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

In [2]:
```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
plt.show()
```

In [3]:
```
df = pd.read_csv("amazon_stock.csv")
df
```

Out[3]:
```
      None  ticker        Date     Open       High       Low      Close   Volume  Adj_Close
   0     0  AMZN     3/27/2018  1572.40  1575.9600  1482.320  1497.0500  6793279  1497.0500
   1     1  AMZN     3/26/2018  1530.00  1556.9900  1499.250  1555.8600  5547618  1555.8600
   2     2  AMZN     3/23/2018  1539.01  1549.0200  1495.360  1495.5600  7843966  1495.5600
   3     3  AMZN     3/22/2018  1565.47  1573.8500  1542.400  1544.1000  6177737  1544.1000
   4     4  AMZN     3/21/2018  1586.45  1590.0000  1563.170  1581.8600  4667291  1581.8600
 ...   ...   ...           ...      ...        ...       ...        ...      ...        ...
1311  1311  AMZN    01-08-2013   267.07   268.9800   263.567   266.3800  3010700   266.3800
1312  1312  AMZN    01-07-2013   262.97   269.7250   262.670   268.4592  4910000   268.4592
1313  1313  AMZN    01-04-2013   257.58   259.8000   256.650   259.1500  1874200   259.1500
1314  1314  AMZN    01-03-2013   257.27   260.8800   256.370   258.4800  2750900   258.4800
1315  1315  AMZN    01-02-2013   256.08   258.0999   253.260   257.3100  3271000   257.3100

1316 rows × 9 columns
```

In [4]:
```
df.head(10)
```

Out[4]:
```
   None  ticker       Date     Open     High      Low    Close   Volume  Adj_Close
0     0  AMZN    3/27/2018  1572.40  1575.96  1482.32  1497.05  6793279    1497.05
1     1  AMZN    3/26/2018  1530.00  1556.99  1499.25  1555.86  5547618    1555.86
2     2  AMZN    3/23/2018  1539.01  1549.02  1495.36  1495.56  7843966    1495.56
3     3  AMZN    3/22/2018  1565.47  1573.85  1542.40  1544.10  6177737    1544.10
4     4  AMZN    3/21/2018  1586.45  1590.00  1563.17  1581.86  4667291    1581.86
5     5  AMZN    3/20/2018  1550.34  1587.00  1545.41  1586.51  4507049    1586.51
6     6  AMZN    3/19/2018  1554.53  1561.66  1525.35  1544.93  6376619    1544.93
7     7  AMZN    3/16/2018  1583.45  1589.44  1567.50  1571.68  5145054    1571.68
8     8  AMZN    3/15/2018  1595.00  1596.91  1578.11  1582.32  4026744    1582.32
9     9  AMZN    3/14/2018  1597.00  1606.44  1590.89  1591.00  4164395    1591.00
```

In [11]:
```
df1=df.drop(['None','ticker'],axis=1)
df1
```

Out[11]:

	Date	Open	High	Low	Close	Volume	Adj_Close
0	3/27/2018	1572.40	1575.9600	1482.320	1497.0500	6793279	1497.0500
1	3/26/2018	1530.00	1556.9900	1499.250	1555.8600	5547618	1555.8600
2	3/23/2018	1539.01	1549.0200	1495.360	1495.5600	7843966	1495.5600
3	3/22/2018	1565.47	1573.8500	1542.400	1544.1000	6177737	1544.1000
4	3/21/2018	1586.45	1590.0000	1563.170	1581.8600	4667291	1581.8600
...	...	...	...	...	...	...	...
1311	01-08-2013	267.07	268.9800	263.567	266.3800	3010700	266.3800
1312	01-07-2013	262.97	269.7250	262.670	268.4592	4910000	268.4592
1313	01-04-2013	257.58	259.8000	256.650	259.1500	1874200	259.1500
1314	01-03-2013	257.27	260.8800	256.370	258.4800	2750900	258.4800
1315	01-02-2013	256.08	258.0999	253.260	257.3100	3271000	257.3100

1316 rows × 7 columns

In [12]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1316 entries, 0 to 1315
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       1316 non-null   object 
 1   Open       1316 non-null   float64
 2   High       1316 non-null   float64
 3   Low        1316 non-null   float64
 4   Close      1316 non-null   float64
 5   Volume     1316 non-null   int64  
 6   Adj_Close  1316 non-null   float64
dtypes: float64(5), int64(1), object(1)
memory usage: 72.1+ KB
```

In [14]:
```
df1['Date'] = df1['Date'].apply(pd.to_datetime)
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1316 entries, 0 to 1315
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   Date       1316 non-null   datetime64[ns]
 1   Open       1316 non-null   float64       
 2   High       1316 non-null   float64       
 3   Low        1316 non-null   float64       
 4   Close      1316 non-null   float64       
 5   Volume     1316 non-null   int64         
 6   Adj_Close  1316 non-null   float64       
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 72.1 KB
```

In [23]:
```
df1.set_index('Date',inplace=True)
```

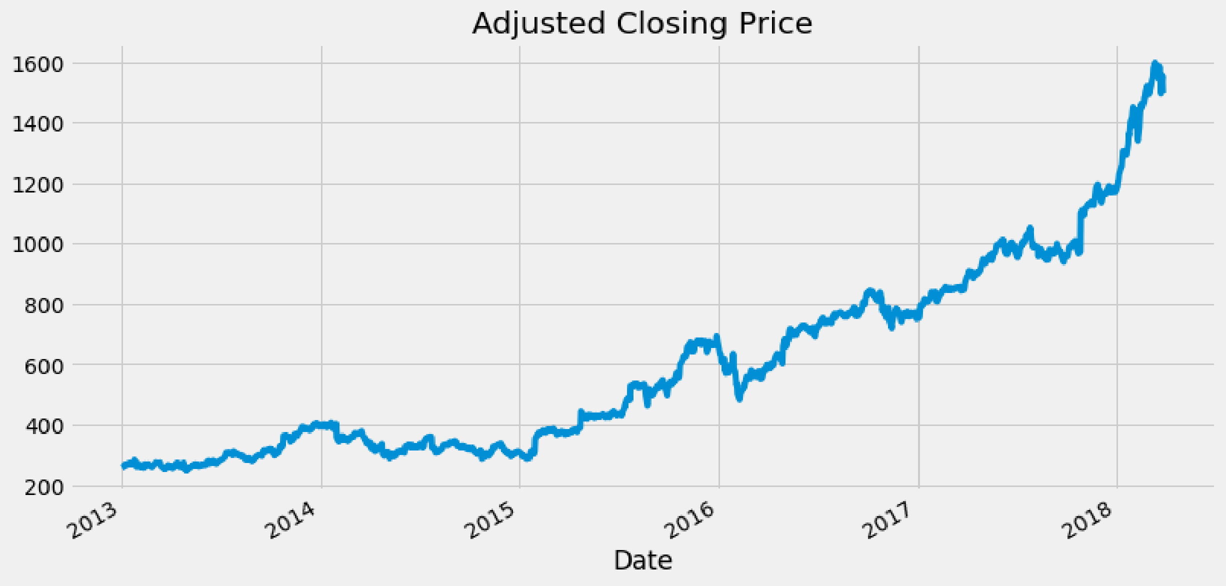
In [25]:
```
df1.head()
```

Out[25]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-03-27	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-03-26	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-03-23	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-03-22	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-03-21	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

In [27]:
```
df1['Adj_Close'].plot(figsize=(12,6),title='Adjusted Closing Price')
```

Out[27]:
```
<AxesSubplot:title={'center':'Adjusted Closing Price'}, xlabel='Date'>
```



In [32]:
```
from datetime import datetime
```

In [38]:
```
my_year = 2020
my_month = 5
my_day = 1
my_hour = 13
my_minute = 36
my_second = 45
test_date =datetime(my_year, my_month, my_day)
test_date
```

Out[38]:
```
datetime.datetime(2020, 5, 1, 0, 0)
```

In [39]:
```
test_date = datetime(my_year, my_month, my_day, my_hour, my_minute, my_second)
print("The day is : ",test_date.day)
print("The hour is : ", test_date.hour)
print("The month is : ", test_date.month)
```

```
The day is :  1
The hour is :  13
The month is :  5
```

In [41]:
```
print(df1.index.max())
```

