In [4]:

```
print("Welcome to Pandas-1")
```

```
Welcome to Pandas-1
```

Agenda

- **Installation of pandas** ✅
    - `Importing pandas` ✅
    - `Importing the dataset` ✅
    - `Dataframe/Series` ✅
- **Basic ops on a DataFrame** ✅
    - `df.info()` ✅
    - `df.head()` ✅
    - `df.tail()` ✅
    - `df.shape()` ✅
- **Basic ops on columns** ✅
    - `Different ways of accessing cols` ✅
    - `Check for Unique values`
    - `Rename column`
    - `Deleting col`
    - `Creating new cols`

In [5]:

```
import numpy as np
```

In [6]:

```
!pip install pandas
```

```
Requirement already satisfied: pandas in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (1.3.4)
Requirement already satisfied: python-dateutil>=2.7.3 in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (from pandas) (1.20.3)
Requirement already satisfied: six>=1.5 in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)

[notice] A new release of pip is available: 23.0 -> 23.0.1
[notice] To update, run: pip install --upgrade pip
```

In [7]:

```
import pandas as pd
```

In [8]:

```
!gdown 1E3bwvYGf1ig32RmcYiWc0IXPN-mD_bI_
```

```
Downloading...
From: https://drive.google.com/uc?id=1E3bwvYGf1ig32RmcYiWc0IXPN-mD_bI_
To: /Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/1_Aug_Beg_Mon/08_Pandas_1/mckinsey.csv
100%|████████████████████████████████████████| 83.8k/83.8k [00:00<00:00, 3.31MB/s]
```

In [9]:

```
df=pd.read_csv("/Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/1_Aug_Beg_Mon/08_Pandas_1/mckinsey.csv")
df
```

Out[9]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
...	...	...	...	...	...	...
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

1704 rows × 6 columns

In [46]:

```
print(df)
```

```
          country  year  population continent  life_exp     gdp_cap
0     Afghanistan  1952     8425333      Asia    28.801  779.445314
1     Afghanistan  1957     9240934      Asia    30.332  820.853030
2     Afghanistan  1962    10267083      Asia    31.997  853.100710
3     Afghanistan  1967    11537966      Asia    34.020  836.197138
4     Afghanistan  1972    13079460      Asia    36.088  739.981106
...           ...   ...         ...       ...       ...         ...
1699     Zimbabwe  1987     9216418    Africa    62.351  706.157306
1700     Zimbabwe  1992    10704340    Africa    60.377  693.420786
1701     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1702     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1703     Zimbabwe  2007    12311143    Africa    43.487  469.709298

[1704 rows x 6 columns]
```

In [10]:

```
type(df)
```

Out[10]:

```
pandas.core.frame.DataFrame
```

In [11]:

```
# method 1 # Recommended
df["country"]
```

Out[11]:

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: country, Length: 1704, dtype: object
```

In [12]:

```
# method 2 # not recommended
df.country
```

Out[12]:

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: country, Length: 1704, dtype: object
```

In [13]:

```
# country Name

# df.country Name

# df.1

# funtions/methods overlapping

# cannot read mutiple columns at once
```

In [14]:

```
type(df["country"])
```

Out[14]:

```
pandas.core.series.Series
```

In [ ]:

In [15]:

```
df.head()
```

Out[15]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106

In [16]:

```
df.head(10)
```

Out[16]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
6	Afghanistan	1982	12881816	Asia	39.854	978.011439
7	Afghanistan	1987	13867957	Asia	40.822	852.395945
8	Afghanistan	1992	16317921	Asia	41.674	649.341395
9	Afghanistan	1997	22227415	Asia	41.763	635.341351

In [33]:

```
df.head(15)
```

Out[33]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
6	Afghanistan	1982	12881816	Asia	39.854	978.011439
7	Afghanistan	1987	13867957	Asia	40.822	852.395945
8	Afghanistan	1992	16317921	Asia	41.674	649.341395
9	Afghanistan	1997	22227415	Asia	41.763	635.341351
10	Afghanistan	2002	25268405	Asia	42.129	726.734055
11	Afghanistan	2007	31889923	Asia	43.828	974.580338
12	Albania	1952	1282697	Europe	55.230	1601.056136
13	Albania	1957	1476505	Europe	59.280	1942.284244
14	Albania	1962	1728137	Europe	64.820	2312.888958

In [ ]:

In [18]:

```
df.tail()
```

Out[18]:

	country	year	population	continent	life_exp	gdp_cap
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

In [19]:

```
df.tail(10)
```

Out[19]:

	country	year	population	continent	life_exp	gdp_cap
1694	Zimbabwe	1962	4277736	Africa	52.358	527.272182
1695	Zimbabwe	1967	4995432	Africa	53.995	569.795071
1696	Zimbabwe	1972	5861135	Africa	55.635	799.362176
1697	Zimbabwe	1977	6642107	Africa	57.674	685.587682
1698	Zimbabwe	1982	7636524	Africa	60.363	788.855041
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

In [ ]:

In [20]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     1704 non-null   object 
 1   year        1704 non-null   int64  
 2   population  1704 non-null   int64  
 3   continent   1704 non-null   object 
 4   life_exp    1704 non-null   float64
 5   gdp_cap     1704 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB
```

In [21]:

```
df.describe()
```

Out[21]:

	year	population	life_exp	gdp_cap
count	1704.00000	1.704000e+03	1704.000000	1704.000000
mean	1979.50000	2.960121e+07	59.474439	7215.327081
std	17.26533	1.061579e+08	12.917107	9857.454543
min	1952.00000	6.001100e+04	23.599000	241.165876
25%	1965.75000	2.793664e+06	48.198000	1202.060309
50%	1979.50000	7.023596e+06	60.712500	3531.846988
75%	1993.25000	1.958522e+07	70.845500	9325.462346
max	2007.00000	1.318683e+09	82.603000	113523.132900

In [22]:

```
df.describe(include="all")
```

Out[22]:

	country	year	population	continent	life_exp	gdp_cap
count	1704	1704.00000	1.704000e+03	1704	1704.000000	1704.000000
unique	142	NaN	NaN	5	NaN	NaN
top	Afghanistan	NaN	NaN	Africa	NaN	NaN
freq	12	NaN	NaN	624	NaN	NaN
mean	NaN	1979.50000	2.960121e+07	NaN	59.474439	7215.327081
std	NaN	17.26533	1.061579e+08	NaN	12.917107	9857.454543
min	NaN	1952.00000	6.001100e+04	NaN	23.599000	241.165876
25%	NaN	1965.75000	2.793664e+06	NaN	48.198000	1202.060309
50%	NaN	1979.50000	7.023596e+06	NaN	60.712500	3531.846988
75%	NaN	1993.25000	1.958522e+07	NaN	70.845500	9325.462346
max	NaN	2007.00000	1.318683e+09	NaN	82.603000	113523.132900

In [23]:

```
 df.shape
```

Out[23]:

```
(1704, 6)
```

In [ ]:

In [24]:

```
#creating a data frame
```

In [ ]:

```
1.Row oriented approach
```

In [25]:

```
df.head(2)
```

Out[25]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030

In [26]:

```
pd.DataFrame([["Afghanistan",1952,8425333,"Asia",28.801,779.445314],
              ["Afghanistan",1957,9240934,"Asia",30.332,820.853030]],
            columns=["country","year","population","continent","life_exp","gdp_cap"])
```

Out[26]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030

In [31]:

```
pd.DataFrame([["Afghanistan",1952,8425333,"Asia",28.801,779.445314]],
            columns=["country","year","population","continent","life_exp","gdp_cap"])
```

Out[31]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314

In [36]:

```
df.head(5)
```

Out[36]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106

In [ ]:

```
2.Column oriented approach
```

In [37]:

```
pd.DataFrame({"country":["Afghanistan","Afghanistan"],
              "year":[1952,1957],
              "population":[8425333,8425333],
              "continent":["Asia","Asia"],
              "life_exp":[28.801,30.332],
              "gdp_cap":[779.445314,820.853030],
})
```

Out[37]:

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	8425333	Asia	30.332	820.853030

In [38]:

```
df.keys()
```

Out[38]:

```
Index(['country', 'year', 'population', 'continent', 'life_exp', 'gdp_cap'], dtype='object')
```

In [39]:

```
df.columns
```

Out[39]:

```
Index(['country', 'year', 'population', 'continent', 'life_exp', 'gdp_cap'], dtype='object')
```

In [40]:

```
df["country"]
```

Out[40]:

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: country, Length: 1704, dtype: object
```

In [ ]:

```
["country","year"]
```

In [42]:

```
df[["country","year"]]
```

Out[42]:

	country	year
0	Afghanistan	1952
1	Afghanistan	1957
2	Afghanistan	1962
3	Afghanistan	1967
4	Afghanistan	1972
...	...	...
1699	Zimbabwe	1987
1700	Zimbabwe	1992
1701	Zimbabwe	1997
1702	Zimbabwe	2002
1703	Zimbabwe	2007

1704 rows × 2 columns

In [45]:

```
df[["country"]]
```

Out[45]:

	country
0	Afghanistan
1	Afghanistan
2	Afghanistan
3	Afghanistan
4	Afghanistan
...	...
1699	Zimbabwe
1700	Zimbabwe
1701	Zimbabwe
1702	Zimbabwe
1703	Zimbabwe

1704 rows × 1 columns

In [ ]:

In [52]:

```
a=np.arange(1,5)
b=np.arange(2,6)
print(a)
print(a)
print(a)

a
```

```
[1 2 3 4]
[1 2 3 4]
[1 2 3 4]
```

Out[52]:

```
array([1, 2, 3, 4])
```

In [48]:

```
print(a)
```

```
[1 2 3 4]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: