

Dividend Tracker Korea

Sangmin SHIM

1) Why do I have to make this application ?

Nowadays people are interested in dividend stocks such as KOSPI and KOSDAQ. In Naver, one of the biggest web sites in Korea, users can FIND the information of enterprise that they want to research. The problem is, given that there are lots of pages (more than 10 pages), users can't SEARCH immediately what they want to check. They have to scroll over the pages until finding the company they need. So, I'd like to make an application for users to SEARCH directly their company interested. To ease the use, I formed a User Interface (UI) like button, message box, scroll etc. Finally, users can see the visualized data with graph. They are able to figure out the changes at one glance. With this application (called "Dividend Tracker Korea") users can find the information at once.

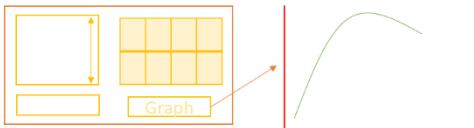
2) How is Dividend Tracker Korea organized ?

1. KOSPI(KOSDAQ)_Write.py

URL : www.example.com
↓
(Requests)
↓
BeautifulSoup : {Text}, **HTML. Parser**
↓
(CSV Writer)
: KOSPI(KOSDAQ) ("+" str(date.today())+"").csv

It consists of **2 python files**,
(1. KOSPI(KOSDAQ) Write.py) and
(2. KOSPI(KOSDAQ) Load.py).

2. KOSPI(KOSDAQ)_Load.py

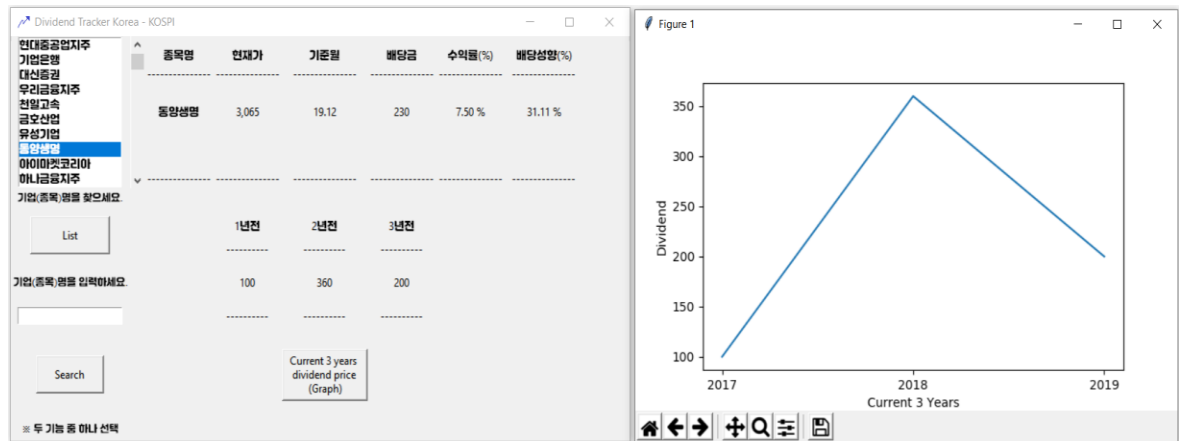
Tkinter : ScrollBox, ListBox, Button, MsgBox

Matplotlib : Plot

First of all, in 1. KOSPI(KOSDAQ)_Write.py, this application gets an URL in which they have the data, information that we want. And then, the application requests an access to the URL for parsing, which means that we could extract the data according to HTML Tag (<>,</>). Through this process we can make a CSV file and save the data.

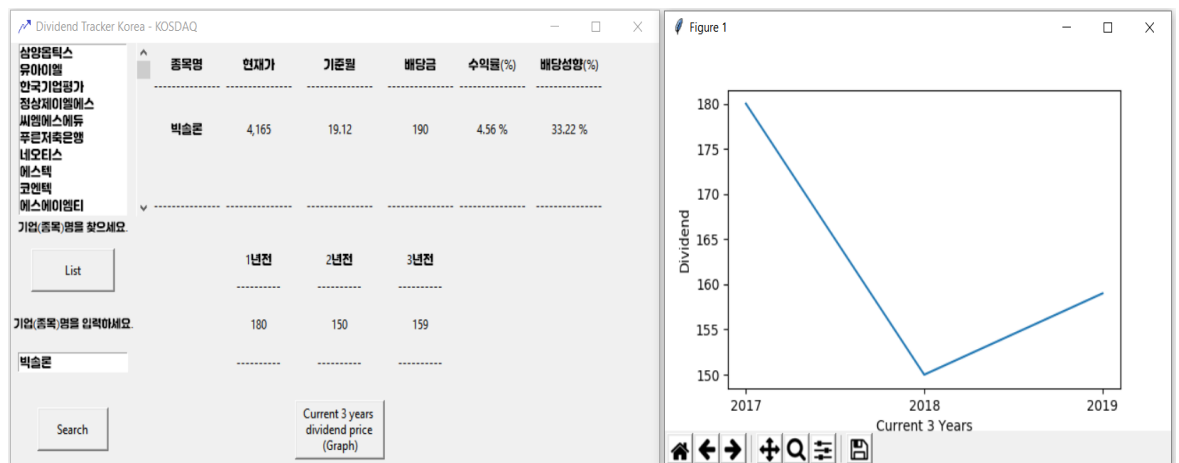
Secondly, in 2. KOSPI(KOSDAQ)_Load.py, this application opens a User Interface (UI). Based on the data taken from 1. KOSPI(KOSDAQ)_Write.py, the application lists the names of enterprises on left. If users could not find easily by scrolling up/down, they search directly by typing its name. On right, it shows the data related in the enterprise. And by clicking the button "Graph", the changes for 3 years show in the form of graph.

Hereby, we can see the result like that.

- List



- Search



3) What kind of libraries (packages) did I use ?

(1) Requests

It allows to access a certain site. It's the first step of Web Crawling.

(2) BeautifulSoup

It allows to parse the data based on HTML's Tag. Once we success requesting the site, this library collects the data. As programmers, they have to consider how the tags are related to each other. We can think about subordinate relation for example.

(3) Csv

It allows to make a CSV file and to write in. Once the HTML Text are well parsed, we can easily use the data.

(4) Tkinter

It allows to design a User Interface (UI). Thanks to this library, I could make a window, scroll box, list box, button, message box and Grid to arrange the data showed on application.

(5) Matplotlib

It's a powerful library to visualize the data. There are various forms of graph, and I chose a simple plot for this application.

4) Characteristic

(1) Up-To-Dateness

The application requests an access to URL every time we execute. Unless the Web Server (URL) changes or fails, it extracts the newest and proper data every time.

(2) Slowness of run

It takes a time to gather the data from Web Server (URL). And sometimes, some Web Server refuses the access of Request. Therefore, it's highly recommended to find other ways to extract the data.