Adding Elements In a Dictionary

Since a dictionary is mutable, we can add or delete entries from the dictionary. This is particularly useful if we have to maintain a dynamic data structure. To assign a value corresponding to a given key *(This includes over-writing the value present in the key or adding a new key-value pair)*, we can use the square bracket operators to simply assign the value to a key.

If we want to update the value of an already existing key in the dictionary then we can simply assign the new value to the given key. This is done as follows:

```
>>> bar= {2:1,3:2,4:3}
>>> bar[3]=4
# This operation updates the value of the key 3 to a new value i.e. 4.
>>> print(bar)
{2:1,3:4,4:3}
```

Now if we want to add a new key-value pair to our dictionary, then we can make a similar assignment. If we have to add a key-value pair as `"man":"boy"`, then we can make the assignment as:

```
>>> bar["man"]="boy"
>>> print(bar)
{2:1,3:2,4:3,"man":"boy"}
```

Adding or concatenation of two dictionaries:

If we have 2 dictionaries and we want to merge the contents of both the dictionaries and form a single dictionary out of it . It is done as follows:

```
a= {1:2,2:3,3:4}
b= {7:2,10:3,6:4}
a.update(b)
print(a)
--> {1:2,2:3,3:4,7:2,10:3,6:4}
```

In this process, the second dictionary is unchanged and the contents of the second dictionary are copied into the first dictionary. The uncommon keys from the second dictionary are added to the first with their corresponding values. However, if these dictionaries have any common key, then the value of the common key present in the first dictionary is updated to the new value from the second.

Deleting an entry:

In order to delete an entry corresponding to any key in a dictionary , we can simple pop the key from the dictionary. The method used here is `.pop()` . This method removes the key-value pair corresponding to any particular key and then returns the value of the removed key-value pair.

```
>>> c={1:2,2:(3,23,3),3:4}
>>> c.pop(2)
(3,23,3)
```

Deleting all the entries from the dictionary:

If we want to clear all the key-value pairs from the given dictionary and thus convert it into an empty dictionary we can use the `.clear()` method.

```
>>> c={1:2,2:(3,23,3),3:4}
>>> c.clear()
>>> print(c)
{}
```

Deleting the entire dictionary:

We can even delete the entire dictionary from the memory by using the `del` keyword. This would remove the presence of the dictionary. This is similar to tuples and lists.