

```
In [1]: import ffn
%matplotlib inline
# download price data from Yahoo! Finance. By default,
# the Adj. Close will be used.
prices = ffn.get('aapl,msft', start='2010-01-01')
# let's compare the relative performance of each stock
# we will rebase here to get a common starting point for both securities
ax = prices.rebase().plot(figsize=(10, 5))
```

```
-----
TypeError                                 Traceback (most recent call last)
/var/folders/q2/vkx_vfw5ln57yp4n5gkgjmv0000gn/T/ipykernel_1007/78685778.py in <module>
      3 # download price data from Yahoo! Finance. By default,
      4 # the Adj. Close will be used.
----> 5 prices = ffn.get('aapl,msft', start='2010-01-01')
      6 # let's compare the relative performance of each stock
      7 # we will rebase here to get a common starting point for both securities

~/opt/anaconda3/lib/python3.9/site-packages/decorator.py in fun(*args, **kw)
    230         if not kwsyntax:
    231             args, kw = fix(args, kw, sig)
--> 232         return caller(func, *(extras + args), **kw)
    233     fun.__name__ = func.__name__
    234     fun.__doc__ = func.__doc__

~/opt/anaconda3/lib/python3.9/site-packages/ffn/utils.py in _memoize(func, *args, **kw)
     32         return cache[key]
     33     else:
--> 34         cache[key] = result = func(*args, **kw)
     35         return result
     36

~/opt/anaconda3/lib/python3.9/site-packages/ffn/data.py in get(tickers, provider, common_dates, forward_fill, clean_tickers, column_names, ticker_f
ield_sep, mrefresh, existing, **kwargs)
     74     # call provider - check if supports memoization
     75     if hasattr(provider, "mcache"):
--> 76         data[ticker] = provider(ticker=t, field=f, mrefresh=mrefresh, **kwargs)
     77     else:
     78         data[ticker] = provider(ticker=t, field=f, **kwargs)

~/opt/anaconda3/lib/python3.9/site-packages/decorator.py in fun(*args, **kw)
    230         if not kwsyntax:
    231             args, kw = fix(args, kw, sig)
--> 232         return caller(func, *(extras + args), **kw)
    233     fun.__name__ = func.__name__
    234     fun.__doc__ = func.__doc__

~/opt/anaconda3/lib/python3.9/site-packages/ffn/utils.py in _memoize(func, *args, **kw)
     32         return cache[key]
     33     else:
--> 34         cache[key] = result = func(*args, **kw)
     35         return result
     36

~/opt/anaconda3/lib/python3.9/site-packages/ffn/data.py in yf(ticker, field, start, end, mrefresh)
    138         field = "Adj Close"
    139
--> 140     tmp = pdata.get_data_yahoo(ticker, start=start, end=end)
    141
    142     if tmp is None:

~/opt/anaconda3/lib/python3.9/site-packages/pandas_datareader/data.py in get_data_yahoo(*args, **kwargs)
     78
     79 def get_data_yahoo(*args, **kwargs):
--> 80     return YahooDailyReader(*args, **kwargs).read()
     81
     82

~/opt/anaconda3/lib/python3.9/site-packages/pandas_datareader/base.py in read(self)
    251     # If a single symbol, (e.g., 'GOOG')
    252     if isinstance(self.symbols, (string_types, int)):
--> 253         df = self._read_one_data(self.url, params=self._get_params(self.symbols))
    254     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    255     elif isinstance(self.symbols, DataFrame):

~/opt/anaconda3/lib/python3.9/site-packages/pandas_datareader/yahoo/daily.py in _read_one_data(self, url, params)
    151         try:
    152             j = json.loads(re.search(ptrn, resp.text, re.DOTALL).group(1))
--> 153             data = j["context"]["dispatcher"]["stores"]["HistoricalPriceStore"]
    154         except KeyError:
    155             msg = "No data fetched for symbol {} using {}"

TypeError: string indices must be integers
```

In []: