```
#Setup
Apikey= "Insert API key"
Secret= "Insert Secret key"

from binance import Client, ThreadedWebsocketManager, ThreadedDepthCacheManager
import pandas as pd
pd.set_option('display.max_rows', 3000)
pd.set_option('display.max_columns', 3000)
pd.set_option('display.width', 1000)
pd.set_option('display.max_rows', 3000)
pd.set_option('display.max_columns', 3000)
pd.set_option('display.width', 3000)
import numpy as np
import mplfinance as mpf

#Authenticate
client = Client (Apikey, Secret)

#Get tickers
tickers = client.get_all_tickers()
tickers_df = pd.DataFrame (tickers, columns = ["symbol", "price"])

print (tickers_df ["symbol"])

#List of symbols

list_of_symbols = []
for i in tickers_df ["symbol"]:
    list_of_symbols.append(i)
print (list_of_symbols)

#Getting the price of each symbol

list_cryptocurrencies = []
list_values = []

for i in list_of_symbols:

    try:

            historical = client.get_historical_klines(""+i+"",
Client.KLINE_INTERVAL_1DAY, "02 Jan 2021")
            hist_df = pd.DataFrame (historical)
            hist_df.columns = ["Open time", "Open", "High", "Low", "Close",
"Volume", "Close time", "Quote asset volume","Number of trades", "Taker buy base
asset volume", "Taker buy quote asset volume", "Can be ignored"]
            hist_df["Open time"] = pd.to_datetime( hist_df["Open time"] / 1000,
unit="s" )
            hist_df["Close time"] = pd.to_datetime( hist_df["Close time"] / 1000,
unit="s" )
```

```
            numeric_columns = ["Open", "High", "Low", "Close", "Volume", "Quote
asset volume", "Taker buy base asset volume","Taker buy quote asset volume"]
            hist_df[numeric_columns] =
hist_df[numeric_columns].apply(pd.to_numeric, axis=1)
            hist_df_reduced = pd.DataFrame (hist_df, columns = ["Open time",
"Close"])

            try:
                difference_close_price_1_day =  hist_df_reduced ["Close"] [1] -
hist_df_reduced ["Close"] [0]
                porcentage_difference_close_price_1_day = 100 - (hist_df_reduced
["Close"] [0] * 100 / hist_df_reduced ["Close"] [1])

            except KeyError:
                continue

            if porcentage_difference_close_price_1_day >= 0:
                if "USDT" in i:
                    print (i, porcentage_difference_close_price_1_day)
                    list_cryptocurrencies.append (i)
                    list_values.append (porcentage_difference_close_price_1_day)

            if porcentage_difference_close_price_1_day < 0:
                continue


    except ValueError:
       continue

excel_file = pd.DataFrame(list(zip(list_cryptocurrencies, list_values)), columns=
["Cryptocurrencies", "Values"])
print (excel_file)

with pd.ExcelWriter ('Path.xlsx') as writer:
    excel_file.to_excel (writer, sheet_name="name", index=False)
```