**List**

Del Keyword in Python List

The del keyword in Python is used to remove an element or multiple elements from a List. We can also remove all the elements i.e. delete the entire list.

Example

Delete an element from a Python List using the del keyword

#Create a List

myList = ["Toyota", "Benz", "Audi", "Bentley"]

print("List = ",myList)

# Delete a single element

del myList[2]

print("Updated List = \n",myList)

Output

List = ['Toyota', 'Benz', 'Audi', 'Bentley']

Updated List =

['Toyota', 'Benz', 'Bentley']

### Example

Delete multiple elements from a Python List using the del keyword

# Create a List

myList = ["Toyota", "Benz", "Audi", "Bentley", "Hyundai", "Honda", "Tata"]

print("List = ",myList)

# Delete multiple element

del myList[2:5]

print("Updated List = \n",myList)

### Output

List = ['Toyota', 'Benz', 'Audi', 'Bentley', 'Hyundai', 'Honda', 'Tata']

Updated List = ['Toyota', 'Benz', 'Honda', 'Tata']

### Example

Delete all the elements from a Python List using the del keyword

# Create a List

myList = ["Toyota", "Benz", "Audi", "Bentley"]

print("List = ",myList)

# Deletes the entire List

del myList

# The above deleted the List completely and all its elements

### Output

List = ['Toyota', 'Benz', 'Audi', 'Bentley']

## Remove() method in Python List

The remove() built-in method in Python is used to remove elements from a List.

### Example

Remove an element from a Python using the remove() method

# Create a List

myList = ["Toyota", "Benz", "Audi", "Bentley"]

print("List = ",myList)

# Remove a single element

myList.remove("Benz")

# Display the updated List

print("Updated List = \n",myList)

### Output

List = ['Toyota', 'Benz', 'Audi', 'Bentley']

Updated List =

['Toyota', 'Audi', 'Bentley']

Del vs Remove()

Let us now see the difference between del and remove() in Python −

|  |  |
| --- | --- |
| **del in Python** | **remove()** |
|  |  |
| The del is a keyword in Python. | The remove() is a built-in method in Python. |
| An indexError is thrown if the index doesn’t exist in the Python list. | The valueError is thrown if the value doesn’t exist in the Python list. |
| The del works on index. | The remove() works on value. |
| The del deletes an element at a specified index number. | It removes the first matching value from the Python List. |
| The del is simple deletion. | The remove() searches the list to find the item. |

mylist = ["apple", "banana", "cherry"]

## List

Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are [Tuple](https://www.w3schools.com/python/python_tuples.asp), [Set](https://www.w3schools.com/python/python_sets.asp), and [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp), all with different qualities and usage.

Lists are created using square brackets:

### Example

Create a List:

thislist = ["apple", "banana", "cherry"]  
print(thislist)

## 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.

**Note:** There are some [list methods](https://www.w3schools.com/python/python_lists_methods.asp) that will change the order, but in general: the order of the items will not change.

## Changeable

The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

## Allow Duplicates

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

### Example

Lists allow duplicate values:

thislist = ["apple", "banana", "cherry", "apple", "cherry"]  
print(thislist)

## List Length

To determine how many items a list has, use the len() function:

### Example

Print the number of items in the list:

thislist = ["apple", "banana", "cherry"]  
print(len(thislist))

## List Items - Data Types

List items can be of any data type:

### Example

String, int and boolean data types:

list1 = ["apple", "banana", "cherry"]  
list2 = [1, 5, 7, 9, 3]  
list3 = [True, False, False]

A list can contain different data types:

### Example

A list with strings, integers and boolean values:

list1 = ["abc", 34, True, 40, "male"]

## type()

From Python's perspective, lists are defined as objects with the data type 'list':

<class 'list'>

### Example

What is the data type of a list?

mylist = ["apple", "banana", "cherry"]  
print(type(mylist))

## The list() Constructor

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

### Example

Using the list() constructor to make a List:

thislist = list(("apple", "banana", "cherry")) # note the double round-brackets  
print(thislist)

## Python Collections (Arrays)

There are four collection data types in the Python programming language:

* **List** is a collection which is ordered and changeable. Allows duplicate members.
* [**Tuple**](https://www.w3schools.com/python/python_tuples.asp) is a collection which is ordered and unchangeable. Allows duplicate members.
* [**Set**](https://www.w3schools.com/python/python_sets.asp) is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.
* [**Dictionary**](https://www.w3schools.com/python/python_dictionaries.asp) is a collection which is ordered\*\* and changeable. No duplicate members.