Visualization

1. Retrieve Data from MongoDB

In [32]:
```
import pymongo

client = pymongo.MongoClient()
```

In [33]:
```
import pandas as pds

db = client.get_database("stock")
collection = db.get_collection("stock")
data = list(collection.find())
df = pds.DataFrame.from_records(data)
df.drop('_id', axis=1, inplace=True)
print(df.head())
print(df.info())
```

```
             Datetime         SPY       SBUX        AAPL  MSFT
0 2020-11-18 14:30:00  360.760010  98.510002  118.910004   NaN
1 2020-11-18 14:31:00  360.679993  98.565498  118.684998   NaN
2 2020-11-18 14:32:00  360.730011  98.669998  118.620003   NaN
3 2020-11-18 14:33:00  360.660004  98.705002  118.377701   NaN
4 2020-11-18 14:34:00  360.695007  98.644997  118.499901   NaN
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1950 entries, 0 to 1949
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Datetime  1950 non-null   datetime64[ns]
 1   SPY       1950 non-null   float64
 2   SBUX      1950 non-null   float64
 3   AAPL      1949 non-null   float64
 4   MSFT      1 non-null      float64
dtypes: datetime64[ns](1), float64(4)
memory usage: 76.3 KB
None
```

2 Visualization of Stock Price with Plot.ly: Line Plots

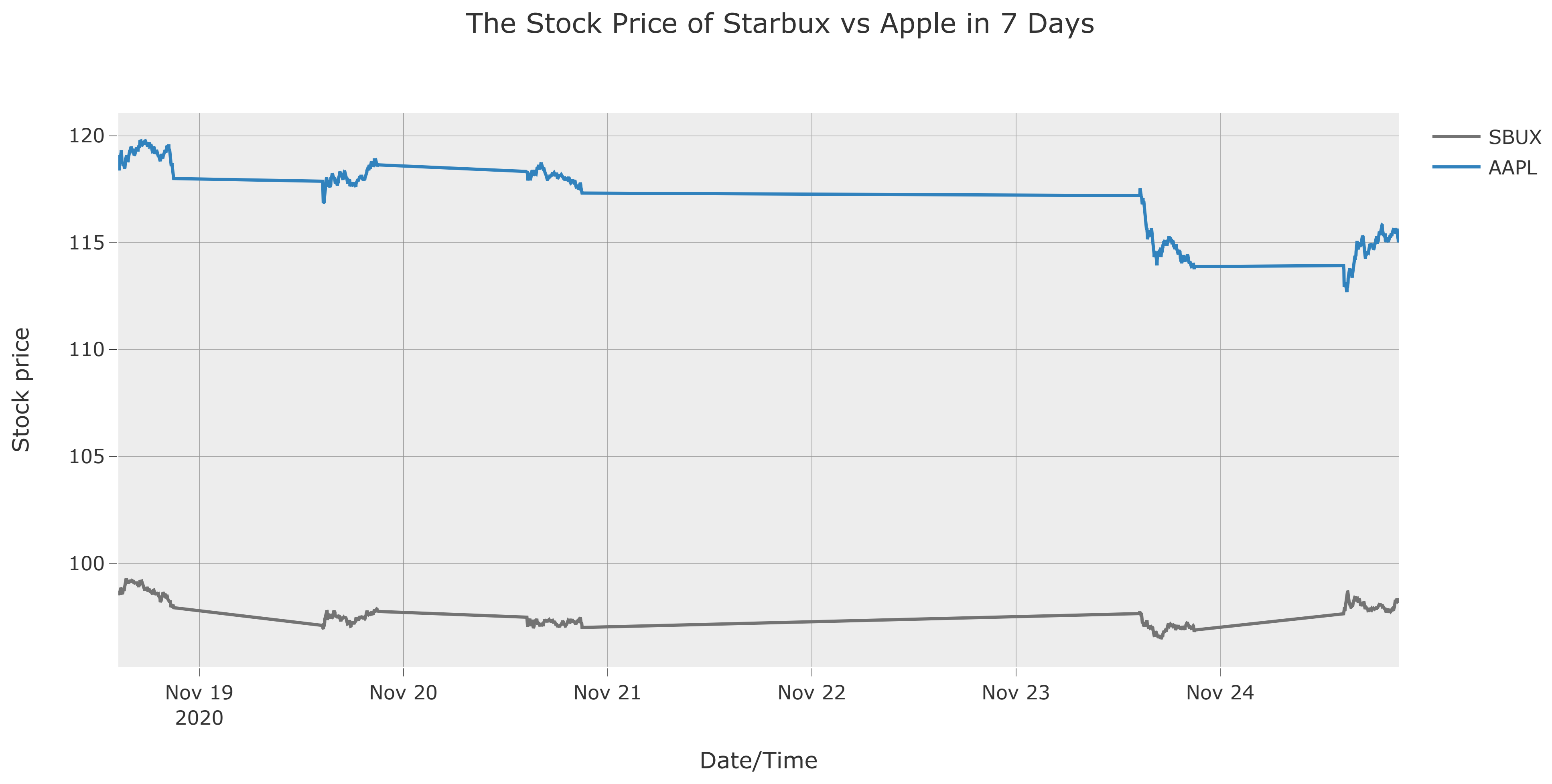
We used the line graph in the scatter type from plotly.graph_objects to create stock price chart in 7 days.

In [38]:
```
import plotly.graph_objects as go

COLORS = ['rgb(115,115,115)', 'rgb(49,130,189)']
def static_stacked_trend_graph(stack=False):
    global df
    sources = ['SBUX', 'AAPL']
    x = df['Datetime']
    fig = go.Figure()
    for i, s in enumerate(sources):
        fig.add_trace(go.Scatter(x=x, y=df[s], mode='lines', name=s,
                                 line={'width': 2, 'color': COLORS[i]},
                                 stackgroup='stack' if stack else None))
    title = 'The Stock Price of Starbux vs Apple in 7 Days'
    if stack:
        title += ' [Stacked]'

    fig.update_layout(template='ggplot2',
                      title=title,
                      yaxis_title='Stock price',
                      xaxis_title='Date/Time')
    return fig
```

In [39]:
```
import plotly.offline as pyo
pyo.init_notebook_mode()
static_stacked_trend_graph(stack=False)
```


The Stock Price of Starbux vs Apple in 7 Days
Stock price
Nov 19
Nov 20
Nov 21
Nov 22
Nov 23
Nov 24
Date/Time
SBUX
AAPL