Dictionary

Dictionary is like a collection of lockers with values inside it.

It should not have same keys but can have same values for any number of keys. If you type same key in it then the value of key will be the last assigned value.

You can’t add two dictionary but can use update to add two dictionary

Built-in Operations

dict1 = {'maths':23,'science':22}	
>>> dict1.items() dict_items([('maths', 23), ('science', 22)])	It will return the keys and values that are present in dictionary

>>> dict1
{'maths': 23, 'science': 22}
>>> dict1.keys()
dict_keys(['maths', 'science'])
	It will return the keys present in the dictionary

>>> dict1
{'maths': 23, 'science': 22}
>>> dict1.values()
	It will return the values present in the dictionary

>>> dict1
{'maths': 23, 'science': 22}
>>> dict1['chemistry'] = 20
>>> dict1
{'maths': 23, 'science': 22, 'chemistry': 20}
>>> dict1['chemistry'] = 25
>>> dict1['chemistry'] = 25
>>> dict1
{'maths': 23, 'science': 22, 'chemistry': 25}
	var_name[key] = value
	Firstly it will check whether the key is present in the dictionary or not.
	Secondly it will add the key and values to the dictionary if key is not present in dictionary.
	if key is present in dictionary then it will change the value of the key in the dictionary

>>> dict1
{'maths': 23, 'science': 22, 'chemistry': 25}
>>> dict2 = {'physics':21}
>>> dict1.update(dict2)
>>> dict1
{'maths': 23, 'science': 22, 'physics': 21, 'chemistry': 25}
	update()
	It is used to add another dictionary to it

>>> dict1
{'maths': 23, 'science': 22, 'physics': 21, 'chemistry': 25}
>>> del dict1['chemistry']
>>> dict1
{'maths': 23, 'science': 22, 'physics': 21}
	del dict_name[key]
	It will delete the respected key and value from the dictionary

>>> dict1.setdefault('chemistry',20)
20
>>> dict1
{'maths': 23, 'science': 22, 'physics': 21, 'chemistry': 20}
>>> dict1.setdefault('chemistry',13)
20
>>> dict1
{'maths': 23, 'science': 22, 'physics': 21, 'chemistry': 20}
	setdefault(key,value)
	It will check whether the key is in dictionary if it is not present then it will add that key and value to it.
	If the key is already present in the dictionary it will not do anything.

>>> dict1
{'maths': 23, 'science': 22, 'chemistry': 20}
>>> dict1.get('maths','maths not in dictonary')
23
>>> dict1.get('physics','physics not in dictonary')
'physics not in dictonary'
	get(key,error_message)
	It will return the value of the key if it is present in the dictionary
	if not then display the error_message

>>> dict1
{'maths': 23, 'science': 22, 'chemistry': 20}
>>> dict1.clear()
>>> dict1
{}
	clear()
	It will clear all the elements in the dictionary (both keys and values)

>>> dict1 = {'maths':23,'science':20}
>>> del dict1
>>> dict1
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'dict1' is not defined
	del dict_name
	It will delete the dictionary completely if you try to access it after deleting it will show error.