In [2]:
```
#hierachical Indexing
```

In [3]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
#multi index from tuples
#example we will take price of pertol in soem states for year 2010 and 2020
pet=[("Tamilnadu",2010),("Tamilnadu",2020),("Kerala",2010),("Kerala",2020),("Andhra Pradesh",2010),("Andhra Pradesh",2020)]
pet_price=[55.92,88.82,57,107.51,58,111.66]
petrol_series=pd.Series(pet_price,index=pet)#not completely accurate data
pet=pd.MultiIndex.from_tuples(pet)#multi index
petrol_series=petrol_series.reindex(pet)#re index for neat representation
print(petrol_series)#printing data
```

```
Tamilnadu       2010     55.92
                2020     88.82
Kerala          2010     57.00
                2020    107.51
Andhra Pradesh  2010     58.00
                2020    111.66
dtype: float64
```

In [11]:
```
#multi index as extra dimension
petrol_series_df=petrol_series.unstack()#unstack will convert series with multiindex to dataframe
print("Converted to dataframe=")
print(petrol_series_df)
print("Series with multiindex=")
print(petrol_series_df.stack())#stack will do the reverse of unstack
#adding new column
petrol_series_df[2030]=petrol_series_df[2020]*1.5#creating new column 2030 which is 1.5 times the price in 2020
print(petrol_series_df)
```

```
Converted to dataframe=
                 2010    2020
Andhra Pradesh  58.00  111.66
Kerala          57.00  107.51
Tamilnadu       55.92   88.82
Series with multiindex=
Andhra Pradesh  2010     58.00
                2020    111.66
Kerala          2010     57.00
                2020    107.51
Tamilnadu       2010     55.92
                2020     88.82
dtype: float64
                 2010    2020     2030
Andhra Pradesh  58.00  111.66  167.490
Kerala          57.00  107.51  161.265
Tamilnadu       55.92   88.82  133.230
```

In [39]:
```
#methods of creating multiindex
#passing two or array indexes to dataframe constructor
p=pd.DataFrame(np.arange(0,8).reshape(4,2),index=[["a","a","b","b"],[0,1,0,1]],columns=["column1","column2"])
print(p)
#explicit multiindex constructors
#from array construct multiindex
array_index=pd.MultiIndex.from_arrays([["a","a","b","b"],[1,2,1,2]])
p1=pd.DataFrame(np.arange(0,8).reshape(4,2),index=array_index)
print("MultiIndex from array=")
print(p1)
#multiindex from tuples
tuple_index=pd.MultiIndex.from_tuples([("a","a1"),("a","a2"),("b","b1"),("b","b2"),("c","c1"),("c","c2")])
p2=pd.DataFrame(np.random.randint(0,9,(4,3)),index=array_index,columns=["i","ii","iii"])
print("MultiIndex from tuples=")
print(p2)
#from cartesian product
product_index=pd.MultiIndex.from_product([["a","b"],[0,1]])
p3=pd.DataFrame(np.random.randint(0,9,(4,3)),index=product_index)
print("MultiIndex from Cartesian product=")
print(p3)
#multiindex naming
p3.index.names=["LEVEL1","LEVEL2"];
print("Naming multiIndex levels=")
print(p3)
```

```
     column1  column2
a 0        0        1
  1        2        3
b 0        4        5
  1        6        7
MultiIndex from array=
     0  1
a 1  0  1
  2  2  3
b 1  4  5
  2  6  7
MultiIndex from tuples=
     i  ii  iii
a 1  0   4    0
  2  5   7    0
b 1  1   3    8
  2  7   4    6
MultiIndex from Cartesian product=
     0  1  2
a 0  3  8  3
  1  4  5  5
b 0  7  3  8
  1  4  2  3
Naming multiIndex levels=
               0  1  2
LEVEL1 LEVEL2
a      0       3  8  3
       1       4  5  5
b      0       7  3  8
       1       4  2  3
```

In [57]:
```
index=pd.MultiIndex.from_product([["A","B","C"],["Cat1","Cat2","Internals","Fat"]]);
columns=pd.MultiIndex.from_product([["First year","Second year","Third Year","Fourth Year"],["sem1","sem2"]])
p=pd.DataFrame(np.arange(0,96).reshape((12,8)),index=index,columns=columns)
print(p)
```

```
            First year      Second year      Third Year      Fourth Year
                  sem1 sem2        sem1 sem2       sem1 sem2        sem1 sem2
A Cat1               0    1           2    3          4    5           6    7
  Cat2               8    9          10   11         12   13          14   15
  Internals         16   17          18   19         20   21          22   23
  Fat               24   25          26   27         28   29          30   31
B Cat1              32   33          34   35         36   37          38   39
  Cat2              40   41          42   43         44   45          46   47
  Internals         48   49          50   51         52   53          54   55
  Fat               56   57          58   59         60   61          62   63
C Cat1              64   65          66   67         68   69          70   71
  Cat2              72   73          74   75         76   77          78   79
  Internals         80   81          82   83         84   85          86   87
  Fat               88   89          90   91         92   93          94   95
```

In [60]:
```
p
print(p["First year"])#get first year marks of all people
#acessing single elements
print("Accesing first year 1st sem marks=")
print(p["First year","sem1"])
```

```
              sem1  sem2
A Cat1           0     1
  Cat2           8     9
  Internals     16    17
  Fat           24    25
B Cat1          32    33
  Cat2          40    41
  Internals     48    49
  Fat           56    57
C Cat1          64    65
  Cat2          72    73
  Internals     80    81
  Fat           88    89
Accesing first year 1st sem marks=
A  Cat1           0
   Cat2           8
   Internals     16
   Fat           24
B  Cat1          32
   Cat2          40
   Internals     48
   Fat           56
C  Cat1          64
   Cat2          72
   Internals     80
   Fat           88
Name: (First year, sem1), dtype: int32
```

In [79]:
```
#sorting of multiIndex
index=pd.MultiIndex.from_product([["a","c","b"],[1,2]])
p=pd.DataFrame(np.random.randint(0,10,(6,2)),index=index)
p.index.names=["Name","Subdivision"]
print("Original dataframe=")
p=p.sort_index()
print("Sorted dataframe=")
print(p)
print("slicing dataframe=")
print(p["a":"b"])
```

```
Original dataframe=
Sorted dataframe=
                  0  1
Name Subdivision
a    1            9  6
     2            1  2
b    1            3  3
     2            1  0
c    1            7  9
     2            1  5
slicing dataframe=
                  0  1
Name Subdivision
a    1            9  6
     2            1  2
b    1            3  3
     2            1  0
```

In [95]:
```
print("Origial dataframe with multiIndex=")
print(petrol_series)
#resetting index using reset index which convert multiindexed data to raw data form
print("Resetting price=")
petrol_series_flat=petrol_series.reset_index(name="Price")
print(petrol_series_flat)
#setting indexes so that they become multiindexed
print("Setting index=")
petrol_series1=petrol_series_flat.set_index(["level_0","level_1"])
print(petrol_series1)
```

```
Origial dataframe with multiIndex=
Tamilnadu       2010     55.92
                2020     88.82
Kerala          2010     57.00
                2020    107.51
Andhra Pradesh  2010     58.00
                2020    111.66
dtype: float64
Resetting price=
          level_0  level_1   Price
0       Tamilnadu     2010   55.92
1       Tamilnadu     2020   88.82
2          Kerala     2010   57.00
3          Kerala     2020  107.51
4  Andhra Pradesh     2010   58.00
5  Andhra Pradesh     2020  111.66
Setting index=
                         Price
level_0        level_1
Tamilnadu      2010      55.92
               2020      88.82
Kerala         2010      57.00
               2020     107.51
Andhra Pradesh 2010      58.00
               2020     111.66
```

In [113]:
```
#aggreagates in mutliindexing
petrol_series=pd.Series(pet_price,index=pet)
petrol_series.index.names=["state","year"]
print(petrol_series)
#petrol price of each state
print("Average Petrol price of each state=")
mean_petrol=petrol_series.mean(level="state")
print(mean_petrol)
#average of petrol prices in 2010
print("Average Petrol price of each year=")
mean_petrol=petrol_series.mean(level="year")
print(mean_petrol)
```

```
state           year
Tamilnadu       2010     55.92
                2020     88.82
Kerala          2010     57.00
                2020    107.51
Andhra Pradesh  2010     58.00
                2020    111.66
dtype: float64
Average Petrol price of each state=
state
Tamilnadu         72.370
Kerala            82.255
Andhra Pradesh    84.830
dtype: float64
Average Petrol price of each year=
year
2010     56.973333
2020    102.663333
dtype: float64

C:\Users\kdmag\AppData\Local\Temp\ipykernel_22164\1977281348.py:7: FutureWarning: Using the level keyword in DataFrame and Seri
es aggregations is deprecated and will be removed in a future version. Use groupby instead. df.median(level=1) should use df.gr
oupby(level=1).median().
  mean_petrol=petrol_series.mean(level="state")
C:\Users\kdmag\AppData\Local\Temp\ipykernel_22164\1977281348.py:11: FutureWarning: Using the level keyword in DataFrame and Ser
ies aggregations is deprecated and will be removed in a future version. Use groupby instead. df.median(level=1) should use df.g
roupby(level=1).median().
  mean_petrol=petrol_series.mean(level="year")
```