

```
In [70]: import pandas_datareader as pdr
import pandas as pd
from datetime import datetime
import yfinance as yf
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [14]: ## Downloading and importing the data
data = 'TSLA'
```

```
In [17]: yf.download(data)
```

[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed

Out[17]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2010-06-29	1.266667	1.666667	1.169333	1.592667	1.592667	281494500
2010-06-30	1.719333	2.028000	1.553333	1.588667	1.588667	257806500
2010-07-01	1.666667	1.728000	1.351333	1.464000	1.464000	123282000
2010-07-02	1.533333	1.540000	1.247333	1.280000	1.280000	77097000
2010-07-06	1.333333	1.333333	1.055333	1.074000	1.074000	103003500
...	...	...	...	...	...	...
2023-09-11	264.269989	274.850006	260.609985	273.579987	273.579987	174667900
2023-09-12	270.760010	278.390015	266.600006	267.480011	267.480011	135999900
2023-09-13	270.070007	274.980011	268.100006	271.299988	271.299988	111673700
2023-09-14	271.320007	276.709991	270.420013	276.040009	276.040009	107709800
2023-09-15	277.549988	278.980011	271.000000	274.390015	274.390015	133422800

3327 rows × 6 columns

```
In [19]: tsla = yf.download(data)
```

[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed

```
In [22]: tsla.tail()
```

Out[22]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-09-11	264.269989	274.850006	260.609985	273.579987	273.579987	174667900
2023-09-12	270.760010	278.390015	266.600006	267.480011	267.480011	135999900
2023-09-13	270.070007	274.980011	268.100006	271.299988	271.299988	111673700
2023-09-14	271.320007	276.709991	270.420013	276.040009	276.040009	107709800
2023-09-15	277.549988	278.980011	271.000000	274.390015	274.390015	133422800

```
In [45]: # Rearranging the column names
data = pd.DataFrame(tsla)
df_tesla = data[['High', 'Low', 'Open', 'Close', 'Volume', 'Adj Close']]
df_tesla.tail()
```

Out[45]:

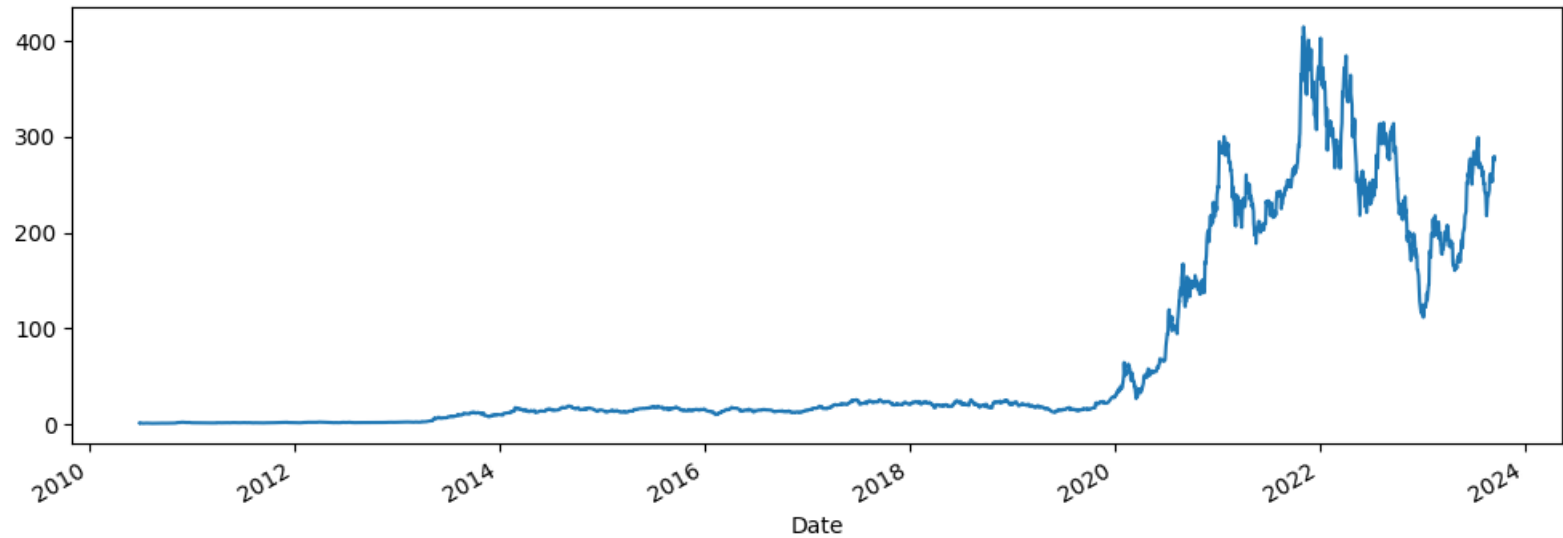
	High	Low	Open	Close	Volume	Adj Close
Date						
2023-09-11	274.850006	260.609985	264.269989	273.579987	174667900	273.579987
2023-09-12	278.390015	266.600006	270.760010	267.480011	135999900	267.480011
2023-09-13	274.980011	268.100006	270.070007	271.299988	111673700	271.299988
2023-09-14	276.709991	270.420013	271.320007	276.040009	107709800	276.040009
2023-09-15	278.980011	271.000000	277.549988	274.390015	133422800	274.390015

```
In [46]: type(df_tesla)
```

Out[46]: pandas.core.frame.DataFrame

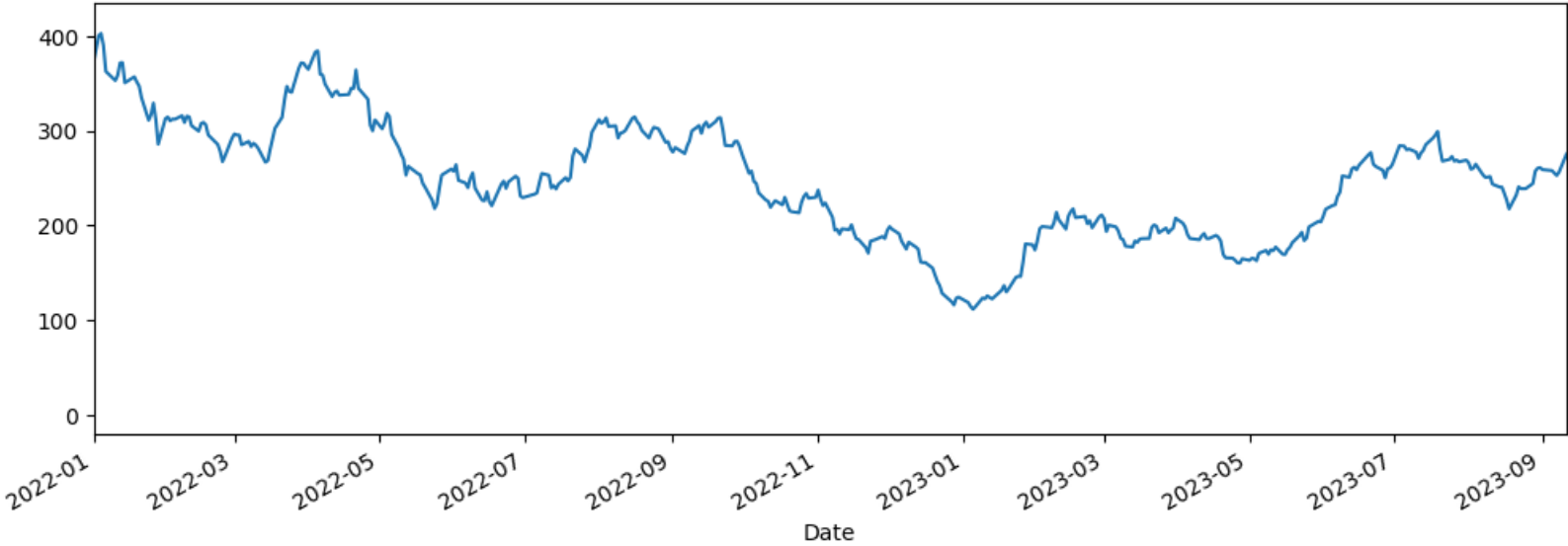
```
In [49]: df_tesla['High'].plot(figsize=(12,4))
```

Out[49]: <Axes: xlabel='Date'>



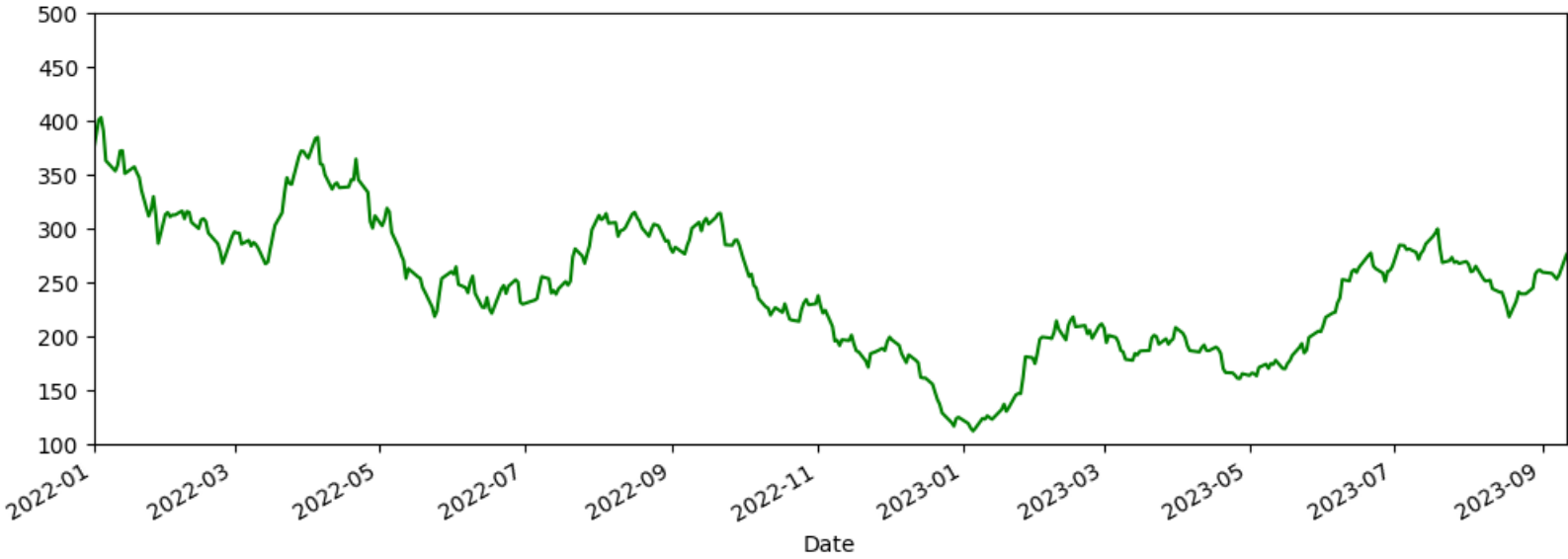
```
In [56]: ## x limit and y limit
df_tesla['High'].plot(xlim = ['2022-01-01','2023-09-11'],figsize=(12,4))
```

Out[56]: <Axes: xlabel='Date'>



```
In [60]: df_tesla['High'].plot(xlim = ['2022-01-01','2023-09-11'],ylim = [100,500], figsize=(12,4), c='green')
```

Out[60]: <Axes: xlabel='Date'>



```
In [62]: df_tesla.index
```

Out[62]: DatetimeIndex(['2010-06-29', '2010-06-30', '2010-07-01', '2010-07-02',  
 '2010-07-06', '2010-07-07', '2010-07-08', '2010-07-09',  
 '2010-07-12', '2010-07-13',  
 ...  
 '2023-09-01', '2023-09-05', '2023-09-06', '2023-09-07',  
 '2023-09-08', '2023-09-11', '2023-09-12', '2023-09-13',  
 '2023-09-14', '2023-09-15'],  
 dtype='datetime64[ns]', name='Date', length=3327, freq=None)

```
In [67]: ## Reviewing data of specific date range
index = df_tesla.loc['2022-01-01':'2023-09-11'].index
share_open = df_tesla.loc['2022-01-01':'2023-09-11']['Open']
```

```
In [68]: share_open
```

Out[68]: Date  
2022-01-03 382.583344  
2022-01-04 396.516663  
2022-01-05 382.216675  
2022-01-06 359.000000  
2022-01-07 360.123322  
...  
2023-09-05 245.000000  
2023-09-06 255.139999  
2023-09-07 245.070007  
2023-09-08 251.220001  
2023-09-11 264.269989  
Name: Open, Length: 424, dtype: float64

```
In [69]: index
```

Out[69]: DatetimeIndex(['2022-01-03', '2022-01-04', '2022-01-05', '2022-01-06',  
 '2022-01-07', '2022-01-10', '2022-01-11', '2022-01-12',  
 '2022-01-13', '2022-01-14',  
 ...  
 '2023-08-28', '2023-08-29', '2023-08-30', '2023-08-31',  
 '2023-09-01', '2023-09-05', '2023-09-06', '2023-09-07',  
 '2023-09-08', '2023-09-11'],  
 dtype='datetime64[ns]', name='Date', length=424, freq=None)

In [78]: `## Subplot on share open prize`  
`figure,axis=plt.subplots()`  
`plt.tight_layout()`  
`figure.autofmt_xdate()`  
`axis.plot(index,share_open)`

Out[78]: [`<matplotlib.lines.Line2D at 0x18f42e07650>`]



## Datetime Index

In [80]: `df_tesla = df_tesla.reset_index()`

In [82]: `df_tesla.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3327 entries, 0 to 3326
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        3327 non-null   datetime64[ns]
1   High        3327 non-null   float64
2   Low         3327 non-null   float64
3   Open        3327 non-null   float64
4   Close       3327 non-null   float64
5   Volume      3327 non-null   int64
6   Adj Close   3327 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 182.1 KB
```

In [83]: `df_tesla.tail()`

Out[83]:

	Date	High	Low	Open	Close	Volume	Adj Close
3322	2023-09-11	274.850006	260.609985	264.269989	273.579987	174667900	273.579987
3323	2023-09-12	278.390015	266.600006	270.760010	267.480011	135999900	267.480011
3324	2023-09-13	274.980011	268.100006	270.070007	271.299988	111673700	271.299988
3325	2023-09-14	276.709991	270.420013	271.320007	276.040009	107709800	276.040009
3326	2023-09-15	278.980011	271.000000	277.549988	274.390015	133422800	274.390015

In [84]: `df_tesla = df_tesla.set_index("Date", drop = True)`

In [86]: `df_tesla.tail()`

Out[86]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2023-09-11	274.850006	260.609985	264.269989	273.579987	174667900	273.579987
2023-09-12	278.390015	266.600006	270.760010	267.480011	135999900	267.480011
2023-09-13	274.980011	268.100006	270.070007	271.299988	111673700	271.299988
2023-09-14	276.709991	270.420013	271.320007	276.040009	107709800	276.040009
2023-09-15	278.980011	271.000000	277.549988	274.390015	133422800	274.390015

