Python - Dictionary

{}

key : values

keys - unique().....

In [1]:

```
d = {'a': 1, 'b': 2, 'c': 3}
```

In [2]:

```
d
```

Out[2]:

```
{'a': 1, 'b': 2, 'c': 3}
```

In [3]:

```
type(d)
```

Out[3]:

```
dict
```

In [4]:

```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__',
'__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__ge
titem__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__it
er__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce_
_', '__reduce_ex__', '__repr__', '__setattr__', '__setitem__', '__size
of__', '__str__', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'ge
t', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'value
s']
```

In [5]:

```
d.keys()
```

Out[5]:

```
dict_keys(['a', 'b', 'c'])
```

In [6]:

```
d.values()
```

Out[6]:

```
dict_values([1, 2, 3])
```

In [7]:

```
1 d1 = {'a': 1, 'b': 1, 'a': 2, 'c': 3, 'b': 4}
```

In [8]:

```
1 d1.keys()
```

Out[8]:

```
dict_keys(['a', 'b', 'c'])
```

In [9]:

```
1 d1.values()
```

Out[9]:

```
dict_values([2, 4, 3])
```

In [10]:

```
1 d1.items()
```

Out[10]:

```
dict_items([('a', 2), ('b', 4), ('c', 3)])
```

In [11]:

```
1 d.items()
```

Out[11]:

```
dict_items([('a', 1), ('b', 2), ('c', 3)])
```

In [13]:

```
1 d['d'] = 4
2 d
```

Out[13]:

```
{'a': 1, 'b': 2, 'c': 3, 'd1': 4, 'd': 4}
```

In [15]:

```
1 d.pop('d1')
```

Out[15]:

```
4
```

In [16]:

```
1 d.popitem()
```

Out[16]:

```
('d', 4)
```

In [17]:

```
[1, 2, 3, 4, 5, 6].pop()
```

Out[17]:

```
6
```

In [18]:

```
len(d1)
```

Out[18]:

```
3
```

In [19]:

```
d1['a']
```

Out[19]:

```
2
```

In [20]:

```
d1['a'] = 5
d1
```

Out[20]:

```
{'a': 5, 'b': 4, 'c': 3}
```

In [22]:

```
for i in d1:
    print(i, d1[i])
```

```
a 5
b 4
c 3
```

In [23]:

```
for i in d1.items():
    print(i)
```

```
('a', 5)
('b', 4)
('c', 3)
```

Nested Dictionary of a class

In [36]:

```
Class = {

    'Class_8': {

            'Section' : {

                "Section"  : '8A',
                "Strength" :  58 ,
            },

                "Section"  : '8A',
                "Strength" :  58 ,

                },


    'Class_9': {

                "Section"  : '9A',
                "Strength" :  64 ,
                },

    'Class_10': {

                "Section"  : '10A',
                "Strength" :  42 ,
                },
}
```

In [37]:

```
Class
```

Out[37]:

```
{'Class_8': {'Section': '8A', 'Strength': 58},
 'Class_9': {'Section': '9A', 'Strength': 64},
 'Class_10': {'Section': '10A', 'Strength': 42}}
```

In [38]:

```
Class.keys()
```

Out[38]:

```
dict_keys(['Class_8', 'Class_9', 'Class_10'])
```

In [42]:

```
Class['Class_8']['Section']
```

Out[42]:

```
'8A'
```

In [35]:

```
Class['Class_9']
```

Out[35]:

```
{'Section': '9A', 'Strength': 64}
```

Pandas

In [43]:

```
data = {'Data Structure' : ['Series', 'Data Frames', 'Panel'],
        'Dimensions'     : [1,2,3],
        'Description'    : ['1D labeled homogenous array',
                            '2D labelled mutubale array',
                            '3D general mutuable array']}
```

In [45]:

```
Class = {'Class' : [8, 9, 10],
         'Section': ['A','B','C'],
         'Strength': [58, 64, 42]}
```

In [46]:

```
import pandas as pd
```

Datasets - dataframes

In [47]:

```
df = pd.DataFrame(data = Class)
```

In [48]:

```
df
```

Out[48]:

	Class	Section	Strength
0	8	A	58
1	9	B	64
2	10	C	42

In [55]:

```
df = df.set_index('Class')
```

In [56]:

```
1 df
```

Out[56]:

Class	Section	Strength
8	A	58
9	B	64
10	C	42

In [60]:

```
1 df['Strength'][9]
```

Out[60]:

64

In [62]:

```
1 df.loc[[8],['Section']]
```

Out[62]:

Class	Section
8	A

In [64]:

```
icc_ind_data = {

                    'Player': ['MA Agarwal'  , 'R Ashwin'   , 'JJ Bumrah'    ,
                               'KD Karthik'  , 'Virat Kohli', 'Kuldeep Yadav',
                               'HH Pandya'   , 'RR Pant'    , 'CA Pujara'    ,
                               'WP Saha'     , 'I Sharma'   , 'RG Sharma'    ,
                               'GH Vihari'   , 'M Vijay'    , 'UT Yadav']    ,


                    'Span'  :  ['2018-2020', '2011-2020', '2018-2020', '2013-201
                                '2004-2018', '2011-2020', '2017-2019', '2013-202
                                '2017-2018', '2018-2020', '2010-2020', '2013-202
                                '2010-2019', '2007-2020', '2013-2019', '2018-202
                                '2018-2020', '2008-2018', '2011-2020'],

                    'Mat'   : [11, 71, 14, 34, 49, 26, 86, 6, 49, 1, 11, 13, 77,
                               32, 4, 1, 9, 61, 46],

                    'Inns'  : [17, 98, 21, 58, 71, 42, 145, 6, 64, 1, 18, 22, 12
                               129, 53, 7, 1, 16, 105, 52],

                    'NO'    : [0, 13, 10, 1, 18, 1, 10, 0, 20, 1, 1, 1, 8, 11, 2
                               1, 1, 1, 22],

                    'Runs'  : [974, 2389, 32, 2315, 1869, 1025, 7240, 51, 497, 1
                               4203, 2006, 1238, 720, 2141, 335, 4, 552, 3982, 34

                    'HS'    : [243, 124, 10, 190, 100, 129, 254, 26, 51, 1, 108,
                               199, 177, 57, 212, 134, 4, 111, 167, 31],

                    'Avg'   : [57.29, 28.10, 2.90, 40.61, 35.26, 25.00, 53.62, 8
                               31.29, 38.76, 48.66, 42.88, 34.58, 30.19, 8.37, 46
                               36.80, 38.28, 11.33],

                    'BF'    : [1740, 4374, 146, 3458, 2931, 2080, 12552, 161, 64
                               12644, 8400, 3553, 2720, 2400, 3613, 388, 12 , 115

                    'SR'    : [55.97, 54.61, 21.91 ,66.94, 63.76, 49.27, 57.68,
                               73.88, 68.57, 46.18, 50.03, 56.45, 45.51, 30.00, 5
                               46.29, 49.85],

                    '100'   : [3, 4, 0, 7, 1, 1, 27, 0, 0, 0, 1, 2, 18, 11, 5, 3

                    '50'    : [4, 11, 0, 5, 14 ,7, 22, 0, 1, 0, 4, 2 ,25, 22, 11

                    '0'     : [0, 4, 6, 4, 4, 4, 10, 2, 13, 0, 2, 1, 7, 6, 6, 6,

                    '4s'    : [116, 270, 3, 316, 179, 134, 811, 6, 50, 0, 68, 83
                               216, 48, 0, 74, 467, 32],

                    '6s'    : [22, 14, 0, 12, 50, 4, 22, 0, 20, 0 ,12 ,19 ,14, 2
                               33, 17]
}
```

In [65]:

```
df= pd.DataFrame(icc_ind_data)
```

In [66]:

```
1 icc_ind_data.keys()
```

Out[66]:

```
dict_keys(['Player', 'Span', 'Mat', 'Inns', 'NO', 'Runs', 'HS', 'Avg',
'BF', 'SR', '100', '50', '0', '4s', '6s'])
```

In [72]:

```
1 df = df.set_index('Player')
```

In [73]:

```
1 df
```

Out[73]:

Player	Span	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	0	4s	6s
MA Agarwal	2018-2020	11	17	0	974	243	57.29	1740	55.97	3	4	0	116	22
R Ashwin	2011-2020	71	98	13	2389	124	28.1	4374	54.61	4	11	4	270	14
JJ Bumrah	2018-2020	14	21	10	32	10	2.9	146	21.91	0	0	6	3	0
S Dhawan	2013-2018	34	58	1	2315	190	40.61	3458	66.94	7	5	4	316	12
RA Jadeja	2012-2020	49	71	18	1869	100	35.26	2931	63.76	1	14	4	179	50
KD Karthik	2004-2018	26	42	1	1025	129	25	2080	49.27	1	7	4	134	4
Virat Kohli	2011-2020	86	145	10	7240	254	53.62	12552	57.68	27	22	10	811	22
Kuldeep Yadav	2017-2019	6	6	0	51	26	8.5	161	31.67	0	0	2	6	0
Mohammad Shammi	2013-2020	49	64	20	497	51	11.29	647	76.81	0	1	13	50	20
S Nadeem	2019-2019	1	1	1	1	1	-	5	20.00	0	0	0	0	0
HH Pandya	2017-2018	11	18	1	532	108	31.29	720	73.88	1	4	2	68	12
RR Pant	2018-2020	13	22	1	814	159	38.76	1187	68.57	2	2	1	83	19
CA Pujara	2010-2020	77	128	8	5840	206	48.66	12644	46.18	18	25	7	691	14
AM Rahane	2013-2020	65	109	11	4203	188	42.88	8400	50.03	11	22	6	475	29
KL Rahul	2014-2019	36	60	2	2006	199	34.58	3553	56.45	5	11	6	237	14
WP Saha	2010-2019	37	50	9	1238	177	30.19	2720	45.51	3	5	6	119	12
I Sharma	2007-2020	97	129	43	720	57	8.37	2400	30.00	0	1	31	81	0
RG Sharma	2013-2019	32	53	7	2141	212	46.54	3613	59.25	6	10	4	216	52
PP Shaw	2018-2020	4	7	1	335	134	55.83	388	86.34	1	2	0	48	2
SN Thakur	2018-2018	1	1	1	4	4	-	12	33.33	0	0	0	0	0
GH Vihari	2018-2020	9	16	1	552	111	36.8	1154	47.83	1	4	1	74	2

Player	Span	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	0	4s	6s
M Vijay	2008-2018	61	105	1	3982	167	38.28	8601	46.29	12	15	8	467	33
UT Yadav	2011-	46	52	22	340	31	11.33	682	49.85	0	0	6	32	17

In [74]:

```
1 df['Span']['Virat Kohli']
```

Out[74]:

```
'2011-2020'
```

In [75]:

```
1 df.values
```

Out[75]:

```
array([['2018-2020', 11, 17, 0, 974, 243, 57.29, 1740, 55.97, 3, 4,
0,
        116, 22],
       ['2011-2020', 71, 98, 13, 2389, 124, 28.1, 4374, 54.61, 4, 11,
4,
        270, 14],
       ['2018-2020', 14, 21, 10, 32, 10, 2.9, 146, 21.91, 0, 0, 6, 3,
0],
       ['2013-2018', 34, 58, 1, 2315, 190, 40.61, 3458, 66.94, 7, 5,
4,
        316, 12],
       ['2012-2020', 49, 71, 18, 1869, 100, 35.26, 2931, 63.76, 1, 1
4, 4,
        179, 50],
       ['2004-2018', 26, 42, 1, 1025, 129, 25.0, 2080, 49.27, 1, 7,
4,
        134, 4],
       ['2011-2020', 86, 145, 10, 7240, 254, 53.62, 12552, 57.68, 27,
22,
        10, 811, 22],
       ['2017-2019', 6, 6, 0, 51, 26, 8.5, 161, 31.67, 0, 0, 2, 6,
0],
       ['2013-2020', 49, 64, 20, 497, 51, 11.29, 647, 76.81, 0, 1, 1
3,
        50, 20],
       ['2019-2019', 1, 1, 1, 1, 1, '-', 5, 20.0, 0, 0, 0, 0, 0],
       ['2017-2018', 11, 18, 1, 532, 108, 31.29, 720, 73.88, 1, 4, 2,
68,
        12],
       ['2018-2020', 13, 22, 1, 814, 159, 38.76, 1187, 68.57, 2, 2,
1,
        83, 19],
       ['2010-2020', 77, 128, 8, 5840, 206, 48.66, 12644, 46.18, 18,
25,
        7, 691, 14],
       ['2013-2020', 65, 109, 11, 4203, 188, 42.88, 8400, 50.03, 11,
22,
        6, 475, 29],
       ['2014-2019', 36, 60, 2, 2006, 199, 34.58, 3553, 56.45, 5, 11,
6,
        237, 14],
       ['2010-2019', 37, 50, 9, 1238, 177, 30.19, 2720, 45.51, 3, 5,
6,
        119, 12],
       ['2007-2020', 97, 129, 43, 720, 57, 8.37, 2400, 30.0, 0, 1, 3
1,
        81, 0],
       ['2013-2019', 32, 53, 7, 2141, 212, 46.54, 3613, 59.25, 6, 10,
4,
        216, 52],
       ['2018-2020', 4, 7, 1, 335, 134, 55.83, 388, 86.34, 1, 2, 0, 4
8,
        2],
       ['2018-2018', 1, 1, 1, 4, 4, '-', 12, 33.33, 0, 0, 0, 0, 0],
       ['2018-2020', 9, 16, 1, 552, 111, 36.8, 1154, 47.83, 1, 4, 1,
```

```
74,
        2],
       ['2008-2018', 61, 105, 1, 3982, 167, 38.28, 8601, 46.29, 12, 1
5,
        8, 467, 33],
       ['2011-2020', 46, 52, 22, 340, 31, 11.33, 682, 49.85, 0, 0, 6,
32,
        1711, dtype=object)
```

