Dictionary

Topics Covered

- Defining dictionary
- Accessing values
- updating dictionary
- deletion
- compare, len, string representation
- Methods
  - clear
  - copy
  - fromkeys
  - get
  - has_key
  - items
  - keys
  - values
  - update

Key Concepts

- key-value
- Keys are immutable
- Duplicate key is **NOT** allowed

Declaring a Dictionary

- Each key is separated from its value by a colon (:)
- the items are separated by commas, and the whole thing is enclosed in curly braces.
- An empty dictionary without any items is written with just two curly braces - {}
- Keys are unique
- values may or may not be unique.
- The values of a dictionary can be of any type

In [1]:

```
my_dict = {}              # Empty Dictionary
my_dict_1 = dict()        # Empty Dictionary
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"]}

print (my_dict)
```

```
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4']}
```

Accessing a dictionary

- Dictionary can be assessed by its key
- If key does not exist, python raises KeyError

In [2]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}
print (my_dict['1701'])
print (my_dict['1702'])
# print my_dict['1703']                   # Key Error is raised as 1703 is not the key in dictionary
```

```
['Sanam', 'CS', '9.01', 'sem-4']
['Sachin', 'EC', '8.0', 'sem-4']
```

Updating Dictionary

- adding a new entry or a key-value pair,
- modifying an existing entry, or
- deleting an existing entry

In [3]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

my_dict['1703'] = ["Ajit","Mech","8.89","sem-5"]             # adding new element
print (my_dict)

my_dict['1704'] = ["Piyush","ELEC","8.89","sem-5"]           # updating exisitng element
print (my_dict)
```

```
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': 'Piyush', '1703': ['Ajit',
'Mech', '8.89', 'sem-5']}
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': ['Piyush', 'ELEC', '8.89',
'sem-5'], '1703': ['Ajit', 'Mech', '8.89', 'sem-5']}
```

Delete element from dictionary

- remove individual dictionary elements
- delete entire dictionary in a single operation

In [4]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

del (my_dict['1701'])       # Value corresponding to key value 1701 is deleted
print (my_dict)             # return list of all key in my_dict

del (my_dict)         # delete complete dictionary
#print (my_dict)      # As dictionary is completely deleted, python will generate NameError - my_dict not defined
```

```
{'1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': 'Piyush'}
```

Dictionary Builtin functions

- `len(my_dict)` - returns length of dictionary i.e. number of keys in dictionary
- `str(my_dict)` - return string representation of entire dictionary
- `type(my_dict)` - standard `type()` function. Return 'dict' for dictionary variable

In [5]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}
print (len(my_dict))
print (str(my_dict))
print (len(str(my_dict))) #It counts individual element including {,",alphabets etc.
print (str(my_dict)[0:4])
print (type(my_dict))
```

```
3
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': 'Piyush'}
102
{'17
<class 'dict'>
```

Dictionary Builtin methods

- `clear()` - clear all elements of dictionary. Results in empty dictionary
- `copy()` - create a copy of dictionary
- `get(key, default=None)` - Returns value for given key. If Key is not present, returns the second argument. Second parameter is optional having default value of None
- `has_key(key)` - returns True is key is present in dictionary, else returns False.
- `items()` - returns list of `(key,value)` tuple
- `keys()` - returns list of all key
- `values()` - returns list of list of values

In [6]:

```
# clear all elements of dictionary
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

my_dict.clear()                # clear all elements of dictionary
print (my_dict)                # empty dictionary
```

```
{}
```

In [7]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

my_dict_copy = my_dict.copy()     # create a copy of dictionary
print (my_dict)
print (my_dict_copy)
```

```
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': 'Piyush'}
{'1701': ['Sanam', 'CS', '9.01', 'sem-4'], '1702': ['Sachin', 'EC', '8.0', 'sem-4'], '1704': 'Piyush'}
```

In [8]:

```
# get values for given key
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

print (my_dict.get('1701'))         # returns value of dictionary stored for the given key
print (my_dict.get('1705'))         # returns None as 1705 is not a key in my_dict
```

```
['Sanam', 'CS', '9.01', 'sem-4']
None
```

In [9]:

```
# check if key is present in dictionary
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

print ('1701' in my_dict)       # returns True for Key present in my_dict
print ('1705' in my_dict)       # returns False for Key present in my_dict
```

```
True
False
```

In [10]:

```
# get list of all keys
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

print (my_dict.keys())              # return list of all key in my_dict
```

```
dict_keys(['1701', '1702', '1704'])
```

In [11]:

```
# get key and value as list of tuple

my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

print (my_dict.items())
print (type(my_dict.items()))        # Note the type of the class.

print (list(my_dict.items())[0])     # In order to use indexing we need to convert it to a list.
print (type(list(my_dict.items())[0]))
```

```
dict_items([('1701', ['Sanam', 'CS', '9.01', 'sem-4']), ('1702', ['Sachin', 'EC', '8.0', 'sem-4']), ('1704', 'Piyus
h')])
<class 'dict_items'>
('1701', ['Sanam', 'CS', '9.01', 'sem-4'])
<class 'tuple'>
```

In [12]:

```
my_dict = {'1701':["Sanam","CS","9.01","sem-4"], '1702':["Sachin","EC","8.0","sem-4"], '1704':'Piyush'}

print (my_dict.values())             # returns 'dict_values' type
```

```
dict_values([['Sanam', 'CS', '9.01', 'sem-4'], ['Sachin', 'EC', '8.0', 'sem-4'], 'Piyush'])
```