```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pandas as pd
>>> import wget
>>>
wget.download("https://raw.githubusercontent.com/lazyprogrammer/machine_learning
_examples/master/tf2.0/sbux.csv")
'sbux (3).csv'
>>> df = pd.read_csv('sbux.csv')
>>> df
            date    open    high     low   close    volume  Name
0     2013-02-08  27.920  28.325  27.920  28.185   7146296  SBUX
1     2013-02-11  28.260  28.260  27.930  28.070   5457354  SBUX
2     2013-02-12  28.000  28.275  27.975  28.130   8665592  SBUX
3     2013-02-13  28.230  28.230  27.750  27.915   7022056  SBUX
4     2013-02-14  27.765  27.905  27.675  27.775   8899188  SBUX
...          ...     ...     ...     ...     ...       ...   ...
1254  2018-02-01  56.280  56.420  55.890  56.000  14690146  SBUX
1255  2018-02-02  55.900  56.320  55.700  55.770  15358909  SBUX
1256  2018-02-05  55.530  56.260  54.570  54.690  16059955  SBUX
1257  2018-02-06  53.685  56.060  53.560  55.610  17415065  SBUX
1258  2018-02-07  55.080  55.430  54.440  54.460  13927022  SBUX

[1259 rows x 7 columns]
>>> df.head()
         date    open    high     low   close   volume  Name
0  2013-02-08  27.920  28.325  27.920  28.185  7146296  SBUX
1  2013-02-11  28.260  28.260  27.930  28.070  5457354  SBUX
2  2013-02-12  28.000  28.275  27.975  28.130  8665592  SBUX
3  2013-02-13  28.230  28.230  27.750  27.915  7022056  SBUX
4  2013-02-14  27.765  27.905  27.675  27.775  8899188  SBUX
>>> df.tail()
            date    open   high    low  close    volume  Name
1254  2018-02-01  56.280  56.42  55.89  56.00  14690146  SBUX
1255  2018-02-02  55.900  56.32  55.70  55.77  15358909  SBUX
1256  2018-02-05  55.530  56.26  54.57  54.69  16059955  SBUX
1257  2018-02-06  53.685  56.06  53.56  55.61  17415065  SBUX
1258  2018-02-07  55.080  55.43  54.44  54.46  13927022  SBUX
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    1259 non-null   object
 1   open    1259 non-null   float64
 2   high    1259 non-null   float64
 3   low     1259 non-null   float64
 4   close   1259 non-null   float64
 5   volume  1259 non-null   int64
 6   Name    1259 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 69.0+ KB
```

```
>>> ### Selecting rows and Columns ###
>>> df.columns
Index(['date', 'open', 'high', 'low', 'close', 'volume', 'Name'],
dtype='object')
>>> df.columns = ['date', 'open', 'high', 'low', 'close', 'volume', 'name']
>>> df['open']
0       27.920
1       28.260
2       28.000
3       28.230
4       27.765
         ...
1254    56.280
1255    55.900
1256    55.530
1257    53.685
1258    55.080
Name: open, Length: 1259, dtype: float64
>>> df['open','high']
Traceback (most recent call last):
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexes\base.py", line 3080, in get_loc
    return self._engine.get_loc(casted_key)
  File "pandas\_libs\index.pyx", line 70, in
pandas._libs.index.IndexEngine.get_loc
  File "pandas\_libs\index.pyx", line 101, in
pandas._libs.index.IndexEngine.get_loc
  File "pandas\_libs\hashtable_class_helper.pxi", line 4554, in
pandas._libs.hashtable.PyObjectHashTable.get_item
  File "pandas\_libs\hashtable_class_helper.pxi", line 4562, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: ('open', 'high')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    df['open','high']
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\frame.py", line 3024, in __getitem__
    indexer = self.columns.get_loc(key)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexes\base.py", line 3082, in get_loc
    raise KeyError(key) from err
KeyError: ('open', 'high')
>>> df[['open','high']]
        open    high
0     27.920  28.325
1     28.260  28.260
2     28.000  28.275
3     28.230  28.230
```

```
4     27.765  27.905
...      ...     ...
1254  56.280  56.420
1255  55.900  56.320
1256  55.530  56.260
1257  53.685  56.060
1258  55.080  55.430

[1259 rows x 2 columns]
>>> df.iloc[0]
date      2013-02-08
open          27.920
high          28.325
low           27.920
close         28.185
volume       7146296
name            SBUX
Name: 0, dtype: object
>>> df.loc[0]
date      2013-02-08
open          27.920
high          28.325
low           27.920
close         28.185
volume       7146296
name            SBUX
Name: 0, dtype: object
>>> df2 = pd.read_csv('sbux.csv', index_col = 'date')
>>> df2
              open    high     low   close    volume  Name
date
2013-02-08  27.920  28.325  27.920  28.185   7146296  SBUX
2013-02-11  28.260  28.260  27.930  28.070   5457354  SBUX
2013-02-12  28.000  28.275  27.975  28.130   8665592  SBUX
2013-02-13  28.230  28.230  27.750  27.915   7022056  SBUX
2013-02-14  27.765  27.905  27.675  27.775   8899188  SBUX
...            ...     ...     ...     ...       ...   ...
2018-02-01  56.280  56.420  55.890  56.000  14690146  SBUX
2018-02-02  55.900  56.320  55.700  55.770  15358909  SBUX
2018-02-05  55.530  56.260  54.570  54.690  16059955  SBUX
2018-02-06  53.685  56.060  53.560  55.610  17415065  SBUX
2018-02-07  55.080  55.430  54.440  54.460  13927022  SBUX

[1259 rows x 6 columns]
>>> df2.iloc[0]
open       27.920
high       28.325
low        27.920
close      28.185
volume    7146296
Name         SBUX
Name: 2013-02-08, dtype: object
>>> df.loc[0]
date      2013-02-08
```

```
open          27.920
high          28.325
low           27.920
close         28.185
volume       7146296
name            SBUX
Name: 0, dtype: object
>>> df.loc['2013-02-13']
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    df.loc['2013-02-13']
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 894, in __getitem__
    return self._getitem_axis(maybe_callable, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 1123, in _getitem_axis
    return self._get_label(key, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 1072, in _get_label
    return self.obj.xs(label, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\generic.py", line 3736, in xs
    loc = index.get_loc(key)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexes\range.py", line 354, in get_loc
    raise KeyError(key)
KeyError: '2013-02-13'
>>> # Oups, mistake
>>> df2.loc[0]
Traceback (most recent call last):
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexes\base.py", line 3080, in get_loc
    return self._engine.get_loc(casted_key)
  File "pandas\_libs\index.pyx", line 70, in
pandas._libs.index.IndexEngine.get_loc
  File "pandas\_libs\index.pyx", line 101, in
pandas._libs.index.IndexEngine.get_loc
  File "pandas\_libs\hashtable_class_helper.pxi", line 4554, in
pandas._libs.hashtable.PyObjectHashTable.get_item
  File "pandas\_libs\hashtable_class_helper.pxi", line 4562, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: 0

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "<pyshell#22>", line 1, in <module>
    df2.loc[0]
```

```
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 894, in __getitem__
    return self._getitem_axis(maybe_callable, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 1123, in _getitem_axis
    return self._get_label(key, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexing.py", line 1072, in _get_label
    return self.obj.xs(label, axis=axis)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\generic.py", line 3736, in xs
    loc = index.get_loc(key)
  File
"C:\Users\Nico\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\c
ore\indexes\base.py", line 3082, in get_loc
    raise KeyError(key) from err
KeyError: 0
>>> # That's what I wanted. Now the index is the date. 0 inst a valid index now
>>> df2.loc[2013-02-13]
SyntaxError: leading zeros in decimal integer literals are not permitted; use an
0o prefix for octal integers
>>> df2.loc['2013-02-13']
open       28.230
high       28.230
low        27.750
close      27.915
volume    7022056
Name          SBUX
Name: 2013-02-13, dtype: object
>>> # Here we go
>>> type(df2.loc['2013-02-13'])
<class 'pandas.core.series.Series'>
>>> # Let's suppose I want all the rows where the open price is greater than 64
>>> df[df['open'] > 64]
            date   open    high    low  close   volume  name
1087  2017-06-05  64.85  64.870  64.18  64.27  6809284  SBUX
1088  2017-06-06  64.22  64.350  64.05  64.16  5448439  SBUX
1089  2017-06-07  64.13  64.295  63.34  63.50  8364994  SBUX
>>> type(df['open'] > 64)
<class 'pandas.core.series.Series'>
>>> df['open'] > 64
0       False
1       False
2       False
3       False
4       False
        ...
1254    False
1255    False
1256    False
```

```
1257    False
1258    False
Name: open, Length: 1259, dtype: bool
>>> df.values
array([['2013-02-08', 27.92, 28.325, ..., 28.185, 7146296, 'SBUX'],
       ['2013-02-11', 28.26, 28.26, ..., 28.07, 5457354, 'SBUX'],
       ['2013-02-12', 28.0, 28.275, ..., 28.13, 8665592, 'SBUX'],
       ...,
       ['2018-02-05', 55.53, 56.26, ..., 54.69, 16059955, 'SBUX'],
       ['2018-02-06', 53.685, 56.06, ..., 55.61, 17415065, 'SBUX'],
       ['2018-02-07', 55.08, 55.43, ..., 54.46, 13927022, 'SBUX']],
      dtype=object)
>>> df[['open', 'close']].values
array([[27.92 , 28.185],
       [28.26 , 28.07 ],
       [28.   , 28.13 ],
       ...,
       [55.53 , 54.69 ],
       [53.685, 55.61 ],
       [55.08 , 54.46 ]])
>>> smalldf = df[['open', 'close']]
>>> smalldf.to_csv('output1.cvs')
>>> # output1.csv will have an index column. If we dont want it, we should do:
>>> smalldf = df[['open', 'close'], index = False]
SyntaxError: invalid syntax
>>> smalldf.to_csv('output1.cvs', index = False)
>>>
```