

# 유한별\_\_PORTFOLIO

---

1. TODAY CHECK
2. US STOCK ANALYSIS
3. INSTA AUTO DM

**+github-address+**

<https://github.com/hanbyulyou/Portfolio> YouHanByul

# 1. TODAY CHECK

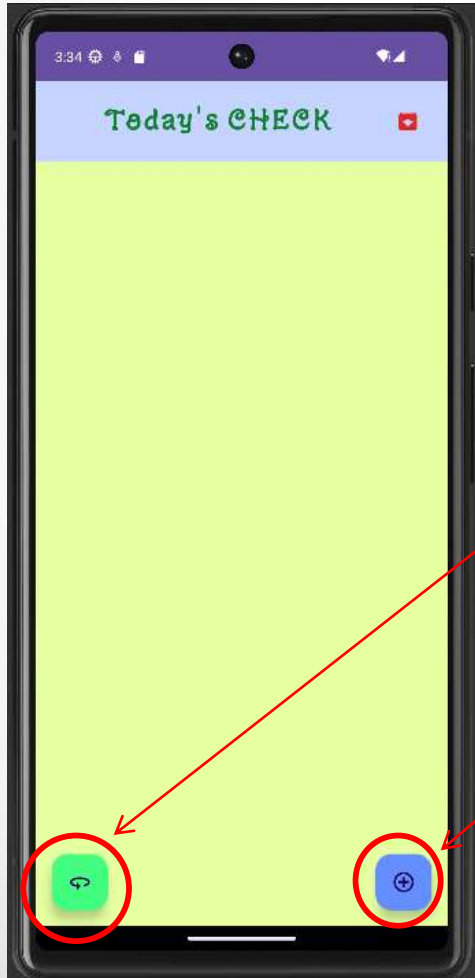
---

# 言語: Kotlin

# 環境: Android Studio

- Androidアプリとして開発した「TODAY CHECK」というアプリです。
- やることを曜日ごとに設定して追加できます。
- やることを毎日（ボタンを押して）チェックできます
- チェックしたことを確認できます。

# 1. TODAY CHECK



- 起動時に表示される画面です。
- 削除ボタン: 追加したすべてのタスクを削除できるボタンです。
- 追加ボタン: タスクを追加できるボタンです。

# 1. TODAY CHECK

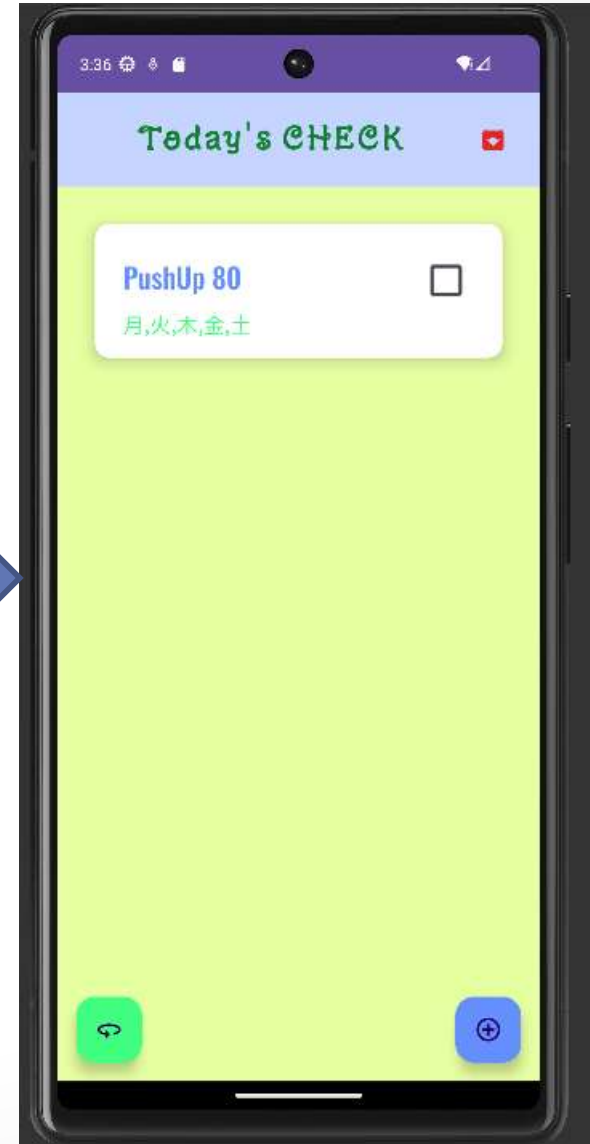


- 追加ボタンを押した後に表示される画面です。
- 曜日ごとに選択してタスクを追加できます。
- SAVEボタン: タスクを追加
- CANCELボタン: キャンセル

# 1. TODAY CHECK

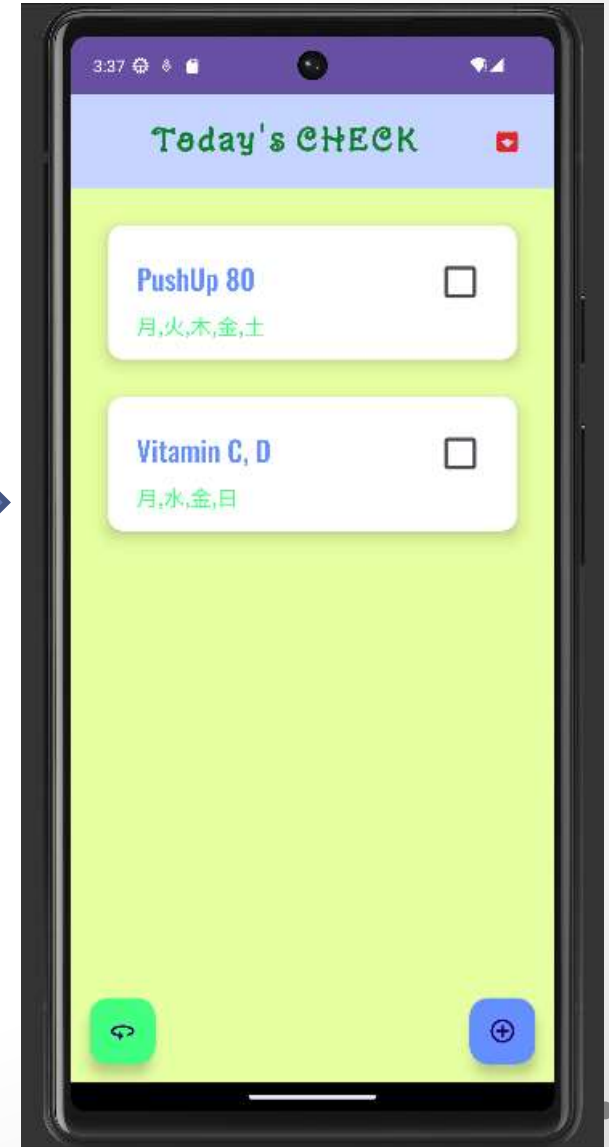


- 日付を指定してやることを追加

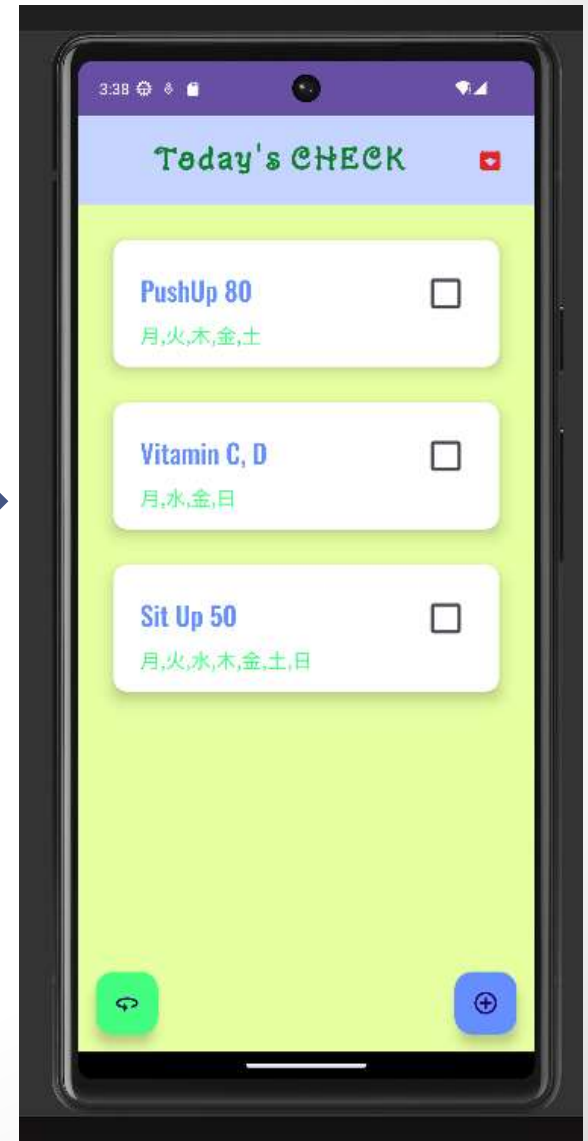


- 追加したやることがメイン画面に表示される

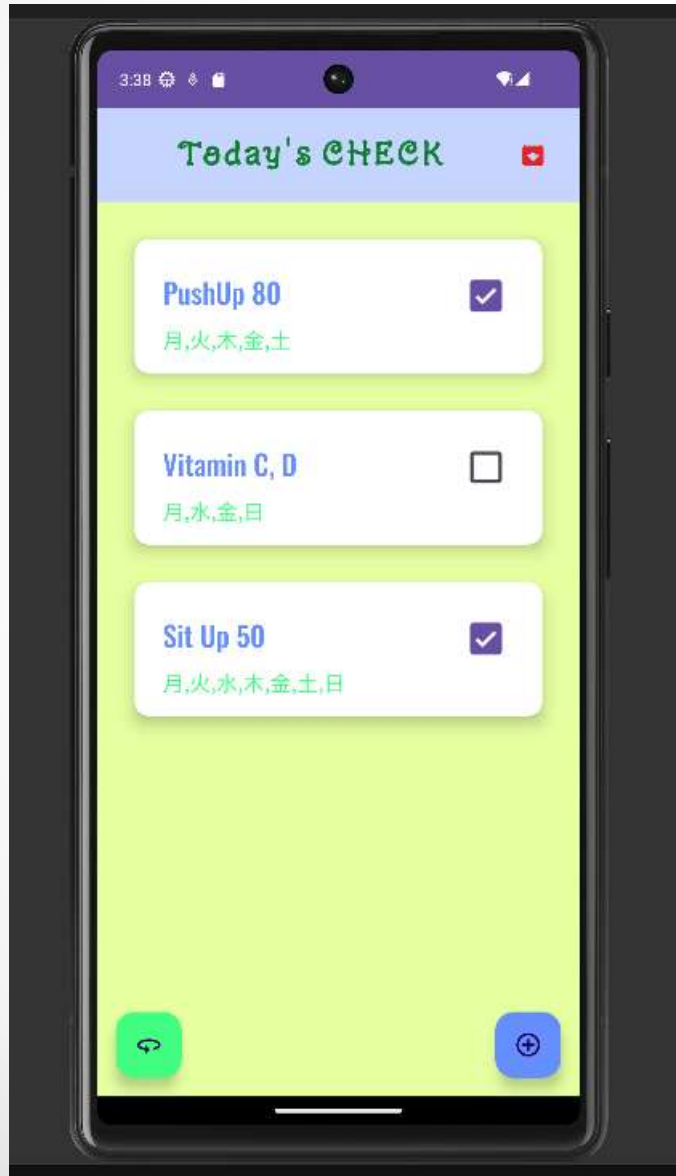
# 1. TODAY CHECK



# 1. TODAY CHECK



# 1. TODAY CHECK



- 今日やったことを次のようにチェックして確認できます。
- 日付が翌日になると、チェックが解除されるようにコード構造を作成しました。



## 2. US STOCK ANALYSIS

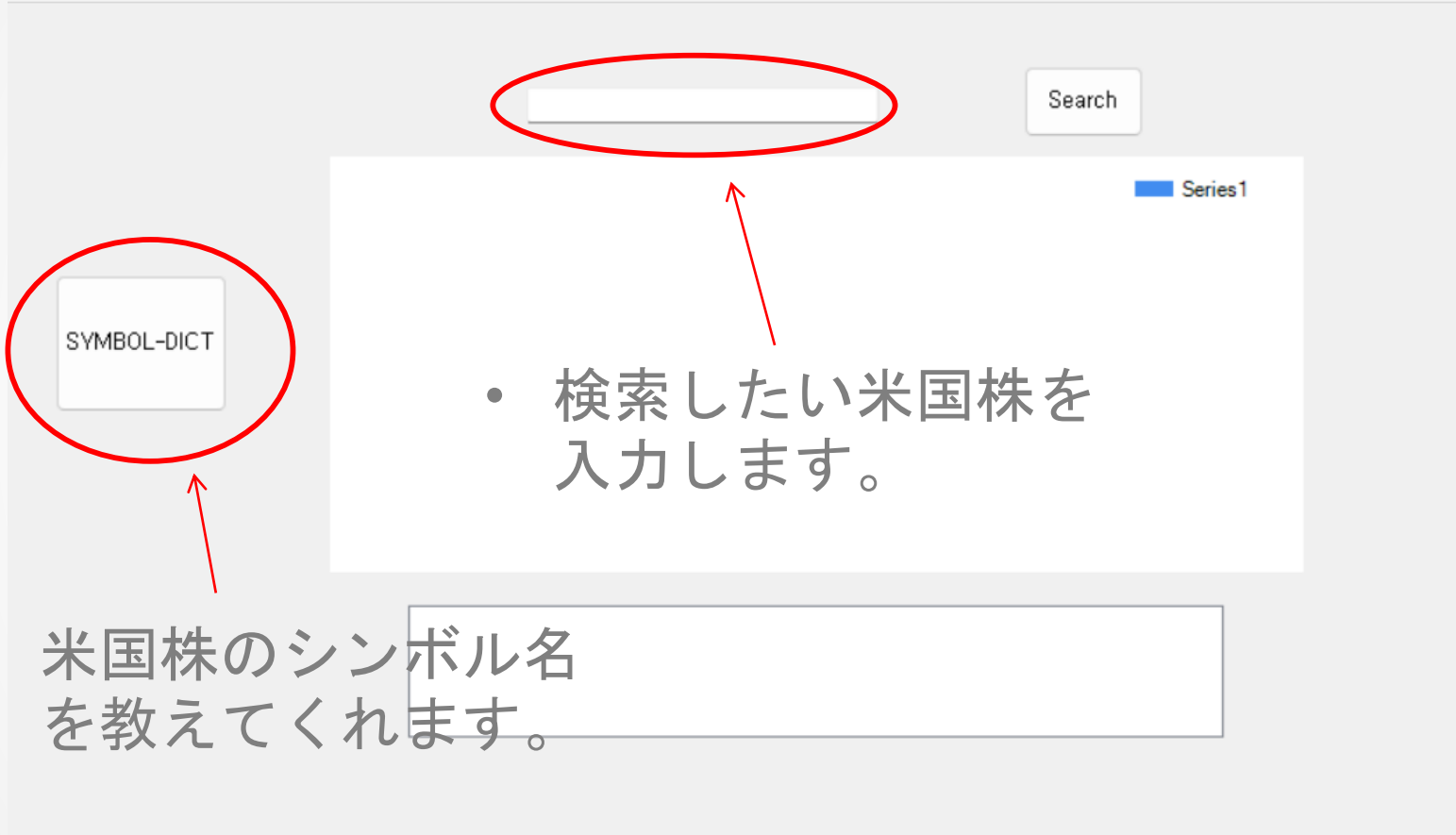
# 言語: C#

# 環境: Visual Studio

- 特定の米国株を検索して、チャートを日単位で表示できます。
- 買い推奨: 前日より株価が 3% 上昇した場合は、買い推奨を行います。
- 売り推奨: 前日より株価が3%上昇した場合、売り推奨をします。

# 2. US STOCK ANALYSIS

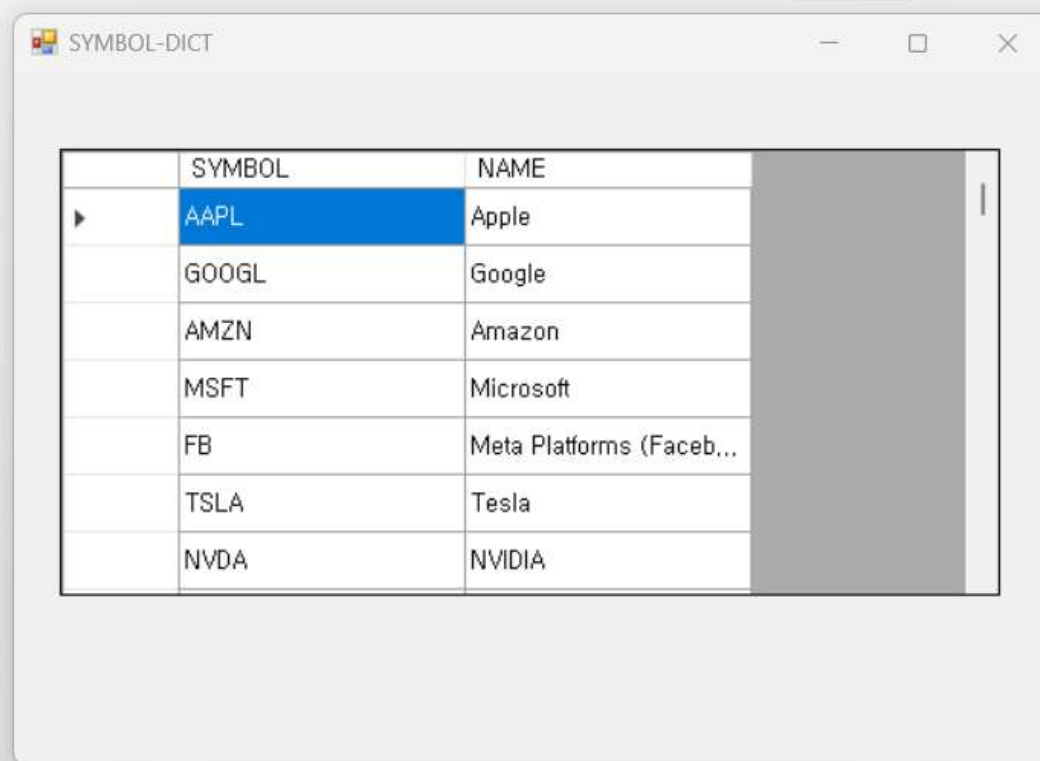
US-STOCK : TRADE HELP PROGRAM



- 米国株のシンボル名を教えてください。
- 実行したときに表示されるメイン画面です。

# 2. US STOCK ANALYSIS

SYMBOL-DICT

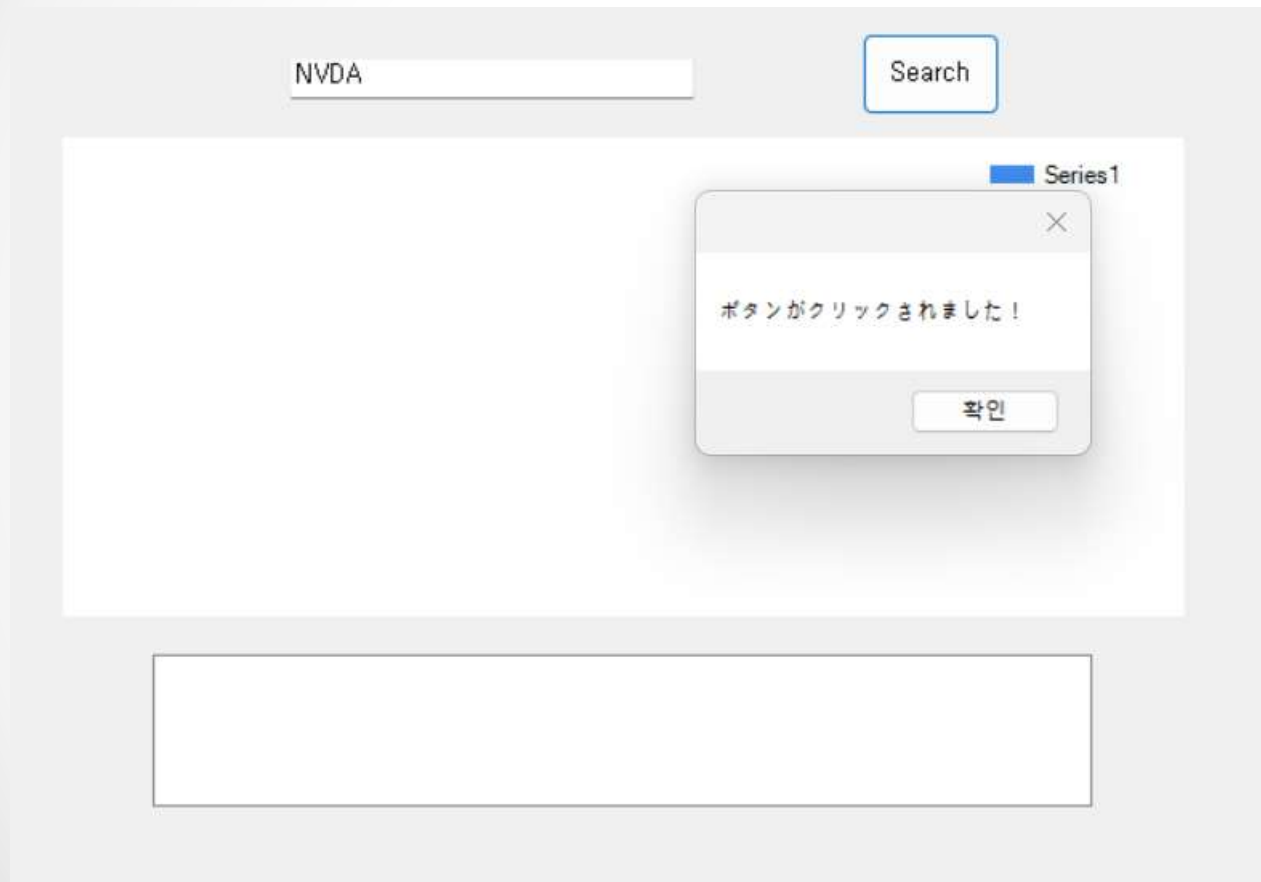


The screenshot shows a web application window titled "SYMBOL-DICT". Inside the window is a table with two columns: "SYMBOL" and "NAME". The table lists several major US companies. The first row, "AAPL" for "Apple", is highlighted in blue. To the right of the table is a large, empty grey rectangular area, likely a placeholder for additional information or a chart.

SYMBOL	NAME
AAPL	Apple
GOOGL	Google
AMZN	Amazon
MSFT	Microsoft
FB	Meta Platforms (Faceb...
TSLA	Tesla
NVDA	NVIDIA

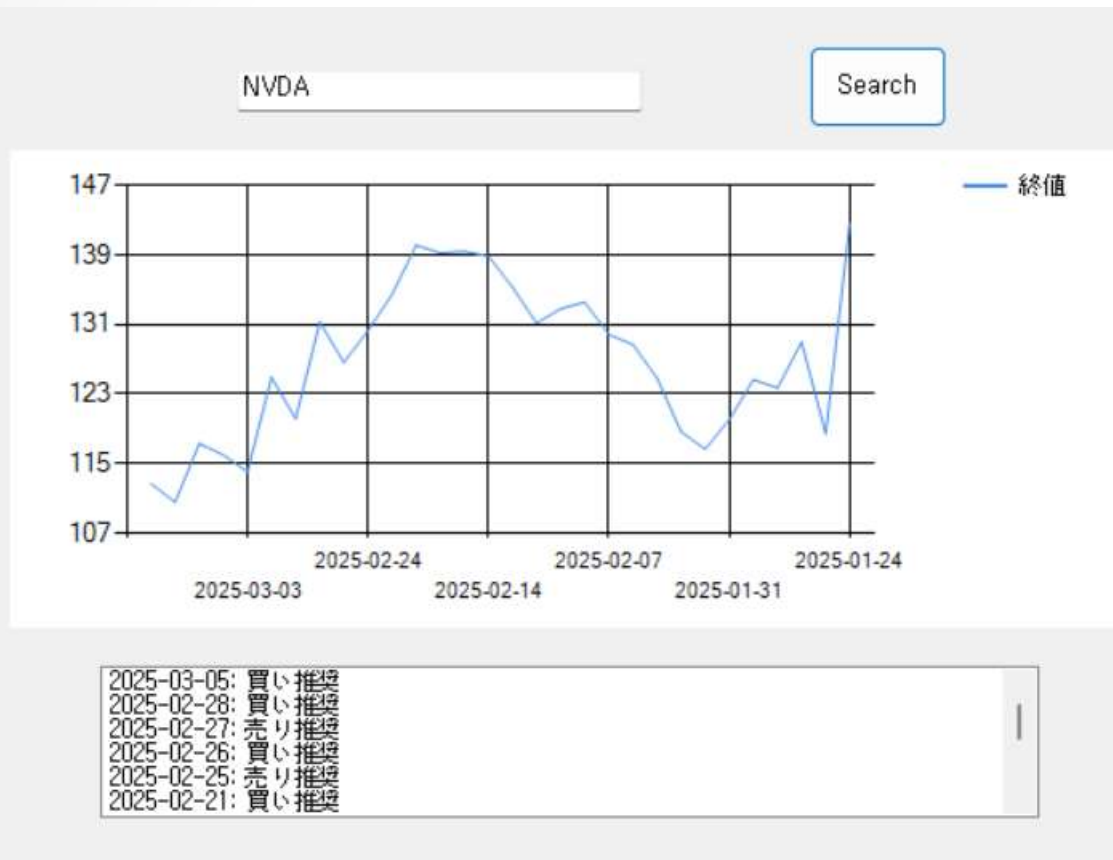
- 右側のNAMEは私たちが一般的に知っている会社名で、左側のシンボル名で検索する必要があります。

# 2. US STOCK ANALYSIS



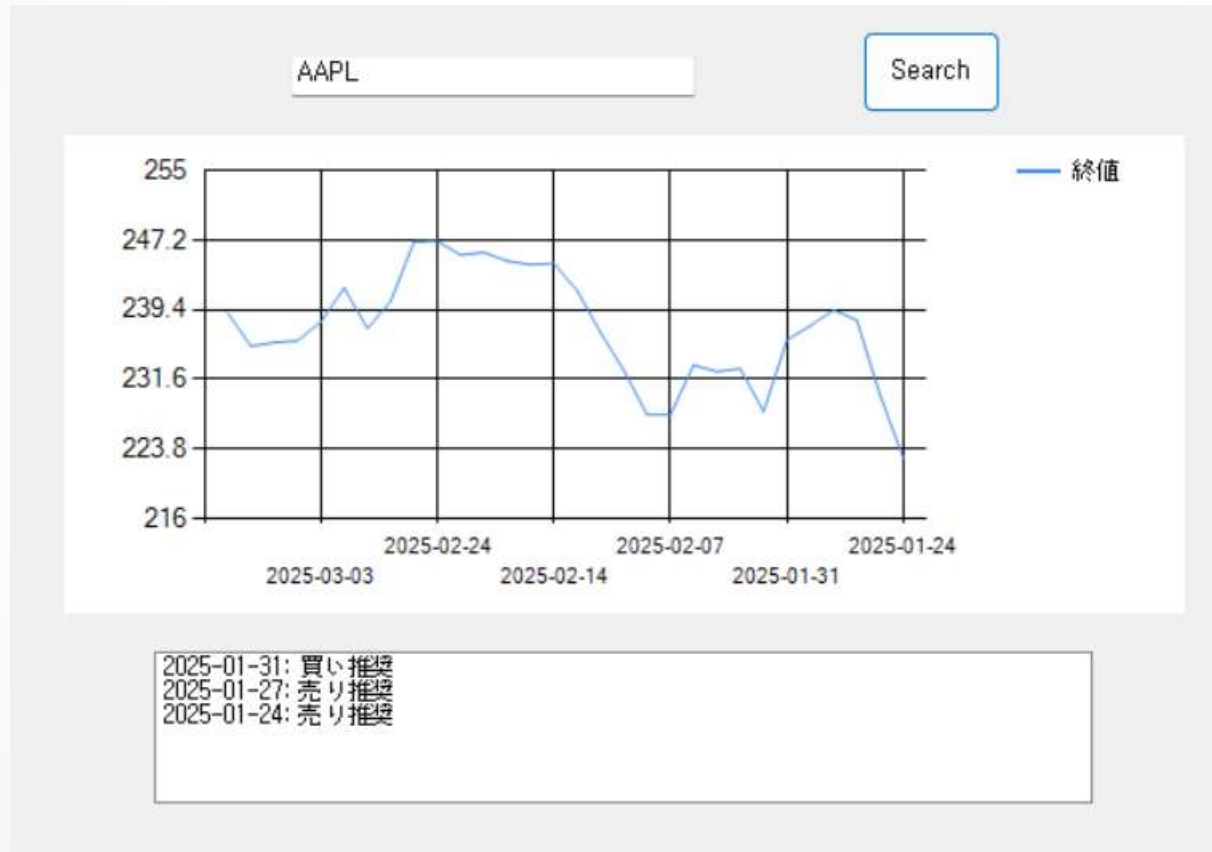
- シンボル名を入力します
- 「Search」ボタンを押すと、次のように検索が実行されます。

# 2. US STOCK ANALYSIS

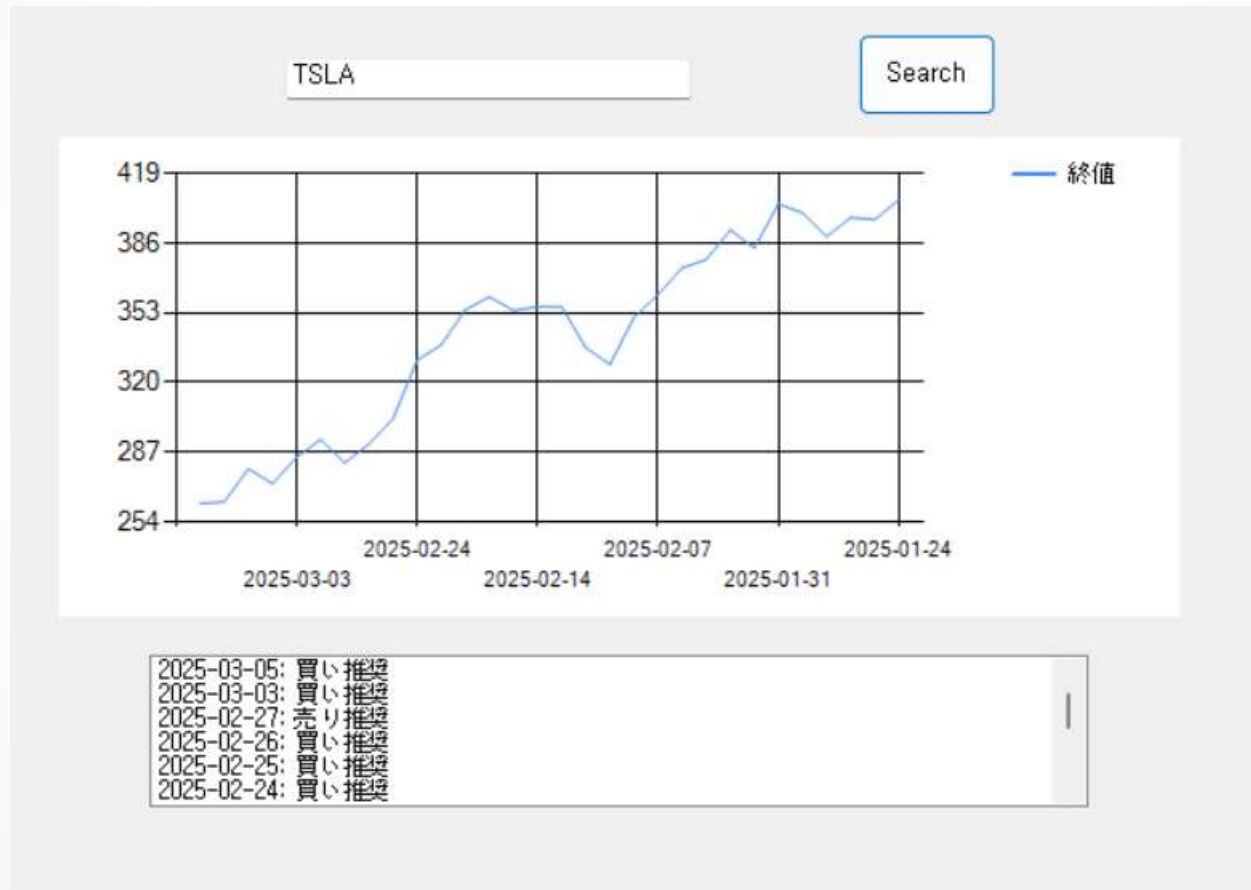


- 検索した会社の最近のチャートが表示されます
- 日付ごとに買い推奨と売り推奨があります

# 2. US STOCK ANALYSIS



# 2. US STOCK ANALYSIS



# 3. INSTA AUTO DM

# 言語: Python

# 環境: selenium, webdriver, jupyter notebook

- ログインから自動DMを送信することまでマクロが可能なプログラムです。
- xlsx(Excelファイル)で値をLoadしてマクロを実行させます。
- まだ一度もDMを利用していないという仮定で使用可能です。



# 3. INSTA AUTO DM

```
chrome_service = Service("C:\\Users\\*\\*\\chromedriver.exe")
```

1. 特定のパスにあるクロム ドライバーを読み込みます。（クロム ドライバーとクロムのバージョンを一致させる必要があります。）

```
file_path = ("C:\\Users\\*\\*\\insta_auto_dm_contents.xlsx")  
df = pd.read_excel(file_path)
```

2. 特定のパスにあるexcelファイルを読むように定義します。

# 3. INSTA AUTO DM

```
def open_url(browser, url):  
    browser.get(url)  
    time.sleep(.5)
```

```
url = "instagram-address"
```

```
# Open the browser
```

```
browser = open_browser()
```

```
open_url(browser, url)
```

```
# Login - Id
```

```
login_id = 'your-id'
```

```
# Login - Pw
```

```
login_pw = 'your-password'
```

3. URLを開く関数を定義します。

4. urlはinstagramログインページのアドレスに設定します。

5. ログインIDとログインパスワードを実際の本人のものとして定義します。

# 3. INSTA AUTO DM

```
def login_page_field(browser, login_id, login_pw):  
  
    # Login - Id  
    if find_xpath('/html/body/div[1]/div/div/div[2]/div/div/div[1]/input', login_id):  
        find_xpath('/html/body/div[1]/div/div/div[2]/div/div/div[1]/input', login_id)  
  
    # Login - Pw  
    if find_xpath('/html/body/div[1]/div/div/div[2]/div/div/div[2]/input', login_pw):  
        find_xpath('/html/body/div[1]/div/div/div[2]/div/div/div[2]/input', login_pw)  
  
    # Login - Button  
    find_xpath('/html/body/div[1]/div/div/div[2]/div/div/div[3]/button', login_pw)  
  
login_page_field(browser, login_id, login_pw)
```

6. ログイン ページ  
フィールドを定義しま  
す。

7. 各IDとパスワードの  
インプットウィンドウ  
を「xpath」値に抽出し、  
値を入れます。

8. ログインボタンの「x  
path」値を抽出し、ク  
リックします。

9. ログイン ページ  
フィールドを実行しま  
す

10. フィールドが実行さ  
れた後は、メイン画面  
に接続されます。

### 3. INSTA AUTO DM

```
# Input Receiver
if find_xpath('/html/body/div[7]/div') != None:
    find_xpath('/html/body/div[7]/div')
# Select Receiver
time.sleep(1)
find_xpath('/html/body/div[7]/div')
# Click Button "Chat"
time.sleep(1)
find_xpath('/html/body/div[7]/div')
# Input Content
time.sleep(3)
find_xpath('/html/body/div[1]/div')
# Click Button "Chat"
time.sleep(1)
find_xpath('/html/body/div[1]/div')
```

11. 受取人の値を入れてくれます。受け取る人のリストが出てきます。
12. 一番上にある受取人のボタンをクリックします。
13. [Chat] ボタンをクリックします。
14. DMのメッセージ内容を入れてくれます。
15. [Chat] ボタンをクリックしてメッセージを送信します。

# 3. INSTA AUTO DM

```
# Loop through each row in the DataFrame
for index, row in df.iterrows():
    # Receiver
    receiver_name = row['NAME']
    # Content
    content_list = row['CONTENT']

    # Process the page fields for this row
    auto_dm__field(browser, receiver_name, content_list)

    # Before executing next row, add delay
    time.sleep(1)
```

16. xlsxファイルで、「NAME」列の項目が受信者の名前に入ります。

17. xlsxファイルで「CONTENT」列の項目がDMのメッセージ内容に入ります。

18. for 関数を使用して、xlsxの該当する列のすべての値が処理されます。