

# 파이썬을 활용한 업무자동화

12회차: 업무자동화로 컴퓨터가 스스로 일하게 하기

# 목차

12회차: 자동화 스케줄링

- 복습
- Cron
  - 설치가이드
  - 설정 방법
  - 설정 방법 예시
- Cron을 설정해보자
  - 뉴스/파일입출력/이메일/엑셀을 활용한 예제
- 입력값을 프로그램에 전달하기

# 복습

01

지난 강의 때 뭐했지..?

## GUI

```
from tkinter import *

class GUIT():
    tkhandler = None

    # __init__ 함수는 클래스 변수가 생성될때
    # 자동으로 실행되는 함수
    def __init__(self):
        self.tkhandler = Tk()

        # 창 크기를 지정
        self.tkhandler.geometry("500x500")

        # 제목 지정
        self.tkhandler.title("패스트캠퍼스 자동화 프로그램")

    def run(self):
        self.tkhandler.mainloop()

g = GUIT()
g.run()
```

# 복습

02

지난 강의 때 뭐했지..?

```
# 이메일 주소를 입력할 수 있는 텍스트박스를 추가
# relief는 텍스트박스의 디자인을 의미
# 그 외 다른 값은 일부 환경에서 보이지 않는 경우가 있음
# bd는 테두리 두께
self.txt = Text(self.tkhandler, width=40, height=1, relief=RIDGE, bd=1)
self.txt.pack()

self.btn_email = Button(self.tkhandler, text="이메일 보내기", width=30, command=self.auto_email)
self.btn_email.pack()
```

```
def auto_email(self):
    from my_email import Email

    # 1.0과 END는 첫문자부터 끝 문자까지 가져온다는 의미
    # 맨 마지막에 개행문자가 추가되므로 strip()으로 제거
    recv = self.txt.get("1.0", END).strip()

    self.append_log(recv + '에게 이메일을 전송합니다.')
    e = Email()
    e.send_mail('이태화', recv)

    self.append_log('성공적으로 이메일을 전송하였습니다.')
```

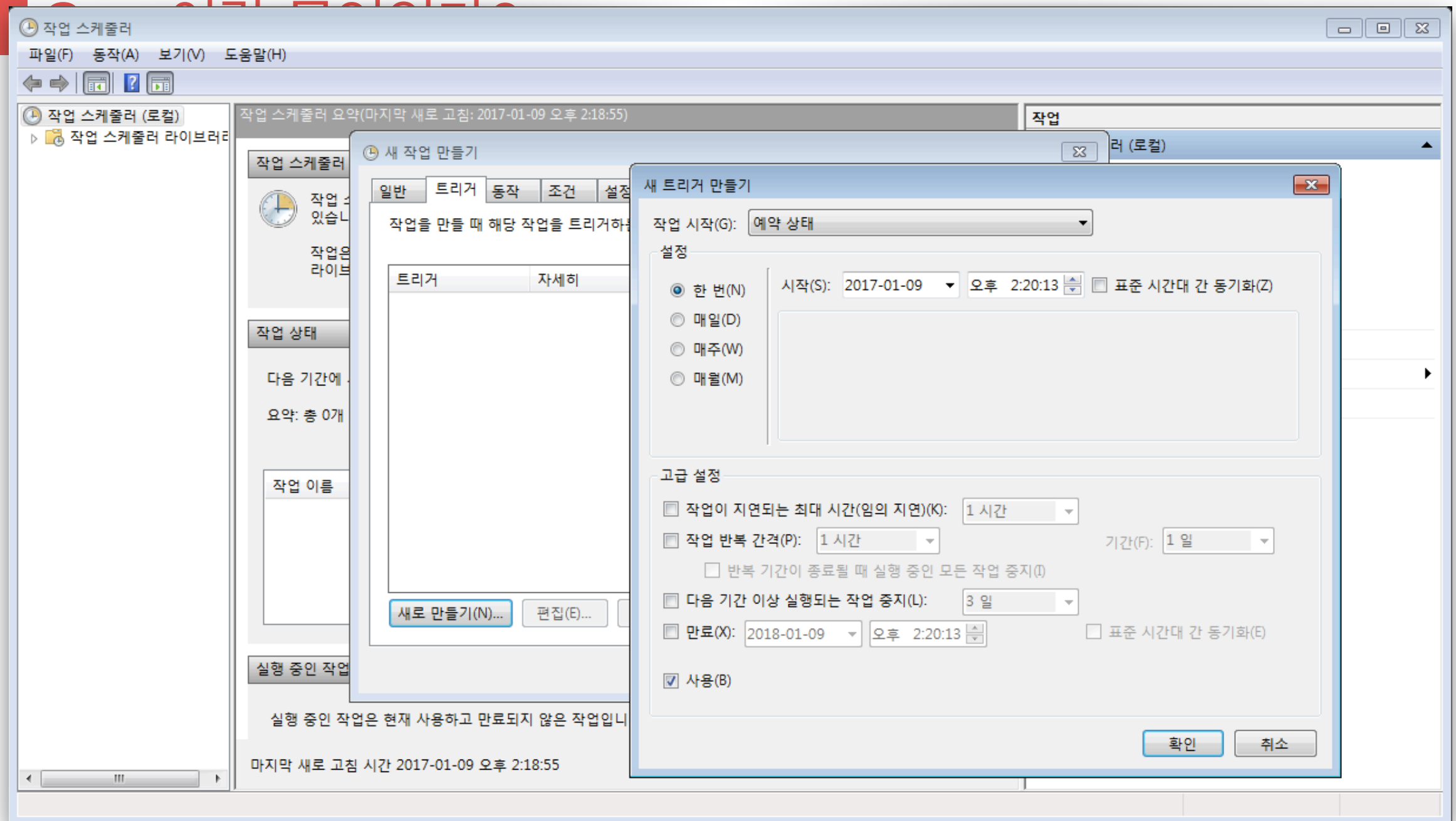
## ■ Cron이란 무엇인가?

- 리눅스/유닉스(맥) 계열의 운영체제에서 사용되는 잡 스케줄러
  - ▶ 설정한 날짜,시간에 따라 스크립트를 실행해줌
- 윈도우에는 작업 스케줄러라는 별도의 스케줄러가 있음
  - ▶ 작업 스케줄러도 충분히 좋으나..!
  - ▶ 더 가벼운 cron 을 사용
  - ▶ 복잡한 GUI 대신 간단한 텍스트 편집의 설정
- 무엇보다 모든 운영체제에서 동일한 설정방법

# Cron

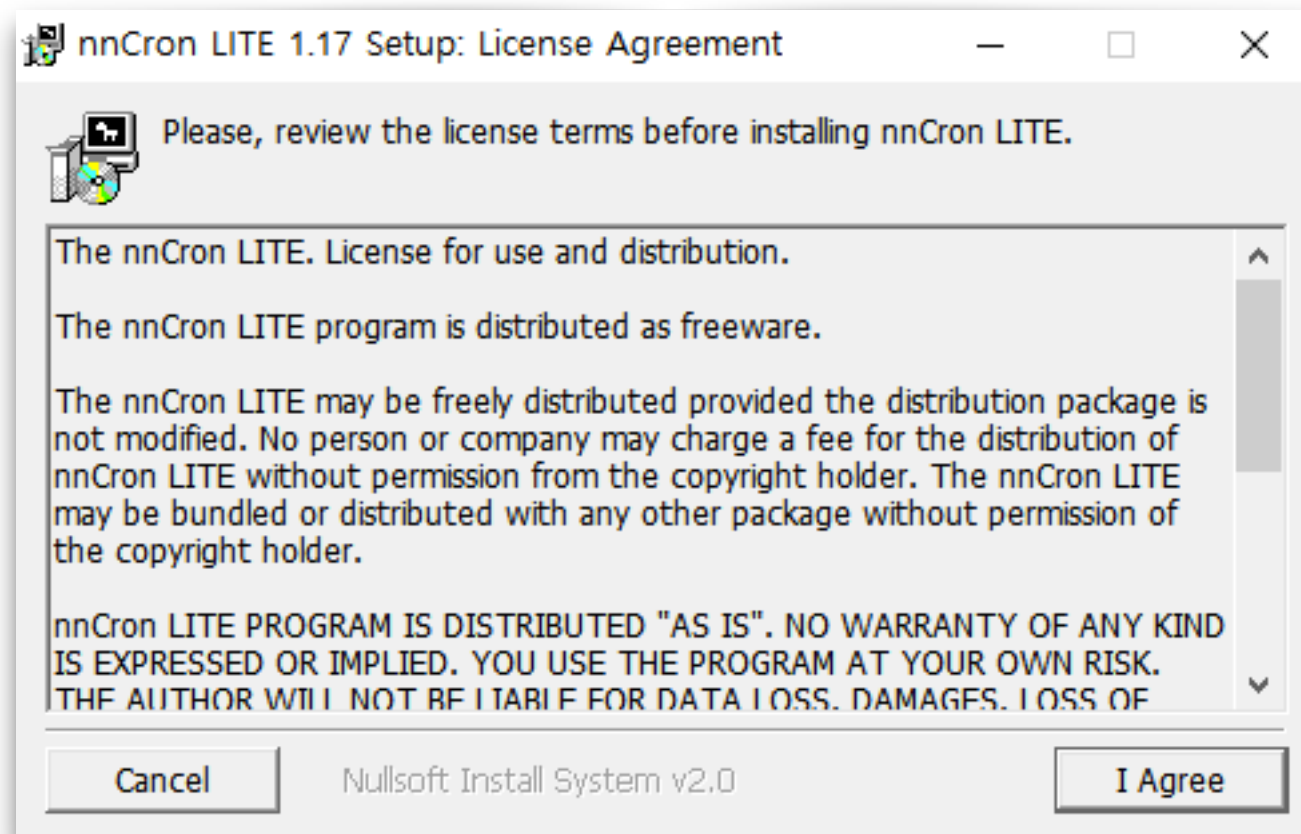
자동화 스케줄링

04



## 설치 가이드

- 윈도우에는 cron 설치가 필요!
- <http://www.nncron.ru/download.shtml> 에서 nncronlt117.exe 다운
- Next만 눌러서 설치하면 완료!



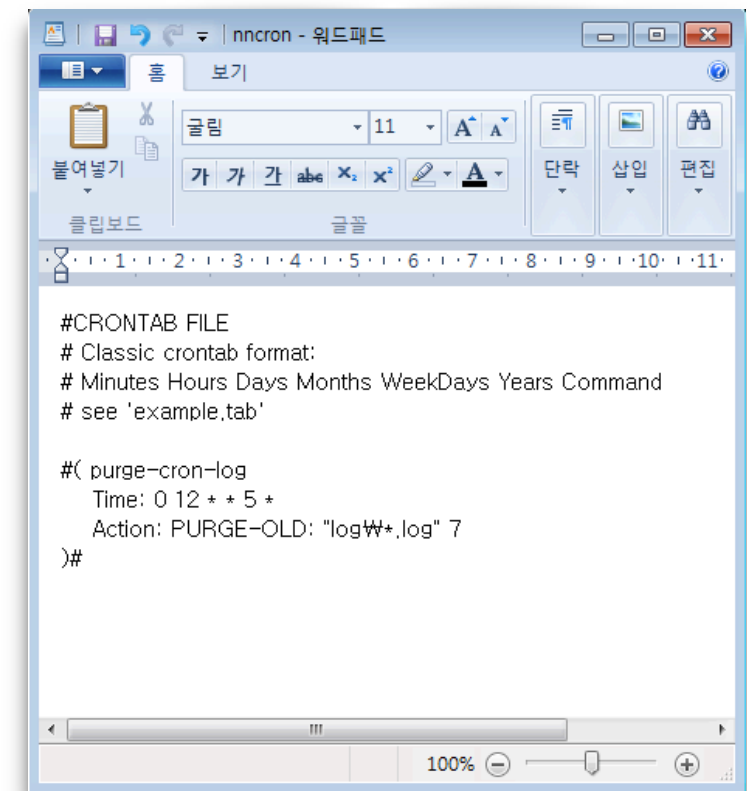
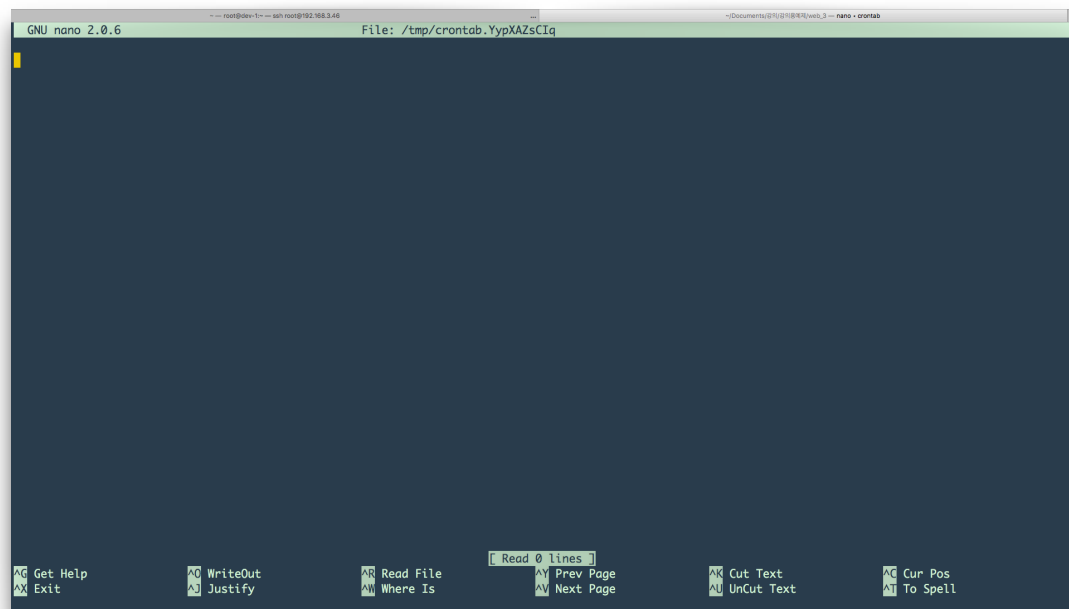
# Cron

자동화 스케줄링

06

## ■ Cron 설정 방법

- 리눅스/맥 사용자는 터미널에서 아래와 같이 입력
  - ▶ `EDITOR=nano crontab -e`
- 윈도우 사용자는 아래 경로의 파일을 관리자권한으로 워드패드 실행
  - ▶ `C:\Program Files (x86)\cron\cron.tab`



리눅스/맥 사용자중 VIM이 익숙한 분은 `crontab -e`만 입력!



# Cron

자동화 스케줄링

07

## ■ Cron 설정 방법

- 분 시 일 월 요일 명령어 형태로 입력
- 띄어쓰기로 구분

```
0 7 * 1 1-6 python /Users/Alghost/test.py
```

분  
시  
일  
월  
요일

실행할 프로그램

# Cron

자동화 스케줄링

08

## Cron 설정 방법

필드	가능한 형태	의미
분, 시, 일, 월	10	매 10분 / 매 10시 / 매 10일 등을 의미
	1,2,3,4,5	매 1,2,3,4,5분 / 매 1,2,3,4,5시 등을 의미
