In [ ]:
```
Time series :

where data behavour of data totally dependent on time .
Stock market , Rainfall, Crop

Resampling : we take average of daily based data using week wise , month wise , quater wise

Stock market live data : yahoofinance
```

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

In [2]:
```
df= pd.read_csv('DJI.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-09-23	26851.449219	27011.070313	26831.339844	26949.990234	26949.990234	204240000
1	2019-09-24	27034.070313	27079.679688	26704.960938	26807.769531	26807.769531	301750000
2	2019-09-25	26866.710938	27016.560547	26755.859375	26970.710938	26970.710938	237220000
3	2019-09-26	27004.109375	27015.070313	26803.839844	26891.119141	26891.119141	229180000
4	2019-09-27	26987.259766	27012.539063	26715.820313	26820.250000	26820.250000	217780000

In [4]:
```
df['Date'][0]
```

Out[4]:
```
'2019-09-23'
```

In [5]:
```
type(df['Date'][0])
```

Out[5]:
```
str
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253 entries, 0 to 252
Data columns (total 7 columns):
Date         253 non-null object
Open         253 non-null float64
High         253 non-null float64
Low          253 non-null float64
Close        253 non-null float64
Adj Close    253 non-null float64
Volume       253 non-null int64
dtypes: float64(5), int64(1), object(1)
memory usage: 13.9+ KB
```

In [12]:

```
## converting the data type of date column
##  we have to convert date columnas index
## if any column converted as index it is automatically comes on
## x axis while plotting
df1= pd.read_csv('DJI.csv',parse_dates=['Date'], index_col='Date')
```

In [13]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 253 entries, 2019-09-23 to 2020-09-22
Data columns (total 6 columns):
Open         253 non-null float64
High         253 non-null float64
Low          253 non-null float64
Close        253 non-null float64
Adj Close    253 non-null float64
Volume       253 non-null int64
dtypes: float64(5), int64(1)
memory usage: 13.8 KB
```

```
In [14]: df1.head()
```

Out[14]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-09-23	26851.449219	27011.070313	26831.339844	26949.990234	26949.990234	204240000
2019-09-24	27034.070313	27079.679688	26704.960938	26807.769531	26807.769531	301750000
2019-09-25	26866.710938	27016.560547	26755.859375	26970.710938	26970.710938	237220000
2019-09-26	27004.109375	27015.070313	26803.839844	26891.119141	26891.119141	229180000
2019-09-27	26987.259766	27012.539063	26715.820313	26820.250000	26820.250000	217780000

```
In [10]: df1['Date'][0]

Out[10]: Timestamp('2019-09-23 00:00:00')
```

```
In [11]: type(df1['Date'][0])

Out[11]: pandas._libs.tslib.Timestamp
```

```
In [15]: df1.index

Out[15]: DatetimeIndex(['2019-09-23', '2019-09-24', '2019-09-25', '2019-09-26',
                        '2019-09-27', '2019-09-30', '2019-10-01', '2019-10-02',
                        '2019-10-03', '2019-10-04',
                        ...
                        '2020-09-09', '2020-09-10', '2020-09-11', '2020-09-14',
                        '2020-09-15', '2020-09-16', '2020-09-17', '2020-09-18',
                        '2020-09-21', '2020-09-22'],
                       dtype='datetime64[ns]', name='Date', length=253, freq=None)
```

if we want to filter any month data

In [16]:

```
df1['2020-05']
```

Out[16]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-05-01	24120.779297	24120.779297	23645.300781	23723.689453	23723.689453	418160000
2020-05-04	23581.550781	23769.560547	23361.160156	23749.759766	23749.759766	355360000
2020-05-05	23958.880859	24169.720703	23868.910156	23883.089844	23883.089844	369710000
2020-05-06	23978.880859	24054.589844	23661.140625	23664.640625	23664.640625	377450000
2020-05-07	23837.210938	24094.619141	23834.390625	23875.890625	23875.890625	369890000
2020-05-08	24107.820313	24349.900391	24107.050781	24331.320313	24331.320313	335760000
2020-05-11	24256.449219	24366.210938	24070.220703	24221.990234	24221.990234	352400000
2020-05-12	24292.839844	24382.089844	23761.580078	23764.779297	23764.779297	359480000
2020-05-13	23702.160156	23708.900391	23067.640625	23247.970703	23247.970703	469950000
2020-05-14	23049.060547	23630.859375	22789.619141	23625.339844	23625.339844	472700000
2020-05-15	23454.830078	23730.080078	23354.150391	23685.419922	23685.419922	491510000
2020-05-18	24059.980469	24708.539063	24059.980469	24597.369141	24597.369141	484190000
2020-05-19	24577.480469	24599.500000	24202.960938	24206.859375	24206.859375	374100000
2020-05-20	24455.939453	24649.480469	24455.939453	24575.900391	24575.900391	366510000
2020-05-21	24564.269531	24718.460938	24370.880859	24474.119141	24474.119141	346770000
2020-05-22	24461.980469	24481.640625	24294.070313	24465.160156	24465.160156	255660000
2020-05-26	24781.839844	25176.419922	24781.839844	24995.109375	24995.109375	424380000
2020-05-27	25298.630859	25551.560547	25009.869141	25548.269531	25548.269531	411230000
2020-05-28	25697.359375	25758.789063	25358.730469	25400.640625	25400.640625	377800000
2020-05-29	25324.150391	25482.800781	25031.669922	25383.109375	25383.109375	527620000

In [17]:

```
df1['2020-08']
```

Out[17]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-08-03	26542.320313	26707.259766	26534.380859	26664.400391	26664.400391	439380000
2020-08-04	26664.609375	26832.720703	26597.820313	26828.470703	26828.470703	346350000
2020-08-05	26924.779297	27221.669922	26924.779297	27201.519531	27201.519531	366000000
2020-08-06	27170.820313	27394.099609	27145.250000	27386.980469	27386.980469	322460000
2020-08-07	27321.679688	27456.240234	27223.550781	27433.480469	27433.480469	321170000
2020-08-10	27488.210938	27803.859375	27488.210938	27791.439453	27791.439453	360340000
2020-08-11	27961.640625	28154.880859	27624.509766	27686.910156	27686.910156	428220000
2020-08-12	27860.240234	28043.890625	27843.320313	27976.839844	27976.839844	340210000
2020-08-13	27922.509766	27986.099609	27789.779297	27896.720703	27896.720703	360160000
2020-08-14	27828.929688	27977.810547	27759.390625	27931.019531	27931.019531	275540000
2020-08-17	27970.050781	27999.810547	27816.400391	27844.910156	27844.910156	286860000
2020-08-18	27853.480469	27891.119141	27668.789063	27778.070313	27778.070313	276170000
2020-08-19	27811.259766	27920.419922	27647.669922	27692.880859	27692.880859	311460000
2020-08-20	27622.679688	27781.460938	27526.250000	27739.730469	27739.730469	304550000
2020-08-21	27758.130859	27959.480469	27686.779297	27930.330078	27930.330078	372720000
2020-08-24	28077.580078	28314.939453	28041.750000	28308.460938	28308.460938	383880000
2020-08-25	28347.419922	28400.740234	28094.570313	28248.439453	28248.439453	338420000
2020-08-26	28257.880859	28353.800781	28153.910156	28331.919922	28331.919922	327070000
2020-08-27	28384.070313	28634.220703	28363.929688	28492.269531	28492.269531	404340000
2020-08-28	28601.289063	28733.349609	28487.980469	28653.869141	28653.869141	369740000
2020-08-31	28643.660156	28643.660156	28363.550781	28430.050781	28430.050781	517320000

Data between any two month ¶

```
In [18]: df1['2020-02':'2020-05']
```

Out[18]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-02-03	28319.650391	28630.390625	28319.650391	28399.810547	28399.810547	307910000
2020-02-04	28696.740234	28904.880859	28696.740234	28807.630859	28807.630859	332750000
2020-02-05	29048.730469	29308.890625	29000.849609	29290.849609	29290.849609	357540000
2020-02-06	29388.580078	29408.050781	29246.929688	29379.769531	29379.769531	263700000
2020-02-07	29286.919922	29286.919922	29056.980469	29102.509766	29102.509766	252860000
2020-02-10	28995.660156	29278.070313	28995.660156	29276.820313	29276.820313	250510000
2020-02-11	29390.710938	29415.390625	29210.470703	29276.339844	29276.339844	279540000
2020-02-12	29406.750000	29568.570313	29406.750000	29551.419922	29551.419922	309530000
2020-02-13	29436.029297	29535.400391	29345.929688	29423.310547	29423.310547	291150000
2020-02-14	29440.470703	29463.039063	29283.179688	29398.080078	29398.080078	231000000
2020-02-18	29282.779297	29330.160156	29116.810547	29232.189453	29232.189453	256600000
2020-02-19	29312.699219	29409.089844	29274.380859	29348.029297	29348.029297	240640000
2020-02-20	29296.250000	29368.449219	28959.650391	29219.980469	29219.980469	287780000
2020-02-21	29146.529297	29146.529297	28892.699219	28992.410156	28992.410156	311210000
2020-02-24	28402.929688	28402.929688	27912.439453	27960.800781	27960.800781	452580000
2020-02-25	28037.650391	28149.199219	26997.619141	27081.359375	27081.359375	513270000
2020-02-26	27159.460938	27542.779297	26890.970703	26957.589844	26957.589844	472450000
2020-02-27	26526.000000	26775.310547	25752.820313	25766.640625	25766.640625	664980000
2020-02-28	25270.830078	25494.240234	24681.009766	25409.359375	25409.359375	915990000
2020-03-02	25590.509766	26706.169922	25391.960938	26703.320313	26703.320313	637200000
2020-03-03	26762.470703	27084.589844	25706.279297	25917.410156	25917.410156	647080000
2020-03-04	26383.679688	27102.339844	26286.310547	27090.859375	27090.859375	457590000
2020-03-05	26671.919922	26671.919922	25943.330078	26121.279297	26121.279297	477370000
2020-03-06	25457.210938	25994.380859	25226.619141	25864.779297	25864.779297	599780000

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-03-09	24992.359375	24992.359375	23706.070313	23851.019531	23851.019531	750430000
2020-03-10	24453.000000	25020.990234	23690.339844	25018.160156	25018.160156	654860000
2020-03-11	24604.630859	24604.630859	23328.320313	23553.220703	23553.220703	663960000
2020-03-12	22184.710938	22837.949219	21154.460938	21200.619141	21200.619141	908260000
2020-03-13	21973.820313	23189.759766	21285.369141	23185.619141	23185.619141	843080000
2020-03-16	20917.529297	21768.279297	20116.460938	20188.519531	20188.519531	770130000
...	...	...	...	...	...	...
2020-04-17	23817.150391	24264.210938	23817.150391	24242.490234	24242.490234	525950000
2020-04-20	24095.099609	24108.689453	23627.189453	23650.439453	23650.439453	423410000
2020-04-21	23365.250000	23365.250000	22941.880859	23018.880859	23018.880859	485140000
2020-04-22	23437.339844	23613.099609	23339.599609	23475.820313	23475.820313	352880000
2020-04-23	23543.089844	23885.359375	23483.349609	23515.259766	23515.259766	389290000
2020-04-24	23628.240234	23826.000000	23417.679688	23775.269531	23775.269531	376020000
2020-04-27	23866.150391	24207.650391	23840.609375	24133.779297	24133.779297	389390000
2020-04-28	24357.169922	24512.240234	24031.199219	24101.550781	24101.550781	400250000
2020-04-29	24490.369141	24764.769531	24453.990234	24633.859375	24633.859375	455290000
2020-04-30	24585.570313	24585.570313	24186.900391	24345.720703	24345.720703	478280000
2020-05-01	24120.779297	24120.779297	23645.300781	23723.689453	23723.689453	418160000
2020-05-04	23581.550781	23769.560547	23361.160156	23749.759766	23749.759766	355360000
2020-05-05	23958.880859	24169.720703	23868.910156	23883.089844	23883.089844	369710000
2020-05-06	23978.880859	24054.589844	23661.140625	23664.640625	23664.640625	377450000
2020-05-07	23837.210938	24094.619141	23834.390625	23875.890625	23875.890625	369890000
2020-05-08	24107.820313	24349.900391	24107.050781	24331.320313	24331.320313	335760000
2020-05-11	24256.449219	24366.210938	24070.220703	24221.990234	24221.990234	352400000
2020-05-12	24292.839844	24382.089844	23761.580078	23764.779297	23764.779297	359480000
2020-05-13	23702.160156	23708.900391	23067.640625	23247.970703	23247.970703	469950000

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-05-14	23049.060547	23630.859375	22789.619141	23625.339844	23625.339844	472700000
2020-05-15	23454.830078	23730.080078	23354.150391	23685.419922	23685.419922	491510000
2020-05-18	24059.980469	24708.539063	24059.980469	24597.369141	24597.369141	484190000
2020-05-19	24577.480469	24599.500000	24202.960938	24206.859375	24206.859375	374100000
2020-05-20	24455.939453	24649.480469	24455.939453	24575.900391	24575.900391	366510000
2020-05-21	24564.269531	24718.460938	24370.880859	24474.119141	24474.119141	346770000
2020-05-22	24461.980469	24481.640625	24294.070313	24465.160156	24465.160156	255660000
2020-05-26	24781.839844	25176.419922	24781.839844	24995.109375	24995.109375	424380000
2020-05-27	25298.630859	25551.560547	25009.869141	25548.269531	25548.269531	411230000
2020-05-28	25697.359375	25758.789063	25358.730469	25400.640625	25400.640625	377800000
2020-05-29	25324.150391	25482.800781	25031.669922	25383.109375	25383.109375	527620000

82 rows × 6 columns

Plot

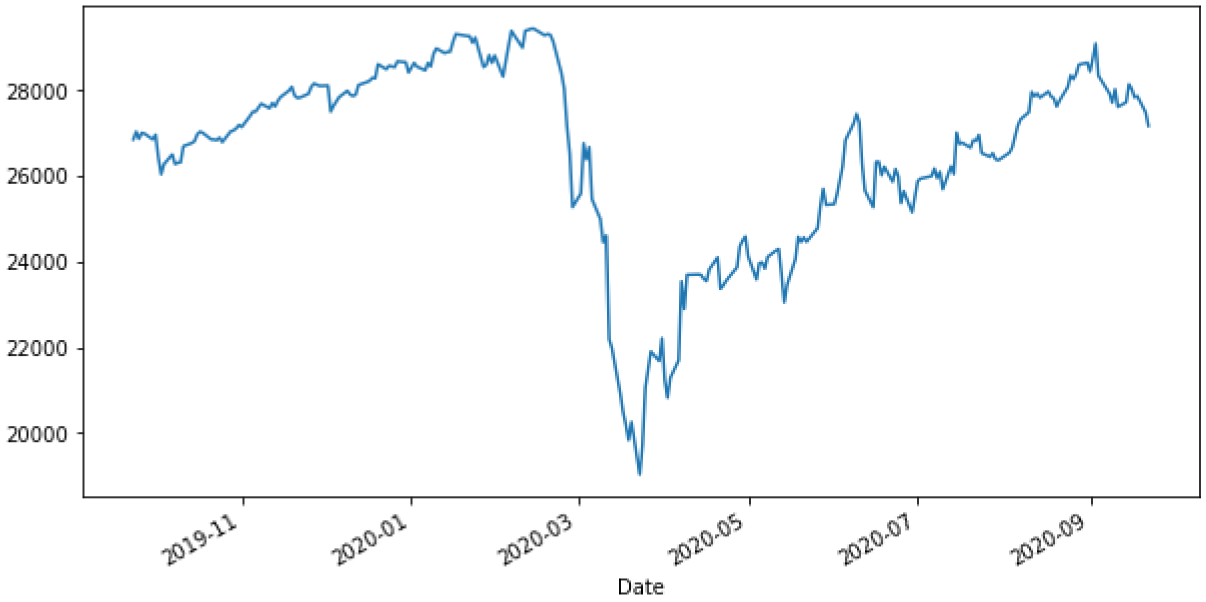
```
In [19]: df1.head()
```

Out[19]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-09-23	26851.449219	27011.070313	26831.339844	26949.990234	26949.990234	204240000
2019-09-24	27034.070313	27079.679688	26704.960938	26807.769531	26807.769531	301750000
2019-09-25	26866.710938	27016.560547	26755.859375	26970.710938	26970.710938	237220000
2019-09-26	27004.109375	27015.070313	26803.839844	26891.119141	26891.119141	229180000
2019-09-27	26987.259766	27012.539063	26715.820313	26820.250000	26820.250000	217780000

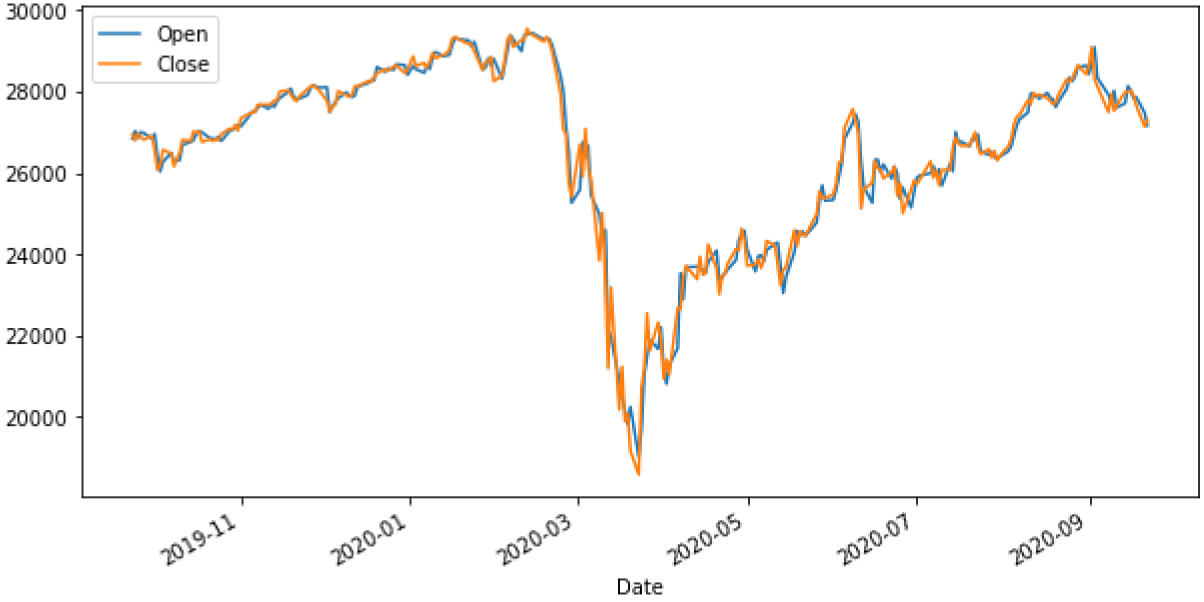
```
In [22]: df1['Open'].plot(figsize=(10,5))
```

```
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91eda35780>
```



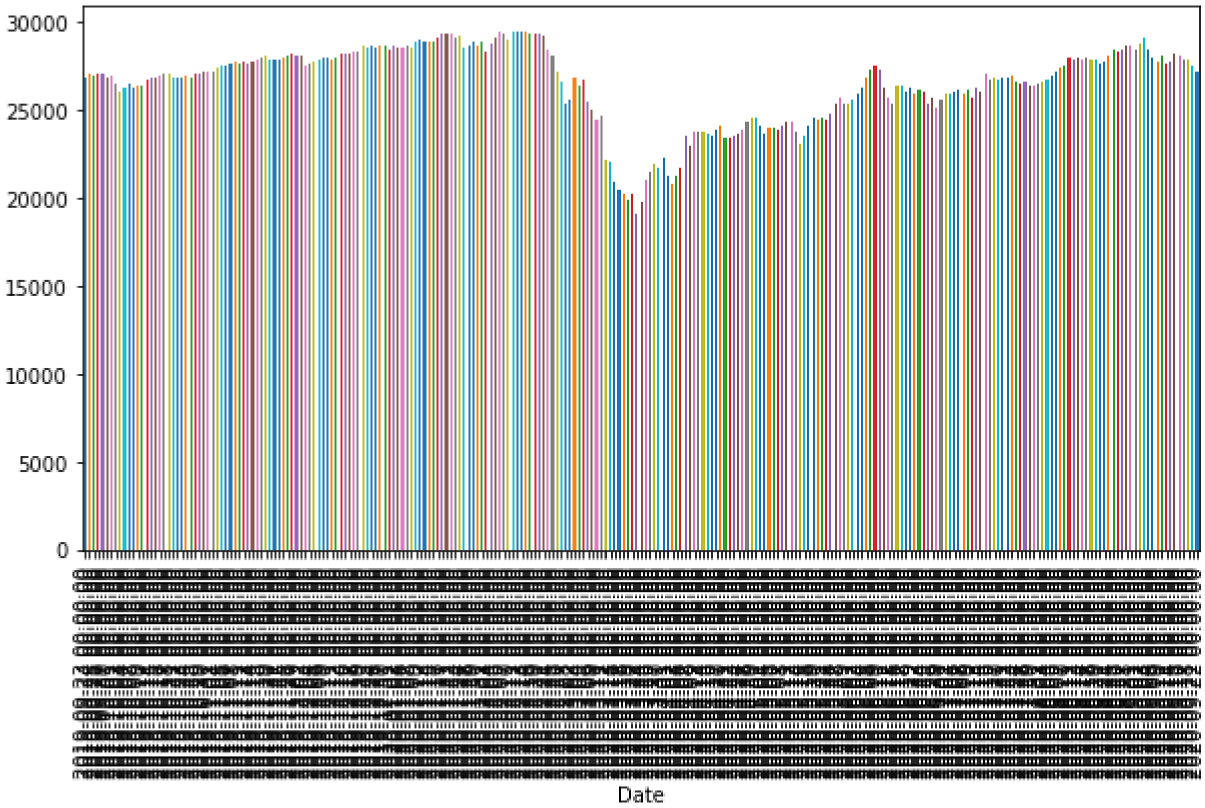
```
In [23]: df1[['Open','Close']].plot(figsize=(10,5))
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed9e8748>
```



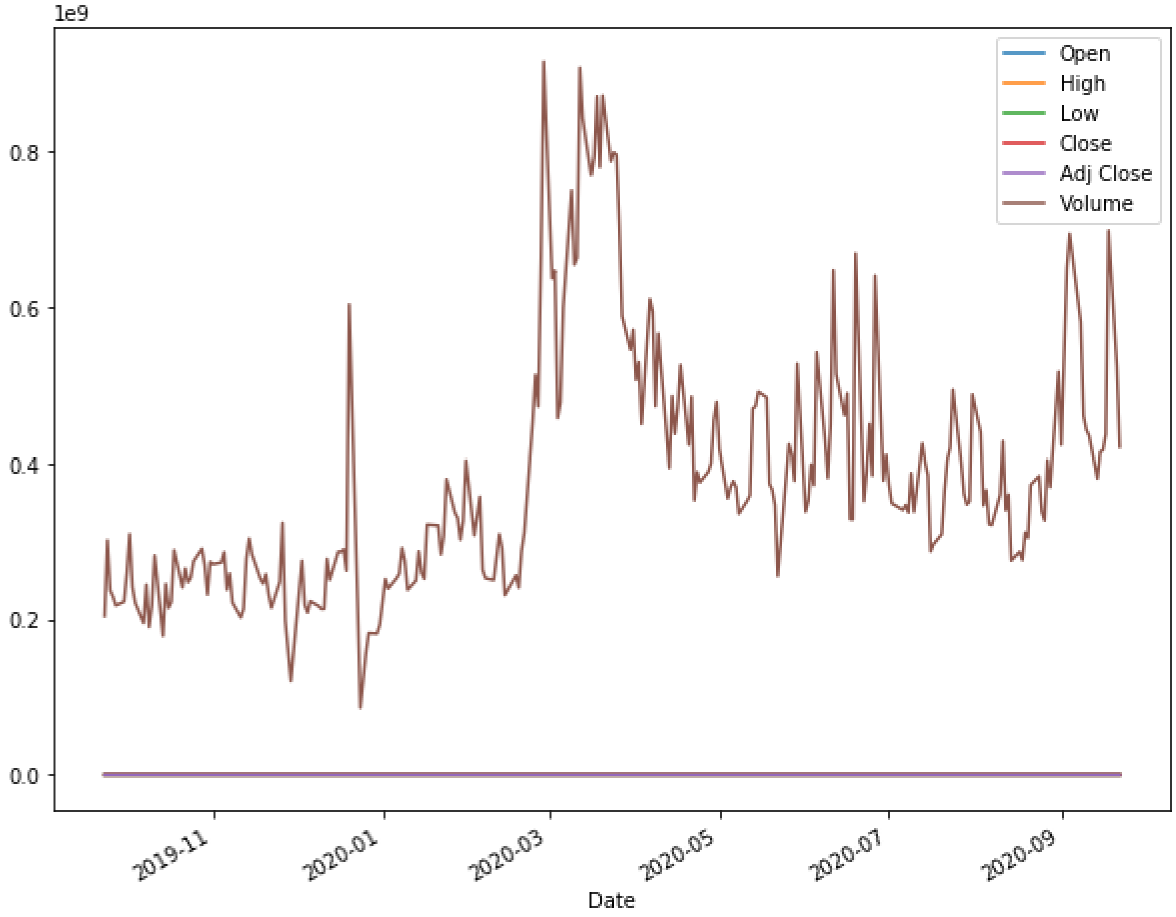
```
In [24]: df1['Open'].plot.bar(figsize=(10,5))
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed9e8358>
```



```
In [25]: df1.plot(figsize=(10,8))
```

```
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed898128>
```



In [26]:

```
df1.head()
```

Out[26]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-09-23	26851.449219	27011.070313	26831.339844	26949.990234	26949.990234	204240000
2019-09-24	27034.070313	27079.679688	26704.960938	26807.769531	26807.769531	301750000
2019-09-25	26866.710938	27016.560547	26755.859375	26970.710938	26970.710938	237220000
2019-09-26	27004.109375	27015.070313	26803.839844	26891.119141	26891.119141	229180000
2019-09-27	26987.259766	27012.539063	26715.820313	26820.250000	26820.250000	217780000

Resampling

In [ ]:

```
Resampling is the process of taking the mean / average
by changing its interval
```

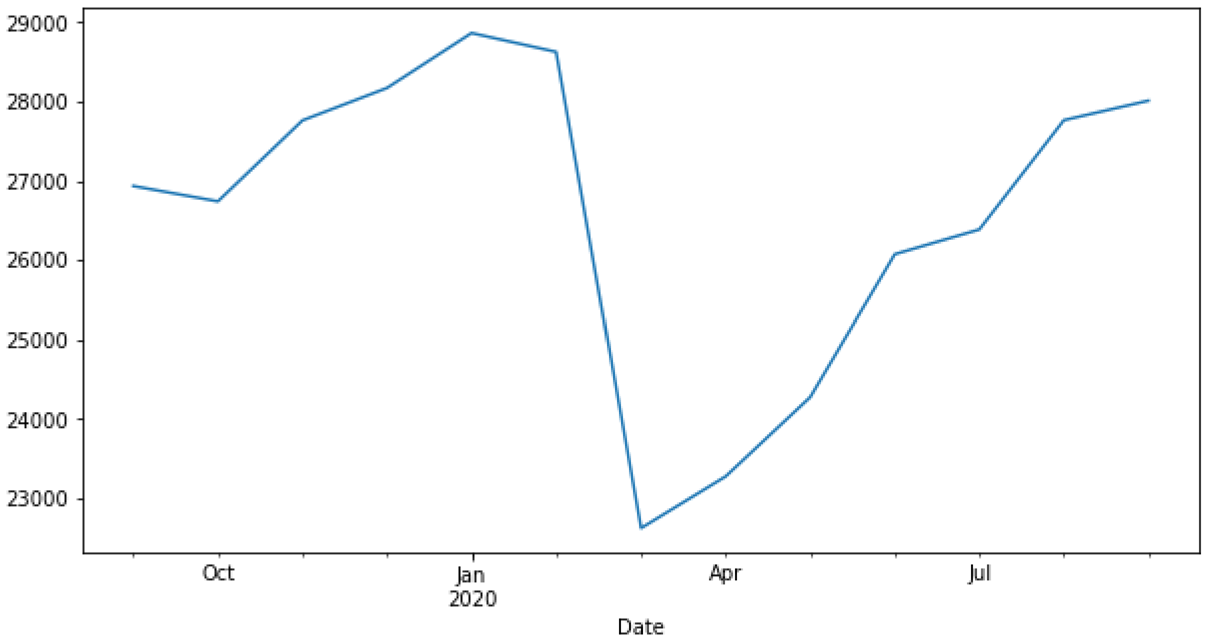
Resampling of Data Month wise

```
In [28]: df1['Open'].resample('M').mean()
```

```
Out[28]: Date
         2019-09-30    26932.654948
         2019-10-31    26739.434188
         2019-11-30    27759.577930
         2019-12-31    28168.544550
         2020-01-31    28860.893880
         2020-02-29    28623.440584
         2020-03-31    22627.505593
         2020-04-30    23279.394438
         2020-05-31    24278.104688
         2020-06-30    26073.461825
         2020-07-31    26384.962269
         2020-08-31    27762.535342
         2020-09-30    28008.383203
         Freq: M, Name: Open, dtype: float64
```

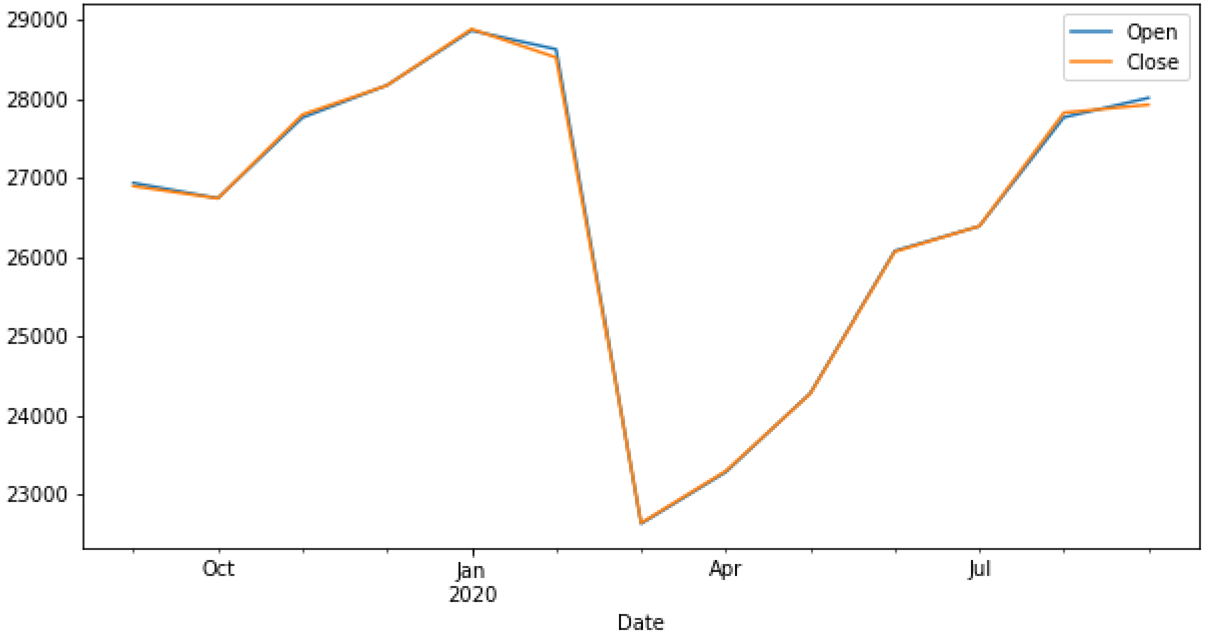
```
In [29]: df1['Open'].resample('M').mean().plot(figsize=(10,5))
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed5a8400>
```



