


```
In [ ]: # Set Index
        # reset Index
        # Index
        # multi index
        # Missing value
```

```
In [1]: import pandas as pd
```

```
In [9]: df = pd.read_csv("gas_prices.csv")
        df.head(1)
```

```
Out[9]:
```

	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.0	2.05	2.82	1.16



```
In [7]: df.index
```

```
Out[7]: RangeIndex(start=0, stop=19, step=1)
```

Set Index

```
In [10]: # apply the set method
         df.set_index('Year', inplace=True)
```

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

```
Out[12]:
```

	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
Year										
1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.82	1.16
1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.01	1.14
1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.06	1.13
1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.84	1.11
1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.99	1.11

```
In [13]: df.columns
```

```
Out[13]: Index(['Australia', 'Canada', 'France', 'Germany', 'Italy', 'Japan', 'Mexico',
               'South Korea', 'UK', 'USA'],
              dtype='object')
```

```
In [14]: df.index
```

```
Out[14]: Index([1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
               2002, 2003, 2004, 2005, 2006, 2007, 2008],
              dtype='int64', name='Year')
```

```
In [15]: # accessing the set index value
df.loc[1990]
```

```
Out[15]: Australia      NaN
Canada      1.87
France      3.63
Germany     2.65
Italy       4.59
Japan       3.16
Mexico      1.00
South Korea  2.05
UK          2.82
USA         1.16
Name: 1990, dtype: float64
```

Reset Index

```
In [16]: df.reset_index(inplace=True)
```

```
In [17]: df.head()
```

```
Out[17]:
```

	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.82	1.16
1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.01	1.14
2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.06	1.13
3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.84	1.11
4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.99	1.11



```
In [18]: df.columns
```

```
Out[18]: Index(['Year', 'Australia', 'Canada', 'France', 'Germany', 'Italy', 'Japan',
               'Mexico', 'South Korea', 'UK', 'USA'],
              dtype='object')
```

```
In [19]: df.index
```

```
Out[19]: RangeIndex(start=0, stop=19, step=1)
```

Multi Index

```
In [23]: data = pd.MultiIndex.from_frame(df)
```

```
In [29]: data
```

```
Out[29]: MultiIndex([(1990, nan, 1.87, 3.63, 2.65, 4.59, 3.16, 1.0, 2.05, 2.82, 1.16),
(1991, 1.96, 1.92, 3.45, 2.9, 4.5, 3.46, 1.3, 2.49, 3.01, 1.14),
(1992, 1.89, 1.73, 3.56, 3.27, 4.53, 3.58, 1.5, 2.65, 3.06, 1.13),
(1993, 1.73, 1.57, 3.41, 3.07, 3.68, 4.16, 1.56, 2.88, 2.84, 1.11),
(1994, 1.84, 1.45, 3.59, 3.52, 3.7, 4.36, 1.48, 2.87, 2.99, 1.11),
(1995, 1.95, 1.53, 4.26, 3.96, 4.0, 4.43, 1.11, 2.94, 3.21, 1.15),
(1996, 2.12, 1.61, 4.41, 3.94, 4.39, 3.64, 1.25, 3.18, 3.34, 1.23),
(1997, 2.05, 1.62, 4.0, 3.53, 4.07, 3.26, 1.47, 3.34, 3.83, 1.23),
(1998, 1.63, 1.38, 3.87, 3.34, 3.84, 2.82, 1.49, 3.04, 4.06, 1.06),
(1999, 1.72, 1.52, 3.85, 3.42, 3.87, 3.27, 1.79, 3.8, 4.29, 1.17),
(2000, 1.94, 1.86, 3.8, 3.45, 3.77, 3.65, 2.01, 4.18, 4.58, 1.51),
(2001, 1.71, 1.72, 3.51, 3.4, 3.57, 3.27, 2.2, 3.76, 4.13, 1.46),
(2002, 1.76, 1.69, 3.62, 3.67, 3.74, 3.15, 2.24, 3.84, 4.16, 1.36),
(2003, 2.19, 1.99, 4.35, 4.59, 4.53, 3.47, 2.04, 4.11, 4.7, 1.59),
(2004, 2.72, 2.37, 4.99, 5.24, 5.29, 3.93, 2.03, 4.51, 5.56, 1.88),
(2005, 3.23, 2.89, 5.46, 5.66, 5.74, 4.28, 2.22, 5.28, 5.97, 2.3),
(2006, 3.54, 3.26, 5.88, 6.03, 6.1, 4.47, 2.31, 5.92, 6.36, 2.59),
(2007, 3.85, 3.59, 6.6, 6.88, 6.73, 4.49, 2.4, 6.21, 7.13, 2.8),
(2008, 4.45, 4.08, 7.51, 7.75, 7.63, 5.74, 2.45, 5.83, 7.42, 3.2
7)],
names=['Year', 'Australia', 'Canada', 'France', 'Germany', 'Italy',
'Japan', 'Mexico', 'South Korea', 'UK', 'USA'])
```

```
In [30]: data[18]
```

```
Out[30]: (np.int64(2008),
np.float64(4.45),
np.float64(4.08),
np.float64(7.51),
np.float64(7.75),
np.float64(7.63),
np.float64(5.74),
np.float64(2.45),
np.float64(5.83),
np.float64(7.42),
np.float64(3.27))
```

```
In [32]: df.reset_index(inplace=True)
```

```
In [33]: df
```

Out[33]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
0	0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.8
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.5
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4

Missing Value

```

In [ ]: pandas --->NaN
        Python --->None

# Missing value

1- Identify
   - isnull()
   - isna()

2- delete
   - drop()
   - dropna()
   - drop_duplicates()

```

```
3- fill
   - fillna()
```

```
In [35]: # identify the NaN value

df.isna().sum()
```

```
Out[35]: index      0
        Year      0
        Australia  1
        Canada    0
        France    0
        Germany   0
        Italy     0
        Japan     0
        Mexico    0
        South Korea 0
        UK        0
        USA       0
        dtype: int64
```

```
In [37]: df.isnull().sum()
```


```
Out[37]: index      0
        Year      0
        Australia  1
        Canada    0
        France    0
        Germany   0
        Italy     0
        Japan     0
        Mexico    0
        South Korea 0
        UK        0
        USA       0
        dtype: int64
```

```
In [38]: # dropna
```

```
df.dropna()
```

Out[38]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.9
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4

In [40]: `df.drop_duplicates(inplace=True)`In [41]: `df`

Out[41]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
0	0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.8
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.5
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4



In [48]:

```
# Fillna
df["Name"] = None
df.fillna(0)
```

C:\Users\jitud\AppData\Local\Temp\ipykernel_16904\94207218.py:4: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`

```
df.fillna(0)
```

Out[48]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
0	0	1990	0.00	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.8
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.5
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4



In [50]:

```
df.fillna({  
    "Australia" :0,  
    "Name": "Jeff"  
})
```


Out[50]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
0	0	1990	0.00	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.8
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.5
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4

In [52]: `df.fillna(method='bfill')`

C:\Users\jitud\AppData\Local\Temp\ipykernel_16904\2831856154.py:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffmpeg() or obj.bfill() instead.
df.fillna(method='bfill')

Out[52]:

	index	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	U
0	0	1990	1.96	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.8
1	1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.0
2	2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.0
3	3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.8
4	4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.9
5	5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.2
6	6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.3
7	7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.8
8	8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.0
9	9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.2
10	10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.5
11	11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.7
12	12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.7
13	13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.7
14	14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.5
15	15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.9
16	16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.3
17	17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.7
18	18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.4



In []: