**List**

Python has six built-in types of sequences:

The most common ones are lists and tuples.

# List

Datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Items in a list need not be of the same type.

# Defining a list

#!/usr/bin/python

list1 = [1, 2, 3, 4, 5]

list2 = ['a', 'b', 'c', 'd', 'e']

list3 = ["word1", "word2", "word3", "word4", "word5"]

list4 = [1, 2, 'a', 'b', "word1", "word3"]

print (list1)

print (list2)

print (list3)

print (list4)

Output:

[1, 2, 3, 4, 5]

['a', 'b', 'c', 'd', 'e']

['word1', 'word2', 'word3', 'word4', 'word5']

[1, 2, 'a', 'b', 'word1', 'word3']

# Accessing Values in Lists

Use the square brackets for slicing along with the index or indices to obtain value available at that index. Example:

#!/usr/bin/python

list = [1, 2, 'a', 'b', "word1", "word3"]

print (list)

print ("")

print ("Element at index 2 : ", list[2])

print ("Element at index -2: ", list[-2])

print ("")

print ("Element in range 1:4 : ", list[1:4])

print ("Element in range -1:-4 : ", list[-1:-4])

print ("Element in range -4:-1 : ", list[-4:-1])

print ("")

print ("Element in range 1: : ", list[1:])

print ("Element in range -4: : ", list[-4:])

Output:

[1, 2, 'a', 'b', 'word1', 'word3']

Element at index 2 : a

Element at index -2: word1

Element in range 1:4 : [2, 'a', 'b']

Element in range -1:-4 : []

Element in range -4:-1 : ['a', 'b', 'word1']

Element in range 1: : [2, 'a', 'b', 'word1', 'word3']

Element in range -4: : ['a', 'b', 'word1', 'word3']

# Updating Lists

update single or multiple elements of lists by giving the slice on the left-hand side of the assignment operator

**append()** method to add elements in a list

#!/usr/bin/python

list = [1, 2, 'a', 'b', "word1", "word3"]

print ("Before Update: ", list)

print ("")

list[3] = 'd'

list[4] = 'word2'

#list[6] = 'word4' # IndexError: list assignment index out of range

print ("After Update: ", list)

Output:

Before Update: [1, 2, 'a', 'b', 'word1', 'word3']

After Update: [1, 2, 'a', 'd', 'word2', 'word3']

# Delete List Elements

del statement to delete an element, if you know exactly which element(s) you are deleting

**remove()** method if you do not know

#!/usr/bin/python

list = [1, 2, 'a', 'b', "word1", "word3"]

print ("Before Update: ", list)

print ("list[5]: ", list[5])

print ("")

del list[3]

print ("After Update: ", list)

#print ("list[5]: ", list[5]) # IndexError: list index out of range

print ("")

#list[5] = "word3"; # IndexError: list assignment index out of range

print ("After Update: ", list)

#print ("list[5]: ", list[5]) # IndexError: list index out of range

Output:

Before Update: [1, 2, 'a', 'b', 'word1', 'word3']

list[5]: word3

After Update: [1, 2, 'a', 'word1', 'word3']

After Update: [1, 2, 'a', 'word1', 'word3']

# List Operations

1. Indexing
2. Slicing
3. Adding
4. Multiplying
5. Checking for membership

1 & 2 check [Accessing Values in Lists](#_Accessing_Values_in_1)

3 & 4

|  |  |
| --- | --- |
| + | Concatenation - Adds lists on either side of the operator |
| \* | Repetition - Creates new list, concatenating multiple copies of the same list |

#!/usr/bin/python

list1 = [1, 2, 3]

list2 = [4, 5, 6]

list = list1 + list2

print ("list1 : ", list1)

print ("list2 : ", list2)

print ("list1 + list2 : ", list1 + list2)

print ("list1 \* 2 : ", list1 \* 2)

Output:

list1 : [1, 2, 3]

list2 : [4, 5, 6]

list1 + list2 : [1, 2, 3, 4, 5, 6]

list1 \* 2 : [1, 2, 3, 1, 2, 3]