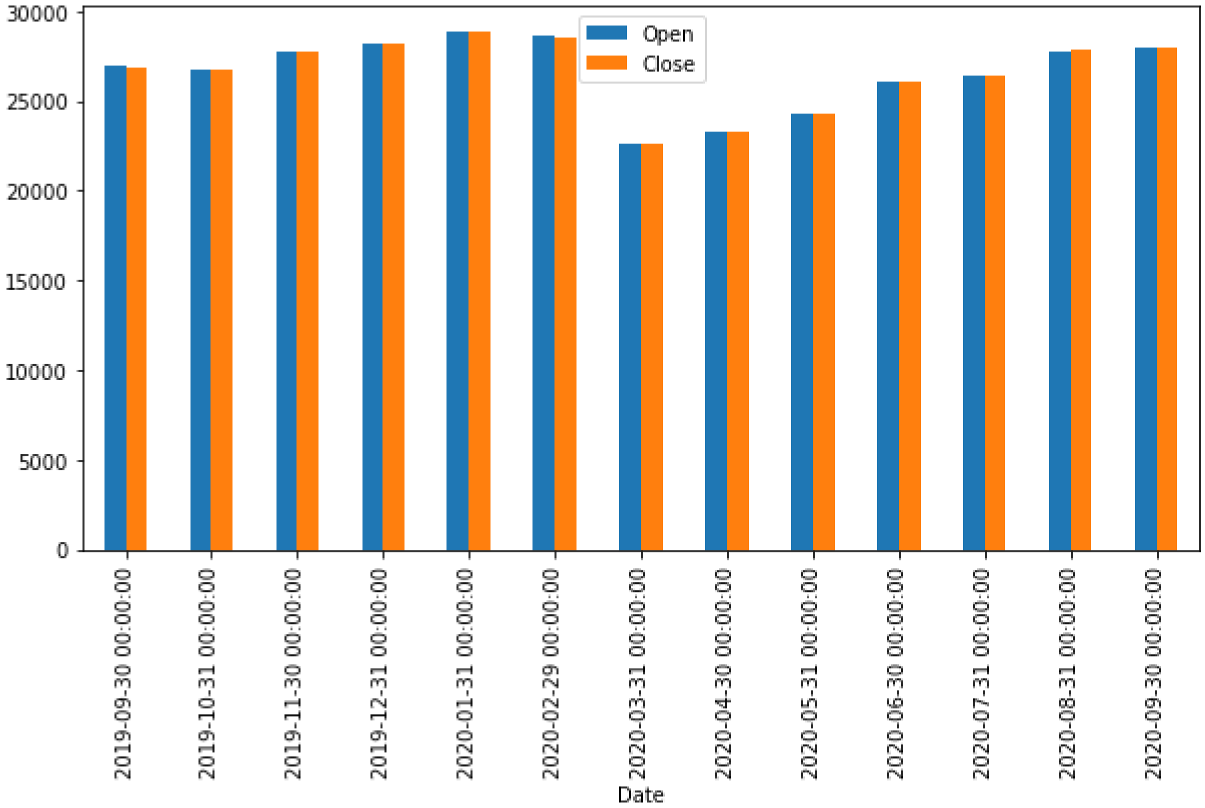
```
In [30]: df1[['Open','Close']].resample('M').mean().plot(figsize=(10,5))
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed659400>
```



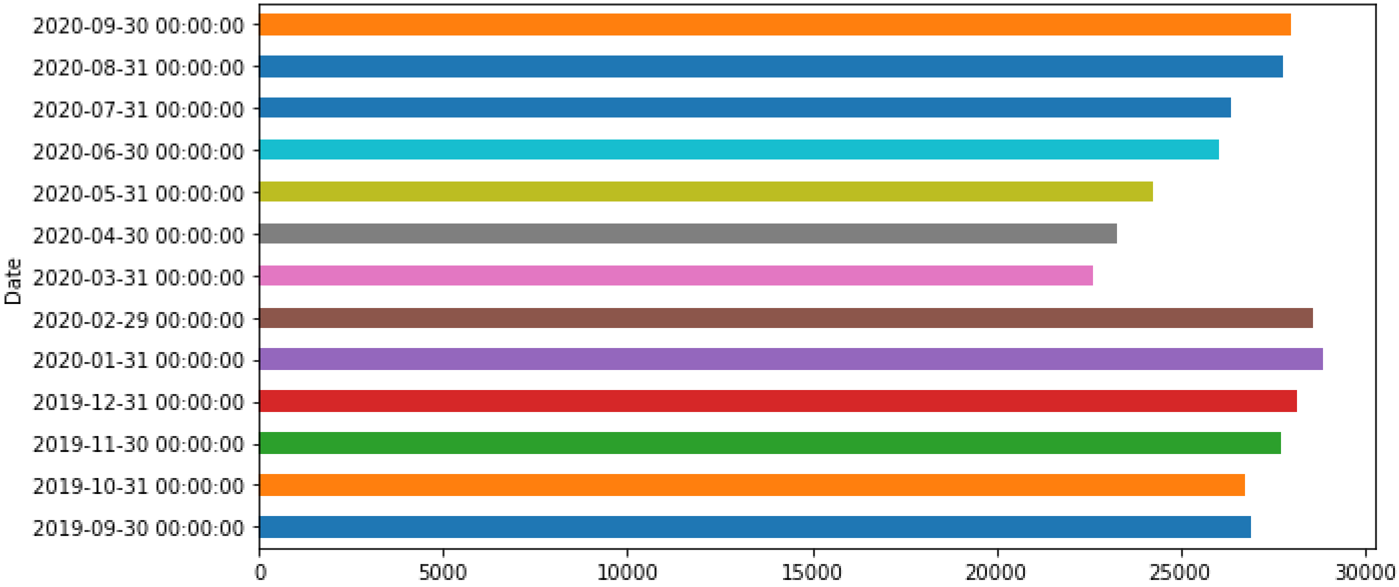
```
In [31]: df1[['Open','Close']].resample('M').mean().plot.bar(figsize=(10,5))

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed53cac8>
```



In [32]:
```
# bar horizontal
df1['Open'].resample('M').mean().plot.barh(figsize=(10,5))
```

Out[32]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f91ed5ec4a8>
```



Weekely Resample

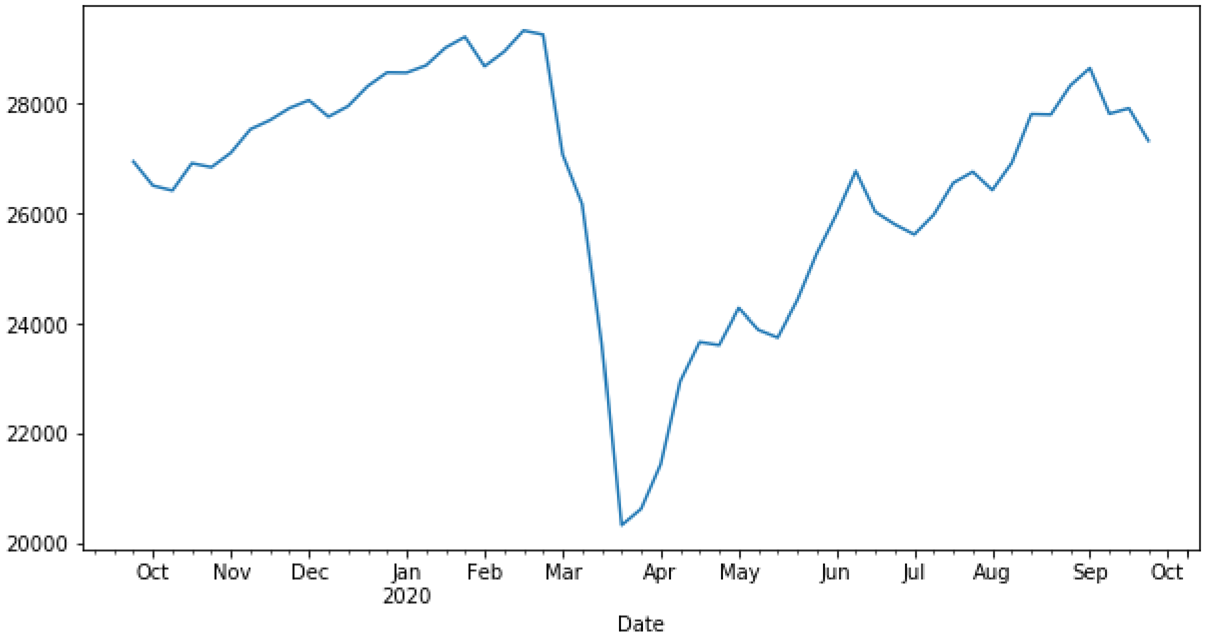
```
In [33]: df1['Open'].resample('W').mean() ### Weekely
```

```
Out[33]: Date
         2019-09-29    26948.719922
         2019-10-06    26510.289453
         2019-10-13    26419.739844
         2019-10-20    26917.362109
         2019-10-27    26844.375781
         2019-11-03    27108.685938
         2019-11-10    27536.278125
         2019-11-17    27701.005469
         2019-11-24    27920.808203
         2019-12-01    28064.537597
         2019-12-08    27764.416406
         2019-12-15    27955.398438
         2019-12-22    28318.362109
         2019-12-29    28569.787598
         2020-01-05    28565.425293
         2020-01-12    28698.062109
         2020-01-19    29022.314063
         2020-01-26    29218.522949
         2020-02-02    28682.099609
         2020-02-09    28948.124219
         2020-02-16    29333.924219
         2020-02-23    29259.564453
         2020-03-01    27079.374219
         2020-03-08    26173.158203
         2020-03-15    23641.704297
         2020-03-22    20335.285938
         2020-03-29    20633.548047
         2020-04-05    21443.882422
         2020-04-12    22953.800293
         2020-04-19    23670.206250
         2020-04-26    23613.803906
         2020-05-03    24284.007813
         2020-05-10    23892.868750
         2020-05-17    23751.067969
         2020-05-24    24423.930078
         2020-05-31    25275.495117
         2020-06-07    25979.136328
         2020-06-14    26774.823828
         2020-06-21    26031.427734
```

```
2020-06-28    25804.868359
2020-07-05    25620.177246
2020-07-12    25980.683984
2020-07-19    26560.048047
2020-07-26    26761.474219
2020-08-02    26428.461719
2020-08-09    26924.841797
2020-08-16    27812.306250
2020-08-23    27803.120313
2020-08-30    28333.648047
2020-09-06    28650.361719
2020-09-13    27818.327637
2020-09-20    27917.847656
2020-09-27    27327.525391
Freq: W-SUN, Name: Open, dtype: float64
```

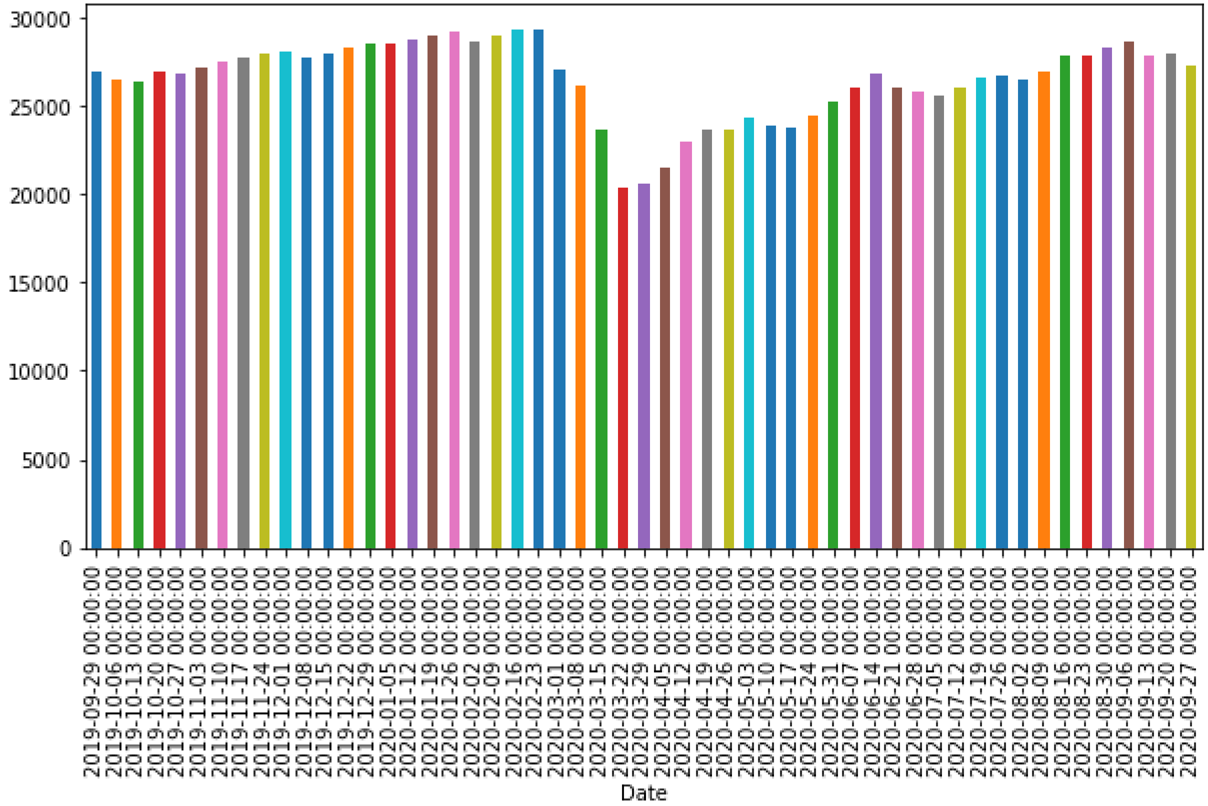
```
In [36]: df1['Open'].resample('W').mean().plot(figsize=(10,5))
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed41b278>
```



```
In [37]: df1['Open'].resample('W').mean().plot.bar(figsize=(10,5))
```

```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed305be0>
```



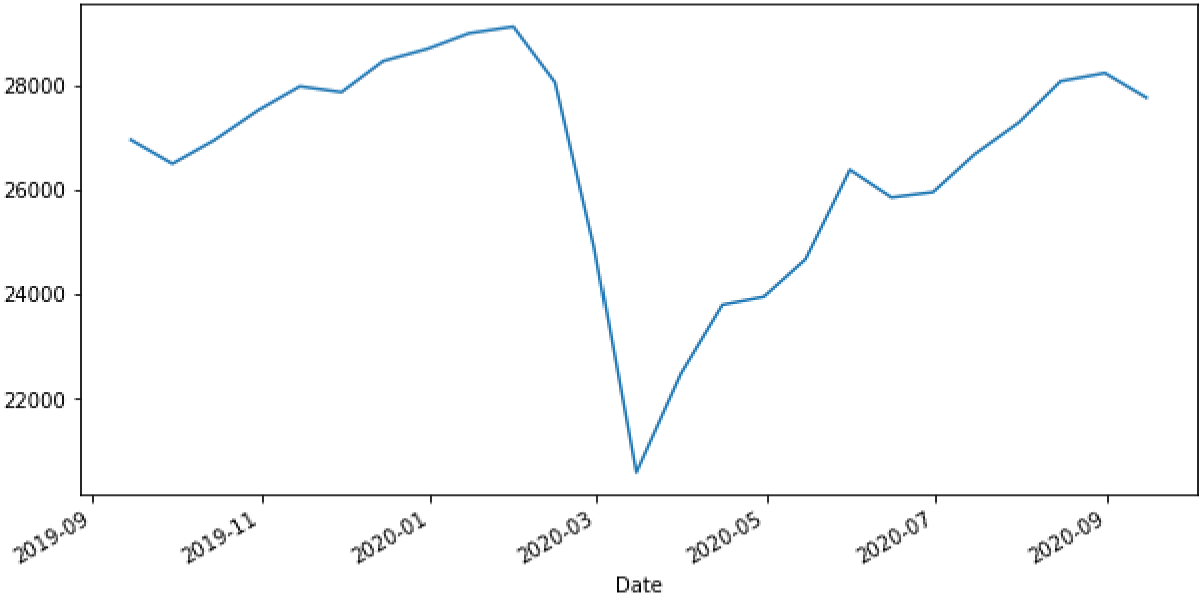
15 Days Period - SM

In [38]:
```
df1['Open'].resample('SM').mean()
```

Out[38]:
```
Date
2019-09-15    26948.719922
2019-09-30    26492.416016
2019-10-15    26937.864258
2019-10-31    27515.836115
2019-11-15    27970.573047
2019-11-30    27859.907422
2019-12-15    28452.572070
2019-12-31    28686.176172
2020-01-15    28983.510121
2020-01-31    29111.207386
2020-02-15    28048.347656
2020-02-29    24907.431250
2020-03-15    20592.944602
2020-03-31    22474.589258
2020-04-15    23794.930930
2020-04-30    23951.927557
2020-05-15    24667.646094
2020-05-31    26376.980078
2020-06-15    25848.539063
2020-06-30    25950.091992
2020-07-15    26672.612468
2020-07-31    27281.370029
2020-08-15    28068.384180
2020-08-31    28224.385938
2020-09-15    27754.258138
Freq: SM-15, Name: Open, dtype: float64
```

```
In [39]: df1['Open'].resample('SM').mean().plot(figsize=(10,5))
```

```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed453208>
```



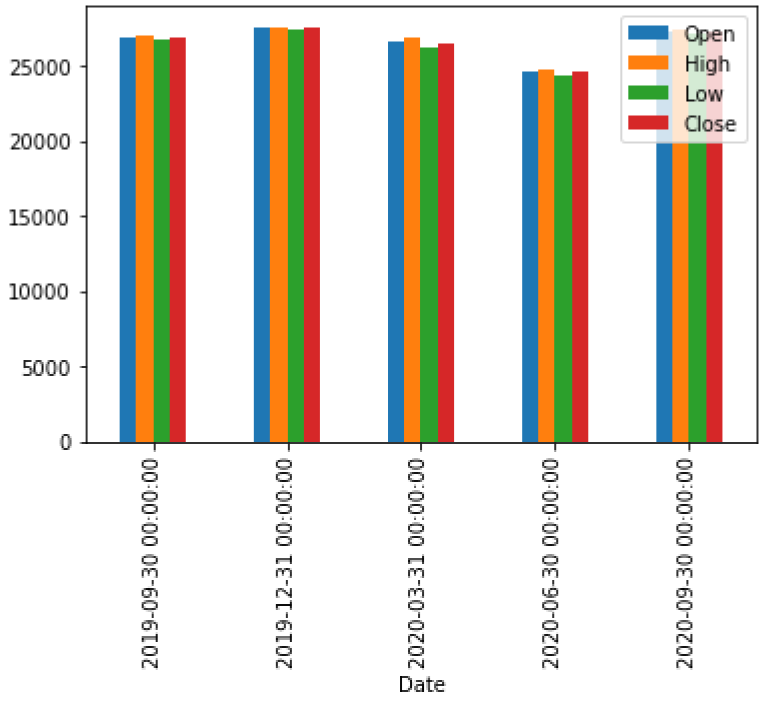
For Quater - Q

```
In [40]: df1['Open'].resample('Q').mean()
```

```
Out[40]: Date
         2019-09-30    26932.654948
         2019-12-31    27527.155945
         2020-03-31    26576.278478
         2020-06-30    24572.151383
         2020-09-30    27303.588968
         Freq: Q-DEC, Name: Open, dtype: float64
```

```
In [43]: df1[['Open','High','Low','Close']].resample('Q').mean().plot.bar()
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f91ed1b0a90>
```



In [45]:
```
df1[['Open','High','Low','Close']].resample('Q').mean()
```

Out[45]:

	Open	High	Low	Close
Date				
2019-09-30	26932.654948	27022.296550	26777.358399	26892.778320
2019-12-31	27527.155945	27624.305176	27439.647644	27537.412354
2020-03-31	26576.278478	26867.389176	26232.766791	26554.484501
2020-06-30	24572.151383	24804.904545	24310.395833	24570.829644
2020-09-30	27303.588968	27474.735722	27113.351731	27302.885608

In [ ]:
```

```