In [77]:

```
1 df.index
```

Out[77]:

```
Index(['MA Agarwal', 'R Ashwin', 'JJ Bumrah', 'S Dhawan', 'RA Jadeja',
       'KD Karthik', 'Virat Kohli', 'Kuldeep Yadav', 'Mohammad Shamm
i',
       'S Nadeem', 'HH Pandya', 'RR Pant', 'CA Pujara', 'AM Rahane',
       'KL Rahul', 'WP Saha', 'I Sharma', 'RG Sharma', 'PP Shaw', 'SN
Thakur',
       'GH Vihari', 'M Vijay', 'UT Yadav'],
      dtype='object', name='Player')
```

In [81]:

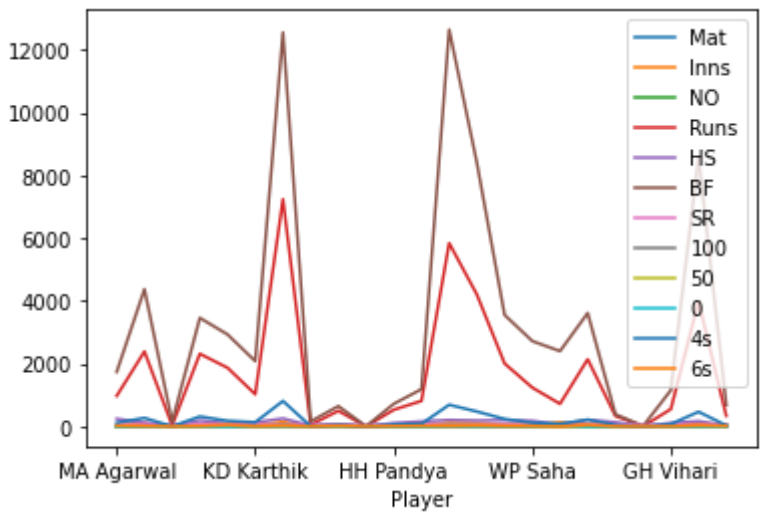
```
1 import matplotlib.pyplot as plt
2 %matplotlib inline
```

In [82]:

```
1 df.plot()
```

Out[82]:

```
<AxesSubplot:xlabel='Player'>
```



In [93]:

```
1 a = 'ThisIsAstring'
```

In [85]:

```
a[].islower()
```

Out[85]:

```
False
```

In [86]:

```
lower = []
upper = []
```

In [87]:

```
len(a)
```

Out[87]:

```
13
```

In [89]:

```
a.split()
```

Out[89]:

```
['T', 'h', 'i', 's', 'I', 's', 'A', 's', 't', 'r', 'i', 'n', 'g']
```

In [90]:

```
a[a.islower()]
```

Out[90]:

```
'T'
```

In [96]:

```
lower, upper = [], []
for i in a:
    if i.islower() == True:
        lower.append(i)
    else:
        upper.append(i)
```

In [97]:

```
print(lower)
```

```
['h', 'i', 's', 's', 's', 't', 'r', 'i', 'n', 'g']
```

In [98]:

```
print(upper)
```

```
['T', 'I', 'A']
```

In [99]:

```
1 len(lower), len(upper)
```

Out[99]:

```
(10, 3)
```

In [ ]:

```
1
```