A company's stock share is a piece of the company more precisely:

An investor can buy a stock and sell it later. If the stock price increases, the investor profits, If it decreases,the investor with incur a loss. Determining the stock price is complex; it depends on the number of outstanding shares, the size of the company's future profits, and much more. People trade stocks throughout the day the stock ticker is a report of the price of a certain stock, updated continuously throughout the trading session by the various stock market exchanges.

```
!pip install yfinance

#!pip install pandas
```

```
Requirement already satisfied: yfinance in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.2.9)
Requirement already satisfied: cryptography>=3.3.2 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (38.0.2)
Requirement already satisfied: pytz>=2022.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (2022.6)
Requirement already satisfied: appdirs>=1.4.4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (1.4.4)
Requirement already satisfied: html5lib>=1.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (1.1)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (4.11.2)
Requirement already satisfied: frozendict>=2.3.4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (2.3.4)
Requirement already satisfied: multitasking>=0.0.7 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (4.9.2)
Requirement already satisfied: numpy>=1.16.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (1.21.6)
Requirement already satisfied: pandas>=1.3.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (1.3.5)
Requirement already satisfied: requests>=2.26 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance) (2.28.1)
Requirement already satisfied: soupsieve>1.2 in
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
beautifulsoup4>=4.11.1->yfinance) (2.3.2.post1)
Requirement already satisfied: cffi>=1.12 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
cryptography>=3.3.2->yfinance) (1.15.1)
Requirement already satisfied: webencodings in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: six>=1.9 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<3,>=2 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.26->yfinance) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.26->yfinance) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.26->yfinance) (1.26.13)
Requirement already satisfied: idna<4,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.26->yfinance) (3.4)
Requirement already satisfied: pycparser in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
cffi>=1.12->cryptography>=3.3.2->yfinance) (2.21)
```

```
import yfinance as yf
import pandas as pd
```

Using the yfinance Library to Extract Stock Data

Using the `Ticker` module we can create an object that will allow us to access functions to extract data. To do this we need to provide the ticker symbol for the stock, here the company is Apple and the ticker symbol is `AAPL`.

```
apple = yf.Ticker("AAPL")
```

Now we can access functions and variables to extract the type of data we need. You can view them and what they represent here https://aroussi.com/post/python-yahoo-finance.

Stock Info

Using the attribute info we can extract information about the stock as a Python dictionary.

```
apple_info=apple.info
apple_info
```


```
{'zip': '95014', 'sector': 'Technology', 'fullTimeEmployees': 164000,
'longBusinessSummary': 'Apple Inc. designs, manufactures, and markets
smartphones, personal computers, tablets, wearables, and accessories
worldwide. It also sells various related services. In addition, the
company offers iPhone, a line of smartphones; Mac, a line of personal
computers; iPad, a line of multi-purpose tablets; and wearables, home,
and accessories comprising AirPods, Apple TV, Apple Watch, Beats
products, and HomePod. Further, it provides AppleCare support and
cloud services store services; and operates various platforms,
including the App Store that allow customers to discover and download
applications and digital content, such as books, music, video, games,
and podcasts. Additionally, the company offers various services, such
as Apple Arcade, a game subscription service; Apple Fitness+, a
personalized fitness service; Apple Music, which offers users a
curated listening experience with on-demand radio stations; Apple
News+, a subscription news and magazine service; Apple TV+, which
offers exclusive original content; Apple Card, a co-branded credit
card; and Apple Pay, a cashless payment service, as well as licenses
its intellectual property. The company serves consumers, and small and
mid-sized businesses; and the education, enterprise, and government
markets. It distributes third-party applications for its products
through the App Store. The company also sells its products through its
retail and online stores, and direct sales force; and third-party
cellular network carriers, wholesalers, retailers, and resellers.
Apple Inc. was incorporated in 1977 and is headquartered in Cupertino,
California.', 'city': 'Cupertino', 'phone': '408 996 1010', 'state':
'CA', 'country': 'United States', 'companyOfficers': [], 'website':
'https://www.apple.com', 'maxAge': 1, 'address1': 'One Apple Park
Way', 'industry': 'Consumer Electronics', 'ebitdaMargins': 0.32329,
'profitMargins': 0.24558, 'grossMargins': 0.43058997,
'operatingCashflow': 109189996544, 'revenueGrowth': -0.055,
'operatingMargins': 0.29408002, 'ebitda': 125287997440,
'targetLowPrice': 118, 'recommendationKey': 'buy', 'grossProfits':
170782000000, 'freeCashflow': 84729126912, 'targetMedianPrice': 171.5,
'earningsGrowth': -0.105, 'currentRatio': 0.938, 'returnOnAssets':
0.19569999, 'numberOfAnalystOpinions': 38, 'targetMeanPrice': 168.9,
'debtToEquity': 195.868, 'returnOnEquity': 1.47943, 'targetHighPrice':
210, 'totalCash': 51355000832, 'totalDebt': 111109996544,
'totalRevenue': 387537010688, 'totalCashPerShare': 3.246,
'financialCurrency': 'USD', 'revenuePerShare': 24.084, 'quickRatio':
0.769, 'recommendationMean': 2, 'shortName': 'Apple Inc.', 'longName':
'Apple Inc.', 'isEsgPopulated': False, 'gmtOffSetMilliseconds': '-
18000000', 'quoteType': 'EQUITY', 'messageBoardId': 'finmb_24937',
'market': 'us_market', 'annualHoldingsTurnover': None,
'enterpriseToRevenue': 6.462, 'beta3Year': None, 'enterpriseToEbitda':
19.988, '52WeekChange': -0.13212836, 'morningStarRiskRating': None,
'forwardEps': 6.61, 'revenueQuarterlyGrowth': None,
```

```
'sharesOutstanding': 15821899776, 'fundInceptionDate': None,
'annualReportExpenseRatio': None, 'totalAssets': None, 'bookValue':
3.581, 'sharesShort': 121868919, 'sharesPercentSharesOut': 0.0077,
'fundFamily': None, 'lastFiscalYearEnd': 1663977600,
'heldPercentInstitutions': 0.61239, 'netIncomeToCommon': 95171002368,
'trailingEps': 5.89, 'lastDividendValue': 0.23, 'SandP52WeekChange': -
0.09077883, 'priceToBook': 42.370842, 'heldPercentInsiders':
0.00072999997, 'nextFiscalYearEnd': 1695513600, 'yield': None,
'mostRecentQuarter': 1672444800, 'shortRatio': 1.47,
'sharesShortPreviousMonthDate': 1671062400, 'floatShares':
15805174737, 'beta': 1.277894, 'enterpriseValue': 2504245641216,
'priceHint': 2, 'threeYearAverageReturn': None, 'lastSplitDate':
1598832000, 'lastSplitFactor': '4:1', 'legalType': None,
'lastDividendDate': 1667520000, 'morningStarOverallRating': None,
'earningsQuarterlyGrowth': -0.134, 'priceToSalesTrailing12Months':
6.1946516, 'dateShortInterest': 1673568000, 'pegRatio': 3.15,
'ytdReturn': None, 'forwardPE': 22.954613, 'lastCapGain': None,
'shortPercentOfFloat': 0.0077, 'sharesShortPriorMonth': 121757434,
'impliedSharesOutstanding': 0, 'category': None,
'fiveYearAverageReturn': None, 'trailingAnnualDividendYield':
0.005889968, 'payoutRatio': 0.1545, 'volume24Hr': None, 'navPrice':
None, 'trailingAnnualDividendRate': 0.91, 'toCurrency': None,
'expireDate': None, 'algorithm': None, 'dividendRate': 0.92,
'exDividendDate': 1675987200, 'circulatingSupply': None, 'startDate':
None, 'trailingPE': 25.76061, 'lastMarket': None, 'maxSupply': None,
'openInterest': None, 'volumeAllCurrencies': None, 'strikePrice':
None, 'ask': 151.98, 'askSize': 900, 'fromCurrency': None,
'fiveYearAvgDividendYield': 0.96, 'bid': 151.91, 'tradeable': False,
'dividendYield': 0.006, 'bidSize': 1000, 'coinMarketCapLink': None,
'preMarketPrice': 152.5, 'logo_url':
'https://logo.clearbit.com/apple.com', 'trailingPegRatio': None}
```

We can get the 'country' using the key country

```
apple_info['country']
```

```
'United States'
```

Extracting Share Price

A share is the single smallest part of a company's stock that you can buy, the prices of these shares fluctuate over time. Using the history() method we can get the share price of the stock over a certain period of time. Using the `period` parameter we can set how far back from the present to get data. The options for `period` are 1 day (1d), 5d, 1 month (1mo) , 3mo, 6mo, 1 year (1y), 2y, 5y, 10y, ytd, and max.

```
apple_share_price_data = apple.history(period="max")
```

The format that the data is returned in is a Pandas DataFrame. With the `Date` as the index the share `Open`, `High`, `Low`, `Close`, `Volume`, and `Stock Splits` are given for each day.

```
apple_share_price_data.head()
```

```
                                 Open      High       Low     Close
Volume  \
Date

1980-12-12 00:00:00-05:00  0.099874  0.100308  0.099874  0.099874
469033600
1980-12-15 00:00:00-05:00  0.095098  0.095098  0.094663  0.094663
175884800
1980-12-16 00:00:00-05:00  0.088149  0.088149  0.087715  0.087715
105728000
1980-12-17 00:00:00-05:00  0.089886  0.090320  0.089886  0.089886
86441600
1980-12-18 00:00:00-05:00  0.092492  0.092927  0.092492  0.092492
73449600

                           Dividends  Stock Splits
Date
1980-12-12 00:00:00-05:00        0.0           0.0
1980-12-15 00:00:00-05:00        0.0           0.0
1980-12-16 00:00:00-05:00        0.0           0.0
1980-12-17 00:00:00-05:00        0.0           0.0
1980-12-18 00:00:00-05:00        0.0           0.0
```

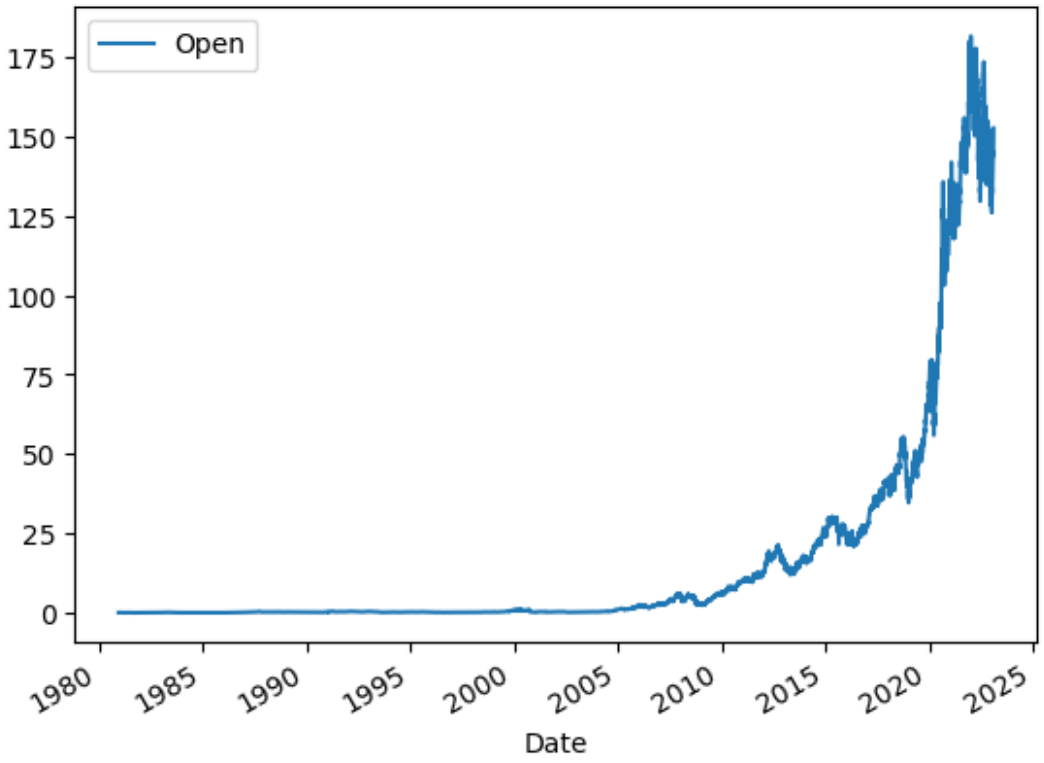
We can reset the index of the DataFrame with the `reset_index` function. We also set the `inplace` paramter to `True` so the change takes place to the DataFrame itself.

```
apple_share_price_data.reset_index(inplace=True)
```

We can plot the `Open` price against the `Date`:

```
apple_share_price_data.plot(x="Date", y="Open")
```

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



Extracting Dividends

Dividends are the distribution of a companys profits to shareholders. In this case they are defined as an amount of money returned per share an investor owns. Using the variable `dividends` we can get a dataframe of the data. The period of the data is given by the period defined in the 'history` function.

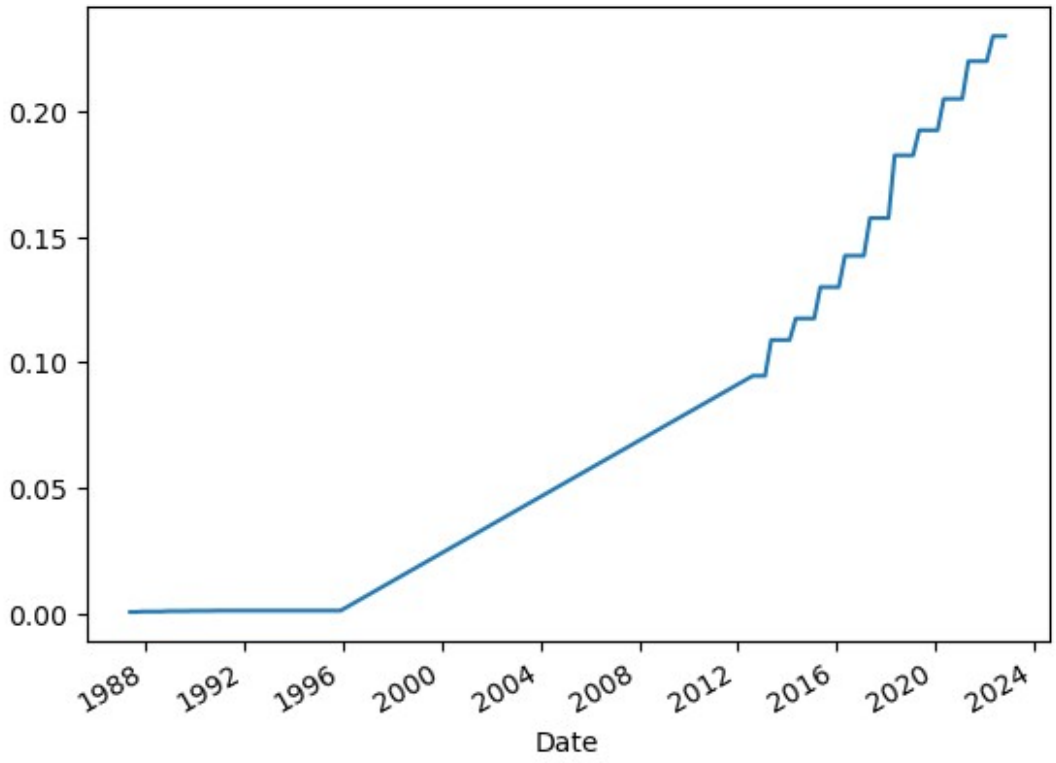
```
apple.dividends
```

```
Date
1987-05-11 00:00:00-04:00    0.000536
1987-08-10 00:00:00-04:00    0.000536
1987-11-17 00:00:00-05:00    0.000714
1988-02-12 00:00:00-05:00    0.000714
1988-05-16 00:00:00-04:00    0.000714
                               ...
2021-11-05 00:00:00-04:00    0.220000
2022-02-04 00:00:00-05:00    0.220000
2022-05-06 00:00:00-04:00    0.230000
2022-08-05 00:00:00-04:00    0.230000
2022-11-04 00:00:00-04:00    0.230000
Name: Dividends, Length: 77, dtype: float64
```

We can plot the dividends overtime:

```
apple.dividends.plot()
```

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



Exercise

Now using the `Ticker` module create an object for AMD (Advanced Micro Devices) with the ticker symbol is `AMD` called; name the object amd.

```
amd = yf.Ticker('AMD')
```

Question 1 Use the key 'country' to find the country the stock belongs to, remember it as it will be a quiz question.

```
amd.info['country']
```

```
'United States'
```

Question 2 Use the key 'sector' to find the sector the stock belongs to, remember it as it will be a quiz question.

```
amd.info['sector']
```

```
'Technology'
```

Question 3 Obtain stock data for AMD using the `history` function, set the `period` to max. Find the `Volume` traded on the first day (first row).

```
amd_share_price_data = amd.history(period="max")
amd_share_price_data.head(1)
```

```
                             Open      High    Low     Close  Volume
Dividends  \
Date

1980-03-17 00:00:00-05:00   0.0  3.302083  3.125  3.145833  219600
0.0

                             Stock Splits
Date
1980-03-17 00:00:00-05:00             0.0
```

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Azim Hirjani

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab