**Python List with examples**

 A list is a data type that allows you to store various types data in it. List is a compound data type which means you can have different-2 data types under a list, for example we can have integer, float and string items in a same list.

**1. Create a List in Python**

Lets see how to create a list in Python. To create a list all you have to do is to place the items inside a square bracket [] separated by comma ,.

# list of floats

num\_list = [11.22, 9.9, 78.34, 12.0]

# list of int, float and strings

mix\_list = [1.13, 2, 5, "beginnersbook", 100, "hi"]

# an empty list

nodata\_list = []

As we have seen above, a list can have data items of same type or different types. This is the reason list comes under compound data type.

**2. Accessing the items of a list**

**Syntax to access the list items:**

list\_name[index]

**Example:**

# a list of numbers

numbers = [11, 22, 33, 100, 200, 300]

# prints 11

print(numbers[0])

# prints 300

print(numbers[5])

# prints 22

print(numbers[1])

Output:

11

300

22

**Points to Note:**  
1. The index cannot be a float number.  
**For example:**

# a list of numbers

numbers = [11, 22, 33, 100, 200, 300]

# error

print(numbers[1.0])

Output:

TypeError: list indices must be integers or slices, not float

2. The index must be in range to avoid IndexError. The range of the index of a list having 10 elements is 0 to 9, if we go beyond 9 then we will get IndexError. However if we go below 0 then it would not cause issue in certain cases, we will discuss that in our next section.  
**For example:**

# a list of numbers

numbers = [11, 22, 33, 100, 200, 300]

# error

print(numbers[6])

Output:

IndexError: list index out of range

**3. Negative Index to access the list items from the end**

Unlike other programming languages where negative index may cause issue, Python allows you to use negative indexes. The idea behind this to allow you to access the list elements starting from the end. For example an index of -1 would access the last element of the list, -2 second last, -3 third last and so on.

**3.1 Example of Negative indexes in Python**

# a list of strings

my\_list = ["hello", "world", "hi", "bye"]

# prints "bye"

print(my\_list[-1])

# prints "world"

print(my\_list[-3])

# prints "hello"

print(my\_list[-4])

Output:

bye

world

hello

**4. How to get a sublist in Python using slicing**

We can get a sublist from a list in Python using slicing operation. Lets say we have a list n\_list having 10 elements, then we can slice this list using colon : operator. Lets take an example to understand this:

**4.1 Slicing example**

# list of numbers

n\_list = [1, 2, 3, 4, 5, 6, 7]

# list items from 2nd to 3rd

print(n\_list[1:3])

# list items from beginning to 3rd

print(n\_list[:3])

# list items from 4th to end of list

print(n\_list[3:])

# Whole list

print(n\_list[:])

Output:

[2, 3]

[1, 2, 3]

[4, 5, 6, 7]

[1, 2, 3, 4, 5, 6, 7]

**5. List Operations**

There are various operations that we can perform on Lists.

**5.1 Addition**

There are several ways you can add elements to a list.

# list of numbers

n\_list = [1, 2, 3, 4]

# 1. adding item at the desired location

# adding element 100 at the fourth location

n\_list.insert(3, 100)

# list: [1, 2, 3, 100, 4]

print(n\_list)

# 2. adding element at the end of the list

n\_list.append(99)

# list: [1, 2, 3, 100, 4, 99]

print(n\_list)

# 3. adding several elements at the end of list

# the following statement can also be written like this:

# n\_list + [11, 22]

n\_list.extend([11, 22])

# list: [1, 2, 3, 100, 4, 99, 11, 22]

print(n\_list)

Output:

[1, 2, 3, 100, 4]

[1, 2, 3, 100, 4, 99]

[1, 2, 3, 100, 4, 99, 11, 22]

**5.2 Update elements**

We can change the values of elements in a List. Lets take an example to understand this:

# list of numbers

n\_list = [1, 2, 3, 4]

# Changing the value of 3rd item

n\_list[2] = 100

# list: [1, 2, 100, 4]

print(n\_list)

# Changing the values of 2nd to fourth items

n\_list[1:4] = [11, 22, 33]

# list: [1, 11, 22, 33]

print(n\_list)

Output:

[1, 2, 100, 4]

[1, 11, 22, 33]

**5.3 Delete elements**

# list of numbers

n\_list = [1, 2, 3, 4, 5, 6]

# Deleting 2nd element

del n\_list[1]

# list: [1, 3, 4, 5, 6]

print(n\_list)

# Deleting elements from 3rd to 4th

del n\_list[2:4]

# list: [1, 3, 6]

print(n\_list)

# Deleting the whole list

del n\_list

Output:

[1, 3, 4, 5, 6]

[1, 3, 6]

**5.4 Deleting elements using remove(), pop() and clear() methods**

remove(item): Removes specified item from list.  
pop(index): Removes the element from the given index.  
pop(): Removes the last element.  
clear(): Removes all the elements from the list.

# list of chars

ch\_list = ['A', 'F', 'B', 'Z', 'O', 'L']

# Deleting the element with value 'B'

ch\_list.remove('B')

# list: ['A', 'F', 'Z', 'O', 'L']

print(ch\_list)

# Deleting 2nd element

ch\_list.pop(1)

# list: ['A', 'Z', 'O', 'L']

print(ch\_list)

# Deleting all the elements

ch\_list.clear()

# list: []

print(ch\_list)

Output:

['A', 'F', 'Z', 'O', 'L']

['A', 'Z', 'O', 'L']

[]