In [5]:
```
import numpy as np
import pandas as pd
#generate a 5*4 array of random numbers
#between 0 and 100
np.random.seed(42)
random_array=np.random.randint(0,100,size=(5,4))
df=pd.DataFrame(random_array,columns=['A','B','C','D'],index=['V','W','X','Y',
'Z'])
print(df)
```

```
    A   B   C   D
V  51  92  14  71
W  60  20  82  86
X  74  74  87  99
Y  23   2  21  52
Z   1  87  29  37
```

1.Selecting specific Rows

In [7]:
```
#using Label_Based Indexing(.loc)
#select rows 'W' to 'Y'
print(df.loc['W':'Y'])
```

```
    A   B   C   D
W  60  20  82  86
X  74  74  87  99
Y  23   2  21  52
```

In [8]:
```
#using Label_Based Indexing(.loc)
#select first 3 rows
print(df.iloc[:3])
```

```
    A   B   C   D
V  51  92  14  71
W  60  20  82  86
X  74  74  87  99
```

2.selecting specific columns

In [9]:
```
#select columns 'A' and 'C'
print(df[['A','C']])
```

```
    A   C
V  51  14
W  60  82
X  74  87
Y  23  21
Z   1  29
```

3.Selecting specific rows and columns

In [10]:
```
#select rows 'W' to 'Y' and columns 'B and 'D
print(df.loc['W':'Y',['B','D']])
#select first 3 rows and first 2 columns
print(df.iloc[:3,:2])
```

```
    B   D
W  20  86
X  74  99
Y   2  52
    A   B
V  51  92
W  60  20
X  74  74
```

4.condition slicing

In [12]:
```
#select rows where column 'A' values are greater than 50
print(df[df['A']>50])
#select rows where column 'c' values are less than 50
print(df[df['C']<30])
```

```
    A   B   C   D
V  51  92  14  71
W  60  20  82  86
X  74  74  87  99
    A   B   C   D
V  51  92  14  71
Y  23   2  21  52
Z   1  87  29  37
```

5.Dropping a row

In [15]:
```
#drop row 'X'
df_dropped_row=df.drop(index='X')
#display dataframe after dropping row
print(df_dropped_row)
```

```
    A   B   C   D
V  51  92  14  71
W  60  20  82  86
Y  23   2  21  52
Z   1  87  29  37
```

6.Dropping multiple rows

In [16]:
```
#drop row 'W' and 'Y'
df_dropped_row=df.drop(index=['W',"Y"])
print(df_dropped_row)
```

```
    A   B   C   D
V  51  92  14  71
X  74  74  87  99
Z   1  87  29  37
```

7.Dropping a column

In [19]:
```
#drop column 'C'
df_dropped_col=df.drop(columns=['A'])
print(df_dropped_col)
```

```
    B   C   D
V  92  14  71
W  20  82  86
X  74  87  99
Y   2  21  52
Z  87  29  37
```

8.Dropping Multiple columns

In [20]:
```
#drop column 'A'and 'D'
df_dropped_col=df.drop(columns=['A','D'])
print(df_dropped_col)
```

```
    B   C
V  92  14
W  20  82
X  74  87
Y   2  21
Z  87  29
```

9.Dropping Rows/Columns In-Place

In [22]:
```
#drop row 'X' permanently
df.drop(index='X',inplace=True)
#drop column 'C' permanently
df.drop(columns='C',inplace=True)
print(df)
```

```
    A   B   D
V  51  92  71
W  60  20  86
Y  23   2  52
Z   1  87  37
```

In [35]:
```
#adding a new column first
print(df)
df['F']=[5,10,15,20,25]
df.loc['Extra']=[55,65,75,85]
print(df)
```

```
        A   B   D   F
V      51  92  71   5
W      60  20  86  10
Y      23   2  52  15
Z       1  87  37  20
Extra  55  65  75  25
        A   B   D   F
V      51  92  71   5
W      60  20  86  10
Y      23   2  52  15
Z       1  87  37  20
Extra  55  65  75  85
```