In [3]:
```
import pandas

mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}

myvar = pandas.DataFrame(mydataset)

print(myvar)
```

```
    cars  passings
0    BMW         3
1  Volvo         7
2   Ford         2
```

```
import pandas as pd

mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}

myvar = pandas.DataFrame(mydataset)

print(myvar)
```

In [8]:
```
import pandas
veg_menu={
    'veg':['panneer','dal','cauliflower','carrot'],
    'cost':['150','50','78','90']
}
nonveg_menu={
    'non-veg':['chicken','crabs','prawns','egg'],
    'cost':['300','550','870','170']
}
print(pandas.DataFrame(veg_menu))
print(pandas.DataFrame(nonveg_menu))
```

```
          veg cost
0      panneer  150
1          dal   50
2  cauliflower   78
3       carrot   90
   non-veg cost
0  chicken  300
1    crabs  550
2   prawns  870
3      egg  170
```

In [22]:
```
import pandas
order=pandas.read_csv('E:\\menu1.csv')
print(order.to_string())
```

```
         order  cost
0      panneer   600
1          dal   200
2  cauliflower   399
3       carrot   400
4      chicken   800
5        crabs   800
6       prawns   670
```

In [14]:
```
import pandas
details={
    'name':['manaswini','sai','ratna'],
    'age':['18','45','37'],
    'gender':['female','male','female']
}
mydataset=pandas.DataFrame(details)
filename='familydata.xlsx'
mydataset.to_excel(filename)
print(mydataset)
```

```
        name age  gender
0  manaswini  18  female
1        sai  45    male
2      ratna  37  female
```

In [15]:
```
import pandas as pd
a = [1,2,3]
m= pd.Series(a)
print(m)
```

```
0    1
1    2
2    3
dtype: int64
```

In [16]:
```
import pandas as pd
a = [1, 7, 2]
m = pd.Series(a, index = ["x", "y", "z"])
print(m)
```

```
x    1
y    7
z    2
dtype: int64
```

In [17]:
```
import pandas as pd
calories = {"day1": 420, "day2": 380, "day3": 390}
myvar = pd.Series(calories)
print(myvar)
```

```
day1    420
day2    380
day3    390
dtype: int64
```

In [19]:
```
import pandas as pd
calories = {"day1": 420, "day2": 380, "day3": 390}
m = pd.Series(calories, index = ["day1", "day2"])
print(m)
```

```
day1    420
day2    380
dtype: int64
```

In [ ]:
```
import pandas as pd
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}
m = pd.DataFrame(data)
print(m)
```

In [25]:
```
import pandas as pd
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}
m = pd.DataFrame(data)
print(m)
print(m.loc[0])
print(m.loc[[0, 1]])
```

```
   calories  duration
0       420        50
1       380        40
2       390        45
calories    420
duration     50
Name: 0, dtype: int64
   calories  duration
0       420        50
1       380        40
```

In [26]:
```
import pandas as pd
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(df)
print(df.loc["day2"])
```

```
      calories  duration
day1       420        50
day2       380        40
day3       390        45
calories    380
duration     40
Name: day2, dtype: int64
```

In [27]:
```
import pandas as pd
df = pd.read_csv('data.csv')
print(df)
```

```
      name  price
0    apple   1000
1  samsung    450
2      pen     10
3    bench   4500
4   iphone  75000
5    watch   9000
6   marker     20
```

In [28]:
```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.to_string())
```

```
      name  price
0    apple   1000
1  samsung    450
2      pen     10
3    bench   4500
4   iphone  75000
5    watch   9000
6   marker     20
```

In [29]:
```
import pandas as pd
print(pd.options.display.max_rows)
```

```
60
```

In [30]:
```
import pandas as pd
pd.options.display.max_rows = 9999
df = pd.read_csv('data.csv')
print(df)
```

```
      name  price
0    apple   1000
1  samsung    450
2      pen     10
3    bench   4500
4   iphone  75000
5    watch   9000
6   marker     20
```

In [32]:
```
import pandas as pd

df = pd.read_json('data.json')

print(df.to_string())
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7584/855483702.py in <module>
      1 import pandas as pd
      2
----> 3 df = pd.read_json('data.json')
      4
      5 print(df.to_string())

E:\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **k
wargs)
    205                 else:
    206                     kwargs[new_arg_name] = new_arg_value
--> 207             return func(*args, **kwargs)
    208
    209         return cast(F, wrapper)

E:\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **k
wargs)
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312
    313         return wrapper

E:\anaconda3\lib\site-packages\pandas\io\json\_json.py in read_json(path_or_bu
f, orient, typ, dtype, convert_axes, convert_dates, keep_default_dates, numpy,
 precise_float, date_unit, encoding, encoding_errors, lines, chunksize, compres
sion, nrows, storage_options)
    612
    613     with json_reader:
--> 614         return json_reader.read()
    615
    616

E:\anaconda3\lib\site-packages\pandas\io\json\_json.py in read(self)
    746                 obj = self._get_object_parser(self._combine_lines(data_
lines))
    747         else:
--> 748             obj = self._get_object_parser(self.data)
    749         self.close()
    750         return obj

E:\anaconda3\lib\site-packages\pandas\io\json\_json.py in _get_object_parser(se
lf, json)
    768         obj = None
    769         if typ == "frame":
--> 770             obj = FrameParser(json, **kwargs).parse()
    771
    772         if typ == "series" or obj is None:
```

```
E:\anaconda3\lib\site-packages\pandas\io\json\_json.py in parse(self)
    883
    884         else:
--> 885             self._parse_no_numpy()
    886
    887         if self.obj is None:

E:\anaconda3\lib\site-packages\pandas\io\json\_json.py in _parse_no_numpy(self)
   1138         if orient == "columns":
   1139             self.obj = DataFrame(
-> 1140                 loads(json, precise_float=self.precise_float), dtype=No
ne
   1141             )
   1142         elif orient == "split":

ValueError: Expected object or value
```

In [33]:
```
import pandas as pd
data = {
  "Duration":{
    "0":60,
    "1":60,
    "2":60,
    "3":45,
    "4":45,
    "5":60
  },
  "Pulse":{
    "0":110,
    "1":117,
    "2":103,
    "3":109,
    "4":117,
    "5":102
  },
  "Maxpulse":{
    "0":130,
    "1":145,
    "2":135,
    "3":175,
    "4":148,
    "5":127
  },
  "Calories":{
    "0":409,
    "1":479,
    "2":340,
    "3":282,
    "4":406,
    "5":300
  }
}
df = pd.DataFrame(data)
print(df)
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130       409
1        60    117       145       479
2        60    103       135       340
3        45    109       175       282
4        45    117       148       406
5        60    102       127       300
```

In [40]:
```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.head(6))
```

```
      name  price
0    apple   1000
1  samsung    450
2      pen     10
3    bench   4500
4   iphone  75000
5    watch   9000
```

In [42]:
```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.head())
print(df.tail())
print(df.info())
```

```
      name  price
0    apple   1000
1  samsung    450
2      pen     10
3    bench   4500
4   iphone  75000
     name  price
2     pen     10
3   bench   4500
4  iphone  75000
5   watch   9000
6  marker     20
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   name    7 non-null      object
 1   price   7 non-null      int64
dtypes: int64(1), object(1)
memory usage: 240.0+ bytes
None
```

In [12]:
```
#EMPTY CELL DELETE
import pandas as pd
df = pd.read_csv('E:\\details.csv')
new_df = df.dropna()
print(new_df.to_string())
```

```
      NAME   AGE GENDER
0     MANU  18.0      F
1  DEDIPYA  23.0      F
3  JAHNAVI  19.0      F
```

In [11]:
```
#Replace NULL values with the number 130:
import pandas as pd
df = pd.read_csv('E:\\details.csv')
df.fillna(30, inplace = True)
print(df.to_string())
```

```
      NAME   AGE GENDER
0     MANU  18.0      F
1  DEDIPYA  23.0      F
2   HARSHI  30.0      F
3  JAHNAVI  19.0      F
```

In [14]:
```
import pandas as pd
df = pd.read_csv('E:\\Book1.csv')
new_df = df.dropna()
print(new_df.to_string())
filename='Book2.xlsx'
new_df.to_excel(filename)
```

```
   S.NO      NAME     CITY       PHONE
0     1      MANU    VIZAG  1234567890
2     3  HARSHITA  PMPALEM  8779742657
```

In [25]:
```
import pandas as pd
df = pd.read_csv('E:\\Book3.csv')
df["calories"].fillna(2000 ,inplace = True)
print(df.to_string())
```

```
    name  calories  pulse
0   manu     200.0  110.0
1  deepu    2000.0  112.0
2   dedi    2300.0  111.0
3  harsh    1456.0    NaN
```

In [28]:
```
import pandas as pd
df = pd.read_csv('E:\\mean.csv')
x = df["marks"].mean()
df["marks"].fillna(x, inplace = True)
print(df.to_string())
```

```
   s.no      marks  age
0     1  23.000000   18
1     2  34.000000   19
2     3  56.000000   20
3     4  37.666667   44
```

In [29]:
```
import pandas as pd
df = pd.read_csv('E:\\mean.csv')
x = df["marks"].mode()
df["marks"].fillna(x, inplace = True)
print(df.to_string())
```

```
   s.no  marks  age
0     1   23.0   18
1     2   34.0   19
2     3   56.0   20
3     4    NaN   44
```

In [30]:
```
import pandas as pd
df = pd.read_csv('E:\\mean.csv')
x = df["marks"].median()
df["marks"].fillna(x, inplace = True)
print(df.to_string())
```

```
   s.no  marks  age
0     1   23.0   18
1     2   34.0   19
2     3   56.0   20
3     4   34.0   44
```

In [22]:
```
# family members
import pandas as pd
df = pd.read_csv('E:\\familymembers.csv')
print(df.to_string())
df.drop_duplicates(inplace=True)
```

```
    s.n0      name  age gender  phno  address
0      1      manu   18      f  4534      nad
1      2     deepu   19      m  2335      nad
2      3    chandu   20      f  3456  pmpalem
3      4   jahnavi   21      f  4556      kpc
4      5       sai   22      m  4562      kgh
5      6     ratna   23      f  3695      cvp
6      7    dedipa   24      f  3577      svp
7      8  harshita   24      f  6789      fes
8      9    hemika   25      f  8765      bgt
9     10      rama   26      f   967     lkhg
10    11   likhita   27      f  6784     gujg
11    12   nikitha   28      f  2398     khjg
12    13    sonika   29      f  3457     jhcf
13    14    preeti   20      f  5679    kjgvh
14    15    pavani   30      f  9783    kufgh
15    16     pandu   32      m  6798   jlkgve
16    17     geetu   31      f  6798   jgghgk
17    18   meghana   33      f  8576    ljhjf
18    19     ramya   34      f  4139    mnhfl
19    20      dsai   23      m  4567    gjxmh
```

In [24]:
```
import pandas as pd
df = pd.read_csv('E:\\passport.csv')
print(df.to_string())
df.drop_duplicates(subset ="name",
                     keep = False, inplace = True)
```

```
        name  age gender       dob passport license
0       manu   18      f  01-02-03      yes     yes
1      deepu   18      m  03-03-03      yes     yes
2     chandu   34      f  04-04-83      yes      no
3       rosy   27      f  03-07-00       no     yes
4        sai   32      m  06-06-03       no     yes
5      ratna   66      f  07-07-76      yes     yes
6     hemika   33      f  02-03-82      yes      no
7     dedipa   22      f  05-06-92      yes     yes
8   harshita   13      f  06-06-54      yes     yes
9      snaha   33      f  03-05-06       no     yes
10     sreya   31      f  04-05-07      yes      no
11     sreya   13      f  04-05-21      yes     yes
12     ramya   32      f  02-02-82      yes     yes
13    reshma   23      f  01-01-00      yes      no
14   meghana   42      f  04-03-84      yes     yes
15    deepak   44      m  09-03-82      yes     yes
16      rama   74      m  02-11-64      yes     yes
17    sonika   24      f  03-09-54       no     yes
18    harini   19      f  05-08-92      yes      no
19     asdsf   45      m  07-03-34       no     yes
```

In [21]:
```
#COLOUMN DUPLICATE
import pandas as pd
df = pd.read_csv('E:\\passport.csv')
df.drop_duplicates('age',inplace=True)
print(df.to_string())
```

```
        name  age gender       dob passport license
0       manu   18      f  01-02-03      yes     yes
2     chandu   34      f  04-04-83      yes      no
3       rosy   27      f  03-07-00       no     yes
4        sai   32      m  06-06-03       no     yes
5      ratna   66      f  07-07-76      yes     yes
6     hemika   33      f  02-03-82      yes      no
7     dedipa   22      f  05-06-92      yes     yes
8   harshita   13      f  06-06-54      yes     yes
10     sreya   31      f  04-05-07      yes      no
13    reshma   23      f  01-01-00      yes      no
14   meghana   42      f  04-03-84      yes     yes
15    deepak   44      m  09-03-82      yes     yes
16      rama   74      m  02-11-64      yes     yes
17    sonika   24      f  03-09-54       no     yes
18    harini   19      f  05-08-92      yes      no
19     asdsf   45      m  07-03-34       no     yes
```

In [ ]:

In [ ]: