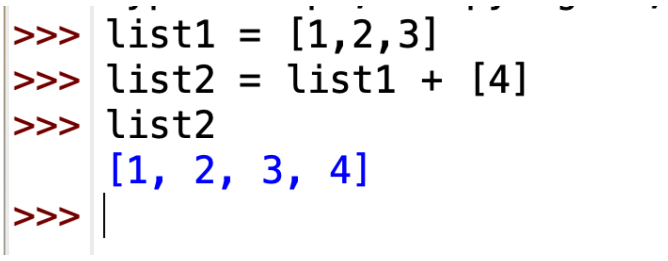
Operators on list

- '+' is also known as concatenation operator
- In list '+' will join two list to make a new list which contains all the elements of two list.
- The '+' doesn't modify the existing list it just simply generate a separate list I.e, a new list.
- If you want to add an element to a list you need to first make it as a list then concatenate it with the list in which you want to add it

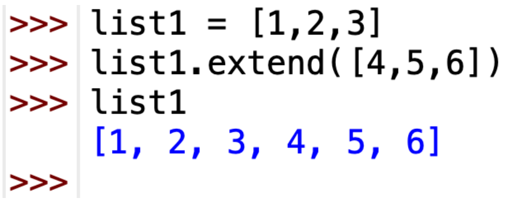
Example :



```
>>> list1 = [1,2,3]
>>> list2 = list1 + [4]
>>> list2
[1, 2, 3, 4]
>>> 
```

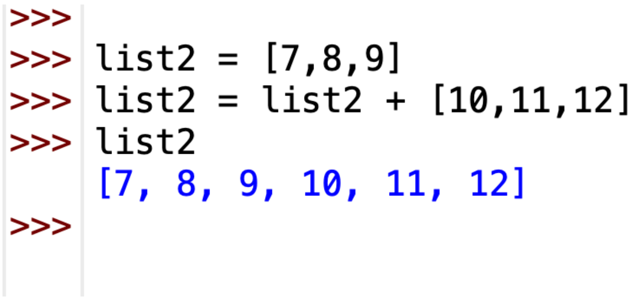
# adding 4 to list1 and storing this in another list I.e list2

- If you want to expand/ modify an existing string then you can use a method called **extend( )** to include more elements.
- By using **extend( )** you can change a list.



```
>>> list1 = [1,2,3]
>>> list1.extend([4,5,6])
>>> list1
[1, 2, 3, 4, 5, 6]
>>>
```

- You can modify the list as follows as well

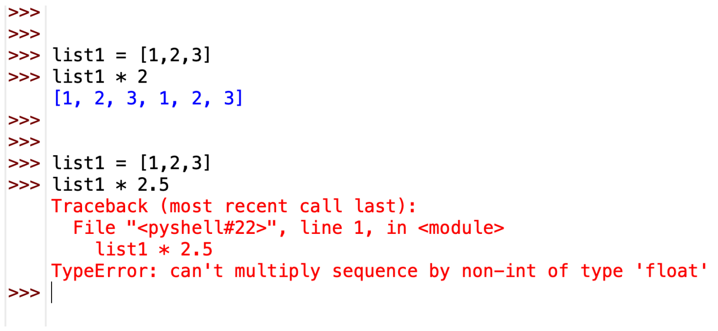


```
>>>
>>> list2 = [7,8,9]
>>> list2 = list2 + [10,11,12]
>>> list2
[7, 8, 9, 10, 11, 12]
>>>
```

Here, list2 is modified simply , by assigning the old list2 to new list2 and concatenating old list2 with another list hence , modified new list2 is created by concatenation.

- ‘ * ’ is a repetition operator.
- Float values cannot be used for repetition only integer values should be used.

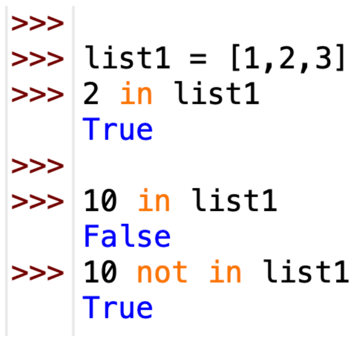
Example:



```
>>>
>>>
>>> list1 = [1,2,3]
>>> list1 * 2
    [1, 2, 3, 1, 2, 3]
>>>
>>>
>>> list1 = [1,2,3]
>>> list1 * 2.5
    Traceback (most recent call last):
      File "<pyshell#22>", line 1, in <module>
        list1 * 2.5
    TypeError: can't multiply sequence by non-int of type 'float'
>>>
```

- The other two operators are **in , not in**
- They check if an element is present in the list or not and return boolean type of result.
- Let us see this with an example

Example:



```
>>>
>>> list1 = [1,2,3]
>>> 2 in list1
    True
>>>
>>> 10 in list1
    False
>>> 10 not in list1
    True
```

- If it is present then it will return true if not than it will return false