	1-12	매 1,2,3,...,12분 / 매 1,2,3,...,12시 등을 의미
	*	모든 분 / 모든 시 / 모든 일 / 모든 월 등을 의미
요일	0	일요일을 의미 (0~6: 일,월,화,수,목,금,토)
	1-5	월,화,수,목,금 을 의미
	*	모든 요일을 의미
프로그램 실행 부분	cmd / 터미널에서 실행가능한 프로그램	해당 프로그램을 실행

## ■ Cron 설정시 유의

- Cron으로 인해 실행되는 프로그램은 시스템 내부에서 실행
- 따라서, 모든 경로는 절대 경로를 쓰는 습관을 들여야함
- 절대 경로란 파일의 위치를 디스크의 최상위부터 표현한 경로를 의미
  - ▶ Win 예) C:\Users\Alghost\Documents\test.py
  - ▶ Mac 예) /Users/Alghost/Documents/test.py

## ■ Cron 설정 방법 예시

- 월-금 아침 9시에 email.py 를 실행해줘!
- 0 9 \* \* 1-5 python /Users/Alghost/Documents/email.py
- 만약 정상적으로 동작하지 않을 때는?
  - ▶ python 경로도 절대 경로를 사용
  - ▶ which 명령어로 절대 경로를 알아낸 후에 사용
  - ▶ 강의자료 Appendix 참고!

# Cron을 설정해보자

뉴스/파일입출력/이메일/엑셀을 활용한 예제

11

## ■ 복습 + 자동화

- 상품의 링크 리스트를 파일에 저장
- 각 상품들의 가격을 모니터링하여 엑셀에 저장
- 가격이 변동된 경우 이메일 발송
  - ▶ 변동된 가격을 이메일로 발송하기 위해 이메일 예제를 수정
  - ▶ 함수에 내용을 전달할 수 있도록 변경

# Cron을 설정해보자

뉴스/파일입출력/이메일/엑셀을 활용한 예제

12

## ■ 복습 + 자동화

- 상품의 링크 리스트를 파일에 저장
- 각 상품들의 가격을 모니터링하여 엑셀에 저장
- 가격이 변동된 경우 이메일 발송

```
http://www.11st.co.kr/product/SellerProductDetail.tmall?  
method=getSellerProductDetail&prdNo=1510196699&trTypeCd=PW02&trCtgrNo=585021&lCtgr  
No=1001395&mCtgrNo=1002815  
http://www.11st.co.kr/product/SellerProductDetail.tmall?  
method=getSellerProductDetail&prdNo=1229084406&trTypeCd=PW02&trCtgrNo=585021&lCtgr  
No=1001395&mCtgrNo=1002815  
http://www.11st.co.kr/product/SellerProductDetail.tmall?  
method=getSellerProductDetail&prdNo=1724601078&trTypeCd=PW02&trCtgrNo=585021&lCtgr  
No=1001395&mCtgrNo=1002815
```

**products.txt**

```

...
def send_mail(self, name, addr, cc='', hidden_cc='', contents='',
attachment=None):
    ...

    if not contents:
        contents = name + self.cont_layout
    #text = MIMEText(contents)
    # html로 보내고자 할때 사용
    text = MIMEText(contents)
    msg.attach(text)

    # 첨부파일이 있을 경우에 이를 추가하기 위한 코드 수행
    if attachment:
        ...
        # 파일 데이터를 Multipart에 추가
        msg.attach(file_data)

    # 서버와 통신하기 위한 클래스 변수인 smtp를 생성
    smtp = smtplib.SMTP_SSL(SMTP_SERVER, SMTP_PORT)
    # SMTP 서버에 로그인
    smtp.login(SMTP_USER, SMTP_PASSWORD)

    # 모든 수신 대상자를 콤마로 연결
    targets = ','.join((addr, cc, hidden_cc))
    # 메일 전송 (모든 대상자를 리스트형태로 만들어서 발송)
    smtp.sendmail(SMTP_USER, targets.split(','), msg.as_string())
    # 닫기
    smtp.close()

```

```
from selenium import webdriver
from openpyxl import Workbook, load_workbook
from os.path import exists
from datetime import datetime
import string, os

# 절대 경로 사용
# 현재 파일 경로 가져오기
base_dir = os.path.dirname(os.path.realpath(__file__))

# 현재 시간을 가져오는 함수
today = datetime.now()
# 현재 시간으로 파일명을 생성 (년,월,일 정보를 today로 부터 가져옴)
today_file_name = '%d_%d_%d.xlsx'%(today.year,today.month,today.day)
file_path = "%s/%s"%(base_dir,today_file_name)

# 위 파일명으로 파일이 있는지 확인
if exists(file_path):
    # 파일이 있을 경우 load_workbook으로 가져옴
    result_xlsx = load_workbook(file_path)
else:
    # 파일이 없을 경우 새로운 Workbook 클래스 변수 생성
    result_xlsx = Workbook()

# 시트 선택과 최근 가격을 씌
worksheet = result_xlsx.active
worksheet['A2'] = '최근 가격'
```



