**Python In-built Data Structures:**

**Data Structure:**  A Data Structure is a collection of data elements(such as members or characters – or even other data structures) that is structured in some way, for example by numbering the elements. The most basic data structure in Python is the “Sequence”.

There are four inbuilt data structure in the python.

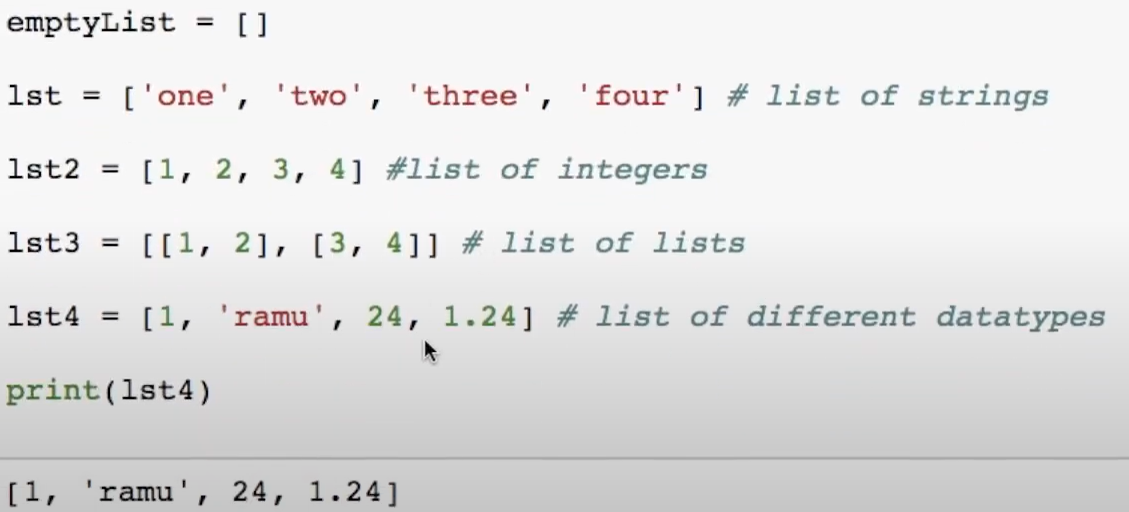
* List
* Tuple
* Set
* Dictionary

**Python List:**

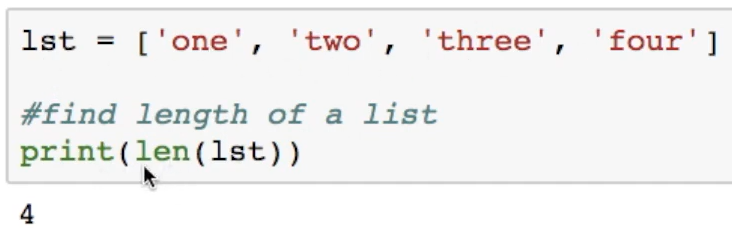
* List is the one of the Sequence Data Structure.
* Lists are collection of items (Strings, integers or even other lists)
* Lists are enclosed in [].
* Each item in the list has an assigned **index** value.
* Each item in a list is separated by a comma.
* Lists are **mutable**, which means items in a list can be changed.

**List Functions:**

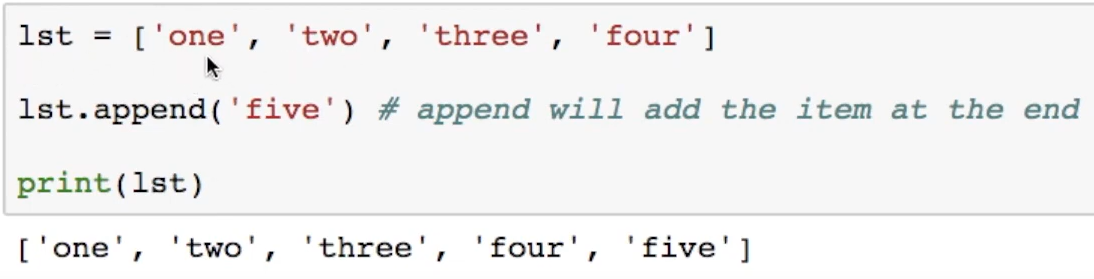
List Creation:



List Length: To find the length of list.

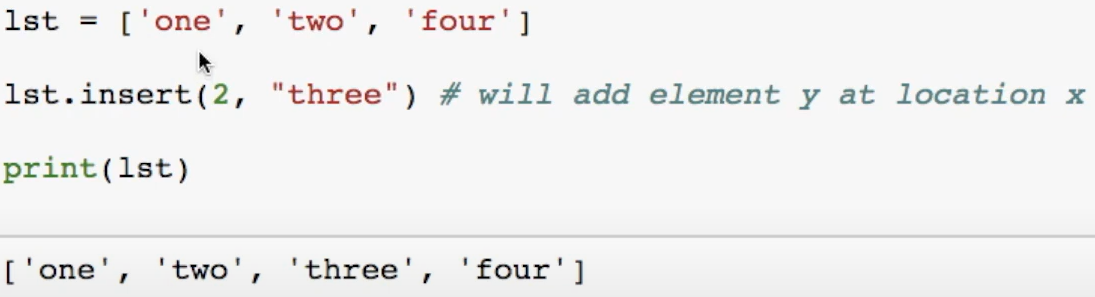


List append: Append will add the items at the end.

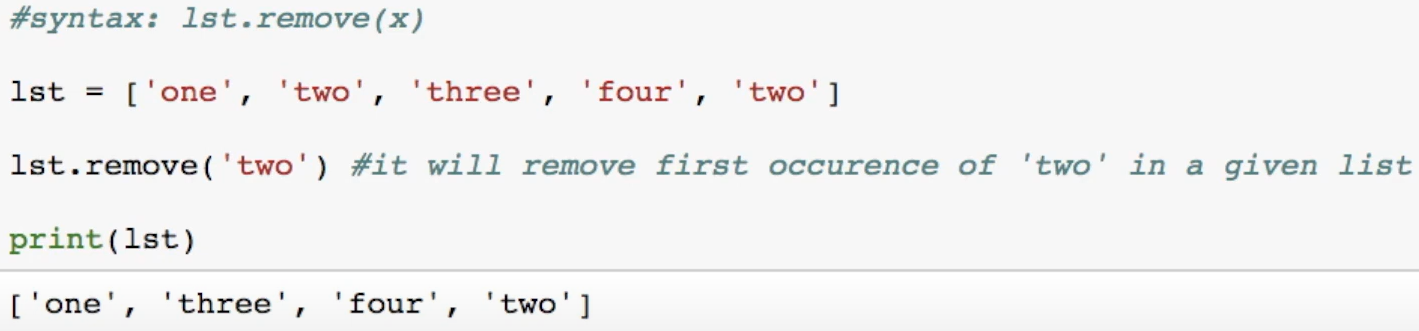


List Insert:

Syntax: **lst.insert(x,y)** # Will add element y at index x

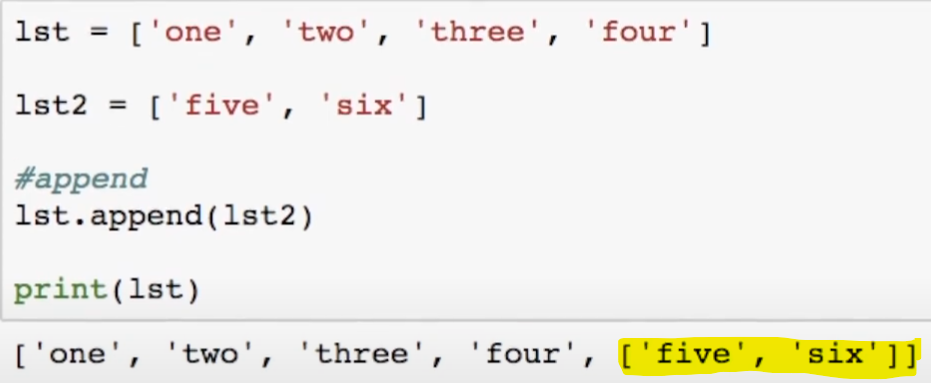


List Remove: SYNTAX: **lst.remove(x)** # It Always remove the first occurance of ‘two in the given list.

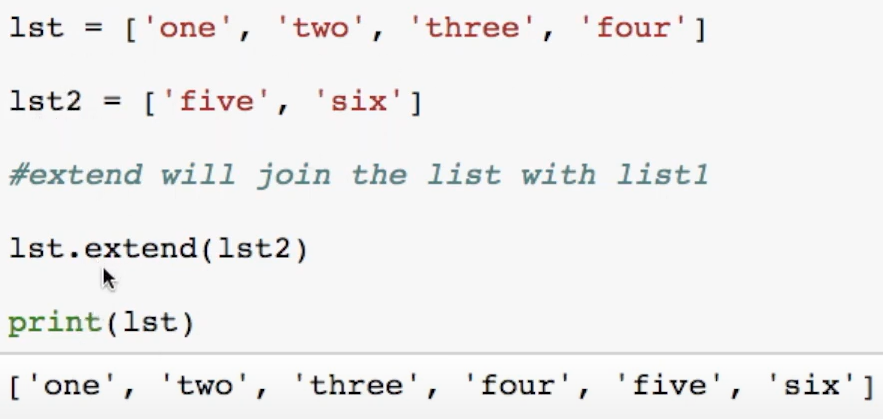


List Append & Extend:

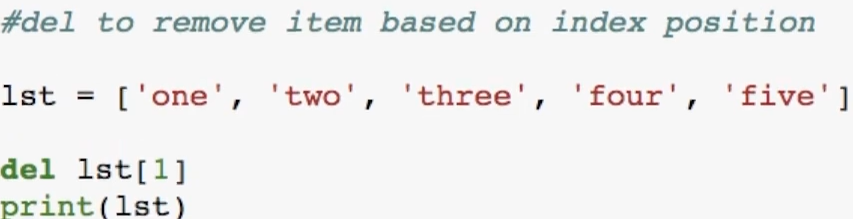
Append: # Always appends the element/list/ at the end.

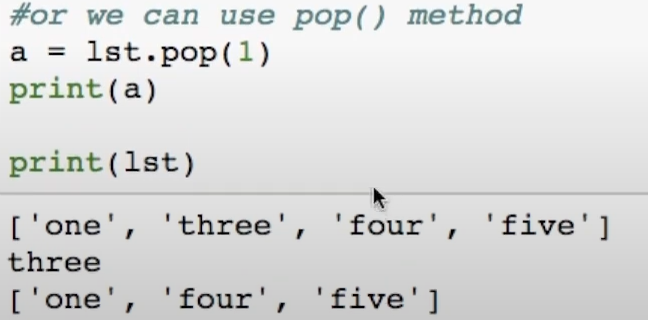


Extend:



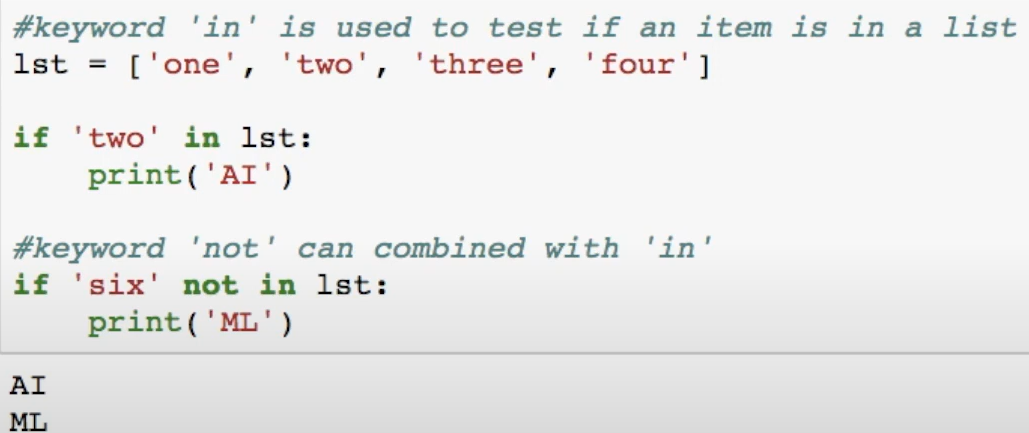
List Delete/pop:



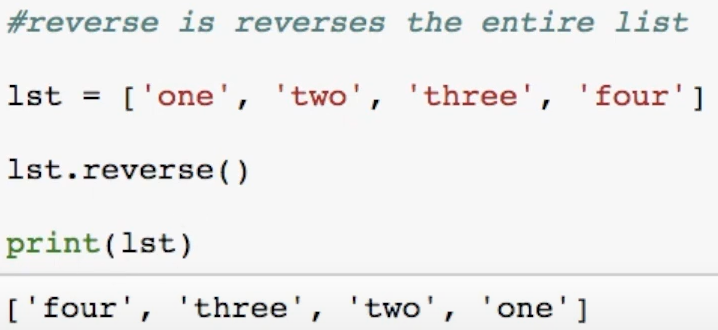
****

**List Related keywords in Python:**

**In & not :**



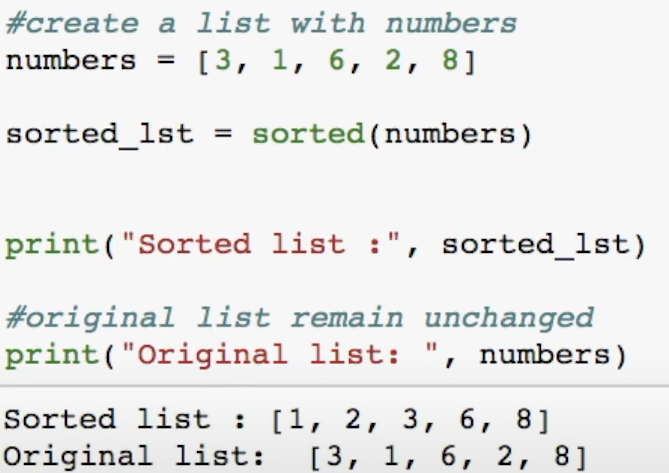
**List Reverse:**

****

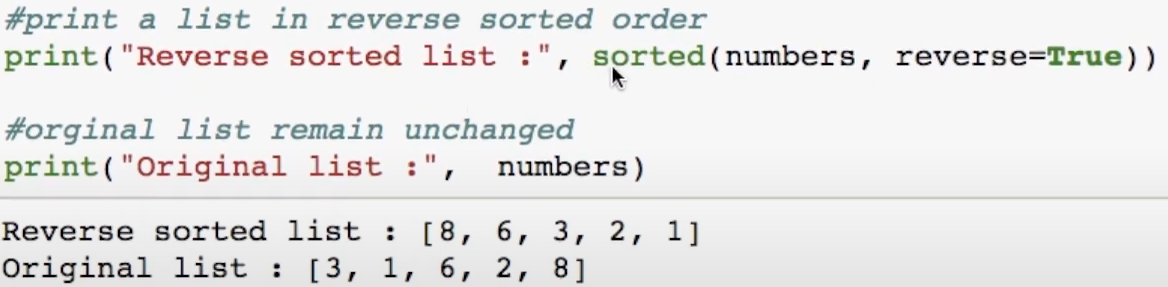
**List Sorting:**

* The easiest way to sort a List is with the sorted(list) function.
* That takes a list and returns a new list wit those elements in sorted order.
* The original list is not changed.
* The sorted() optional argument **reverse=True**, e.g, **sorted(list,reverse=True)**, makes it sort backwards.

