

```

In [10]: 1 # import pandas as pd
          2 import pandas as pd
          3
          4 # import numpy as np
          5 import numpy as np
          6
          7 # simple array
          8 v = np.array([1,2,3,65,765])
          9 data=v
         10 print("array content is:", v)
         11 print(v[0])
         12 #Creating data series
         13 ser = pd.Series(v) ###Series function is used to create a data series
         14 print("Series content is: \n", ser)
         15 print(ser[0])
         16 #retrieve the first N elements of series
         17 print(ser[:5])
         18
         19 ""NOTE: Default index starts from 0""
         20
         21 ### EXPLICITLY ASSIGNING INDEX
         22 ser = pd.Series(data,index=[11,12,13,14,15])
         23 #ser = pd.Series(data,index=['a','b','c','d','e'])
         24 print(ser)
         25 newser = pd.Series(data,index=range(1,6))
         26 print(newser)
         27
         28 ""NOTE: Data series is a single dimensional datatype holding heterogeneous data.
         29 The axis labels are collectively referred to as the index.""

```

```
array content is: [ 1  2  3 65 765]
```

```
1
```

```
Series content is:
```

```
0      1
```

```
1      2
```

```
2      3
```

```
3     65
```

```
4    765
```

```
dtype: int32
```

```
1
```

```
0      1
```

```
1      2
2      3
3      65
4      765
dtype: int32
11     1
12     2
13     3
14     65
15     765
dtype: int32
1      1
2      2
3      3
4      65
5      765
dtype: int32
```

Out[10]: 'NOTE: Data series is a single dimensional datatype holding heterogeneous data. \n\nThe axis labels are collectively referred to as the index.'

In [4]:

```
1  # importing pandas module
2  import pandas as pd
3
4  ### reading csv file
5  df = pd.read_csv("WBCDdata.csv")
6
7  """read_csv() automatically converts the file content into a 2D dataframe(table)"""
8
9
10 ### Extracting country column(series) from the dataframe df
11 ###Indexing a Series using indexing operator [] :
12 ser = pd.Series(df['radius_mean'])
13
14 ### Reading first 20 instances of the series
15 data = ser.head(20)
16 print(data)
17
18 # using indexing operator
19 print(data[3:6])
20
21 # using .iloc() function
22 print(data.iloc[3:6])
```

```
0    17.99
1    20.57
2    19.69
3    11.42
4    20.29
5    12.45
6    18.25
7    13.71
8    13.00
9    12.46
10   16.02
11   15.78
12   19.17
13   15.85
14   13.73
15   14.54
16   14.68
17   16.13
```

```
18    19.81
19    13.54
Name: radius_mean, dtype: float64
3     11.42
4     20.29
5     12.45
Name: radius_mean, dtype: float64
3     11.42
4     20.29
5     12.45
Name: radius_mean, dtype: float64
```

In [3]:

```
1 #####PERFORMING OPERATIONS ON SERIES DATA
2 # importing pandas module
3 import pandas as pd
4
5 # creating a series
6 data1 = pd.Series([5, 2, 3,7], index=['a', 'b', 'c', 'd'])
7
8 # creating a series
9 data2 = pd.Series([1, 6, 4, 9], index=['a', 'b', 'd', 'e'])
10
11 print("data2 details \n", data2, "\n\n", "data1 details \n", data1)
12 print(data1+data2)
13 data2.add(data1, fill_value=0)
```

data2 details

```
a    1
b    6
d    4
e    9
```

dtype: int64

data1 details

```
a    5
b    2
c    3
d    7
```

dtype: int64

a 6.0

b 8.0

c NaN

d 11.0

e NaN

dtype: float64

Out[3]:

a 6.0

b 8.0

c 3.0

d 11.0

e 9.0

dtype: float64

In [6]:

```
1  ### DATATYPE CONVERSION OPERATIONS
2
3  # importing pandas module
4  import pandas as pd
5
6  # reading csv file from url
7  data = pd.read_csv("WBCDdata.csv")
8
9  # dropping null value columns to avoid errors
10 data.dropna(inplace = True)
11
12 # storing dtype before converting
13 before = data.dtypes
14
15 ### converting dtypes using astype
16 data["diagnosis"] = data["diagnosis"].astype(str)
17 data["radius_mean"] = data["radius_mean"].astype(str)
18
19 # storing dtype after converting
20 after = data.dtypes
21
22 # printing to compare
23 print("BEFORE CONVERSION\n", before, "\n")
24 print("AFTER CONVERSION\n", after, "\n")
25
26 ### Converting to List
27 # converting to list
28 country_list = data["diagnosis"].tolist()
29 country_list
```

BEFORE CONVERSION

id	int64
diagnosis	object
radius_mean	float64
texture_mean	float64
perimeter_mean	float64
area_mean	float64
smoothness_mean	float64
compactness_mean	float64
concavity_mean	float64

```

concave points_mean    float64
symmetry_mean          float64
fractal_dimension_mean float64
radius_se              float64
texture_se             float64
perimeter_se          float64
area_se                float64
smoothness_se          float64
compactness_se         float64
concavity_se           float64
concave points_se      float64
symmetry_se            float64
fractal_dimension_se   float64
radius_worst           float64
texture_worst          float64
perimeter_worst        float64
area_worst             float64
smoothness_worst       float64
compactness_worst      float64
concavity_worst        float64
concave points_worst   float64
symmetry_worst         float64
fractal_dimension_worst float64
Unnamed: 32            float64
dtype: object

```

AFTER CONVERSION

```

id                int64
diagnosis          object
radius_mean        object
texture_mean       float64
perimeter_mean     float64
area_mean          float64
smoothness_mean    float64
compactness_mean   float64
concavity_mean     float64
concave points_mean float64
symmetry_mean      float64
fractal_dimension_mean float64
radius_se          float64
texture_se         float64
perimeter_se       float64

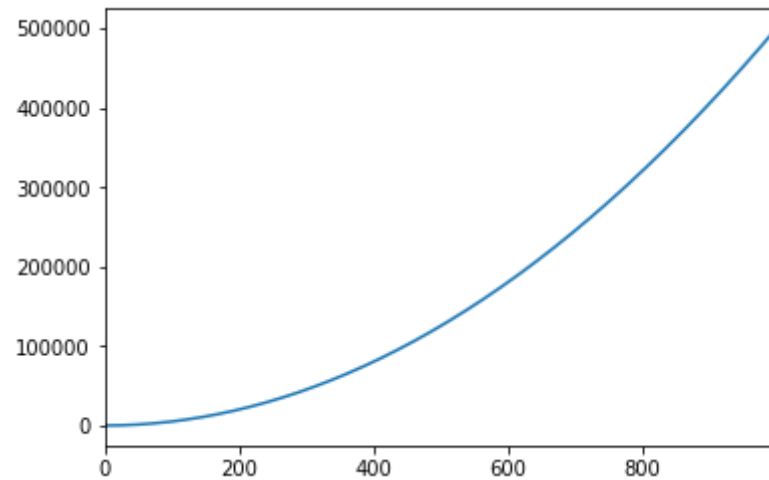
```

area_se	float64
smoothness_se	float64
compactness_se	float64
concavity_se	float64
concave points_se	float64
symmetry_se	float64
fractal_dimension_se	float64
radius_worst	float64
texture_worst	float64
perimeter_worst	float64
area_worst	float64
smoothness_worst	float64
compactness_worst	float64
concavity_worst	float64
concave points_worst	float64
symmetry_worst	float64
fractal_dimension_worst	float64
Unnamed: 32	float64
dtype:	object

Out[6]: []


```
In [8]: 1 import matplotlib.pyplot as plt
        2 import pandas as pd
        3 import numpy as np
        4 x=range(1,1000)
        5 ts = pd.Series(x)
        6 ts = ts.cumsum()
        7 ts.plot()
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x254eb413550>



In []:

1

In []:

1