# Cron을 설정해보자

15

뉴스/파일입출력/이메일을 활용한 예제

## ■ 뉴스 , 지도하기

```
# 크롬 옵션으로 윈도우 크기를 지정하고, 창이 뜨지 않도록 설정
# 기존의 Options 클래스 변수가 아닌 크롬전용옵션 클래스를 사용할수있음
opts = webdriver.ChromeOptions()
opts.add_argument('window-size=1920,1080')
opts.add_argument('headless')
driver = webdriver.Chrome(base_dir+'/chromedriver', chrome_options=opts)
try:
    # 모든 상품 링크를 가져옴
    # readlines함수는 모든 내용을 가져오는데 행단위로 리스트를 만들어 반환
    product_urls = open('products.txt', 'r').readlines()

    # strftime은 시간을 원하는 형태로 표기
    # %H: 시, %M: 분, %S: 초
    row = [today.strftime('%H:%M:%S')]

    # 가격 변화가 생긴 상품에 대한 정보를 추가할 리스트 변수
    changes = []
    # 이 컬럼 인덱스를 통해 각 상품들을 엑셀의 컬럼으로 변환
    # 1 => B, 2 => C, 3 => D ...
    column_idx = 1
```

# Cron을 설정해보자

16

뉴스/파일입출력/이메일/엑셀을 활용한 예제

## 복습 + 자동화

```
for product in product_urls:
    driver.get(product)

    # xpath로 가격을 가져옴
    price_xpath = "//div[contains(@class, 'prdc_default_info')]/strong[@class='sale_price']"
    price_tag = driver.find_element_by_xpath(price_xpath)
    # 가격정보를 row에 추가, 즉 엑셀에 컬럼으로 추가
    row.append(price_tag.text)
```



```
<div class="prdc_default_info no_discount">
  <h3 class="hide">상품가격정보</h3>
  <div class="price_info ">
    <!-- 할인율 -->
    <!-- //할인율 -->
    <span class="price_detail">
      <!-- 정가 S -->
      <!-- //정가 E -->
      <!-- 정상가 S -->
      <!-- //정상가 E -->
      <!-- 판매가 S -->
      <span class="hide">정상가</span>
      <span class="normal_price">...</span>
      <!-- //판매가 E -->
      <!-- 휴대폰 -->
      <!-- //휴대폰 E -->
      <!-- 할인모음가 S -->
      <span class="hide">판매가</span>
      <strong class="sale_price">1,490,000</strong>
    </div>
  </div>
```

# Cron을 설정해보자

17

뉴스/파일입출력/이메일/엑셀을 활용한 예제

## ■ 복습 + 자동화

```
# 상품명 가져옴
title_tag = driver.find_element_by_xpath("//div[contains(@class, 'heading')]/h2")
```

품 5종 증정

접이식 전기 자전거(모야2)



1,49

1,...

```
><div class="brand_shop" style="display:none;" data-
log-actionid-area="atf">...</div>
<!-- //브랜드샵 -->
▼<div class="heading">
  <!-- 상품명 -->
  <h2>모야2S 3초 접이식 휴대용 전기 자전거 어버이날 선물</h2> ==
  <!-- //상품명 -->
  ><div class="add_benefit_ico">...</div>
</div>
<!-- //prdc_heading -->
```

auto.py

# Cron을 설정해보자

18

뉴스/파일입출력/이메일/엑셀을 활용한 예제

## ■ 복습 + 자동화

```
# 컬럼 인덱스를 컬럼 문자로 변경
# ascii_uppercase는 'ABCD...'라서 인덱스로 알파벳 접근 가능
column = string.ascii_uppercase[column_idx]
# 제목을 갱신
worksheet[column+'1'] = title_tag.text

# 이전 가격을 엑셀로부터 가져옴
# 만약 값이 없는 경우 None이 넘어옴
prev_price = worksheet[column+'2'].value
# 최근 가격은 price_tag의 text임
curr_price = price_tag.text
# 이전 가격과 최근 가격이 다른 경우 changes에 튜플로 추가
if prev_price and prev_price != curr_price:
    changes.append((title_tag.text, prev_price, curr_price))

# 최근 상품 금액을 갱신하고 컬럼 인덱스를 증가함
worksheet[column+'2'] = price_tag.text
column_idx += 1
```