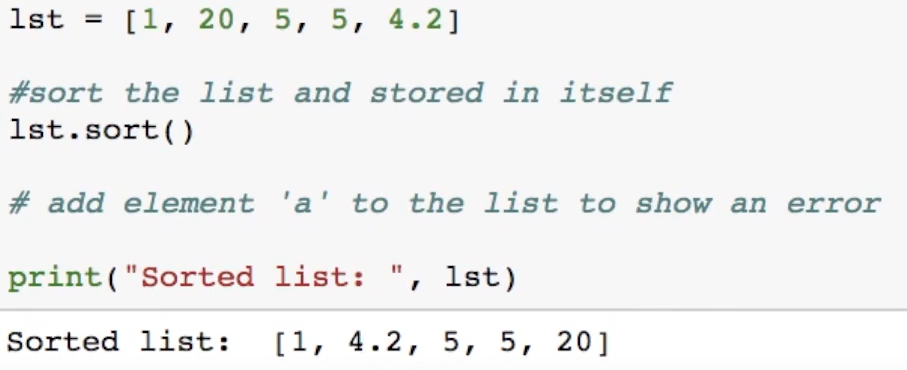
**Sort Ascending:**



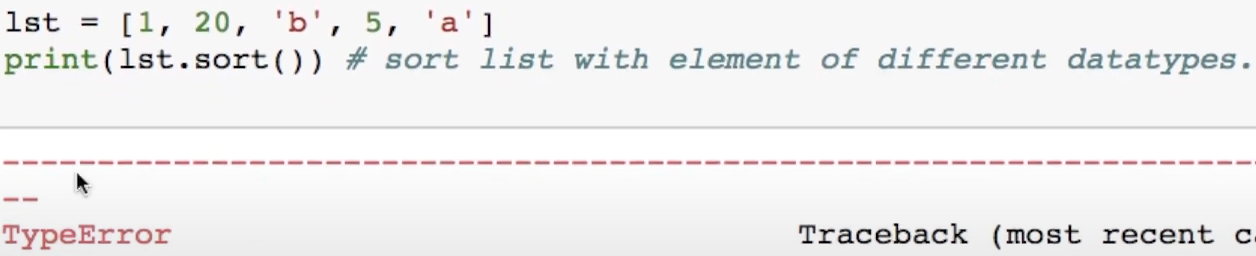
**Sort Descending:**

****

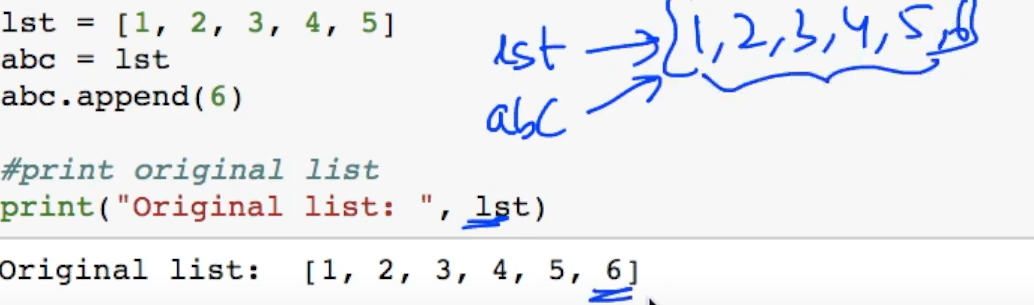
**Using sort(): op will be sorted in asc order.**

****

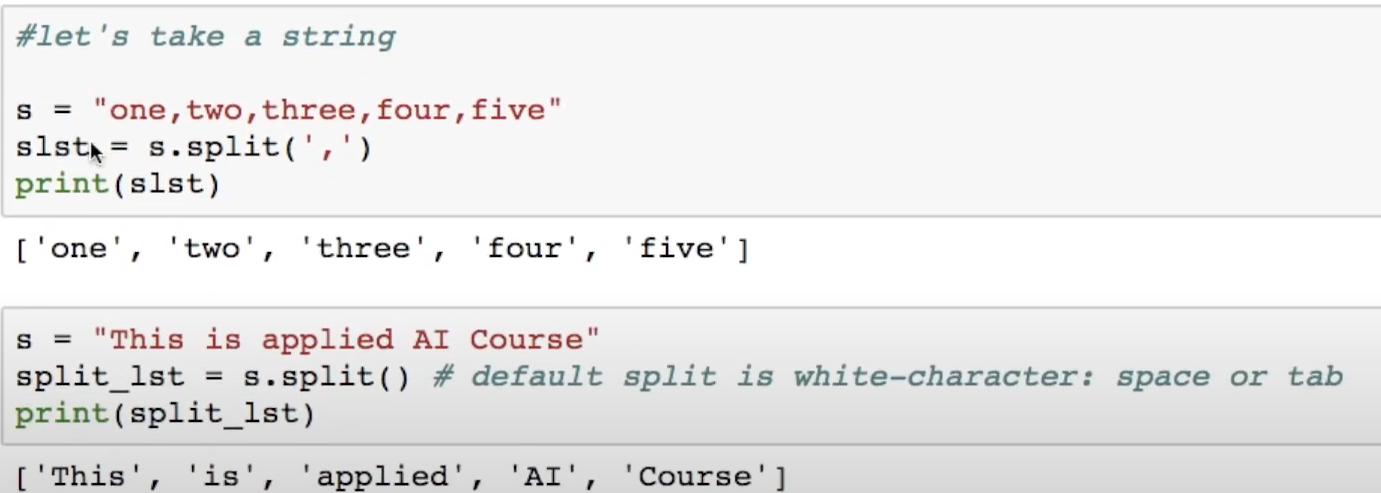
**Using Sort we cannot compare list of multiple datatype:**

****

**List having Multilpe References:**

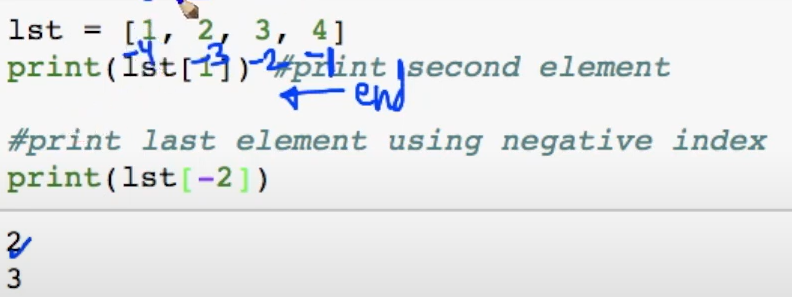
****

**String Split to create a list:**

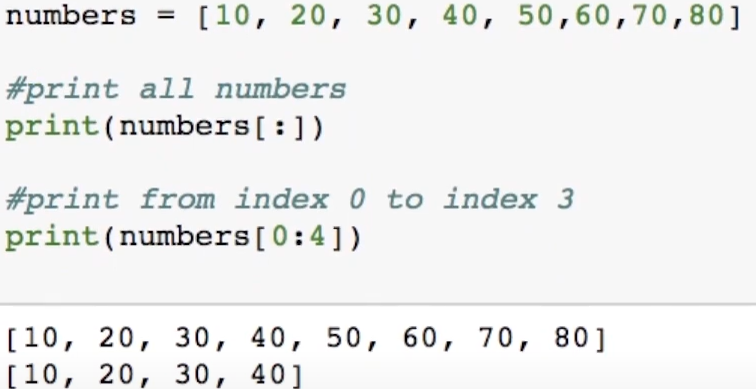
****

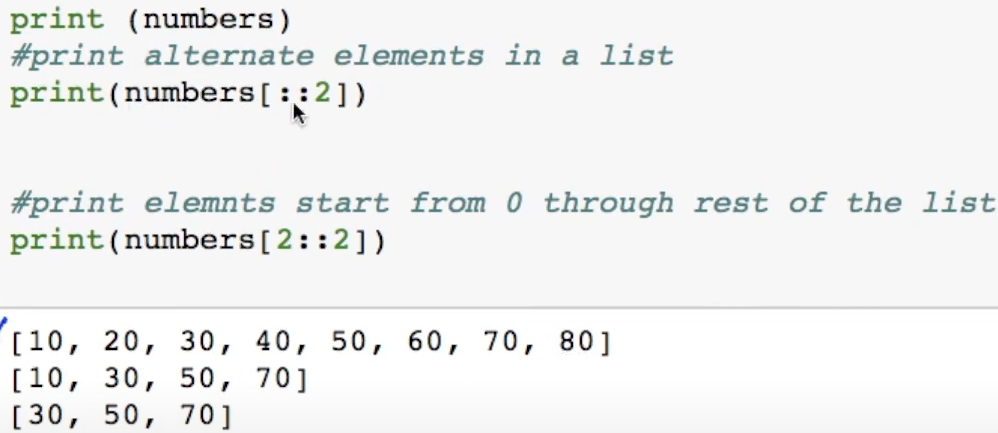
**List Indexing:**

* Each Item in the list has an assigned index value starting from 0.
* Accessing elements in a list is called indexing.

****

**List Slicing:**  Accessing parts of segments is called Slicing. The Key point to remember is that the **:end** value represents the first value that is not in the selected slice. **[start : end : step]**

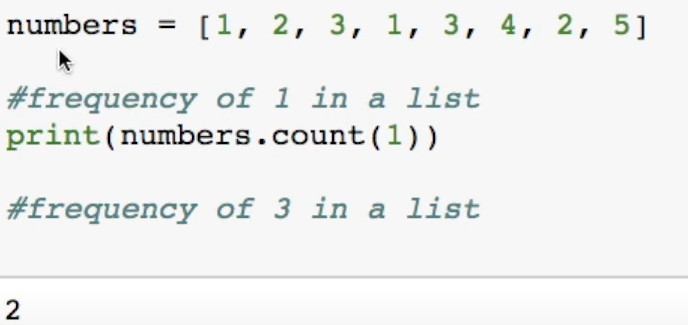


****

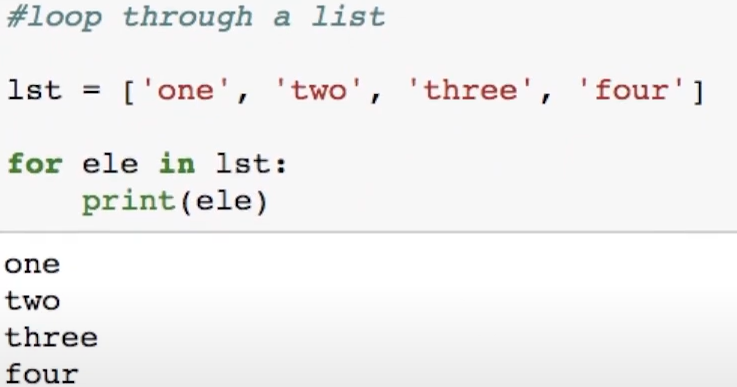
**List extend using “+” :**

****

**List Count:**

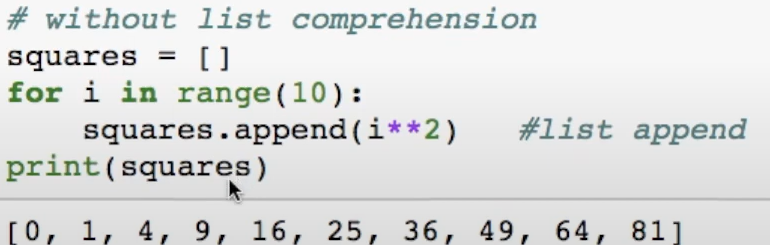
****

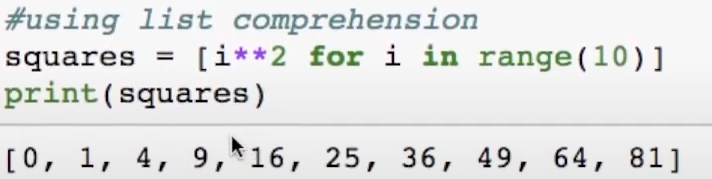
**List Looping:**

****

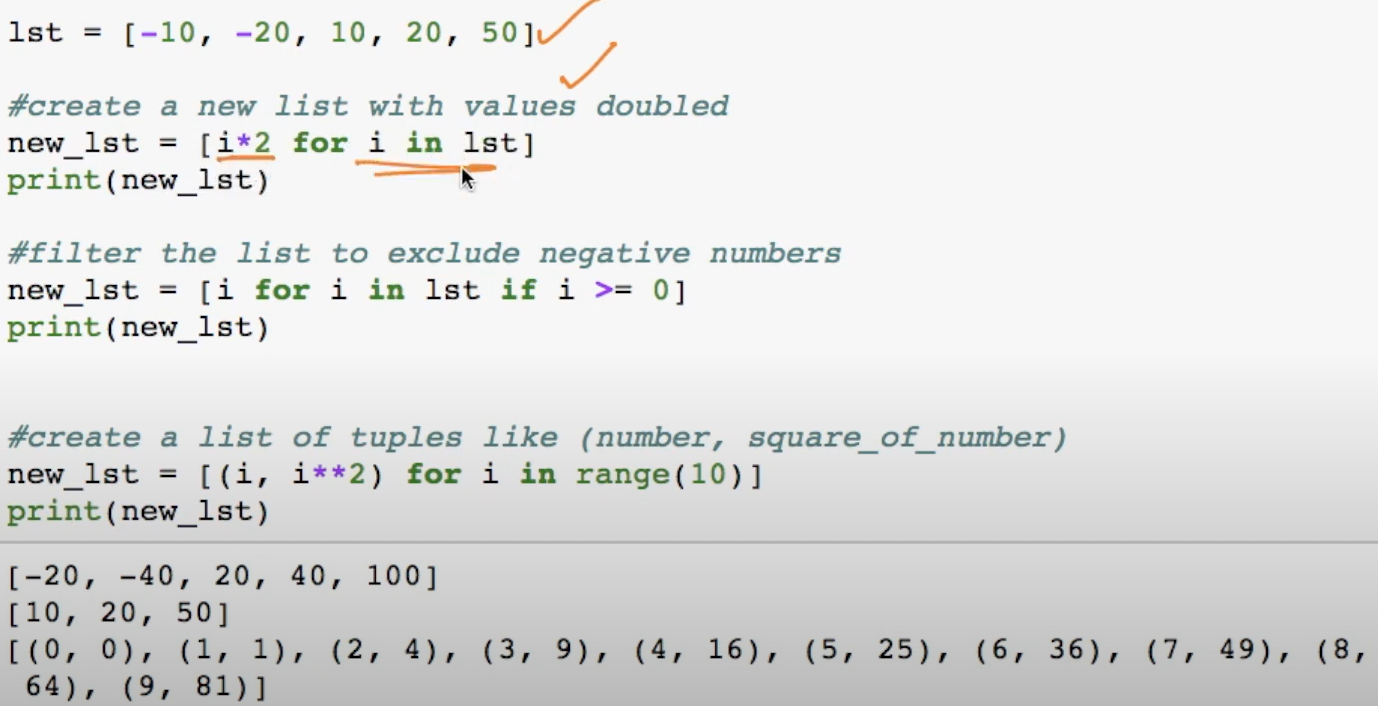
**List Comprehensions:**

* List comprehensions provide a concise way to create lists.
* Common applications are to make new lists where each element is the result of some operations applied to each member of another sequence or iterable, or to create a subsequence of those elements that satisfy a certain condition.





