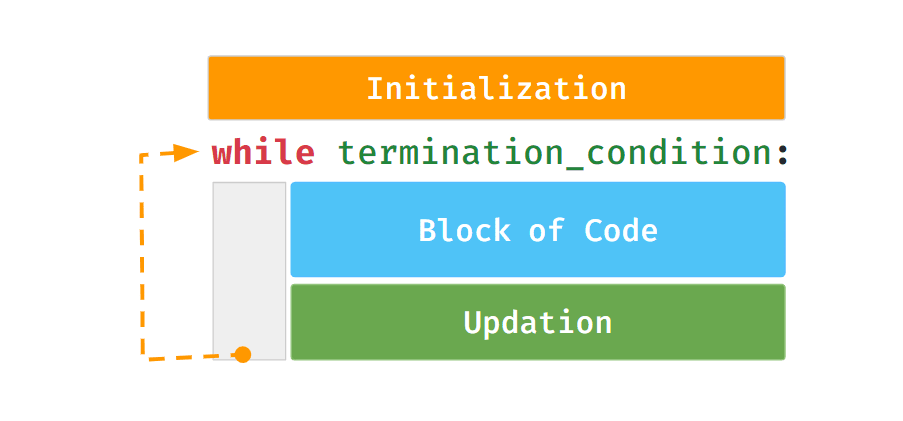
So far we have seen that Python executes code in a sequence and each block of code is executed once.  
**Loops allow us to execute a block of code several times.**

While Loop

Allows us to execute a block of code several times as long as the condition is

True

.



While Loop Example

The following code snippet prints the next three consecutive numbers after a given number.

**Code**



1

2

3

4

5

6

a = int(input())

counter = 0

while counter < 3:

a = a + 1

print(a)

counter = counter + 1

PYTHON

**Input**



4

**Output**



5

6

7

Possible Mistakes

**1. Missing Initialization**

**Code**



1

2

3

4

5

6

a = int(input())

while counter < 3:

a = a + 1

print(a)

counter = counter + 1

print("End")

PYTHON

**Input**



5

**Output**



NameError: name 'counter' is not defined

**2. Incorrect Termination Condition**

**Code**



1

2

3

4

5

6

7

a = int(input())

counter = 0

condition = (counter < 3)

while condition:

a = a + 1

print(a)

counter = counter + 1

PYTHON

**Input**



10

**Output**



Infinite Loop

While block will keep repeating as the value in condition variable is

True

.

**3. Not Updating Counter Variable**

**Code**



1

2

3

4

5

6

a = int(input())

counter = 0

while counter < 3:

a = a + 1

print(a)

print("End")

PYTHON

**Input**



10

**Output**



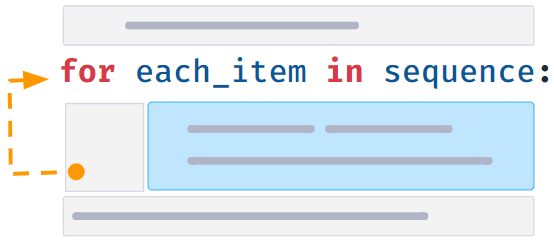
Infinite Loop

As the value of counter is not updating.

## For Loop

for

statement iterates over each item of a sequence.



*Examples of sequences:*

* Sequence of Characters (string)
* Sequence of numbers, etc.

### For Syntax

#### Code



1

2

3

word = "Python"

for each\_char in word:

print(each\_char)

PYTHON

#### Output



P

y

t

h

o

n

## Range

Generates a sequence of integers starting from 0. Syntax:

range(n)

Stops before n (n is not included).

#### Code



1

2

for number in range(3):

print(number)

PYTHON

#### Output



0

1

2

### Range with Start and End

Generates a sequence of numbers starting from

start

Syntax:

range(start, end)

Stops before

end

(end is not included).

#### Code



1

2

for number in range(5, 8):

print(number)

PYTHON

#### Output



5

6

7

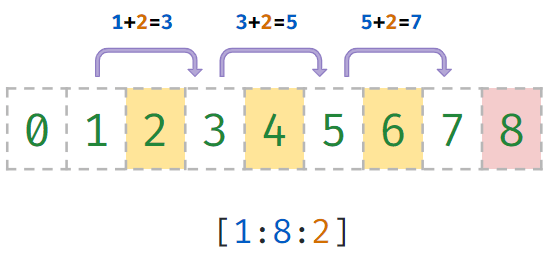
Notes

**Extended Slicing and String Methods**

Extended Slicing

Syntax:

variable[start\_index:end\_index:step]



Step determines the increment between each index for slicing.

**Code**



1

2

3

a = "Waterfall"

part = a[1:6:3]

print(part)

PYTHON

**Output**



ar

Methods

Python has a set of built-in reusable utilities. They simplify the most commonly performed operations are:

**String Methods**

* isdigit()
* strip()
* lower()
* upper()
* startswith()
* endswith()
* replace()

 and more...

Isdigit

Syntax:

str\_var.isdigit()

Gives

True

if all the characters are digits. Otherwise,

False

**Code**



1

2

is\_digit = "4748".isdigit()

print(is\_digit)

PYTHON

**Output**



True

Strip

Syntax:

str\_var.strip()

Removes all the leading and trailing spaces from a string.

**Code**



1

2

3

mobile = " 9876543210 "

mobile = mobile.strip()

print(mobile)

PYTHON

**Output**



9876543210

Strip - Specific characters

Syntax:

str\_var.strip(chars)

We can also specify characters that need to be removed.

**Code**



1

2

3

name = "Ravi."

name = name.strip(".")

print(name)

PYTHON

**Output**



Ravi

Strip - Multiple Characters

Removes all spaces, comma(,) and full stop(.) that lead or trail the string.

**Code**



1

2

3

name = ", .. ,, ravi ,, .. ."

name = name.strip(" ,.")

print(name)

PYTHON

**Output**



ravi

Replace

Syntax:

str\_var.replace(old,new)

Gives a new string after replacing all the occurrences of the old substring with the new substring.

**Code**



1

2

3

sentence = "teh cat and teh dog"

sentence = sentence.replace("teh","the")

print(sentence)

PYTHON

**Output**



the cat and the dog

Startswith

Syntax:

str\_var.startswith(value)

Gives

True

if the string starts with the specified value. Otherwise,

False

**Code**



1

2

3

url = "https://onthegomodel.com"

is\_secure\_url = url.startswith("https://")

print(is\_secure\_url)

PYTHON

**Output**



True

Endswith

Syntax:

str\_var.endswith(value)

Gives

True

if the string ends with the specified value. Otherwise,

False

**Code**



1

2

3

gmail\_id = "rahul123@gmail.com"

is\_gmail = gmail\_id.endswith("@gmail.com")

print(is\_gmail)

PYTHON

**Output**



True

Upper

Syntax:

str\_var.upper()

Gives a new string by converting each character of the given string to uppercase.

**Code**



1

2

name = "ravi"

print(name.upper())

PYTHON

**Output**



RAVI

Lower

Syntax:

str\_var.lower()

Gives a new string by converting each character of the given string to lowercase.

**Code**



1

2

name = "RAVI"

print(name.lower())

PYTHON

**Output**



ravi

Notes