

GetRegular.py (sounds like a laxative)

Use this program to collect share prices each Friday.

- a. Uses schedule to control the collection of data each Friday.
- b. Collects the share price for all shares listed in a file called sharestoget.txt.
- c. The share price collected is the most recent Closing price.
- d. The data is saved in a file using the share ticker as the file name.
- e. If the share data file already exists the new data is appended to it.
- f. Each record has an incrementing index (pointer) used when plotting.
- g. A watch/bought/sold price will be added to each record.

The Raspberry Pi used to test these instruction notes had a new memory card with the latest Raspian-Wheezy image (2015-05-05). The programs work with Python 2.7

Prerequisites

1. Update and Upgrade your Raspberry Pi before installing more libraries.
2. The Ystockquote function is required.
3. The Schedule function is required.
4. A text file sharestoget.txt is required to hold the list of shares.

1. Update and Upgrade . (Skip this step if you installed GetHistorical.)

- a. Type `sudo apt-get update`
- b. Type `sudo apt-get upgrade`

Choose Y to continue when asked. This process will take a while.

2. Install Ystockquote function using pip installer (skip if you installed GetHistorical)

- a. Type `sudo apt-get install python-pip`

Choose Y to continue when asked.

- b. Type `sudo pip install ystokequote`

3. Install Schedule function

- a. Type `sudo pip install schedule`

4. Create the Directories and the `sharestoget.txt` file. (skip if you installed GetHistorical)

- a. Using File Manager create a directory called 'Shares'
- b. Using File Manager create a directory called 'SharesData' within the directory called 'Shares'
- c. Using a Text Editor Leafpad or similar create a text file called 'sharestoget.txt' save it in the sharesdata directory.
- d. Edit the 'sharestoget.txt' file to have at least two shares to collect. I suggest using something like the following as I know these shares exist.

```
COST.L , 273,1  
RR.L , 875,1
```

Note the space either side of the first comma. After typing the last line, make sure you press return so the cursor moves to the next blank line before saving the file.

Program Listing.

Using Midori, Epiphany or similar to navigate to the python file then cut and paste the listing into Idle.

<http://www.rknorman.co.uk/GetRegular.py>

Save the program as `GetRegular.py` in the 'Shares' directory.

More notes below.

Notes on using GetRegular

1. The Raspberry Pi must be connected to the Internet.
2. Remember it collects the data at the scheduled day/time. (Friday 19:00).
3. If the ticker reference(s) is incorrect you will get an error.
4. If the GetRegular program fails to run and you miss a Friday data collection you can use the GerHistorical program to collect the data. If there are any overlaps in the data collected use ReOrder.py to fix the Index and delete any duplicate records.
5. If the watch/bought/sold value was incorrect during the data collection it can be modified using SetWatch.py
6. GetRegular will collect data even if the Friday is a holiday such as Good Friday, Christmas Day, Boxing Day and New Years Day. This is because it collects the most recent Closing price available at the collection time. So if Christmas Day is a Friday it collects Thursday's price data but saves it with Friday's date.

I have two directories with the programs and the data files in them. The 'Shares' and 'SharesData' directories are the main ones. Another pair of directories 'TestShares' and 'TestSharesData' are used as my test environment so I can modify and test programs without messing up data already collected.

Sources and Contributors

I have borrowed code and ideas from the following sites for this program:

Getting share price data.

<http://www.instructables.com/id/Getting-Stock-Prices-on-Raspberry-Pi-Python>

Scott Kildall

Copying files <http://pythoncentral.io/how-to-copy-a-file-in-python-with-shutil/>

Using Schedule <https://pypi.python.org/pypi/schedule>

Disclaimer

I will not be responsible for any monetary loss, loss of data, corruption of files or any other problems resulting from these programs or the installation processes.