**Python Loops**

Python has two primitive loop commands:

* while loops
* for loops

PYTHON WHILE LOOPS

## The while Loop

With the while loop we can execute a set of statements as long as a condition is true.

Print i as long as i is less than 6:

i = 1  
while i < 6:  
  print(i)  
  i += 1  
**OUTPUT**



**The break Statement**

With the break statement we can stop the loop even if the while condition is true:

Example

Exit the loop when i is 3:

i = 1  
while i < 6:  
  print(i)  
  if i == 3:  
    break  
  i += 1

**OUTPUT**

****

**The continue Statement**

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

Example

Continue to the next iteration if i is 3:

i = 0  
while i < 6:  
  i += 1  
  if i == 3:  
    continue  
  print(i)

OUTPUT



**The else Statement**

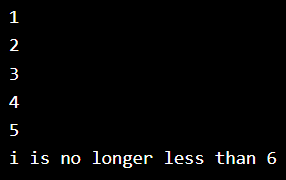
With the else statement we can run a block of code once when the condition no longer is true:

Example

Print a message once the condition is false:

i = 1  
while i < 6:  
  print(i)  
  i += 1  
else:  
  print("i is no longer less than 6")

**OUTPUT**

****

**Python For Loops**

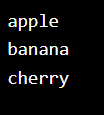
A **for loop** is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

Example:

Print each fruit in a fruit list:

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

**OUTPUT**



**Looping Through a String**

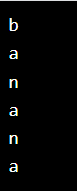
Even strings are iterable objects, they contain a sequence of characters

Example

Loop through the letters in the word "banana":

for x in "banana":  
  print(x)

OUTPUT



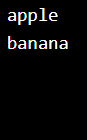
**The break Statement**

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  
**OUTPUT**



**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)

**OUTPUT**

****

**The range() Function**

To loop through a set of code a specified number of times, we can use the range() function,

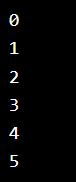
The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

**Example**

Using the range() function:

for x in range(6):  
  print(x)

**OUTPUT**

****

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)

**OUTPUT**

****

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)

**OUTPUT**

****

**Nested Loops**

A nested loop is a loop inside a loop.

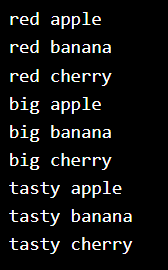
The "inner loop" will be executed one time for each iteration of the "outer loop":

**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)

**OUTPUT**

****