Ques 10 : How to use the "axis" parameter in pandas?

In [4]:
```
import pandas as pd
drinks_ds = pd.read_csv('http://bit.ly/drinksbycountry')
drinks_ds.head(4)
```

Out[4]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe

In [5]:
```
# Dropping the 'continent' column
drinks_ds.drop('continent', axis=1, inplace=True)
drinks_ds.head(2)
```

Out[5]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
0	Afghanistan	0	0	0	0.0
1	Albania	89	132	54	4.9

In [6]:
```
drinks_ds.drop(1, axis=0,inplace=True)  # Dropping 2nd row
drinks_ds.head(3)
```

Out[6]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
0	Afghanistan	0	0	0	0.0
2	Algeria	25	0	14	0.7
3	Andorra	245	138	312	12.4

In [7]:
```
# We can use aggregation along row-wise (default) or column-wise.
# axis = 0 / axis = 'index' means row-wise
# axis = 1 / axis = 'columns' means column-wise
drinks_ds.mean(axis=0)
```

Out[7]:
```
beer_servings                 106.250000
spirit_servings                80.729167
wine_servings                  49.427083
total_litres_of_pure_alcohol    4.716146
dtype: float64
```

In [8]:
```
drinks_ds.mean(axis='index')
```

Out[8]:
```
beer_servings                 106.250000
spirit_servings                80.729167
wine_servings                  49.427083
total_litres_of_pure_alcohol    4.716146
dtype: float64
```

In [9]:
```
drinks_ds.shape
```

Out[9]: (192, 5)

In [10]:
```
print(drinks_ds.mean(axis=0).shape)
print(drinks_ds.mean(axis=1).shape)
```

```
(4,)
(192,)
```

In [12]:
```
drinks_ds.head()
```

Out[12]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
0	Afghanistan	0	0	0	0.0
2	Algeria	25	0	14	0.7
3	Andorra	245	138	312	12.4
4	Angola	217	57	45	5.9
5	Antigua & Barbuda	102	128	45	4.9

In [11]:
```
drinks_ds.mean(axis=1).head()
```

Out[11]:
```
0      0.000
2      9.925
3    176.850
4     81.225
5     69.975
dtype: float64
```