Tuples in python are very similar to lists but are also very restricted, as once they are created they are unchangeable. Like lists, tuples are ordered, indexed and can hold duplicate members, but the values inside can never be changed or appended. Like lists, count(), index(), and len() methods can be used to retrieve variables from the tuple, but methods such as append(), clear(), extend(), remove() cannot be used as they are methods for modification.

Like lists, it is still possible to iterate through a tuple, for example using a for loop:

myTuple = (‘desk’, ‘chair’, ‘computer’)

for x in myTuple:

print(x)

or even able to iterate through the tuple in reverse:

myTuple = (‘desk’, ‘chair’, ‘computer’)

for x in myTuple[::-1]:

print(x)

If for some reason you were using a tuple and wanted to make changes, you would actually have to convert the tuple into a list, modify that list, then create a new tuple with the modified list:

tuple1 = (‘desk’, ‘chair)

myList = list(tuple1)

myList.append(‘computer’)

tuple1 = tuple(mylist)

This can become rather repetitive, as the original tuple still exists along with the new tuple and the modifier list. If this were done repeatedly it would be a waste of time, commands, and computer resources and one would be better off just using a list to begin with.

Tuples are best suited for storing data that needs to be read-only and protected, so a programmer can read the data but not write to or erase it.