In [2]:

```
# Provides ways to work with large multidimensional arrays
import numpy as np
# Allows for further data manipulation and analysis
import pandas as pd
from pandas_datareader import data as web # Reads stock data
import matplotlib.pyplot as plt # Plotting
import matplotlib.dates as mdates # Styling dates
%matplotlib inline

import datetime as dt # For defining dates
import mplfinance as mpf # Matplotlib finance
```

Scrapping S&P500 Data Symbols

In [3]:

```
# Scrapping data for S&P500 Shares

payload=pd.read_html('https://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
snp_500 = payload[0]

snp_symbol = snp_500["Symbol"].values.tolist()

snp_symbol[1:15]
```

Out[3]:

```
['ABT',
 'ABBV',
 'ABMD',
 'ACN',
 'ATVI',
 'ADBE',
 'AMD',
 'AAP',
 'AES',
 'AFL',
 'A',
 'APD',
 'AKAM',
 'ALK']
```

In [4]:

```
# Reading Downloaded File For 500 Stocks Return

df = pd.read_csv('D:\Python Class\Python4finance\All_Stocks.csv')
```

In [5]:

```
df
```

Out[5]:

	Date	MMM	ABT	ABBV	ABMD	ACN	ATVI	ADBE	AMD	
0	2018-01-02	100.004228	99.779363	98.459257	98.299458	99.540601	98.468830	98.155106	95.064930	99.10
1	2018-01-03	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.00
2	2018-01-04	101.307133	99.830279	99.429714	101.751605	101.184088	99.004752	101.204158	104.935062	103.68
3	2018-01-05	102.096505	100.118805	101.160589	103.319375	102.018735	101.623036	102.375167	102.857142	104.79

	Date	MMM	ABT	ABBV	ABMD	ACN	ATVI	ADBE	AMD	
4	2018-01-08	101.765483	99.830279	99.539770	106.117861	102.834025	102.021126	102.209457	106.320342	104.05
...	...	...	...	...	...	...	...	...	...	
982	2021-11-24	85.604488	226.660919	144.130726	172.495142	248.625681	95.578790	369.156005	1366.233770	222.89
983	2021-11-26	84.587624	228.128855	141.519221	166.469206	242.608504	95.123726	365.720283	1340.346277	216.55
984	2021-11-29	84.929787	232.025243	141.980785	162.965981	250.992769	94.637284	379.744816	1401.818190	217.07
985	2021-11-30	81.946665	227.929504	140.025193	160.749661	245.215713	91.953981	370.001105	1371.168766	210.81
986	2021-12-01	82.279195	229.687410	140.790431	156.117853	247.095669	89.882663	363.129694	1290.995655	208.43

987 rows × 504 columns

Calculating Stocks Return

In [6]:

```
stocks_return = []

for ticker in snp_symbol:
    try:
        stocks_return.append((ticker, df[ticker][986] / df[ticker][0] * 100))
    except:pass
#     else:
#         print(stocks_return)
```

In [7]:

```
# Converting into dataframe

df2 = pd.DataFrame(stocks_return)
```

In [8]:

```
sorted_return = df2.sort_values(by = 1, ascending = False).head(25)
sorted_return
```

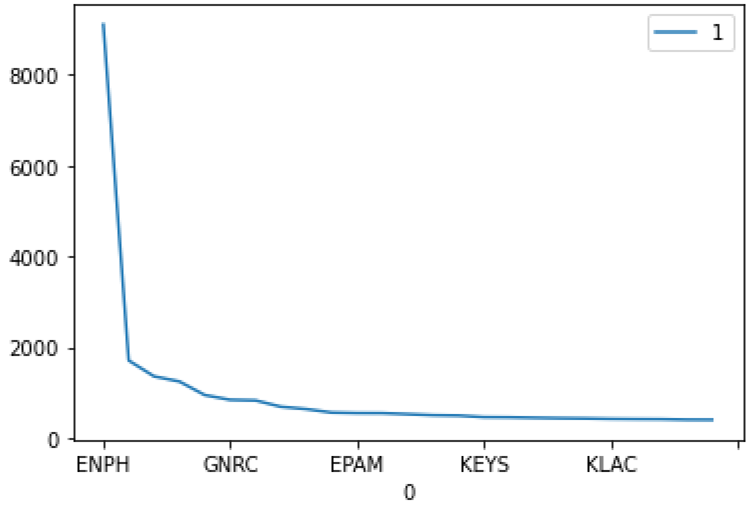
Out[8]:

	0	1
169	ENPH	9110.000358
440	TSLA	1708.108371
7	AMD	1358.014634
178	ETSY	1244.743158
143	DXCM	945.593998
210	GNRC	840.245716
419	SEDG	827.989472
200	FTNT	688.297678
346	NVDA	636.583385

	0	1
499	ZBRA	560.244947
172	EPAM	548.902215
103	CMG	546.878269
361	PAYC	524.726484
327	MSCI	501.945925
321	MPWR	493.756601
270	KEYS	459.478034
413	NOW	455.150707
487	WST	444.628778
374	POOL	436.068415
254	INTU	430.329112
274	KLAC	421.051362
80	CDNS	416.290146
303	MTCH	413.654204
315	MSFT	402.512151
44	AAPL	400.042085

```
In [9]: sorted_return.plot(x = 0, y =1)
```

```
Out[9]: <AxesSubplot:xlabel='0'>
```


ENPH
GNRC
EPAM
KEYS
KLAC