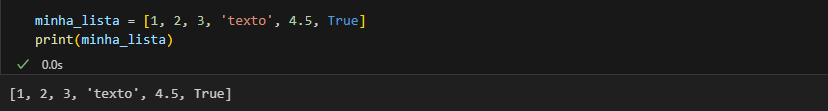
PYTHON lists

[Lists in python](https://www.w3schools.com/python/python_lists.asp)

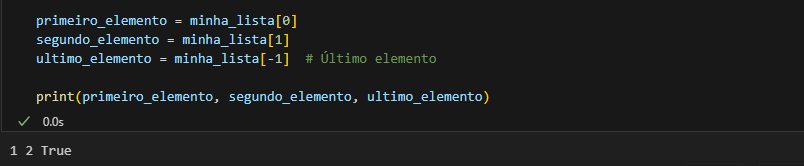
In python, lists are a very versatile and widely used data structure. A list is a changing sequence of elements. Each element can be of any type, and elements can be accessed or modified using indexes;

Creating a list:

You can create a list in python using square brackets " [] " and insert elements separated by commas.

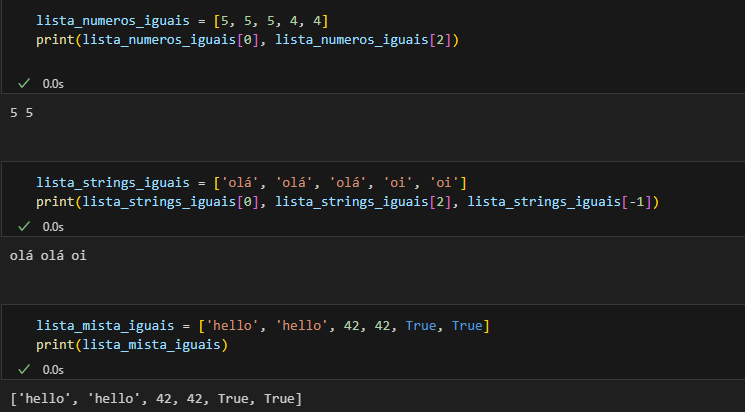


The first item in the list has an index [0], the last has [-1];

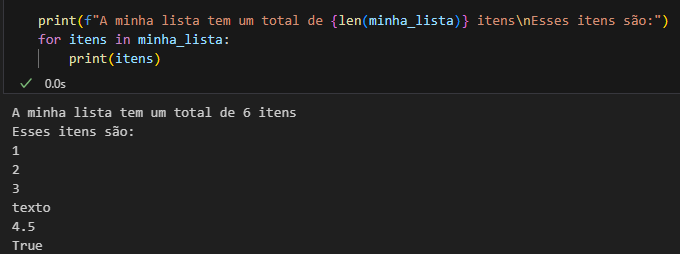


The lists are ordered, this means that the items within it have a set order, and that order will not change;

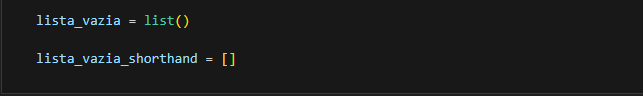
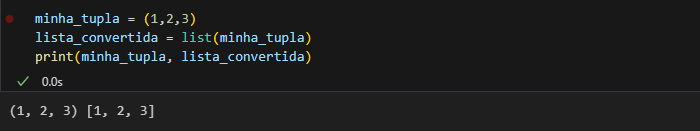
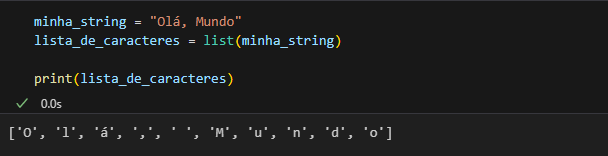
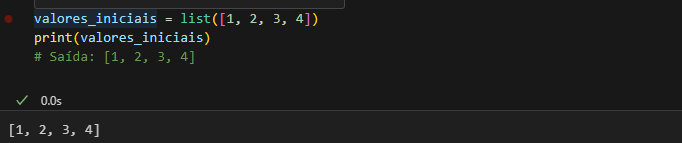
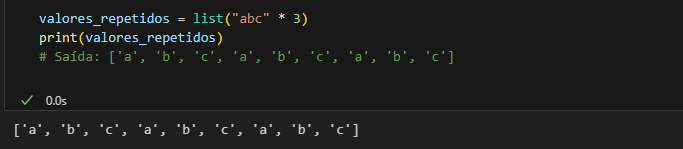
If you add a new item to the list, that new item will be placed at the bottom of the list, now receiving the last position;

As the lists are indexed, this allows them to have items with the same value:

The lenght function can also be used in lists to check how many items are inside them;



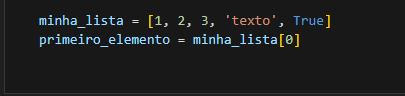
In python, the list() constructor is a built-in function that creates a new list. You can use it to convert other iterable objects, with tuples or strings, into a list, or to create an empty list.

1. Create an empty list:
2.  Convert a Tuple to a List:
3.  Convert a String to a List:
4.  Create a list with starting values:
5.  Create a list with repeating values:

It's important to note that using the list() constructor isn't always necessary, and you can often create lists using square brackets directly. However, the constructor is useful when you need to explicitly convert other iterable types into a list.

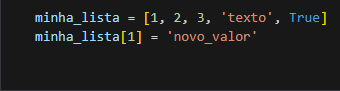
Access to Elements:

The elements of a list are accessed through indexes.



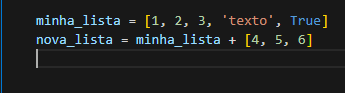
Elements Update:

The elements of a list can be changed through their indexes.



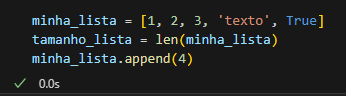
List Operations:

Lists support various operations, such as concatenations and repetition.



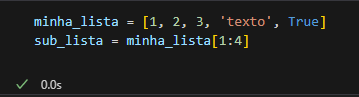
Functions and [Methods](https://www.w3schools.com/python/python_lists_methods.asp):

There are useful functions and methods for working with lists, such as len(), appen(), extend(), remove(), pop(), among others.



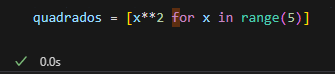
Slicing:

It is possible to use slicing technique to obtain specific portions of a list.



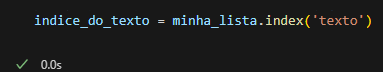
List Comprehension:

It's a concise and powerful construct for creating lists efficiently.



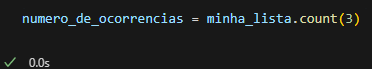
'index()' method:

This method returns the index of the first element corresponding to a specific value.



'count()' method:

Returns the number of occurrences of a given element in the list.



'sort()' method:

Sorts the items in the list.

