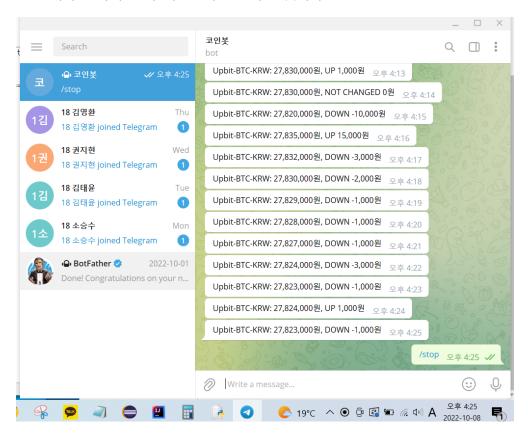
Using upbitpy and telegrambot programs, we created a program that represents current price and market fluctuations. The problem is that if /stop is entered in advance, the system stops even before it starts.

upbitpy와 telegrambot program을 이용해서 현재 가격과 시세변동을 나타내는 프로그램을 만들었다. 문제점은 /stop을 가장 최근의 입력을 가져오는 방식으로 해서 사전에 /stop이 입력되어 있으면 시작도 하기 전에 시스템이 멈춘다는 것이다.



```
kelplyungy-C#Usermidwid#Deakop#HYU#3-2#합(#Wee]
File Edit Fommat Run Options Window Help
Fitelegram upbit 원호 교육 설치
from telegram ext import Updater
from telegram ext import Updater
from telegram ext import Updater
from telegram
smort Updater
smort telegram
smort Updater
smort datetime
smort datetime
smort options
smort pandas as pd #cvc module
smort pandas as pd #cvc module
 # 시작 전에 채팅창을 clear하거나 어떤 채팅이라도 남겨야 합니다.
# 채팅의 마지막이 /stop인 상태에서 시작하면 실행이 안됩니다.
 COINBOT_TOKEN='5743982025:AAF7Up20IPAPMYIBwlgo6SmYffHw46wmw_8'
COINBOT_CHAT_ID='5658191294'
        main():
bot=telegram.Bot(token=COINBOT_TOKEN)
upbit = Upbitpy()
updater = Ubdater(COINBOT_TOKEN)
last_price=0
       #csv
Date=[]
BTC_KRW=[]
data = {
    'Date': Date.
    'BTC-KRW': BTC_KRW
}
while True:
              | le True:
| # /stoe을 일릭하면 코인봇이 while을 빠져나온다.

updates=bot.getUpdates()
| last_message=updates[-]]['message']['text']

if(last_message=='/stop'):

break
              ticker = upbit.get.ticker(['KRM-BTC'])[0] # 비트교인 원화 (USDI-BTC..etc)
price = int(ticker['trade_price']) # int값으로 현재 가격을 낳음
changesprice-lest.price
if last.price(라) # last.price가 0인 경우는 첫번째라서 UPDOWN을 제크x
if change-D:
pose"
                   else:
    pos="NOT CHANGED"
    pos="NOT CHANGED"
    pos="NOT CHANGED"
    #text = "((1) Lubit-ETC-KRM: (1원, () (1원*, format(datetime_datetime_now()_strftime("Xm/Xd 채내됐지도"), format(crice, ','), pos.format(change, ','))
    #text = "(10) Lubit-ETC-KRM: (1원, () (1원*, format(format(price, ','), pos.format(change, ','))    #format(price, ',') → 숫자를 문자으로 바꾸고 45,000 같이 임료를 넣어짐

              else:
text = 'Upbit-BTC-KRW: {}원'.format(format(price,','))
updater.bot.send_message(chat_id=COINBOT_CHAT_ID, text=text)
last_price=price
                                                                                                                                                                                                                                                                                                                                     Ln: 1 Col: 0
 🕟 leejiyun.py - C:#Users#dlwld#Desktop#HYU#3-2#컴네#leejiyun.py (3.10.7)
 |
|le True:
|# /stop을 입력하면 코인봇이 while을 빠져나온다.
|undates=bot.getUndates()
|last_message=undates[-1]['message']['text']
|if(last_message=='/stop'):
             #csv data 일찍
Date.aspend(format(datetime.datetime.now().strftime('%m/%d %H:%M:%S'))) # 현재시간
BTC_KFM:aspend(format(price,','))
              wait(INTERVAL_MIN)
        # /stop을 하고 빠져나와서 모든 data로 csv파일을 만든다.
dfpd.lbateFrame(data)
print(df)
os.chdir(r)C:MUser#Rdiw|dmDesktop제fYUm3-2#검네')
df.to_csv('BTD_KRR.csv')
  INTERVAL_MIN = 1
    ief wait(min):

now - datetime.datetime.now() #현재 시간 넣고

resain_second = 80 - now.second # 현재 시간이 18:44:47이면 remain_second=13

resain_second == 80 - (min - (now.minute % min + 1)) # 2분이라 치면 #7초에 출력되고, 13초후 출력, 그라고 45분의 2분 후인 47분에 출력됨

time_sfeet(remin_second)
      __name__ == '__main__': # nameO| mainOl라면(mainOl 직접 실행된다면)
logging.basicConfig(level=logging.INFO) # logging의 level INFO,MAPNING 등등.. INFO는 예상대로 작용하는지에 대한 확인
main()
```

[] leejiyun.py - C:₩Users₩dlwld₩Desktop₩HYU₩3-2₩컵네₩leejiyun.py (3.10.7)

- a ×

```
#telegram,upbit 관련 모듈 설치
from telegram.ext import Updater
from upbitpy import Upbitpy
from telegram.ext import Updater, CommandHandler
from telegram import Update, Bot
import telegram
import time
import datetime
import logging
import pandas as pd #cvc module
import os
# 시작 전에 채팅창을 clear하거나 어떤 채팅이라도 남겨야 합니다.
# 채팅의 마지막이 /stop인 상태에서 시작하면 실행이 안됩니다.
COINBOT_TOKEN='5743982025:AAF7Up2OIPAPMYIBwlqo6SmYffHw46wmw_8'
COINBOT_CHAT_ID='5658191294'
def main():
   bot=telegram.Bot(token=COINBOT_TOKEN)
   upbit = Upbitpy()
   updater = Updater(COINBOT_TOKEN)
   last_price=0
```

#csv

```
Date=[]
    BTC_KRW=[]
    data = {
       'Date': Date,
       'BTC-KRW': BTC_KRW
       }
   while True:
       #/stop을 입력하면 코인봇이 while을 빠져나온다.
       updates=bot.getUpdates()
       last_message=updates[-1]['message']['text']
        if(last_message=='/stop'):
            break
       ticker = upbit.get_ticker(['KRW-BTC'])[0] # 비트코인 원화 (USDT-BTC..etc)
       price = int(ticker['trade_price']) # int값으로 현재 가격을 넣음
       change=price-last_price
       if last_price!=0: # last_price가 0인 경우는 첫번째라서 UPDOWN을 체크x
            if change > 0:
                pos="UP"
            elif change < 0:
                pos="DOWN"
            else:
                pos="NOT CHANGED"
            #text
                                 '({})
                                            Upbit-BTC-KRW:
                                                                  {}원,
                                                                                       {}원
                                                                              {}
'.format(datetime.datetime.now().strftime('%m/%d
                                                                              %H:%M:%S'),
format(price,','),pos,format(change,','))
```

```
text = 'Upbit-BTC-KRW: {}원, {} {}원'.format( format(price,','),pos,format(change,','))
#format(price,',') -> 숫자를 문자열로 바꾸고 45,000 같이 쉼표를 넣어줌
       else:
           text = 'Upbit-BTC-KRW: {}원'.format(format(price,','))
       updater.bot.send_message(chat_id=COINBOT_CHAT_ID, text=text)
       last_price=price
       #csv data 입력
       Date.append(format(datetime.datetime.now().strftime('%m/%d %H:%M:%S'))) # 현재시간
       BTC_KRW.append(format(price,','))
       wait(INTERVAL_MIN)
    #/stop을 하고 빠져나와서 모은 data로 csv파일을 만든다.
    df=pd.DataFrame(data)
    print(df)
    os.chdir(r'C:\Users\dlwld\Desktop\HYU\3-2\Zlub
    df.to_csv('BTC_KRW.csv')
INTERVAL_MIN = 1
def wait(min):
    now = datetime.datetime.now() #현재 시간 넣고
    remain_second = 60 - now.second # 현재 시간이 19:44:47이면 remain_second=13
```

remain\_second += 60 \* (min - (now.minute % min + 1)) # 2분이라 치면 47초에 출력되고, 13 초후 출력, 그리고 45분의 2분 후인 47분에 출력됨

time.sleep(remain\_second)

if \_\_name\_\_ == '\_\_main\_\_': # name이 main이라면(main이 직접 실행된다면)

logging.basicConfig(level=logging.INFO) # logging의 level INFO,WARNING 등등.. INFO는 예상 대로 작동하는지에 대한 확인

main()