

---

# **finvizfinance**

***Release 0.9.8***

**May 08, 2021**



---

## Contents

---

<b>1</b>	<b>Installation (python &gt;= v3.5)</b>	<b>3</b>
<b>2</b>	<b>Quote Examples</b>	<b>5</b>
<b>3</b>	<b>Screener Example</b>	<b>7</b>
<b>4</b>	<b>News Example</b>	<b>9</b>
<b>5</b>	<b>Insider Example</b>	<b>11</b>
<b>6</b>	<b>Contents</b>	<b>13</b>
6.1	Quote . . . . .	13
6.2	Screener . . . . .	15
6.3	Group . . . . .	17
6.4	News . . . . .	18
6.5	Insider . . . . .	18
6.6	Forex . . . . .	19
6.7	Crypto . . . . .	19
6.8	Future . . . . .	20
6.9	Earnings . . . . .	20
<b>7</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



It is a Finviz Finance information downloader.



# CHAPTER 1

---

## Installation (python >= v3.5)

---

```
> virtualenv -p python3 virtualenvironment  
> source virtualenvironment/bin/activate  
> pip install finvizfinance
```





## CHAPTER 2

---

### Quote Examples

---

```
import pandas as pd
from finvizfinance.quote import finvizfinance

stock = finvizfinance('tsla')
```

Example downloading chart:

```
stock.TickerCharts(out_dir='asset')
```

Example getting individual ticker information

```
stock_fundament = stock.TickerFundament()
stock_description = stock.TickerDescription()
outer_ratings_df = stock.TickerOuterRatings()
news_df = stock.TickerNews()
inside_trader_df = stock.TickerInsideTrader()
```



## CHAPTER 3

---

### Screeener Example

---

```
from finvizfinance.screener.overview import Overview
foverview = Overview()
filters_dict = {'Exchange': 'AMEX', 'Sector': 'Basic Materials'}
foverview.set_filter(filters_dict=filters_dict)
df = foverview.ScreenerView()
df.head()
```



## CHAPTER 4

---

### News Example

---

```
from finvizfinance.news import News
fnews = News()
all_news = fnews.getNews()
# all_news['news'].head()
# all_news['blogs'].head()
```



## CHAPTER 5

---

### Insider Example

---

```
from finvizfinance.insider import Insider
finsider = Insider(option='top owner trade')
finsider.getInsider().head()
```





## 6.1 Quote

Getting individual ticker information.

**class** finvizfinance.quote.**Quote**

quote Getting current price of the ticker

**getCurrent** (*ticker*)

Getting current price of the ticker.

**Returns** price of the ticker

**Return type** price(float)

**class** finvizfinance.quote.**Statements**

Getting statements of ticker

**getStatements** (*ticker*, *statement*='I', *timeframe*='A')

Getting statements of ticker.

**Parameters**

- **ticker** (*str*) – ticker string
- **statement** (*str*) – I(Income Statement), B(Balace Sheet), C(Cash Flow)
- **timeframe** (*str*) – A(Annual), Q(Quarter)

**Returns** statements table

**Return type** df(pandas.DataFrame)

**class** finvizfinance.quote.**finvizfinance** (*ticker*, *verbose*=0)

Getting information from the individual ticker.

**Parameters**

- **ticker** (*str*) – ticker string

- **verbose** (*int*) – choice of visual the progress. 1 for visualize progress.

**TickerCharts** (*timeframe='daily', charttype='advanced', out\_dir="", urlonly=False*)  
Download ticker charts.

**Parameters**

- **timeframe** (*str*) – choice of timeframe (daily, weekly, monthly).
- **charttype** (*str*) – choice of type of chart (candle, line, advanced).
- **out\_dir** (*str*) – output image directory. default none.
- **urlonly** (*bool*) – choice of downloading charts, default: downloading chart

**Returns** url for the chart

**Return type** charturl(str)

**TickerDescription** ()  
Get ticker description.

**Returns** ticker description.

**Return type** description(str)

**TickerFullInfo** ()  
Get all the ticker information.

**Returns** insider information table

**Return type** df(pandas.DataFrame)

**TickerFundament** (*raw=True*)  
Get ticker fundament.

**Parameters** **raw** (*boolean*) – if True, the data is raw.

**Returns** ticker fundament.

**Return type** fundament(dict)

**TickerInsideTrader** ()  
Get insider information table.

**Returns** insider information table

**Return type** df(pandas.DataFrame)

**TickerNews** ()  
Get news information table.

**Returns** news information table

**Return type** df(pandas.DataFrame)

**TickerOuterRatings** ()  
Get outer ratings table.

**Returns** outer ratings table

**Return type** df(pandas.DataFrame)

**TickerSignal** ()  
Get all the trading signals from finviz.

**Returns** get all the ticker signals as list.

**Return type** ticker\_signals(list)

## 6.2 Screener

Getting screener information.

### 6.2.1 Overview

**class** finvizfinance.screener.overview.**Overview**

Getting information from the finviz screener overview page.

**ScreenerView** (*order='ticker', limit=-1, verbose=1, ascend=True*)

Get screener table.

**Parameters**

- **order** (*str*) – sort the table by the choice of order.
- **limit** (*int*) – set the top k rows of the screener.
- **verbose** (*int*) – choice of visual the progress. 1 for visualize progress.
- **ascend** (*bool*) – if True, the order is ascending.

**Returns** screener information table

**Return type** df(pandas.DataFrame)

**compare** (*ticker, compare\_list, order='ticker', verbose=1*)

Get screener table of similar property (Sector, Industry, Country)

**Parameters**

- **ticker** (*str*) – the ticker to compare
- **compare\_list** (*list*) – choice of compare property (Sector, Industry, Country) or combination.
- **order** (*str*) – sort the table by the choice of order
- **verbose** (*int*) – choice of visual the progress. 1 for visualize progress

**Returns** screener information table

**Return type** df(pandas.DataFrame)

**getFilterOptions** (*screen\_filter*)

Get filters options.

**Parameters** **screen\_filter** (*str*) – screen filter for checking options

**Returns** all the available filters

**Return type** filter\_options(list)

**getFilters** ()

Get filters.

**Returns** all the available filters

**Return type** filters(list)

**getOrders** ()

Get orders.

**Returns** all the available orders

**Return type** orders(list)

**getSignal** ()

Get signals.

**Returns** all the available trading signals

**Return type** signals(list)

**set\_filter** (signal=", filters\_dict={}, ticker=")

Update the settings.

**Parameters**

- **signal** (*str*) – ticker signal
- **filters\_dict** (*dict*) – dictionary of filters
- **ticker** (*str*) – ticker string

## 6.2.2 Valuation

**class** finvizfinance.screener.valuation.Valuation

Valuation inherit from overview module. Getting information from the finviz screener valuation page.

## 6.2.3 Financial

**class** finvizfinance.screener.financial.Financial

Financial inherit from overview module. Getting information from the finviz screener financial page.

## 6.2.4 Ownership

**class** finvizfinance.screener.ownership.Ownership

Ownership inherit from overview module. Getting information from the finviz screener ownership page.

## 6.2.5 Technical

**class** finvizfinance.screener.technical.Technical

Technical inherit from overview module. Getting information from the finviz screener technical page.

## 6.2.6 Custom

**class** finvizfinance.screener.custom.Custom

Custom inherit from overview module. Getting information from the finviz screener custom page.

**ScreenerView** (order='ticker', limit=-1, verbose=1, ascend=True, columns=[0, 1, 2, 3, 4, 5, 6, 7, 65, 66, 67])

Get screener table.

**Parameters**

- **order** (*str*) – sort the table by the choice of order.
- **limit** (*int*) – set the top k rows of the screener.
- **verbose** (*int*) – choice of visual the progress. 1 for visualize progress.

- **ascend** (*bool*) – if True, the order is ascending.
- **columns** (*list*) – columns of your choice. Default index: 0,1,2,3,4,5,6,7,65,66,67.

**Returns** screener information table

**Return type** df(pandas.DataFrame)

**getColumns** ()

Get information about the columns

**Returns** return the index and column name.

**Return type** columns(dict)

## 6.2.7 Tickers

**class** finvizfinance.screener.ticker.**Ticker**

Financial inherit from overview module. Getting information from the finviz screener ticker page.

**ScreenerView** (*limit=-1, verbose=1*)

Get screener table.

**Parameters** **verbose** (*int*) – choice of visual the progress. 1 for visualize progress.

**Returns** get all the tickers as list.

**Return type** tickers(list)

## 6.3 Group

Getting group information.

### 6.3.1 Overview

**class** finvizfinance.group.overview.**Overview**

Getting information from the finviz group overview page.

**ScreenerView** (*group='Sector', order='Name'*)

Get screener table.

**Parameters**

- **group** (*str*) – choice of group option.
- **order** (*str*) – sort the table by the choice of order.

**Returns** group information table.

**Return type** df(pandas.DataFrame)

**getGroup** ()

Get groups.

**Returns** all the available groups.

**Return type** groups(list)

**getOrders** ()

Get orders.

**Returns** all the available orders.

**Return type** orders(list)

### 6.3.2 Valuation

**class** finvizfinance.group.valuation.Valuation

Valuation inherit from overview module. Getting information from the finviz group valuation page.

### 6.3.3 Performance

**class** finvizfinance.group.performance.Performance

Performance inherit from overview module. Getting information from the finviz group performance page.

### 6.3.4 Spectrum

**class** finvizfinance.group.spectrum.Spectrum

Spectrum inherit from overview module. Getting information from the finviz group spectrum page.

**ScreenersView** (group='Sector', order='Name', out\_dir="")

Get screener table.

#### Parameters

- **group** (*str*) – choice of group option.
- **order** (*str*) – sort the table by the choice of order.

## 6.4 News

Getting trading news.

**class** finvizfinance.news.News

Getting information from the finviz news page.

**getNews** ()

Get insider information table.

Retrieves table information from finviz finance news.

**Returns** news table

**Return type** news(dict)

## 6.5 Insider

Getting trading insider.

**class** finvizfinance.insider.Insider (option='latest')

Getting information from the finviz insider page.

**Parameters** **option** (*str*) – choose a option (latest, latest buys, latest sales, top week, top week buys, top week sales, top owner trade, top owner buys, top owner sales, insider\_id)

**getInsider()**

Get insider information table.

**Returns** insider information table

**Return type** df(pandas.DataFrame)

## 6.6 Forex

Getting foreign exchange information.

**class** finvizfinance.forex.Forex

Getting information from the finviz forex page.

**chart** (forex, timeframe='D', urlonly=False)

Get forex chart.

**Parameters**

- **forex** (str) – foreign exchange name
- **timeframe** (str) – choice of timeframe(5M, H, D, W, M)
- **urlonly** (bool) – choice of downloading charts, default: downloading chart

**Returns** url for the chart

**Return type** charturl(str)

**performance** (change='percent')

Get forex performance table.

**Parameters** **change** (str) – choose an option of change(percent, PIPS)

**Returns** forex performance table

**Return type** df(pandas.DataFrame)

## 6.7 Crypto

Getting crypto currency information.

**class** finvizfinance.crypto.Crypto

Getting information from the finviz crypto page.

**chart** (crypto, timeframe='D', urlonly=False)

Get crypto chart.

**Parameters**

- **crypto** (str) – crypto currency
- **timeframe** (str) – choice of timeframe(5M, H, D, W, M)
- **urlonly** (bool) – choice of downloading charts, default: downloading chart

**Returns** url for the chart

**Return type** charturl(str)

**performance** ()

Get crypto performance table.

**Returns** crypto performance table

**Return type** df(pandas.DataFrame)

## 6.8 Future

Getting future information.

**class** finvizfinance.future.**Future**

Getting information from the finviz future page.

**performance** (*timeframe*='D')

Get forex performance table.

**Parameters** *timeframe* (*str*) – choice of timeframe(D, W, M, Q, HY, Y)

**Returns** forex performance table

**Return type** df(pandas.DataFrame)

## 6.9 Earnings

Getting earning dates information.

**class** finvizfinance.earnings.**Earnings** (*period*='This Week')

Partition dataframe of ticker information of period of earning dates(This Week, Next Week, Previous Week, This Month) into dates

**Parameters** *period* (*str*) – choose an option of period(This Week, Next Week, Previous Week, This Month).

**outputCSV** (*output\_dir*='earning\_days')

Output dataframes to csv files.

**Parameters** *output\_dir* (*str*) – name of the output directory.

**outputExcel** (*output\_file*='earning\_days.xlsx')

Output dataframes to single Excel file.

**Parameters** *output\_file* (*str*) – name of the output excel file.

**partitionDays** (*mode*='financial')

Partition dataframe to separate dataframes according to the dates.

**Parameters** *mode* (*str*) – choose an option of period(financial, overview, valuation, ownership, performance, technical).



## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### f

- `finvizfinance.crypto`, [19](#)
- `finvizfinance.earnings`, [20](#)
- `finvizfinance.forex`, [19](#)
- `finvizfinance.future`, [20](#)
- `finvizfinance.group.overview`, [17](#)
- `finvizfinance.group.performance`, [18](#)
- `finvizfinance.group.spectrum`, [18](#)
- `finvizfinance.group.valuation`, [18](#)
- `finvizfinance.insider`, [18](#)
- `finvizfinance.news`, [18](#)
- `finvizfinance.quote`, [13](#)
- `finvizfinance.screener.custom`, [16](#)
- `finvizfinance.screener.financial`, [16](#)
- `finvizfinance.screener.overview`, [15](#)
- `finvizfinance.screener.ownership`, [16](#)
- `finvizfinance.screener.technical`, [16](#)
- `finvizfinance.screener.ticker`, [17](#)
- `finvizfinance.screener.valuation`, [16](#)



**C**

`chart()` (*finvizfinance.crypto.Crypto method*), 19  
`chart()` (*finvizfinance.forex.Forex method*), 19  
`compare()` (*finvizfinance.screener.overview.Overview method*), 15  
`Crypto` (*class in finvizfinance.crypto*), 19  
`Custom` (*class in finvizfinance.screener.custom*), 16

**E**

`Earnings` (*class in finvizfinance.earnings*), 20

**F**

`Financial` (*class in finvizfinance.screener.financial*), 16  
`finvizfinance` (*class in finvizfinance.quote*), 13  
`finvizfinance.crypto` (*module*), 19  
`finvizfinance.earnings` (*module*), 20  
`finvizfinance.forex` (*module*), 19  
`finvizfinance.future` (*module*), 20  
`finvizfinance.group.overview` (*module*), 17  
`finvizfinance.group.performance` (*module*), 18  
`finvizfinance.group.spectrum` (*module*), 18  
`finvizfinance.group.valuation` (*module*), 18  
`finvizfinance.insider` (*module*), 18  
`finvizfinance.news` (*module*), 18  
`finvizfinance.quote` (*module*), 13  
`finvizfinance.screener.custom` (*module*), 16  
`finvizfinance.screener.financial` (*module*), 16  
`finvizfinance.screener.overview` (*module*), 15  
`finvizfinance.screener.ownership` (*module*), 16  
`finvizfinance.screener.technical` (*module*), 16  
`finvizfinance.screener.ticker` (*module*), 17  
`finvizfinance.screener.valuation` (*module*), 16

`Forex` (*class in finvizfinance.forex*), 19  
`Future` (*class in finvizfinance.future*), 20

**G**

`getColumns()` (*finvizfinance.screener.custom.Custom method*), 17  
`getCurrent()` (*finvizfinance.quote.Quote method*), 13  
`getFilterOptions()` (*finvizfinance.screener.overview.Overview method*), 15  
`getFilters()` (*finvizfinance.screener.overview.Overview method*), 15  
`getGroup()` (*finvizfinance.group.overview.Overview method*), 17  
`getInsider()` (*finvizfinance.insider.Insider method*), 18  
`getNews()` (*finvizfinance.news.News method*), 18  
`getOrders()` (*finvizfinance.group.overview.Overview method*), 17  
`getOrders()` (*finvizfinance.screener.overview.Overview method*), 15  
`getSignal()` (*finvizfinance.screener.overview.Overview method*), 16  
`getStatements()` (*finvizfinance.quote.Statements method*), 13

**I**

`Insider` (*class in finvizfinance.insider*), 18

**N**

`News` (*class in finvizfinance.news*), 18

**O**

`outputCSV()` (*finvizfinance.earnings.Earnings method*), 20

[outputExcel\(\)](#) (*finvizfinance.earnings.Earnings method*), 20  
[Overview](#) (*class in finvizfinance.group.overview*), 17  
[Overview](#) (*class in finvizfinance.screener.overview*), 15  
[Ownership](#) (*class in finvizfinance.screener.ownership*), 16

## P

[partitionDays\(\)](#) (*finvizfinance.earnings.Earnings method*), 20  
[Performance](#) (*class in finvizfinance.group.performance*), 18  
[performance\(\)](#) (*finvizfinance.crypto.Crypto method*), 19  
[performance\(\)](#) (*finvizfinance.forex.Forex method*), 19  
[performance\(\)](#) (*finvizfinance.future.Future method*), 20

## Q

[Quote](#) (*class in finvizfinance.quote*), 13

## S

[ScreenerView\(\)](#) (*finvizfinance.group.overview.Overview method*), 17  
[ScreenerView\(\)](#) (*finvizfinance.group.spectrum.Spectrum method*), 18  
[ScreenerView\(\)](#) (*finvizfinance.screener.custom.Custom method*), 16  
[ScreenerView\(\)](#) (*finvizfinance.screener.overview.Overview method*), 15  
[ScreenerView\(\)](#) (*finvizfinance.screener.ticker.Ticker method*), 17  
[set\\_filter\(\)](#) (*finvizfinance.screener.overview.Overview method*), 16  
[Spectrum](#) (*class in finvizfinance.group.spectrum*), 18  
[Statements](#) (*class in finvizfinance.quote*), 13

## T

[Technical](#) (*class in finvizfinance.screener.technical*), 16  
[Ticker](#) (*class in finvizfinance.screener.ticker*), 17  
[TickerCharts\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerDescription\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerFullInfo\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerFundament\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14

[TickerInsideTrader\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerNews\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerOuterRatings\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14  
[TickerSignal\(\)](#) (*finvizfinance.quote.finvizfinance method*), 14

## V

[Valuation](#) (*class in finvizfinance.group.valuation*), 18  
[Valuation](#) (*class in finvizfinance.screener.valuation*), 16