

Lists

Now that we have finished working on numbers and strings, let's work with the lists and sublists.

WE'LL COVER THE FOLLOWING ^

- Lists
- Sublist
- Operations on List
 - List Concatenation
 - Traverse a List

Lists

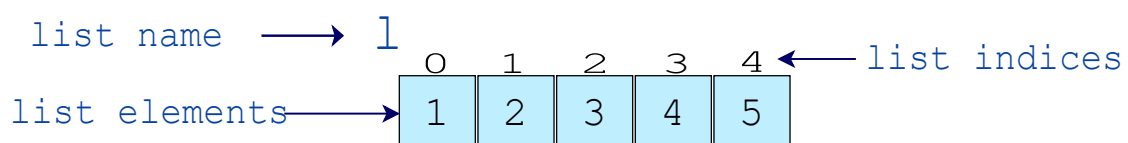
Python lists are data structures that group sequences of elements. Lists **can have elements of several types**, and you can also **mix different types** within the same list (although all elements are usually of the same datatype).

Lists are created using square brackets `[]`, and the elements are separated by commas `(,)`. The elements in a list can be accessed by their positions, starting with 0 as the index of the first element.

```
list=[element1,element2,...]
```

For example, to create list `l` with five elements, write

```
l=[1,2,3,4,5]
```



```
l = [1, 2, 3, 4, 5]
print(l) #prints the entire list

print l[0]#prints value at index 0
print l[1]#prints value at index 1
```



Sublist

Sometimes you want just a small portion of a list—a sublist. Sublists are simply subsets of a list; they can be retrieved using a technique called *slicing*.

A sublist is created by writing the list name, then separating the start and end indexes with a colon (:) and enclosing them within square brackets ([]).

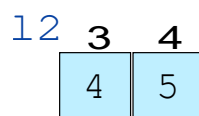
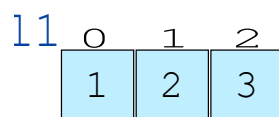
```
sublist=list[startindex:endindex]
```

Given a list `l`, to create sublist `l1` and `l2` write:

```
l=[1,2,3,4,5]

l1=l[0:3]
l2=l[3:5]
```

Note: The start index is inclusive and end index is exclusive in the range.



Write the following python code to print the substring.

```
l = ['a', 'b', 'c', 'd', 'e']  
print l[0:3]
```

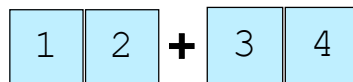


Operations on List

Some basic operations on the lists are explained below:-

List Concatenation

Lists can be concatenated using the **+** operator



1 of 2



2 of 2

—



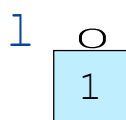
In python,write the following piece of code to concatenate the list:

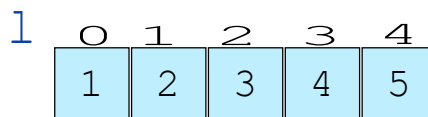
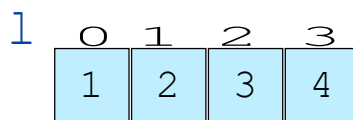
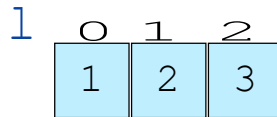
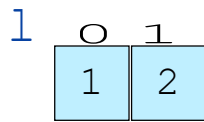
```
print([1,2] + [3,4])
```



Traverse a List

Finally, lists can be iterated over using **for** loops in Python.





-

[]

Use the following piece of code in python to traverse a list element-by-element:

```
l=[1, 2, 4, 8, 10]
for val in l:
    print(val)
```



Now that the basics about the lists and sublists are clear, let's check your knowledge in the upcoming exercises for lists before moving on to the next lesson. (List comprehension)

lesson— List comprehension .