Loop ciklus

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

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



Looping through a string

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



The break statement

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

Exit the loop when x is "banana":

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


the continue statement

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

Do not print banana:  


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

  
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):  


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):  


Else in for loop:

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

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


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

Nested loop

A nested loop is a loop inside a loop.

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


Pass statement

for loops cannot be empty, but if you for some reason have a for loop with no content, put in the pass statement to avoid getting an error.