

```
In [ ]:
```

## dictionary

```
In [4]: dt = {1 : "one", 2.5 : "float", True : 100, 2.5 : False}
```

```
In [5]: dt[2.5]
```

```
Out[5]: False
```

```
In [6]: dt[True]
```

```
Out[6]: 100
```

```
In [10]: dt = {1 : "one", 2.5 : "float", True : 100, 2.5 : False,  
              5 : [4, 8, 3], "dictionary" : {1 : "A", 2 : "B"}}
```

```
In [8]: dt[5]
```

```
Out[8]: [4, 8, 3]
```

```
In [11]: dt['dictionary']
```

```
Out[11]: {1: 'A', 2: 'B'}
```

```
In [ ]:
```

```
In [12]: table = {"column 1" : [1, 2, 3],  
                  "column 2" : ['a', 'b', 'c'],  
                  "column 3" : [True, False, 100.10]}
```

```
In [ ]:
```

```
In [ ]:
```

## working with libraries

```
In [24]: import numpy as np  
         # scipy  
         import pandas as pd  
         import matplotlib.pyplot as plt  
         import seaborn as sns
```

```
In [ ]:
```

```
In [14]: import array  
         import random  
         import math
```

```
In [16]: arr1 = array.array("i", [4, 7, 8])  
         arr1
```

```
Out[16]: array('i', [4, 7, 8])
```

```
In [17]: random.random()
```

```
Out[17]: 0.6206338224155084
```

```
In [18]: math.sqrt(100)
```

```
Out[18]: 10.0
```

```
In [ ]:
```

```
In [20]: numpy.random.random()
```

```
Out[20]: 0.9079736960108741
```

```
In [21]: numpy.sqrt(100)
```

```
Out[21]: 10.0
```

```
In [23]: numpy.array([4, 8])
```

```
Out[23]: array([4, 8])
```

```
In [ ]:
```

```
In [57]: df1 = pd.read_excel("StudentsPerformance.xlsx", sheet_name="StudentsPerformanc  
e")
```

```
In [53]: df1.head(4)
```

```
Out[53]:
```

	Unnamed: 0	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	1	female	group B	bachelor's degree	standard	none	72	72	74
1	2	female	group C	some college	standard	completed	69	90	88
2	3	female	group B	master's degree	standard	none	90	95	93
3	4	male	group A	associate's degree	free/reduced	none	47	57	44

```
In [31]: df1.head(10)
```

```
Out[31]:
```

	Unnamed: 0	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	1	female	group B	bachelor's degree	standard	none	72	72	74
1	2	female	group C	some college	standard	completed	69	90	88
2	3	female	group B	master's degree	standard	none	90	95	93
3	4	male	group A	associate's degree	free/reduced	none	47	57	44
4	5	male	group C	some college	standard	none	76	78	75
5	6	female	group B	associate's degree	standard	none	71	83	78
6	7	female	group B	some college	standard	completed	88	95	92
7	8	male	group B	some college	free/reduced	none	40	43	39
8	9	male	group D	high school	free/reduced	completed	64	64	67
9	10	female	group B	high school	free/reduced	none	38	60	50

```
In [32]: df1.tail()
```

```
Out[32]:
```

	Unnamed: 0	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	996	female	group E	master's degree	standard	completed	88	99	95
996	997	male	group C	high school	free/reduced	none	62	55	55
997	998	female	group C	high school	free/reduced	completed	59	71	65
998	999	female	group D	some college	standard	completed	68	78	77
999	1000	female	group D	some college	free/reduced	none	77	86	86

```
In [33]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
Unnamed: 0    1000 non-null int64
gender        1000 non-null object
race/ethnicity 1000 non-null object
parental level of education 1000 non-null object
lunch         1000 non-null object
test preparation course 1000 non-null object
math score    1000 non-null int64
reading score 1000 non-null int64
writing score 1000 non-null int64
dtypes: int64(4), object(5)
memory usage: 70.4+ KB
```

```
In [36]: df1.describe(include = 'all')
```

```
Out[36]:
```

	Unnamed: 0	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
count	1000.000000	1000	1000	1000	1000	1000	1000.000000	1000.000000	1000.000000
unique	NaN	2	5	6	2	2	NaN	NaN	NaN
top	NaN	female	group C	some college	standard	none	NaN	NaN	NaN
freq	NaN	518	319	226	645	642	NaN	NaN	NaN
mean	500.500000	NaN	NaN	NaN	NaN	NaN	66.08900	69.169000	72.367000
std	288.819436	NaN	NaN	NaN	NaN	NaN	15.16308	14.600192	15.465770
min	1.000000	NaN	NaN	NaN	NaN	NaN	0.00000	17.000000	19.000000
25%	250.750000	NaN	NaN	NaN	NaN	NaN	57.00000	59.000000	61.000000
50%	500.500000	NaN	NaN	NaN	NaN	NaN	66.00000	70.000000	72.000000
75%	750.250000	NaN	NaN	NaN	NaN	NaN	77.00000	79.000000	81.000000
max	1000.000000	NaN	NaN	NaN	NaN	NaN	100.00000	100.000000	100.000000

```
In [ ]:
```

```
In [39]: # single column
df1["race/ethnicity"].head()
```

```
Out[39]: 0    group B
1    group C
2    group B
3    group A
4    group C
Name: race/ethnicity, dtype: object
```

```
In [41]: # multiple columns
df1[["math score", 'gender']].head()
```

```
Out[41]:
```

	math score	gender
0	72	female
1	69	female
2	90	female
3	47	male
4	76	male

```
In [ ]:
```

```
In [51]: df1.head()
```

```
Out[51]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [ ]:

## dropping

```
In [60]: df1.drop(["lunch", "reading score"], axis = 1, inplace = True)
```

```
In [61]: df1.head()
```

Out[61]:

	gender	race/ethnicity	parental level of education	test preparation course	math score	writing score
0	female	group B	bachelor's degree	none	72	74
1	female	group C	some college	completed	69	88
2	female	group B	master's degree	none	90	93
3	male	group A	associate's degree	none	47	44
4	male	group C	some college	none	76	75

In [ ]:

```
In [69]: df1["parental level of education"].value_counts()
```

```
Out[69]: some college      226
associate's degree      222
high school            196
some high school       179
bachelor's degree      118
master's degree         59
Name: parental level of education, dtype: int64
```

```
In [68]: df1[df1["parental level of education"] == "master's degree"]["race/ethnicity"].value_counts()
```

```
Out[68]: group D      23
group C      19
group E       8
group B       6
group A       3
Name: race/ethnicity, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

## slicing dataframe

```
In [ ]: datafra
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```