```
2018-03-27 00:00:00
```

In [42]:
```
print(df1.index.min())
```

```
2013-01-02 00:00:00
```

In [46]:
```
df1.index.argmin()
```

Out[46]:
```
1315
```

In [48]:
```
df1.index.argmax()
```

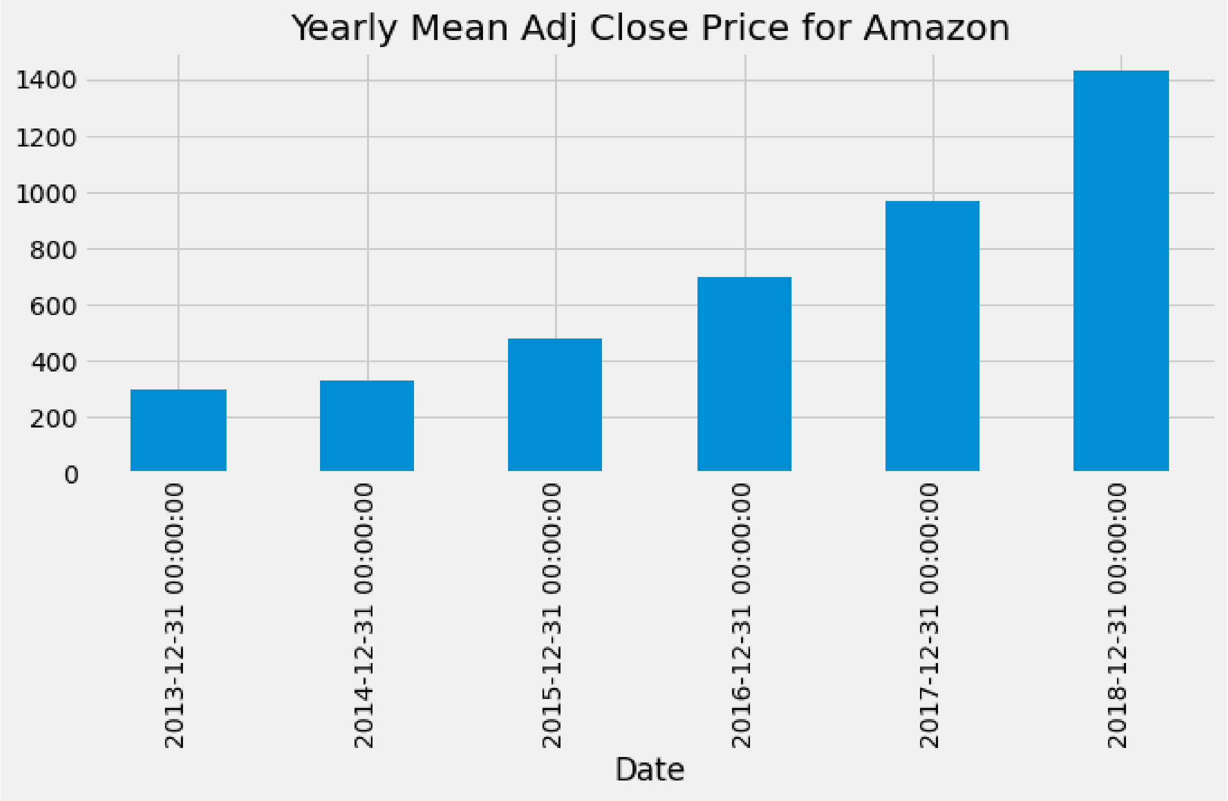
Out[48]:
```
0
```

In [58]:
```
Y = df1.resample('Y').mean()
Y
```

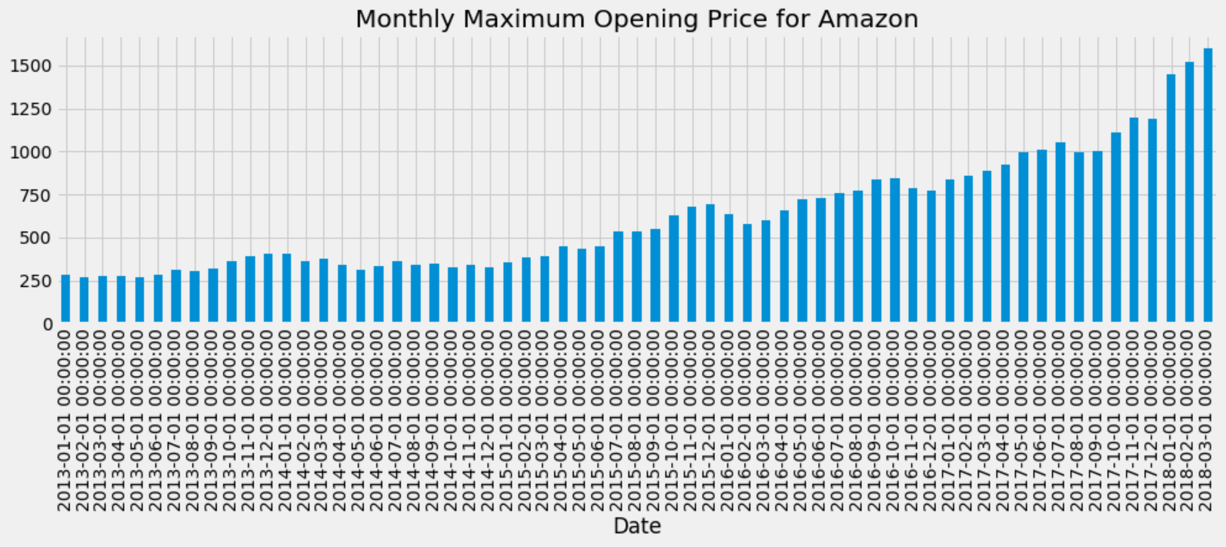
Out[58]:

Date	Open	High	Low	Close	Volume	Adj_Close
2013-12-31	297.877223	300.925966	294.656658	298.032235	2.967880e+06	298.032235
2014-12-31	332.798433	336.317462	328.545440	332.550976	4.083223e+06	332.550976
2015-12-31	478.126230	483.248272	472.875443	478.137321	3.797801e+06	478.137321
2016-12-31	699.669762	705.799103	692.646189	699.523135	4.122043e+06	699.523135
2017-12-31	967.565060	973.789752	959.991826	967.403996	3.466207e+06	967.403996
2018-12-31	1429.770000	1446.701017	1409.469661	1429.991186	5.586829e+06	1429.991186

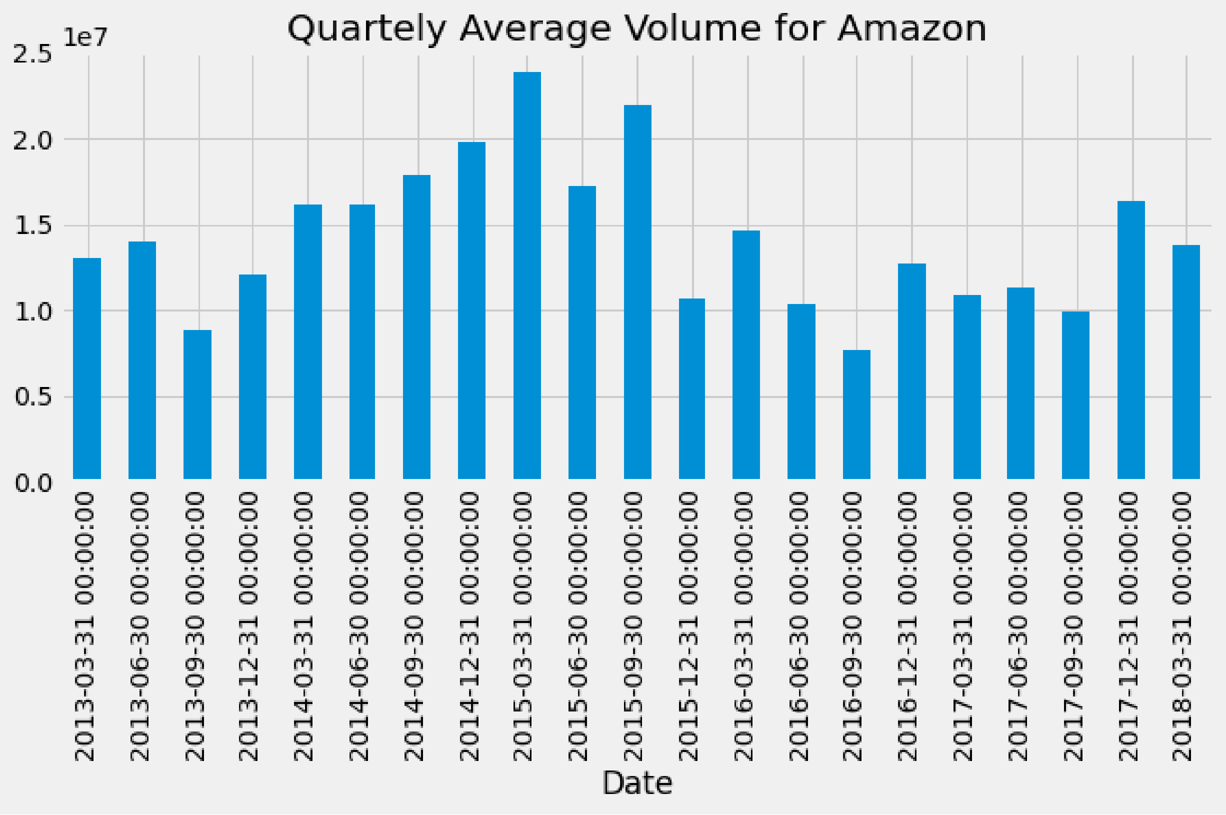
In [60]:
```
df1['Adj_Close'].resample('A').mean().plot(kind='bar', figsize=(10, 4))
plt.title('Yearly Mean Adj Close Price for Amazon')
plt.show()
```



In [67]:
```
df1['Adj_Close'].resample('MS').max().plot(kind='bar', figsize=(15, 4))
plt.title('Monthly Maximum Opening Price for Amazon')
plt.show()
```



In [71]:
```
df1['Volume'].resample('Q').max().plot(kind='bar', figsize=(10, 4))
plt.title('Quartely Average Volume for Amazon')
plt.show()
```



In [72]:
```
df1.head()
```

Out[72]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-03-27	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-03-26	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-03-23	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-03-22	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-03-21	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

In [78]:
```
shiftfor=df1.shift(1)
shiftfor.head()
```

Out[78]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-03-27	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-26	1572.40	1575.96	1482.32	1497.05	6793279.0	1497.05
2018-03-23	1530.00	1556.99	1499.25	1555.86	5547618.0	1555.86
2018-03-22	1539.01	1549.02	1495.36	1495.56	7843966.0	1495.56
2018-03-21	1565.47	1573.85	1542.40	1544.10	6177737.0	1544.10

In [77]:
```
shiftbac=df1.shift(-1)
shiftbac.head()
```

Out[77]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-03-27	1530.00	1556.99	1499.25	1555.86	5547618.0	1555.86
2018-03-26	1539.01	1549.02	1495.36	1495.56	7843966.0	1495.56
2018-03-23	1565.47	1573.85	1542.40	1544.10	6177737.0	1544.10
2018-03-22	1586.45	1590.00	1563.17	1581.86	4667291.0	1581.86
2018-03-21	1550.34	1587.00	1545.41	1586.51	4507049.0	1586.51

In [79]:
```
df1.head(10)
```

Out[79]:

Date	Open	High	Low	Close	Volume	Adj_Close
2018-03-27	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-03-26	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-03-23	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-03-22	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-03-21	1586.45	1590.00	1563.17	1581.86	4667291	1581.86
2018-03-20	1550.34	1587.00	1545.41	1586.51	4507049	1586.51
2018-03-19	1554.53	1561.66	1525.35	1544.93	6376619	1544.93
2018-03-16	1583.45	1589.44	1567.50	1571.68	5145054	1571.68
2018-03-15	1595.00	1596.91	1578.11	1582.32	4026744	1582.32
2018-03-14	1597.00	1606.44	1590.89	1591.00	4164395	1591.00

In [89]:
```
timeshift=df1.tshift(periods=1, freq='D')
timeshift.head()
```

```
<ipython-input-89-0d8fa9439a49>:1: FutureWarning: tshift is deprecated and will
be removed in a future version. Please use shift instead.
  timeshift=df1.tshift(periods=1, freq='D')
```

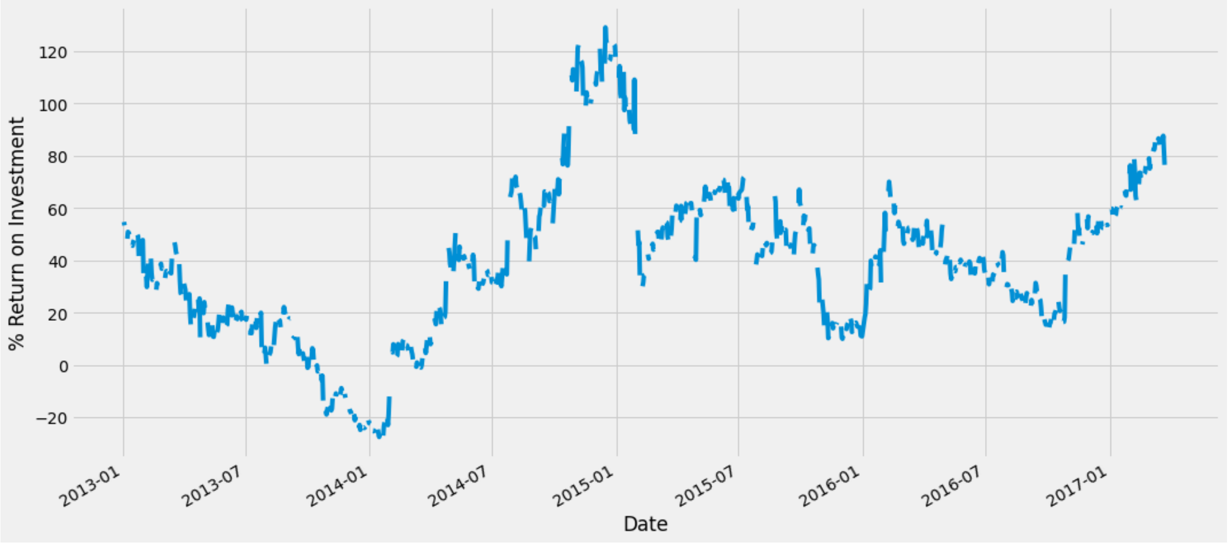
Out[89]:

Date	Open	High	Low	Close	Volume	Adj_Close
2018-03-28	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-03-27	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-03-24	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-03-23	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-03-22	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

```
In [110]: ROI = 100 * (df1['Adj_Close'].tshift(periods=-365, freq = 'D') / df1['Adj_Close']
          ROI.plot(figsize=(16,8))
          plt.ylabel('% Return on Investment')
```

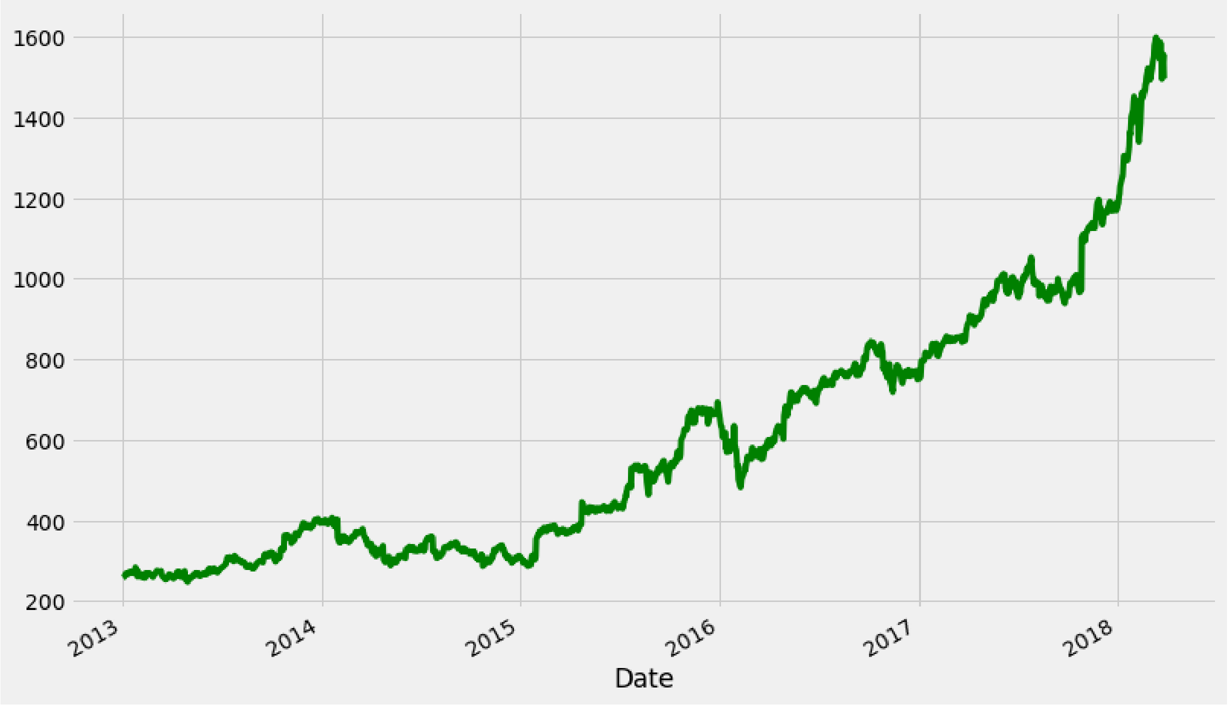
```
<ipython-input-110-340b8dccb74b>:1: FutureWarning: tshift is deprecated and wil
l be removed in a future version. Please use shift instead.
  ROI = 100 * (df1['Adj_Close'].tshift(periods=-365, freq = 'D') / df1['Adj_Clo
se'] - 1)
```

```
Out[110]: Text(0, 0.5, '% Return on Investment')
```



```
In [114]: df1['Adj_Close'].plot(figsize = (12,8), color='green')
```

```
Out[114]: <AxesSubplot:xlabel='Date'>
```



In [117]:

```
df1.rolling(7).mean().head(10)
```

Out[117]:

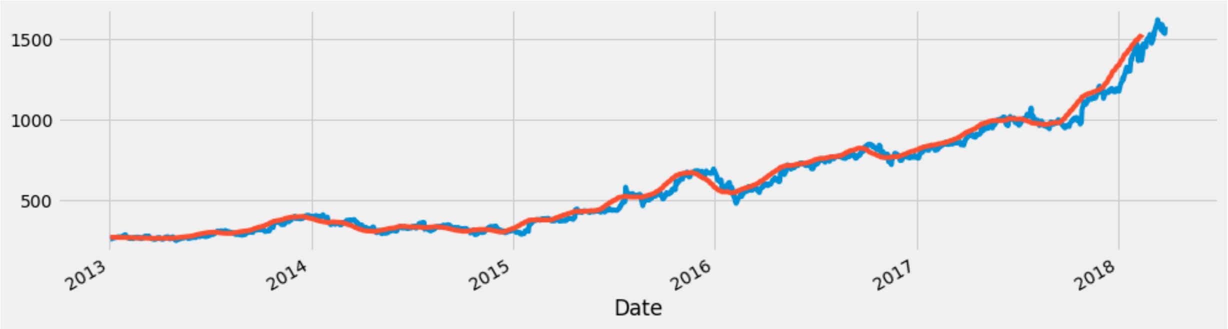
Date	Open	High	Low	Close	Volume	Adj_Close
2018-03-27	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-26	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-23	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-22	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-21	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-20	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-19	1556.885714	1570.640000	1521.894286	1543.695714	5.987651e+06	1543.695714
2018-03-16	1558.464286	1572.565714	1534.062857	1554.357143	5.752191e+06	1554.357143
2018-03-15	1567.750000	1578.268571	1545.328571	1558.137143	5.534923e+06	1558.137143
2018-03-14	1576.034286	1586.471429	1558.975714	1571.771429	5.009270e+06	1571.771429

In [124]:

```
df1["Open"].plot(kind="line")
df1["Open"].rolling(30).mean().plot(kind="line",figsize=(15,4))
```

Out[124]:

```
<AxesSubplot:xlabel='Date'>
```



In [ ]: