

```
In [1]: #Neha Nemade  
#Roll_no:22150
```

```
In [1]: import os  
import pandas as pd  
import numpy as np  
os.chdir("C:\Pandas")  
car=pd.read_csv('Toyota.csv',index_col=0)
```

```
In [2]: car.index
```

```
Out[2]: Int64Index([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  
...  
1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435],  
dtype='int64', length=1436)
```

```
In [3]: car.columns
```

```
Out[3]: Index(['Price', 'Age', 'KM', 'FuelType', 'HP', 'MetColor', 'Automatic', 'CC',  
'Doors', 'Weight'],  
dtype='object')
```

```
In [4]: car.size
```

```
Out[4]: 14360
```

```
In [5]: car.shape
```

```
Out[5]: (1436, 10)
```

```
In [6]: #memory use by each column in bytes
```

```
car.memory_usage()
```

```
Out[6]: Index      11488  
Price      11488  
Age        11488  
KM         11488  
FuelType   11488  
HP         11488  
MetColor   11488  
Automatic  11488  
CC         11488  
Doors      11488  
Weight     11488  
dtype: int64
```

```
In [7]: car.ndim
```

```
Out[7]: 2
```

```
In [8]: #indexing
#head-return first n rows
car.head(5)
```

```
Out[8]:
```

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23.0	46986	Diesel	90	1.0	0	2000	three	1165
1	13750	23.0	72937	Diesel	90	1.0	0	2000	3	1165
2	13950	24.0	41711	Diesel	90	NaN	0	2000	3	1165
3	14950	26.0	48000	Diesel	90	0.0	0	2000	3	1165
4	13750	30.0	38500	Diesel	90	0.0	0	2000	3	1170

```
In [9]: #tail - -returns last n rows
car.tail(5)
```

```
Out[9]:
```

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
1431	7500	NaN	20544	Petrol	86	1.0	0	1300	3	1025
1432	10845	72.0	??	Petrol	86	0.0	0	1300	3	1015
1433	8500	NaN	17016	Petrol	86	0.0	0	1300	3	1015
1434	7250	70.0	??	NaN	86	1.0	0	1300	3	1015
1435	6950	76.0	1	Petrol	110	0.0	0	1600	5	1114

```
In [10]: #Label based scalar lookup:
#selecting data based on label using 'at'
#getting data of row 4 coloum with label 'KM'

car.at[4, 'KM']
```

```
Out[10]: '38500'
```

```
In [11]: #integer based lookup using iat
car.iat[4,9]
```

```
Out[11]: 1170
```

```
In [ ]: #PANDAS 2
```

```
In [12]: '''
1) Data Types
Numeric(int64,float64)
Character(category,object)
2)checking Data types each column
3)count of unique data types
4)selecting data based on data type
5)Concise summary of dataframe
6)Checking Format of each column
7)Getting Unique elements of each column
'''
```

```
Out[12]: '\n1) Data Types\n Numeric(int64,float64)\n Character(category,object)\n 2)checking Data types each column\n 3)count of unique data types\n 4)selecting data based on data type\n 5)Concise summary of dataframe\n 6)Checking Format of each column\n 7)Getting Unique elements of each column\n '
```

```
In [14]: import os
import numpy as np
os.chdir("C:\Pandas")
car=pd.read_csv('Toyota.csv',index_col=0)
```

```
In [15]: #checking data types each column
car.dtypes
```

```
Out[15]: Price          int64
Age          float64
KM           object
FuelType     object
HP           object
MetColor     float64
Automatic    int64
CC           int64
Doors        object
Weight       int64
dtype: object
```

```
In [17]: #count of unique data types

car.dtypes.value_counts()
```

```
Out[17]: object      4
int64      4
float64     2
dtype: int64
```

In [18]: *#selecting data based on data types*

```
car.select_dtypes(exclude=[object])
```

Out[18]:

	Price	Age	MetColor	Automatic	CC	Weight
0	13500	23.0	1.0	0	2000	1165
1	13750	23.0	1.0	0	2000	1165
2	13950	24.0	NaN	0	2000	1165
3	14950	26.0	0.0	0	2000	1165
4	13750	30.0	0.0	0	2000	1170
...	...	...	...	...	...	...
1431	7500	NaN	1.0	0	1300	1025
1432	10845	72.0	0.0	0	1300	1015
1433	8500	NaN	0.0	0	1300	1015
1434	7250	70.0	1.0	0	1300	1015
1435	6950	76.0	0.0	0	1600	1114

1436 rows × 6 columns

In [19]: *#concise summary of data frame and checking format of each column*

```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null  int64
1   Age         1336 non-null  float64
2   KM          1436 non-null  object
3   FuelType    1336 non-null  object
4   HP          1436 non-null  object
5   MetColor    1286 non-null  float64
6   Automatic   1436 non-null  int64
7   CC          1436 non-null  int64
8   Doors       1436 non-null  object
9   Weight      1436 non-null  int64
dtypes: float64(2), int64(4), object(4)
memory usage: 123.4+ KB
```

In [20]: *#Getting unique elements of each column*

```
print(np.unique(car['KM']))
```

```
['1' '10000' '100123' ... '99865' '99971' '??']
```



```
In [27]: # importing data
import os
import numpy as np
os.chdir("C:\Pandas")
car=pd.read_csv('Toyota.csv',index_col=0)
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null   int64
1   Age         1336 non-null   float64
2   KM          1436 non-null   object
3   FuelType    1336 non-null   object
4   HP          1436 non-null   object
5   MetColor    1286 non-null   float64
6   Automatic   1436 non-null   int64
7   CC          1436 non-null   int64
8   Doors       1436 non-null   object
9   Weight      1436 non-null   int64
dtypes: float64(2), int64(4), object(4)
memory usage: 123.4+ KB
```

```
In [28]: #concise summary of dataframe:Now , importing the data considering data frames
car=pd.read_csv('Toyota.csv',index_col=0,na_values=["??","???"])
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null   int64
1   Age         1336 non-null   float64
2   KM          1421 non-null   float64
3   FuelType    1336 non-null   object
4   HP          1436 non-null   object
5   MetColor    1286 non-null   float64
6   Automatic   1436 non-null   int64
7   CC          1436 non-null   int64
8   Doors       1436 non-null   object
9   Weight      1436 non-null   int64
dtypes: float64(3), int64(4), object(3)
memory usage: 123.4+ KB
```

```
In [30]: #converting data types
car['MetColor']=car['MetColor'].astype('object')
car['Automatic']=car['Automatic'].astype('object')
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null   int64
1   Age         1336 non-null   float64
2   KM          1421 non-null   float64
3   FuelType    1336 non-null   object
4   HP          1436 non-null   object
5   MetColor    1286 non-null   object
6   Automatic   1436 non-null   object
7   CC          1436 non-null   int64
8   Doors       1436 non-null   object
9   Weight      1436 non-null   int64
dtypes: float64(2), int64(3), object(5)
memory usage: 123.4+ KB
```

```
In [31]: #category vs data object type

print('Memory whwn data is of object Type')
car['FuelType'].nbytes
```

Memory whwn data is of object Type

Out[31]: 11488

```
In [32]: print('Memory whwn data is of category Type')
car['FuelType'].astype('category').nbytes
```

Memory whwn data is of category Type

Out[32]: 1460

```
In [33]: #cleaning column doors
#useof replace()

print(np.unique(car['Doors']))
car['Doors'].replace('three',3,inplace=True)
car['Doors'].replace('four',4,inplace=True)
car['Doors'].replace('five',5,inplace=True)
```

```
['2' '3' '4' '5' 'five' 'four' 'three']
```

```
In [34]: car['Doors']=car['Doors'].astype('int64')
print(np.unique(car['Doors']))
```

```
[2 3 4 5]
```

In [35]: *#Getting count of missing values*

```
car.isnull().sum()
```

Out[35]:

Price	0
Age	100
KM	15
FuelType	100
HP	0
MetColor	150
Automatic	0
CC	0
Doors	0
Weight	0
dtype: int64	

In [ ]: