# Collections : List - Part 1

Let us get familiar with the ‘Collections’ module on Python.

Collections in Python are containers that store collections of data.
In total, there are for collections on python:

- List
- Dict
- Set
- Tuple

Using these collections we can store any type of data that is related in some manner. Will will study all types of collections in the course...

**List - definition**

Lets begin, by starting with the first Collection type, which is ‘List:

★ In Python, a list is created by placing all the values (elements) inside square brackets [], separated by commas between each value.

★ A list can have any number of values.

★ List can contain a mixed types of values (Integer, Float, String, etc)

**Quick example:**

Creation of a list: my_list = []

A list of integers: my_list = [1, 2, 3]

A Mixed list: my_list = [1, "Hello", 3.4]

**How to access value in a list?**
There are 2 ways to access items inside a list

1. Accessing by index
   We can use the index of a list's value, to access it. But, its important to remember that the first value would have the index 0 (and not 1).

   For example a list with 5 values in it, will have the index '4' as its last one.
   list_of_random_values = [31, 4, 31.5 , 3, "Yes"]

| Value in list cell | 31 | 4 | 31.5 | 3 | "Yes" |
|---|---|---|---|---|---|
| Index of the value | 0 | 1 | 2 | 3 | 4 |

Trying to access other indexes than these, will raise an IndexError.

2. Negative indexing:

   Python allows us to access the values. starting at the end of a list, by adding the prefix '-' to it.

   In other words the index '-1' is referring to the last item of the list (= "Yes"), and index '-2' is referring to the second before last (=3).

   So for the same list :

   List_of_random_values = [31, 4, 31.5 , 3, "Yes"]

   The once accessing to the values using 'Negative indexing", it will look like that

| Value in list cell | 31 | 4 | 31.5 | 3 | "Yes" |
|---|---|---|---|---|---|
| Index of the value | -5 | -4 | -3 | -2 | -1 |

**Accessing the list by index - examples:**

- print(List_of_random_values [0]) -> output : 31
- print(List_of_random_values [3]) -> output : 3
- print(List_of_random_values [-2]) -> output : 3
- print(List_of_random_values [-1]) -> output : "Yes"
- print(List_of_random_values [7]) -> IndexError

# Collections : List - (Advanced)

Let's get familiar with additional manipulations upon a list collections...

First let's declare a new list,
numbers_list = [123 , 111.9 , 10000 , 0.4]

**Get list's length (number of existing cells)**
To find out what is the length of a list we should apply
And command len(numbers_list), and if we print it we would get an integer variable indicating the length.

**Change a value of a specific cell inside a list**
Let's say we want to change the value in cell number '1'
(aka : numbers_list[1]) , and set the value '10' instead of '111.9'.

we can do it by using : numbers_list[1] = 10

**Add an item to the end of an existing list**
Once handling the list -> numbers_list = [123 , 111.9 , 10000 , 0.4]
As we seen in the last video [Lists - practice (basics), bonus assignment],
If we wish to add an item to the end of it, just is '.append(value).

So for adding the number '15', we'd use: numbers_list.append(15)
And we'll have then : numbers_list=[123 , 111.9 , 10000, 0.4 , 15]
Done !

**Add an item to a specific cell item in the list**
Say we want to add '1005' to cell number 2 in the list,
We can achieve it by using numbers_list.insert(1005, 2) - this will cause the existing item in this cell to move forward to the next one.

**Remove an item by mentioning the <u>value</u>**
Let's learn how to remove the item we have just added - '1005',
We can do it by using '.remove()' - `numbers_list.remove(1005)`

**Remove an item by mentioning the <u>cell index</u>**
We have an additional way to remove cells from a list, by using the 'pop()' actions.

We can apply it in 2 ways:

- `numbers_list.pop()` - Will remove the last item of the list, the opposite of 'append()'
- `numbers_list.pop(0)` - Will remove the cell which we mention inside the brackets, in this example cell number '0'