**More Examples:**

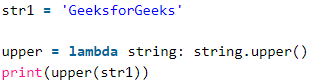
****

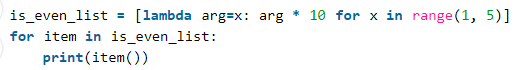
**Nested List Comprehensions:**

****

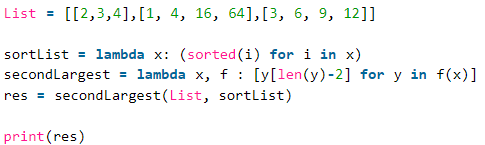
**Lambda Functions in python:**

Python Lambda Functions are anonymous functions means that the function is without a name. As we already know the *def* keyword is used to define a normal function in Python.  Similarly, the *lambda* keyword is used to define an anonymous function in [Python](https://www.geeksforgeeks.org/python-programming-language/).



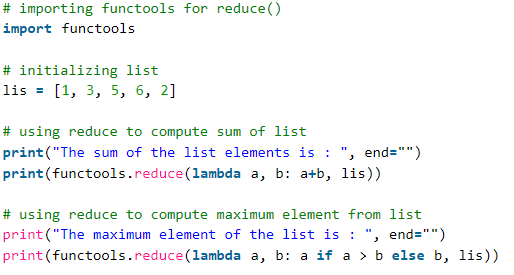






**Reduce():**

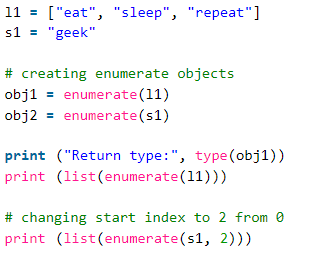
The reduce(fun,seq) function is used to apply a particular function passed in its argument to all of the list elements mentioned in the sequence passed along. This function is defined in “functools” module.

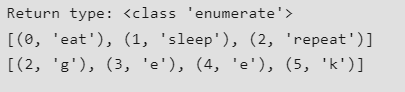




**Enumerate():**

Often, when dealing with iterators, we also need to keep a count of iterations. The enumerate () method adds a counter to an iterable and returns it in the form of an enumerating object. This enumerated object can then be used directly for loops or converted into a list of tuples using the list() function.



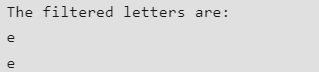


**Filter():**

**Python filter with a custom Fucntion:**

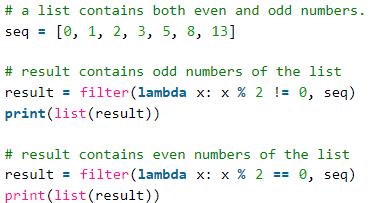
The filter() method filters the given sequence with the help of a function that tests each element in the sequence to be true or not.



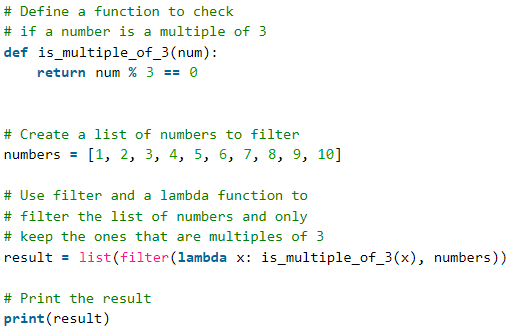


**Filter function in python with lambda:**

Python filter() function is normally used with [Lambda functions](https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/). In this example, we are using the lambda function to filter out the odd and even numbers from a list.



**Filter function in python with lambda and custom function:**



**map, filter, and reduce** - *Map, Filter, and Reduce are built-in Python functions that can be used for functional programming tasks.* With the help of these operations, you may apply a specific function to sequence items using the 'map', filter sequence elements based on a condition using the 'filter', and cumulatively aggregate elements using the 'reduce'.