# Cron을 설정해보자

19

뉴스/파일입출력/이메일/엑셀을 활용한 예제

```
# 시트의 한 행으로 추가
worksheet.append(row)
# 엑셀 저장
result_xlsx.save(file_path)

# 가격 변동 사항이 있을 경우
if changes:
    from auto_email import Email
    contents = str(today)[-7] + ': 가격이 변동된 상품이 있습니다.\n\n'

    for c in changes:
        contents += '(%s) %s => %s\n'%(c[0], c[1], c[2])

    e = Email()
    e.send_mail('이태화', 'alghost.lee@gmail.com', contents=contents)

except Exception as e:
    print(e)
finally:
    driver.quit()
```

# 입력값을 프로그램에 전달하기

20

자동화 스케줄링에 입력값을 받고 싶다!

## ■ 스케줄링에서 입력값 받기

- 스케줄링은 시스템에서 실행!
- 따라서, 배운 내용으로 입력값을 파일입출력 외엔 방법이 없음
- 간단한 입력을 위해 파일을 굳이 쓸필요가..?
- ‘명령행 인자’ 를 활용!

```
alghost:12th Alghost$ python3 args.py alghost name 123 41234
```

위 그림처럼 프로그램을 실행할 수 있음

# 입력값을 프로그램에 전달하기

21

자동화 스케줄링에 입력값을 받고 싶다!

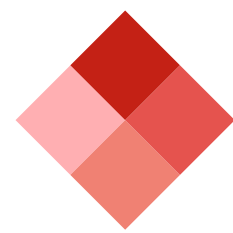
## ■ 스케줄링에서 입력값 받기

- 파이썬의 내부 라이브러리인 sys 활용
- sys.argv 변수에 프로그램 실행시 입력받은 값들이 리스트로 추가됨

```
import sys

print(sys.argv)
```

```
alghost:12th Alghost$ python3 args.py
['args.py']
alghost:12th Alghost$ python3 args.py alghost
['args.py', 'alghost']
alghost:12th Alghost$ python3 args.py alghost.lee@gmail.com
['args.py', 'alghost.lee@gmail.com']
alghost:12th Alghost$ python3 args.py alghost.lee@gmail.com 'this is content!'
['args.py', 'alghost.lee@gmail.com', 'this is content!']
alghost:12th Alghost$ python3 args.py alghost.lee@gmail.com 'this is content!'
123123
['args.py', 'alghost.lee@gmail.com', 'this is content!', '123123']
```



# Good Bye

Thank you

그 동안 잘 따라와 주셔서 감사합니다.  
포럼은 ~2017/12/22 까지 유지합니다.  
(추후 읽기전용으로 변경)

질문에 대한 답변은 48시간 내에 답변해드리겠습니다



# Appendix

cron으로 실행이 안되는 경우

## ■ Windows: cron으로 실행안될 때

- 직접 실행하면 정상 동작하지만 cron으로는 안될 때?
  - ▶ crontab에 한글경로가 들어갈 때 utf-8로 저장되면 안됨
  - ▶ 즉, 워드패드를 이용하여 편집하는 경우에는 문제없음

# Appendix

cron으로 실행이 안되는 경우

## ■ Mac: cron으로 실행안될 때

- 직접 실행하면 정상 동작하지만 cron으로는 안될 때?
  - ▶ 터미널에서 사용하던 파이썬과 다른 버전이 실행될 때 발생
  - ▶ 아래와 같이 python의 절대경로를 구해서 실행하면 됨!
    - \$ which python  
경로가 나옴
    - 해당 경로를 crontab에 추가

```
taehwui-MacBook-Pro:10th Alghost$ which python  
/Library/Frameworks/Python.framework/Versions/2.7/bin/python
```

```
1 * * * * * /Library/Frameworks/Python.framework/Versions/2.7/bin/python /Users/Alghost/python/4th.class/10th/auto2.py
```