

## Python List

A List is a built-in, linear data structure of Python. It is used to store the data in a sequence manner. We can perform several operations to list, such as indexing, iterating, and slicing. The list comes with the following features.

- The list elements are enclosed with a square bracket, and each element is separated by a comma (,).
- It is a mutable type which means we can modify the list items after their creation.
- The lists are ordered, which means items are stored in a specific order. We can use indexing to access the list element.
- We can store the items of different data types. We can combine strings, integers, and objects in the same list.

Below is an example of a list.

### Example -

1. `list = [31, 60, 19, 12]`
2. `print(list)`
3. `print(type(list))`

**Output:**

```
[31, 60, 19, 12]
<class 'list'>
```

### Example - 2

1. `# creating a list containing elements`
2. `# belonging to different data types`
3. `list1 = [1, "Yash", ['a', 'e']]`
4. `print(list1)`

**Output:**

```
[1, 'Yash', ['a', 'e']]
```

## Array in Python

An Array is also a linear data structure that stores the data. It is also ordered, mutable, and enclosed in square brackets. It can store the non-unique items. But there are restrictions to store the different data type values.

To work with the Array in Python, we need to import either an array module or a Numpy.

1. `import array as arr`
2. or
3. `import numpy as np`

Elements are allocated in contiguous memory location that allows us to easy modification, addition, deletion, accessing of element. Moreover, we need to specify the data type. Let's understand the following example.

### Example -

1. Import array as arr
2. `array_1 = arr.array("i", [31, 60, 19, 12])`
3. `print(array_1)`
4. `print(type(array_1))`

### Output:

```
array('i', [31, 60, 19, 12])
<class 'array.array'>
```

### Example - 2: Using Numpy array

1. `import numpy as np`
2. `array_sample = np.array(["str", 'sunil', 'sachin', 'megha', 'deepti'])`
3. `print (array_sample)`
4. `print(type(array_sample))`

```
5. ['numbers' 'sunil' 'sachin' 'megha' 'deepti']
6. <class 'numpy.ndarray'>
```

## Difference between Array and List

Now, we have a brief introduction and features. Here, we will discuss the differences between the Array and List.

Sr. No	List	Array
1.	The list can store the value of different types.	It can only consist of value of same type.
2.	The list cannot handle the direct arithmetic operations.	It can directly handle arithmetic operations.
3.	We need to import the array before work with the array.	The lists are the build-in data structure so we don't need to import it.
4.	The lists are less compatible than the array to store the data.	An array are much compatible than the list.
5.	It consumes a large memory.	It is a more compact in memory size comparatively list.
6.	It is suitable for storing the longer sequence of the data item.	It is suitable for storing shorter sequence of data items.
7.	We can print the entire list using explicit looping.	We can print the entire list without using explicit looping.
8.	It can be nested to contain different types of elements.	It must contain either all nested elements of same size.

## Conclusion

We have discussed the differences between array and list. Both data types are essential in Python and both have some limitations. Python lists are easy to use in python where arrays are typically used for data analysis.