```
In [88]: ## year end frequency
df_tesla.resample(rule='A').min()
```

Out[88]:

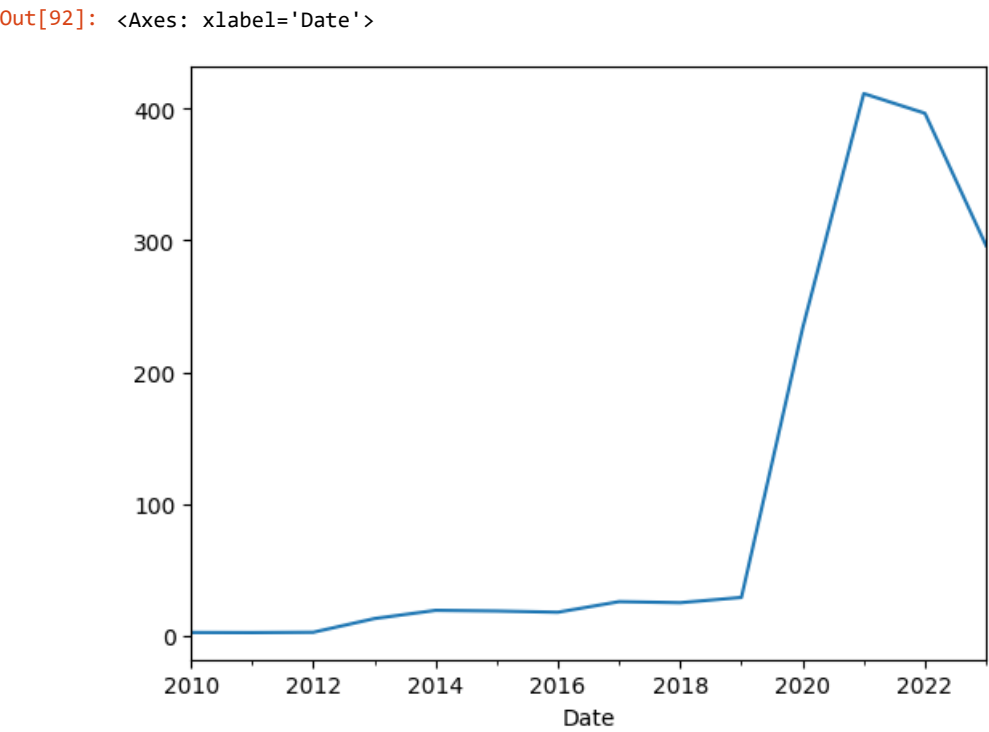
	High	Low	Open	Close	Volume	Adj Close
Date						
2010-12-31	1.108667	0.998667	1.076000	1.053333	1777500	1.053333
2011-12-31	1.484667	1.407333	1.452000	1.455333	3594000	1.455333
2012-12-31	1.790000	1.509333	1.774667	1.519333	5473500	1.519333
2013-12-31	2.225333	2.140667	2.205333	2.194000	6603000	2.194000
2014-12-31	9.800000	9.111333	9.366667	9.289333	19983000	9.289333
2015-12-31	12.619333	12.093333	12.388667	12.333333	10620000	12.333333
2016-12-31	10.331333	9.403333	9.488000	9.578000	24892500	9.578000
2017-12-31	14.688667	14.064000	14.316667	14.466000	32800500	14.466000
2018-12-31	17.355333	16.306000	16.851999	16.704000	46210500	16.704000
2019-12-31	12.445333	11.799333	12.073333	11.931333	36984000	11.931333
2020-12-31	26.990667	23.367332	24.980000	24.081333	52073100	24.081333
2021-12-31	188.736664	179.830002	184.183334	187.666672	29401800	187.666672
2022-12-31	116.269997	108.239998	110.349998	109.099998	41864700	109.099998
2023-12-31	111.750000	101.809998	103.000000	108.099998	83166000	108.099998

```
In [90]: df_tesla.resample(rule='A').max()
```

Out[90]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2010-12-31	2.428000	2.316667	2.391333	2.364667	281494500	2.364667
2011-12-31	2.333333	2.268667	2.308667	2.329333	172767000	2.329333
2012-12-31	2.663333	2.474000	2.546000	2.534000	85206000	2.534000
2013-12-31	12.966667	12.558000	12.930667	12.891333	557458500	12.891333
2014-12-31	19.427999	18.693333	19.177999	19.069332	490225500	19.069332
2015-12-31	19.110001	18.420000	18.680000	18.817333	234744000	18.817333
2016-12-31	17.955999	16.967333	17.763332	17.694668	356136000	17.694668
2017-12-31	25.974001	25.290001	25.779333	25.666668	296871000	25.666668
2018-12-31	25.830667	24.474667	25.000000	25.304667	504745500	25.304667
2019-12-31	29.020666	28.423332	29.000000	28.729334	450091500	28.729334
2020-12-31	239.573334	230.373337	233.330002	235.223328	914082000	235.223328
2021-12-31	414.496674	405.666656	411.470001	409.970001	268189500	409.970001
2022-12-31	402.666656	378.679993	396.516663	399.926666	221923300	399.926666
2023-12-31	299.290009	289.519989	296.040009	293.339996	306590600	293.339996

```
In [92]: df_tesla.resample(rule='A').max()['Open'].plot()
```

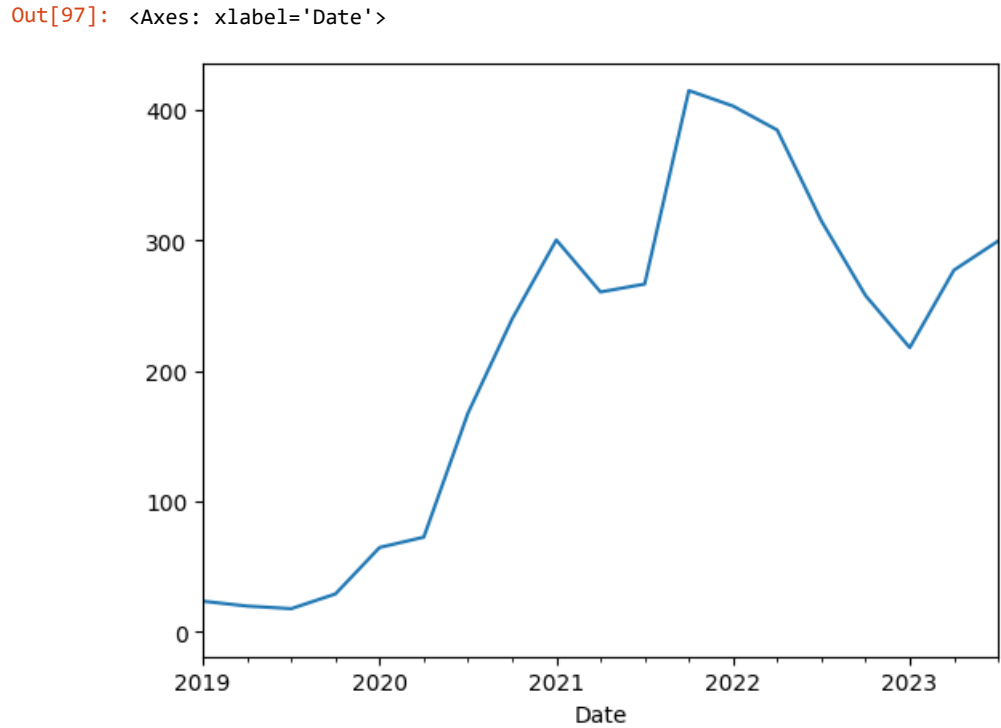


In [105]: *## Quartly end frequency*  
df\_tesla.resample(rule='QS').max().loc[ '2019-01-01': '2023-09-11' ]

Out[105]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2019-01-01	23.466667	22.943333	23.080667	23.153999	362262000	23.153999
2019-04-01	19.744667	19.144667	19.219999	19.454000	398206500	19.454000
2019-07-01	17.738001	17.210667	17.278000	17.658667	336274500	17.658667
2019-10-01	29.020666	28.423332	29.000000	28.729334	450091500	28.729334
2020-01-01	64.599335	60.068001	61.566666	61.161331	914082000	61.161331
2020-04-01	72.512665	66.915337	67.518669	71.987335	487977000	71.987335
2020-07-01	167.496674	156.836670	167.380005	166.106674	584781000	166.106674
2020-10-01	239.573334	230.373337	233.330002	235.223328	666378600	235.223328
2021-01-01	300.133331	290.533325	297.126678	294.363342	268189500	294.363342
2021-04-01	260.263336	244.203339	256.899994	254.106674	147052200	254.106674
2021-07-01	266.333344	258.333344	262.399994	263.786682	100847400	263.786682
2021-10-01	414.496674	405.666656	411.470001	409.970001	188556300	409.970001
2022-01-01	402.666656	378.679993	396.516663	399.926666	151565700	399.926666
2022-04-01	384.290009	362.433319	378.766663	381.816681	144973200	381.816681
2022-07-01	314.666656	305.579987	311.666656	309.320007	142032300	309.320007
2022-10-01	257.500000	242.009995	254.500000	249.440002	221923300	249.440002
2023-01-01	217.649994	206.110001	211.759995	214.240005	306590600	214.240005
2023-04-01	276.989990	261.119995	275.130005	274.450012	211797100	274.450012
2023-07-01	299.290009	289.519989	296.040009	293.339996	175158300	293.339996

In [97]: df\_tesla.resample(rule='QS').max()['High'].plot(xlim = [ '2019-01-01', '2023-09-11' ])



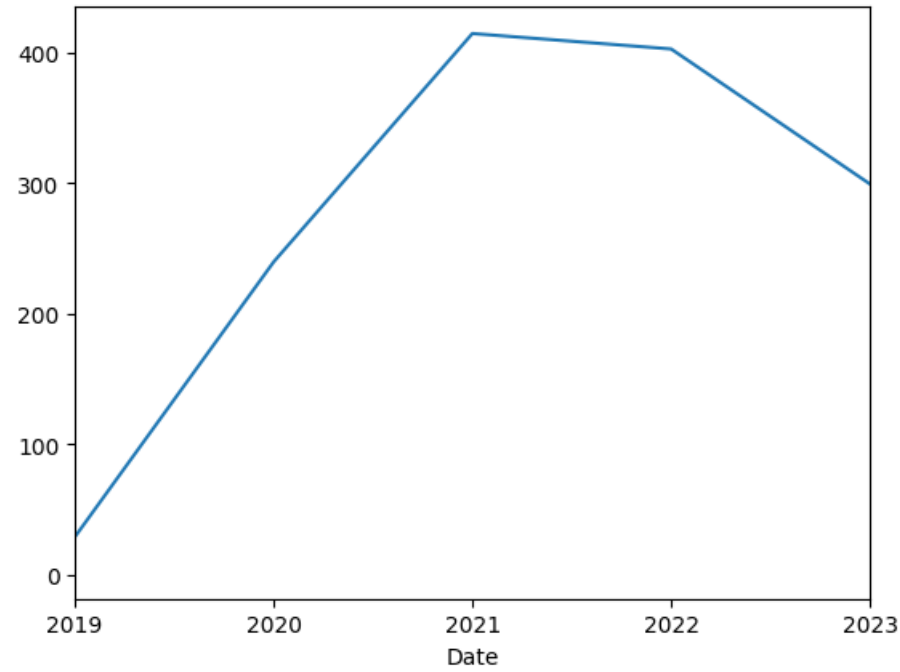
In [99]: *## Business End Frequency*  
df\_tesla.resample(rule='BA').max()

Out[99]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2010-12-31	2.428000	2.316667	2.391333	2.364667	281494500	2.364667
2011-12-30	2.333333	2.268667	2.308667	2.329333	172767000	2.329333
2012-12-31	2.663333	2.474000	2.546000	2.534000	85206000	2.534000
2013-12-31	12.966667	12.558000	12.930667	12.891333	557458500	12.891333
2014-12-31	19.427999	18.693333	19.177999	19.069332	490225500	19.069332
2015-12-31	19.110001	18.420000	18.680000	18.817333	234744000	18.817333
2016-12-30	17.955999	16.967333	17.763332	17.694668	356136000	17.694668
2017-12-29	25.974001	25.290001	25.779333	25.666668	296871000	25.666668
2018-12-31	25.830667	24.474667	25.000000	25.304667	504745500	25.304667
2019-12-31	29.020666	28.423332	29.000000	28.729334	450091500	28.729334
2020-12-31	239.573334	230.373337	233.330002	235.223328	914082000	235.223328
2021-12-31	414.496674	405.666656	411.470001	409.970001	268189500	409.970001
2022-12-30	402.666656	378.679993	396.516663	399.926666	221923300	399.926666
2023-12-29	299.290009	289.519989	296.040009	293.339996	306590600	293.339996

In [100]: df\_tesla.resample(rule='BA').max()['High'].plot(xlim = ['2019-01-01','2023-09-11'])

Out[100]: <Axes: xlabel='Date'>



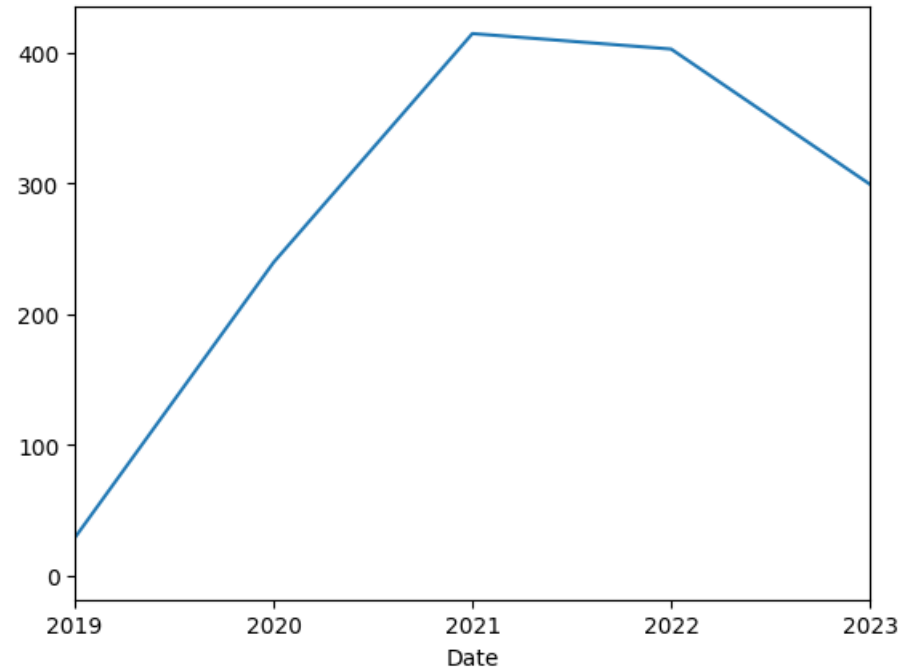
In [106]: *## Business End Quartly Frequency*  
df\_tesla.resample(rule='BQS').max().loc['2019-01-01':'2023-09-11']

Out[106]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2019-01-01	23.466667	22.943333	23.080667	23.153999	362262000	23.153999
2019-04-01	19.744667	19.144667	19.219999	19.454000	398206500	19.454000
2019-07-01	17.738001	17.210667	17.278000	17.658667	336274500	17.658667
2019-10-01	29.020666	28.423332	29.000000	28.729334	450091500	28.729334
2020-01-01	64.599335	60.068001	61.566666	61.161331	914082000	61.161331
2020-04-01	72.512665	66.915337	67.518669	71.987335	487977000	71.987335
2020-07-01	167.496674	156.836670	167.380005	166.106674	584781000	166.106674
2020-10-01	239.573334	230.373337	233.330002	235.223328	666378600	235.223328
2021-01-01	300.133331	290.533325	297.126678	294.363342	268189500	294.363342
2021-04-01	260.263336	244.203339	256.899994	254.106674	147052200	254.106674
2021-07-01	266.333344	258.333344	262.399994	263.786682	100847400	263.786682
2021-10-01	414.496674	405.666656	411.470001	409.970001	188556300	409.970001
2022-01-03	402.666656	378.679993	396.516663	399.926666	151565700	399.926666
2022-04-01	384.290009	362.433319	378.766663	381.816681	144973200	381.816681
2022-07-01	314.666656	305.579987	311.666656	309.320007	142032300	309.320007
2022-10-03	257.500000	242.009995	254.500000	249.440002	221923300	249.440002
2023-01-02	217.649994	206.110001	211.759995	214.240005	306590600	214.240005
2023-04-03	276.989990	261.119995	275.130005	274.450012	211797100	274.450012
2023-07-03	299.290009	289.519989	296.040009	293.339996	175158300	293.339996

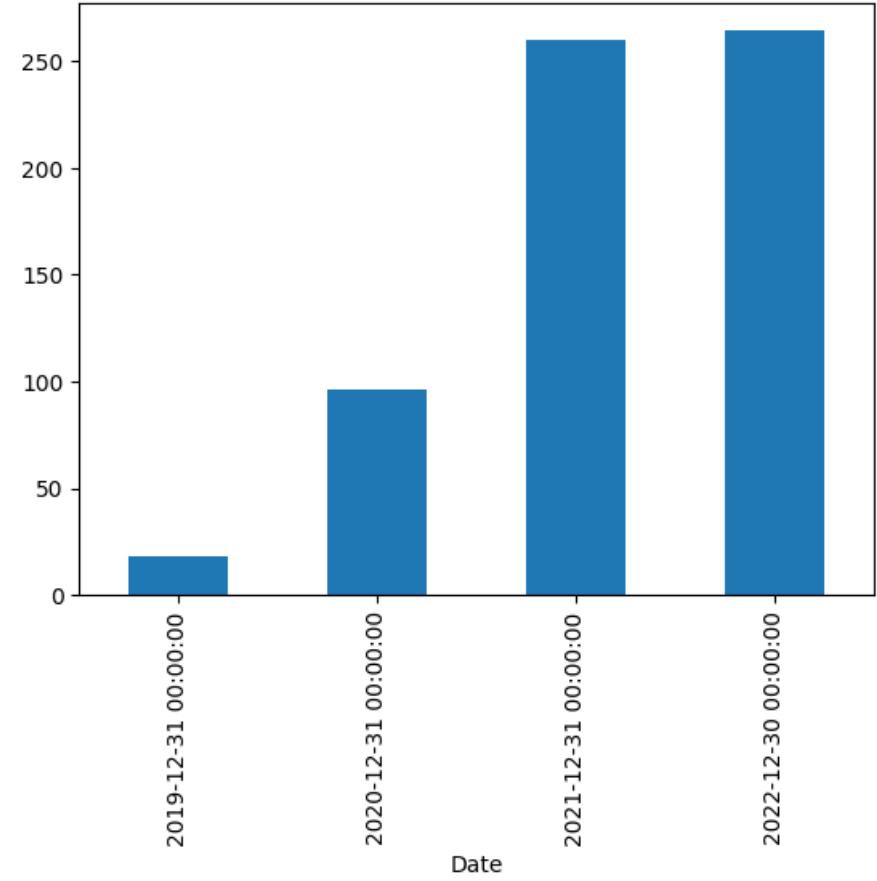
In [102]: df\_tesla.resample(rule='BA').max()['High'].plot(xlim = ['2019-01-01','2023-09-11'])

Out[102]: <Axes: xlabel='Date'>



```
In [115]: ## Plotting Graphs
df_tesla['Open'].resample(rule='BA').mean().loc['2019-01-01':'2023-09-11'].plot(kind='bar')
```

Out[115]: <Axes: xlabel='Date'>

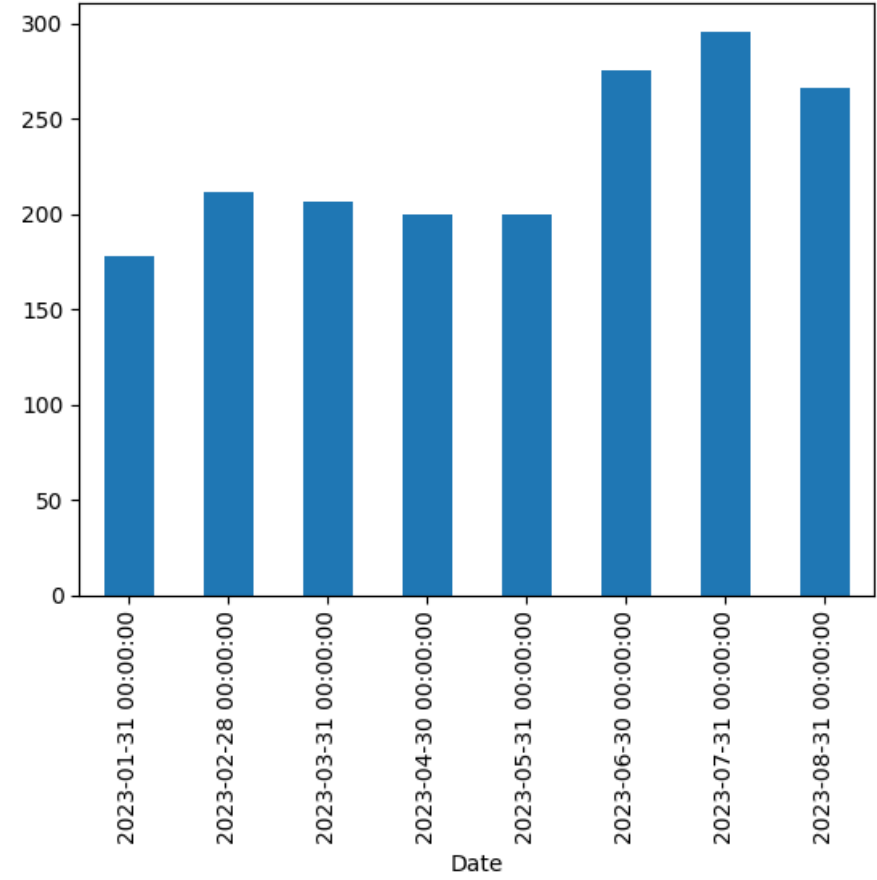


```
In [116]: ## Monthly share price data
df_tesla['Open'].resample(rule='M').max().loc['2023-01-01':'2023-09-11']
```

Out[116]: Date  
2023-01-31 178.050003  
2023-02-28 211.759995  
2023-03-31 206.210007  
2023-04-30 199.910004  
2023-05-31 200.100006  
2023-06-30 275.130005  
2023-07-31 296.040009  
2023-08-31 266.260010  
Freq: M, Name: Open, dtype: float64

```
In [114]: df_tesla['Open'].resample(rule='M').max().loc['2023-01-01':'2023-09-11'].plot(kind='bar')
```

Out[114]: <Axes: xlabel='Date'>



```
In [118]: ## Aggregate Function
df_tesla['High'].rolling(5).max().head(6)
```

Out[118]: Date  
2010-06-29 NaN  
2010-06-30 NaN  
2010-07-01 NaN  
2010-07-02 NaN  
2010-07-06 2.028  
2010-07-07 2.028  
Name: High, dtype: float64

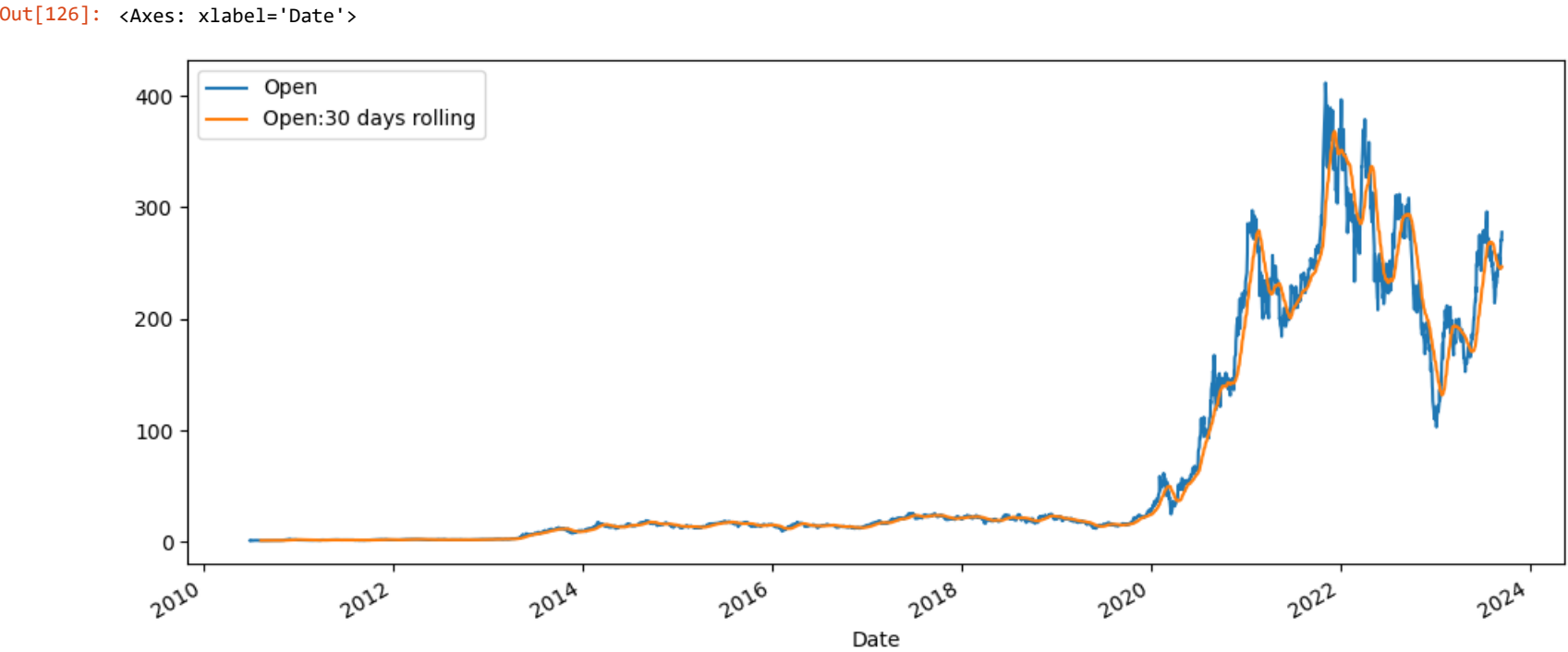
```
In [125]: df_tesla['Open:30 days rolling'] = df_tesla['Open'].rolling(30).mean()
```

```
In [120]: df_tesla.head()
```

Out[120]:

	High	Low	Open	Close	Volume	Adj Close	Open:30 days rolling
Date							
2010-06-29	1.666667	1.169333	1.266667	1.592667	281494500	1.592667	NaN
2010-06-30	2.028000	1.553333	1.719333	1.588667	257806500	1.588667	NaN
2010-07-01	1.728000	1.351333	1.666667	1.464000	123282000	1.464000	NaN
2010-07-02	1.540000	1.247333	1.533333	1.280000	77097000	1.280000	NaN
2010-07-06	1.333333	1.055333	1.333333	1.074000	103003500	1.074000	NaN

```
In [126]: df_tesla[['Open', 'Open:30 days rolling']].plot(figsize=(12,5))
```



```
In [ ]:
```