In [2]:

```
import pandas as pd

purchase_1 = pd.Series({'Name' : 'Chris',
                        'Item Purchased' : 'Dog Food',
                        'Cost' : 22.50})
purchase_2 = pd.Series({'Name' : 'Kevyn',
                        'Item Purchased' : 'Kitty Litter',
                        'Cost' : 2.50})
purchase_3 = pd.Series({'Name' : 'Vinod',
                        'Item Purchased' : 'Bird Seed',
                        'Cost' : 5.00})
df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index = ['Store 1', 'Store 1', 'Store 2'])
df.head()
```

Out[2]:

	Name	Item Purchased	Cost
Store 1	Chris	Dog Food	22.5
Store 1	Kevyn	Kitty Litter	2.5
Store 2	Vinod	Bird Seed	5.0

In [3]:

```
df.loc['Store 1']
```

Out[3]:

	Name	Item Purchased	Cost
Store 1	Chris	Dog Food	22.5
Store 1	Kevyn	Kitty Litter	2.5

In [4]:

```
df.loc[:,['Name', 'Cost']]
```

Out[4]:

	Name	Cost
Store 1	Chris	22.5
Store 1	Kevyn	2.5
Store 2	Vinod	5.0

In [5]:

```
#update value
costs = df['Cost']
costs
```

Out[5]:

```
Store 1    22.5
Store 1     2.5
Store 2     5.0
Name: Cost, dtype: float64
```

In [6]:

```
costs += 2
```

In [7]:

```
costs
```

Out[7]:

```
Store 1    24.5
Store 1     4.5
Store 2     7.0
Name: Cost, dtype: float64
```

In [8]:

```
id(costs)
```

Out[8]:

2543595868752

In [9]:

```
id(df['Cost'])
```

Out[9]:

2543595868752

In [10]:

```
df
```

Out[10]:

	Name	Item Purchased	Cost
Store 1	Chris	Dog Food	22.5
Store 1	Kevyn	Kitty Litter	2.5
Store 2	Vinod	Bird Seed	5.0

In [11]:

```
df['Cost'] = costs
```

In [12]:

```
df
```

Out[12]:

	Name	Item Purchased	Cost
Store 1	Chris	Dog Food	24.5
Store 1	Kevyn	Kitty Litter	4.5
Store 2	Vinod	Bird Seed	7.0

Dataframe Indexing and Loading

In [13]:

```
df = pd.read_csv('olympics.csv')
df.head()
```

Out[13]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	NaN	№ Summer	01 !	02 !	03 !	Total	№ Winter	01 !	02 !	03 !	Total	№ Games	01 !	02 !	03 !
1	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2
2	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8
3	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28
4	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9

In [15]:

```
df = pd.read_csv('olympics.csv', index_col = 0, skiprows = 1)
df.head()
```

Out[15]:

	№ Summer	01 !	02 !	03 !	Total	№ Winter	01 !.1	02 !.1	03 !.1	Total.1	№ Games	01 !.2	02 !.2	0 !.
Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	
Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	
Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	2
Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	
Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	4	

In [16]:

```
df.columns
```

Out[16]:

```
Index(['№ Summer', '01 !', '02 !', '03 !', 'Total', '№ Winter', '01 !.1',
       '02 !.1', '03 !.1', 'Total.1', '№ Games', '01 !.2', '02 !.2', '03 !.2',
       'Combined total'],
      dtype='object')
```

In [17]:

```
for col in df.columns :    #컬럼 리네임하기
    if col [:2] == '01' :  #컬럼에서 두번째 글자까지 01이면
        df.rename(columns = {col : 'Gold' + col[4:]}, inplace = True)    #inplace 해야 df에 바로적용
    if col [:2] == '02' :
        df.rename(columns = {col : 'Silver' + col[4:]}, inplace = True)
    if col [:2] == '03' :
        df.rename(columns = {col : 'Bronze' + col[4:]}, inplace = True)
    if col [:1] == '№' :
        df.rename(columns = {col : '#' + col[1:]}, inplace = True)

df.head()
```

Out[17]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	
Algeria (ALG)	12	5	2	8	15	3	0	0	0	
Argentina (ARG)	23	18	24	28	70	18	0	0	0	
Armenia (ARM)	5	1	2	9	12	6	0	0	0	
Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	

Querying a Dataframe

In [18]:

```
df['Gold'] > 0
```

Out[18]:

```
Afghanistan (AFG)                                False
Algeria (ALG)                                     True
Argentina (ARG)                                   True
Armenia (ARM)                                     True
Australasia (ANZ) [ANZ]                           True
                                                  ...
Independent Olympic Participants (IOP) [IOP]     False
Zambia (ZAM) [ZAM]                               False
Zimbabwe (ZIM) [ZIM]                              True
Mixed team (ZZX) [ZZX]                            True
Totals                                            True
Name: Gold, Length: 147, dtype: bool
```

In [19]:

```
only_gold = df.where(df['Gold'] > 0)
only_gold.head()
```

Out[19]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Afghanistan (AFG)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
Algeria (ALG)	12.0	5.0	2.0	8.0	15.0	3.0	0.0	0.0	0.0	
Argentina (ARG)	23.0	18.0	24.0	28.0	70.0	18.0	0.0	0.0	0.0	
Armenia (ARM)	5.0	1.0	2.0	9.0	12.0	6.0	0.0	0.0	0.0	
Australasia (ANZ) [ANZ]	2.0	3.0	4.0	5.0	12.0	0.0	0.0	0.0	0.0	

In [20]:

```
only_gold['Gold'].count()
```

Out[20]:

```
100
```

In [21]:

```
only_gold.count()
```

Out[21]:

```
# Summer          100
Gold              100
Silver            100
Bronze            100
Total             100
# Winter          100
Gold.1            100
Silver.1          100
Bronze.1          100
Total.1           100
# Games           100
Gold.2            100
Silver.2          100
Bronze.2          100
Combined total    100
dtype: int64
```

In [22]:

```
df.count()
```

Out[22]:

```
# Summer          147
Gold              147
Silver            147
Bronze            147
Total             147
# Winter          147
Gold.1            147
Silver.1          147
Bronze.1          147
Total.1           147
# Games           147
Gold.2            147
Silver.2          147
Bronze.2          147
Combined total    147
dtype: int64
```

In [23]:

```
only_gold.head()
```

Out[23]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Afghanistan (AFG)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
Algeria (ALG)	12.0	5.0	2.0	8.0	15.0	3.0	0.0	0.0	0.0	
Argentina (ARG)	23.0	18.0	24.0	28.0	70.0	18.0	0.0	0.0	0.0	
Armenia (ARM)	5.0	1.0	2.0	9.0	12.0	6.0	0.0	0.0	0.0	
Australasia (ANZ) [ANZ]	2.0	3.0	4.0	5.0	12.0	0.0	0.0	0.0	0.0	

In [24]:

```
only_gold = only_gold.dropna()    #NaN 있으면 제거
only_gold.head()
```

Out[24]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Algeria (ALG)	12.0	5.0	2.0	8.0	15.0	3.0	0.0	0.0	0.0	
Argentina (ARG)	23.0	18.0	24.0	28.0	70.0	18.0	0.0	0.0	0.0	
Armenia (ARM)	5.0	1.0	2.0	9.0	12.0	6.0	0.0	0.0	0.0	
Australasia (ANZ) [ANZ]	2.0	3.0	4.0	5.0	12.0	0.0	0.0	0.0	0.0	
Australia (AUS) [AUS] [Z]	25.0	139.0	152.0	177.0	468.0	18.0	5.0	3.0	4.0	

In [26]:

```
only_gold1 = df[df['Gold'] > 0]
only_gold1.head()
```

Out[26]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tota
Algeria (ALG)	12	5	2	8	15	3	0	0	0	
Argentina (ARG)	23	18	24	28	70	18	0	0	0	
Armenia (ARM)	5	1	2	9	12	6	0	0	0	
Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	
Australia (AUS) [AUS] [Z]	25	139	152	177	468	18	5	3	4	

In [31]:

```
both_gold = df[(df['Gold'] > 0) & (df['Gold.1'] > 0)]
both_gold.head()
```

Out[31]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Total.1
Australia (AUS) [AUS] [Z]	25	139	152	177	468	18	5	3	4	12
Austria (AUT)	26	18	33	35	86	22	59	78	81	218
Belarus (BLR)	5	12	24	39	75	6	6	4	5	15
Belgium (BEL)	25	37	52	53	142	20	1	1	3	5
Bulgaria (BUL) [H]	19	51	85	78	214	19	1	2	3	6

In [34]:

```
any_gold = df[(df['Gold'] > 0) | (df['Gold.1'] > 0)]
len(any_gold)
```

Out[34]:

101

In [38]:

```
#stop the video and resume
#which country get gold medal in winter olympic not in summer olympic?

#winter_gold = df[(df['Gold'] == 0) & (df['Gold.1'] > 0)]
#winter_gold

df[(df['Gold.1'] > 0) & (df['Gold'] == 0)]
```

Out[38]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	To
Liechtenstein (LIE)	16	0	0	0	0	18	2	2	5	

Indexing Dataframes

In [40]:

```
df.head()
```

Out[40]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	
Algeria (ALG)	12	5	2	8	15	3	0	0	0	
Argentina (ARG)	23	18	24	28	70	18	0	0	0	
Armenia (ARM)	5	1	2	9	12	6	0	0	0	
Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	

In [41]:

```
df['country'] = df.index
df.head()
```

Out[41]:

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tot
Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	
Algeria (ALG)	12	5	2	8	15	3	0	0	0	
Argentina (ARG)	23	18	24	28	70	18	0	0	0	
Armenia (ARM)	5	1	2	9	12	6	0	0	0	
Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	

In [43]:

```
df = df.set_index('Combined total')
df.head()
```

Out[43]:

Combined total	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Total.1	Ga
2	13	0	0	2	2	0	0	0	0	0	
15	12	5	2	8	15	3	0	0	0	0	
70	23	18	24	28	70	18	0	0	0	0	
12	5	1	2	9	12	6	0	0	0	0	
12	2	3	4	5	12	0	0	0	0	0	

In [44]:

```
df = df.sort_index()
df
```

Out[44]:

Combined total	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Total.1	G
1	13	0	0	1	1	0	0	0	0	0	
1	13	0	1	0	1	1	0	0	0	0	
1	11	0	1	0	1	0	0	0	0	0	
1	9	0	1	0	1	0	0	0	0	0	
1	16	0	0	1	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
782	15	174	182	217	573	11	78	78	53	209	
806	27	236	272	272	780	22	10	4	12	26	
1204	9	395	319	296	1010	9	78	57	59	194	
2681	26	976	757	666	2399	22	96	102	84	282	
17579	27	4809	4775	5130	14714	22	959	958	948	2865	

147 rows × 15 columns

In [46]:

```
df = df.reset_index()
df.head()
```

Out[46]:

	Combined total	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Total.1
0	1	13	0	0	1	1	0	0	0	0	0
1	1	13	0	1	0	1	1	0	0	0	0
2	1	11	0	1	0	1	0	0	0	0	0
3	1	9	0	1	0	1	0	0	0	0	0
4	1	16	0	0	1	1	0	0	0	0	0

In [47]:

```
df
```

Out[47]:

	Combined total	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Tota
0	1	13	0	0	1	1	0	0	0	0	
1	1	13	0	1	0	1	1	0	0	0	
2	1	11	0	1	0	1	0	0	0	0	
3	1	9	0	1	0	1	0	0	0	0	
4	1	16	0	0	1	1	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
142	782	15	174	182	217	573	11	78	78	53	2
143	806	27	236	272	272	780	22	10	4	12	
144	1204	9	395	319	296	1010	9	78	57	59	1
145	2681	26	976	757	666	2399	22	96	102	84	2
146	17579	27	4809	4775	5130	14714	22	959	958	948	28

147 rows × 16 columns

In [74]:

```
purchase_1 = pd.Series({'Name' : 'Chris',
                        'Item Purchased' : 'Dog Food',
                        'Cost' : 22.50})
purchase_2 = pd.Series({'Name' : 'Kevyn',
                        'Item Purchased' : 'Kitty Litter',
                        'Cost' : 2.50})
purchase_3 = pd.Series({'Name' : 'Vinod',
                        'Item Purchased' : 'Bird Seed',
                        'Cost' : 5.00})
df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index = ['Store 1', 'Store 1', 'Store 2'])
df.head()
```

Out[74]:

	Name	Item Purchased	Cost
Store 1	Chris	Dog Food	22.5
Store 1	Kevyn	Kitty Litter	2.5
Store 2	Vinod	Bird Seed	5.0

In [75]:

```
df['store'] = df.index
df
```

Out[75]:

	Name	Item Purchased	Cost	store
Store 1	Chris	Dog Food	22.5	Store 1
Store 1	Kevyn	Kitty Litter	2.5	Store 1
Store 2	Vinod	Bird Seed	5.0	Store 2

In [76]:

```
df = df.reset_index()
df
```

Out[76]:

	index	Name	Item Purchased	Cost	store
0	Store 1	Chris	Dog Food	22.5	Store 1
1	Store 1	Kevyn	Kitty Litter	2.5	Store 1
2	Store 2	Vinod	Bird Seed	5.0	Store 2

In [77]:

```
df = df.drop('index', axis = 1)
df
```

Out[77]:

	Name	Item Purchased	Cost	store
0	Chris	Dog Food	22.5	Store 1
1	Kevyn	Kitty Litter	2.5	Store 1
2	Vinod	Bird Seed	5.0	Store 2

In [78]:

```
df
```

Out[78]:

	Name	Item Purchased	Cost	store
0	Chris	Dog Food	22.5	Store 1
1	Kevyn	Kitty Litter	2.5	Store 1
2	Vinod	Bird Seed	5.0	Store 2

In [79]:

```
df1 = df.reset_index(drop = True)
df1
```

Out[79]:

	Name	Item Purchased	Cost	store
0	Chris	Dog Food	22.5	Store 1
1	Kevyn	Kitty Litter	2.5	Store 1
2	Vinod	Bird Seed	5.0	Store 2

In [80]:

```
df['store']
```

Out[80]:

```
0    Store 1
1    Store 1
2    Store 2
Name: store, dtype: object
```

In [81]:

```
df['store']
```

Out[81]:

```
0    Store 1
1    Store 1
2    Store 2
Name: store, dtype: object
```

In [82]:

```
df['store'].unique()
```

Out[82]:

```
array(['Store 1', 'Store 2'], dtype=object)
```

In [83]:

```
df = df.set_index(['store', 'Name'])
df
```

Out[83]:

		Item Purchased	Cost
store	**Name**		
Store 1	**Chris**	Dog Food	22.5
	Kevyn	Kitty Litter	2.5
Store 2	**Vinod**	Bird Seed	5.0

In [86]:

```
df.loc['Store 1', 'Chris']
```

Out[86]:

```
Item Purchased    Dog Food
Cost                  22.5
Name: (Store 1, Chris), dtype: object
```