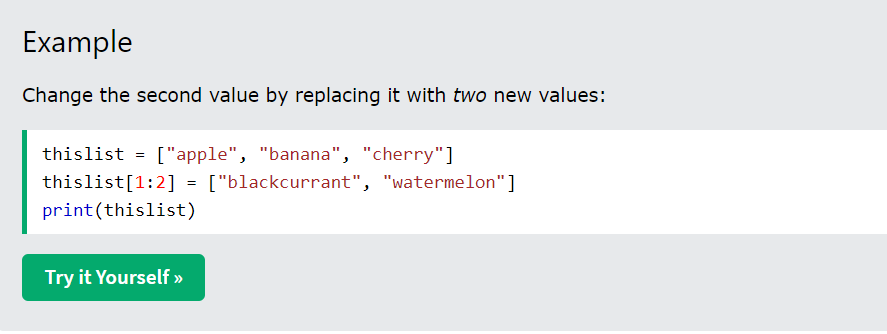


All of list, tuples, set and dictionary are iterable

All of them allows for…in… loop

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Starting List**



Same thing we did with strings

**.append( )**

This method appends new items at the end of list

**.pop ( )**

this method removes the last item of list and returns that removed item if not provided any index value

**.pop( index\_value )**

Removes the item of the given index, and returns that item

**.remove( name\_of\_item )**

Removes the named item

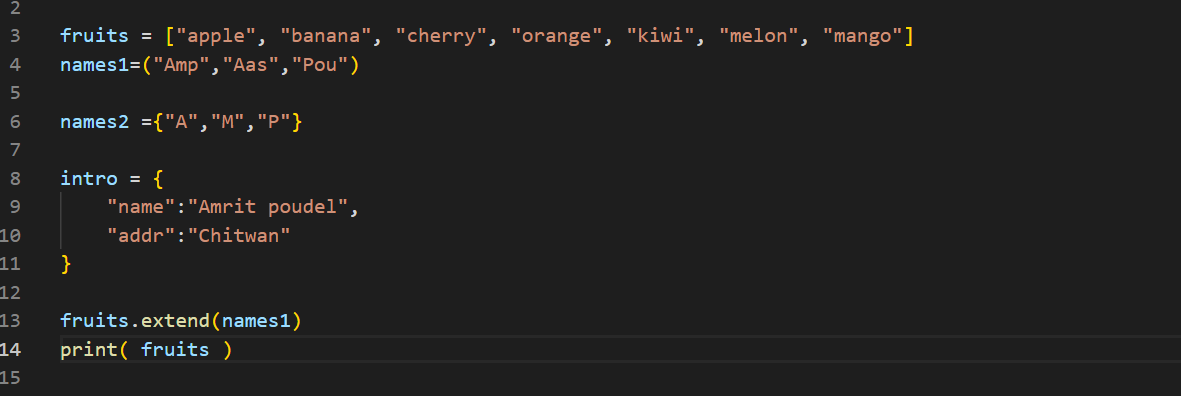
**.insert( index\_number, item\_to\_be\_inserted )**

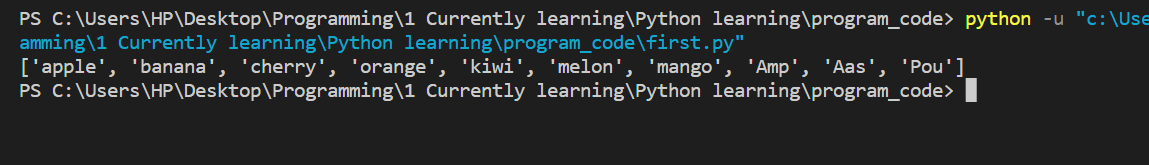
Inserts item at the designated index position

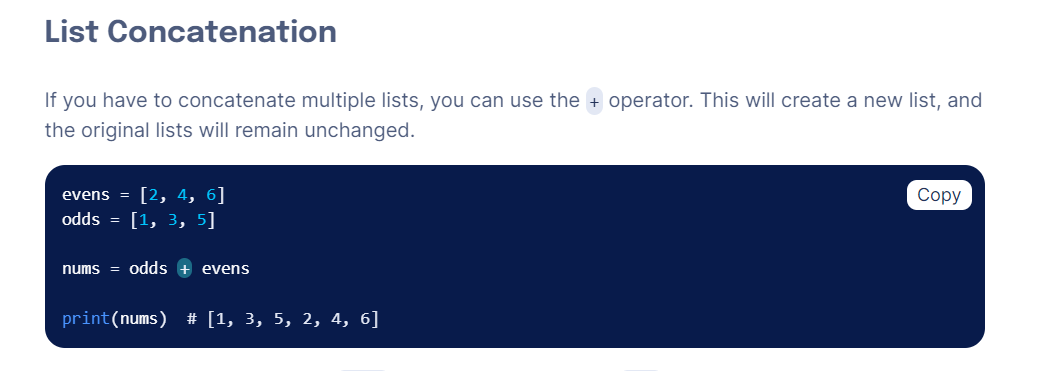
**.extend( any\_iterable )**

Extends the given list by appending elements from the provided iterable

All list, tuples, set and dictionary are iterable





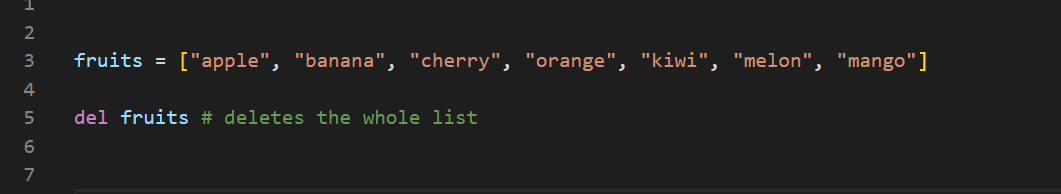


**del keyword**

it can delete the specified list item

it can also delete the whole list

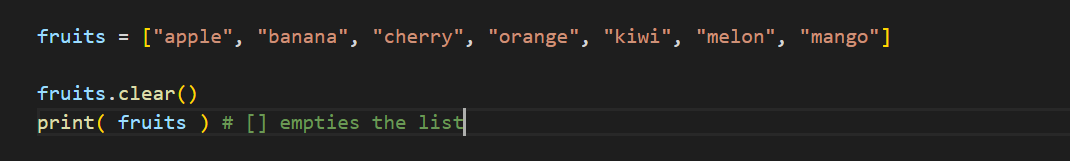




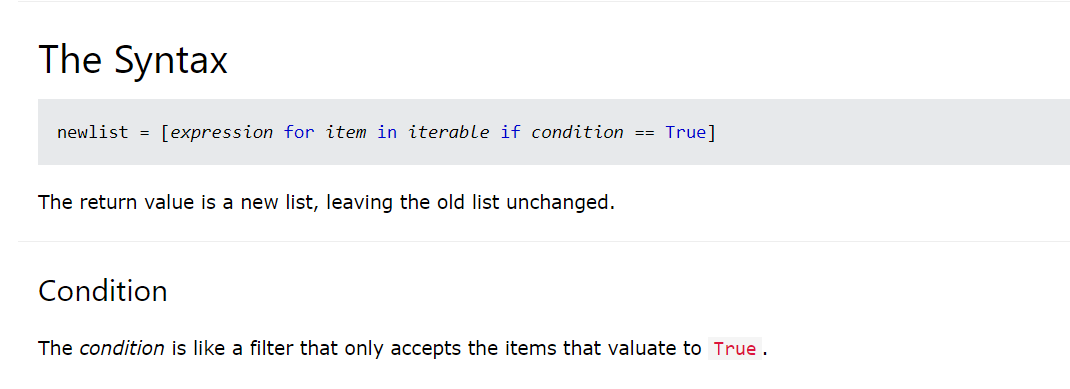
**.clear( )**

It clears the list

It empties the list



**List Comprehension**

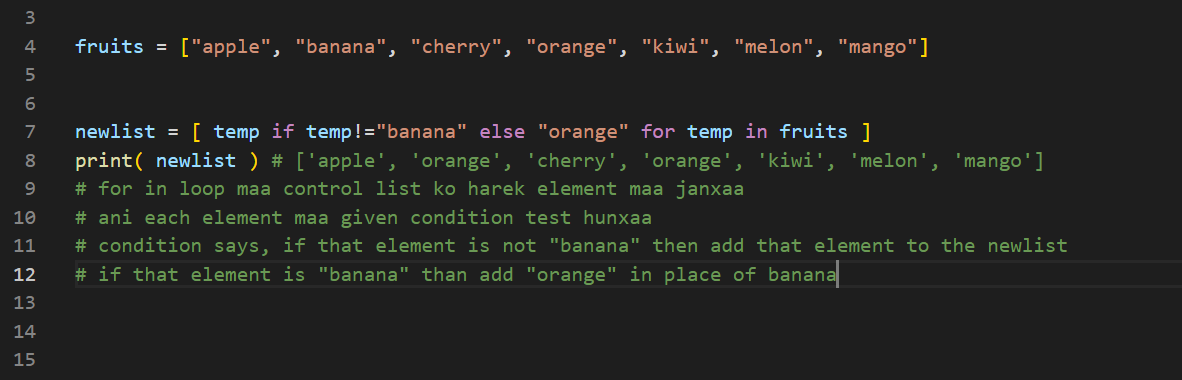


Using this we can get our desired list from the existing list

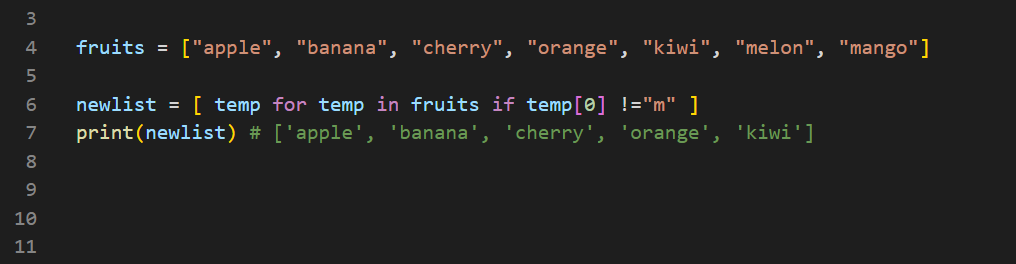


**This is super useful**

**Case1: Using condition before the for…in.. iteration**

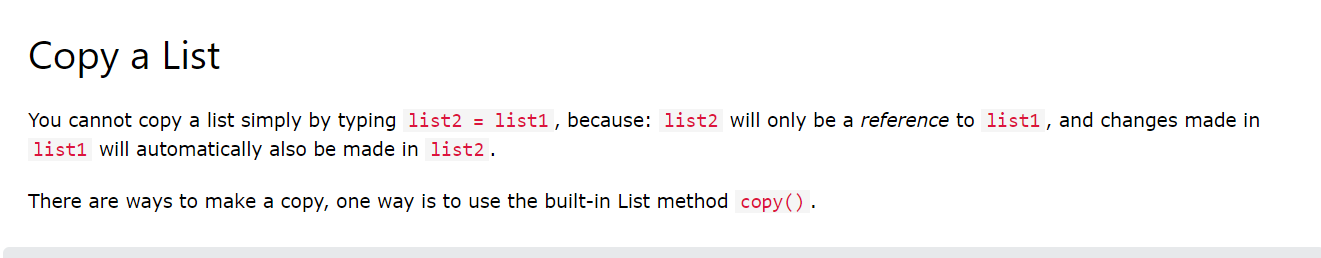


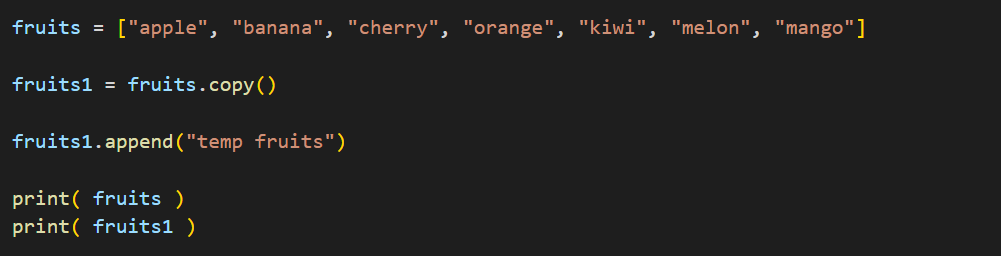
**Case2: Using condition after the for…in… iteration**



**Here we do not need to specify else condition**

We can drive different list from the given list based on our need

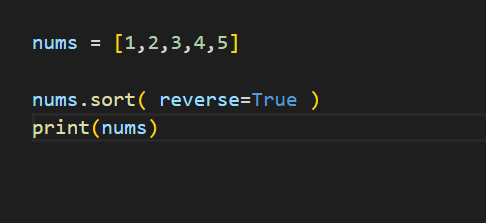




Sorting list in Ascending order



Sorting list in descending order



Reverse the list

