

### Introduction

Lists are used to store multiple items in a single variable and they are created using square brackets:

```
### Example
my_list = ['Luke', 'San', 'Paras']
print(my_list)
```

**Output:** ['Luke', 'San', 'Paras']

### List Items

List items are ordered, changeable, and allow duplicate values. \ List items are indexed, the first item has index [0], the second item has index [1] etc.

#### Ordered

When we say that lists are ordered, it means that the items have a defined order, and that order will not change. \ If you add new items to a list, the new items will be placed at the end of the list

### Duplicates

Since lists are indexed, lists can have items with the same value:

```
# example
my_list = ['Luke', 'San', 'San', 'Paras']
print(my_list)
```

**Output:** ['Luke', 'San', 'San', 'Paras']

### Length

To determine number of items a list has, use the len() function:

```
print( len (my_list))
```

**Output:** 4

### The list() Constructor

It is also possible to use the list() constructor when creating a new list.

```
new_list = list( ('Luke', 'San', 'Paras'))
```

```
# note the double round- brackets
print(new_list)
```

**Output:** ['Luke', 'San', 'Paras']

### List Items - Same Data Types

List items can be of any data type:

```
my_list2 = [0, 1, 3, 5]
my_list3 = [True, False, False, True]
my_list2, my_list3`
```

**Output:** ([0, 1, 3, 5], [True, False, False, True])

### List Item - Different Data Type

```
# A list can also contain different data types
my_list4 = ['Luke', 0, True]
print(my_list4)
```

**Output:** ['Luke', 0, True]

### Advantages

- List are mutable, i.e., we can update its data.
- lists keep the order of the elements while dictionary does not. So, it is wise to use a list data structure when you are concerned with the order of the data elements
- Lists are highly useful for array operations.

### Disadvantages

- Lists have the limitation that one can only append at the end

### Best practices

Always keep in mind that the input to extend must be an iterable object. This could be a set, a tuple, or another list. You will receive an unexpected response if you attempt to pass in something that is not iterable.

it is typically preferable to use the extend method rather than the append method when adding numerous members to a list at once. This is due to the inefficiency of append, which must generate a new list for each element that is added.

Remember that the extend method alters the existing list. You can use the following code to make a new list with the extra elements: my list + new elements = new list



By San (Bhartik)  
[cheatography.com/bhartik/](https://cheatography.com/bhartik/)

Not published yet.  
Last updated 5th July, 2022.  
Page 1 of 1.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>