In [29]:
```
import pandas as pd
```

In [30]:
```
import pandas_datareader as dr
```

In [31]:
```
import matplotlib.pyplot as plt
```

In [32]:
```
import numpy as np
```

In [33]:
```
import cvxopt as opt
from cvxopt import blas, solvers
```

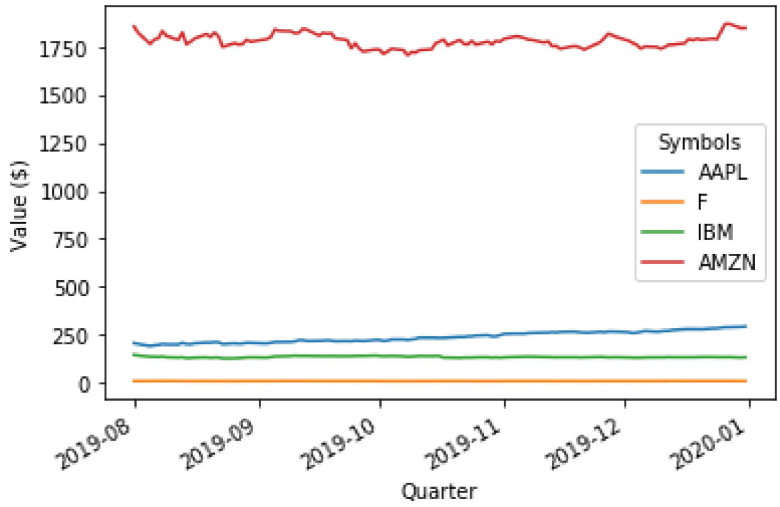
In [34]:
```
import datetime
start = datetime.datetime(2019,8,1)
end = datetime.datetime(2020,1,1)
```

In [35]:
```
stock_data = dr.data.get_data_yahoo(['AAPL', 'F', 'IBM', 'AMZN'], start, end)
#(stock name, start, end)
selected = stock_data["Adj Close"]
print(selected)
```

```
Symbols            AAPL         F         IBM         AMZN
Date
2019-08-01  206.529373  9.006506  145.317749  1855.319946
2019-08-02  202.159607  8.977483  142.349915  1823.239990
2019-08-05  191.576981  8.929112  136.075882  1765.130005
2019-08-06  195.203613  9.170962  136.046875  1787.829956
2019-08-07  197.225006  9.219334  134.480789  1793.400024
...                ...       ...         ...          ...
2019-12-24  283.596924  9.311640  133.585083  1789.209961
2019-12-26  289.223602  9.291973  133.515808  1868.770020
2019-12-27  289.113831  9.203478  133.872086  1869.800049
2019-12-30  290.829773  9.095318  131.437500  1846.890015
2019-12-31  292.954712  9.144482  132.654785  1847.839966

[106 rows x 4 columns]
```

In [36]:
```
selected.plot()
plt.xlabel('Quarter')
plt.ylabel('Value ($)')
plt.show()
```



In [ ]:
```
```

In [37]:
```
returns_quarterly = df.pct_change()
expected_returns = returns_quarterly.mean()
cov_quarterly = returns_quarterly.cov()
```

In [38]:
```
cov_quarterly = returns_quarterly.cov()
print(cov_quarterly)
```

```
Symbols      AAPL         F        IBM      AMZN
Symbols
AAPL     0.000218  0.000088  0.000108  0.000110
F        0.000088  0.000209  0.000081  0.000053
IBM      0.000108  0.000081  0.000171  0.000081
AMZN     0.000110  0.000053  0.000081  0.000142
```

In [48]:
```
def return_portfolios(expected_returns, cov_matrix):
    port_returns = []
    port_volatility = []
    stock_weights = []

    selected = (expected_returns.axes)[0]

    num_assets = len(selected)
    num_portfolios = 500

    for single_portfolio in range(num_portfolios):
        weights = np.random.random(num_assets)
        weights /= np.sum(weights)
        returns = np.dot(weights, expected_returns)
        volatility = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
        port_returns.append(returns)
        port_volatility.append(volatility)
        stock_weights.append(weights)

    portfolio = {'Returns': port_returns,
                 'Volatility': port_volatility}

    for counter,symbol in enumerate(selected):
        portfolio[symbol +' Weight'] = [Weight[counter] for Weight in stock_we
ights]

    df = pd.DataFrame(portfolio)

    column_order = ['Returns', 'Volatility'] + [stock+' Weight' for stock in s
elected]

    df = df[column_order]

    return df
```

In [40]:
```
```

In [46]:
```
random_portfolios = return_portfolios(expected_returns, cov_quarterly)
print(random_portfolios)
```

	Returns	Volatility	AAPL Weight	F Weight	IBM Weight	AMZN Weight
0	0.000623	0.010642	0.244149	0.180640	0.345710	0.229501
1	0.001520	0.011567	0.455518	0.343044	0.172318	0.029119
2	0.000537	0.010546	0.204065	0.278262	0.308720	0.208953
3	0.001551	0.011215	0.436143	0.274790	0.034438	0.254628
4	0.001067	0.010984	0.362201	0.113107	0.276071	0.248620
5	0.000762	0.010455	0.234871	0.236767	0.150655	0.377707
6	0.002071	0.012214	0.599246	0.043524	0.017361	0.339870
7	0.000732	0.010558	0.255576	0.207184	0.265881	0.271359
8	0.000169	0.010251	0.084611	0.180654	0.234998	0.499738
9	0.001622	0.011413	0.466541	0.129163	0.036671	0.367624
10	0.001232	0.011217	0.326415	0.455441	0.015827	0.202317
11	0.001344	0.011402	0.446541	0.063755	0.275916	0.213789
12	0.000546	0.010371	0.149437	0.285789	0.071069	0.493705
13	0.000976	0.011086	0.344943	0.224044	0.345324	0.085689
14	0.000227	0.010322	0.056361	0.382446	0.098967	0.462226
15	0.000850	0.010729	0.291471	0.238388	0.279544	0.190597
16	0.001216	0.011077	0.391635	0.222525	0.245979	0.139861
17	-0.000272	0.010987	0.015031	0.346886	0.528407	0.109676
18	0.001294	0.011202	0.397342	0.067528	0.132379	0.402752
19	0.001323	0.011273	0.404432	0.055466	0.123213	0.416889

In [53]:
```
random_portfolios.plot.scatter(x='Volatility', y='Returns', figsize = (10,5))
plt.xlabel('Volatility (Std. Deviation)')
plt.ylabel('Expected Returns')
plt.title('Efficient Frontier')
plt.show()
```