

# Python 으로 만들어보는 업무자동화

조민성

**BACK TO US,  
BACK TO PYTHON**

---

# Table of Contents

- 들어가며
- 자동화... ?
- Let's Do This!
  - Problem Solving
  - Planning
  - Python
- 그 외의 하고 싶은 이야기
  - selenium 의 변화
  - WA
- 마치며

# The technical level for this session

---

Lv. beginner



# 환경설정

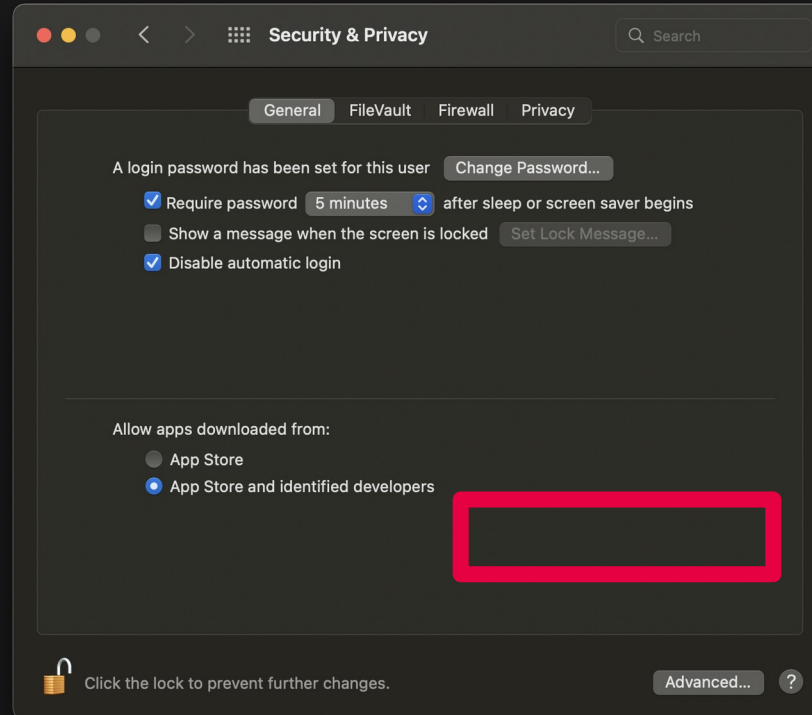
---

```
# pyenv  
$ curl https://pyenv.run | bash  
$ pyenv install 3.11.5
```

```
# venv  
$ cd <source-folder>  
$ python3 -m venv .venv  
$ . .venv/bin/activate
```

```
# pip install  
$ pip install requests  
$ pip install bs4  
$ pip install selenium  
$ pip install pandas
```

# Only in Mac OS



# 반복 업무 어떻게 하시나요?

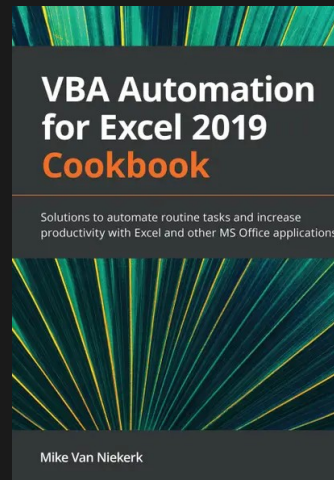
고수



매크로 (키 / 마)



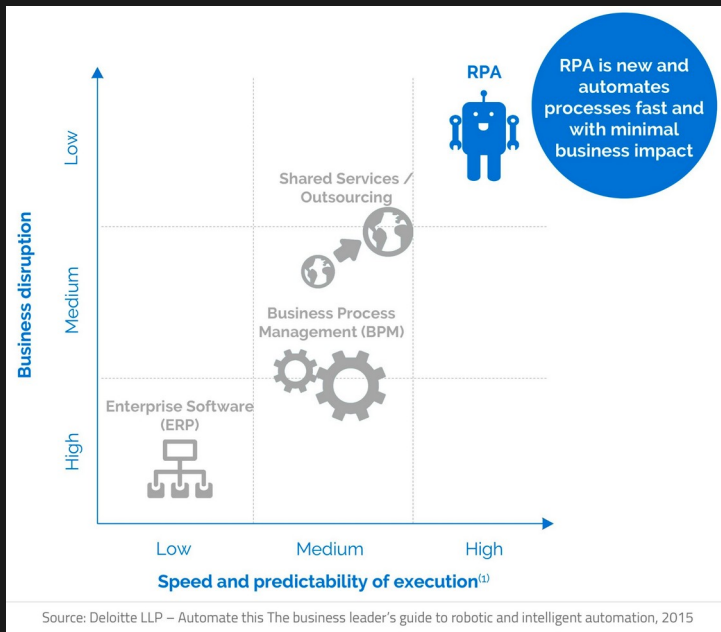
VBA + Excel



Code  
(with Python)



# RPA(Robot Process Automation)



- 자주 반복되거나, 계속 진행되거나 예측 가능한 태스크와 프로세스에 적용 (규칙성이 있는 단순 반복, BI 작성 등)
- 일반적인 RPA 개념과 국내의 RPA에 대한 인식
- 국내의 경우, ERP/BPM과의 연동을 고려한 GUI 관리툴 환경에도 방점을 두는 추세

## ‘전체 상품 목록을 정리해주지 않을까?’

어느\_날  
갑자기  
**미션**이  
찾아왔다



대기업 MZ세대 ‘요요요’ 주의보

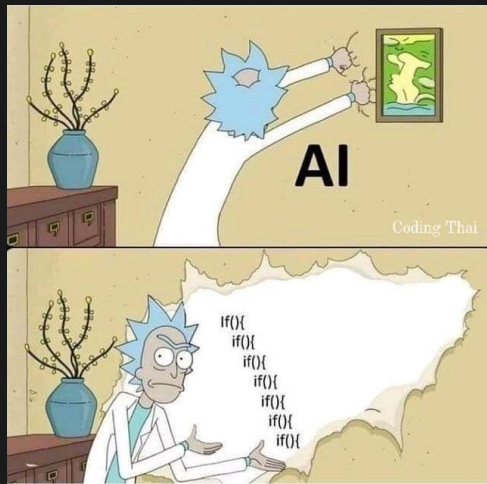


- 이걸‘요’?** 지시 받은 업무의 정확한 내용과 목적에 대한 설명 요구
- 제가‘요’?** 많은 임직원 중 해당 업무를 수행해야 하는 사람이 왜 자신 인지 설명 요구
- 왜‘요’?** 해당 업무를 해야 하는 이유와 필요성, 기대 효과 등에 대한 설명 요구

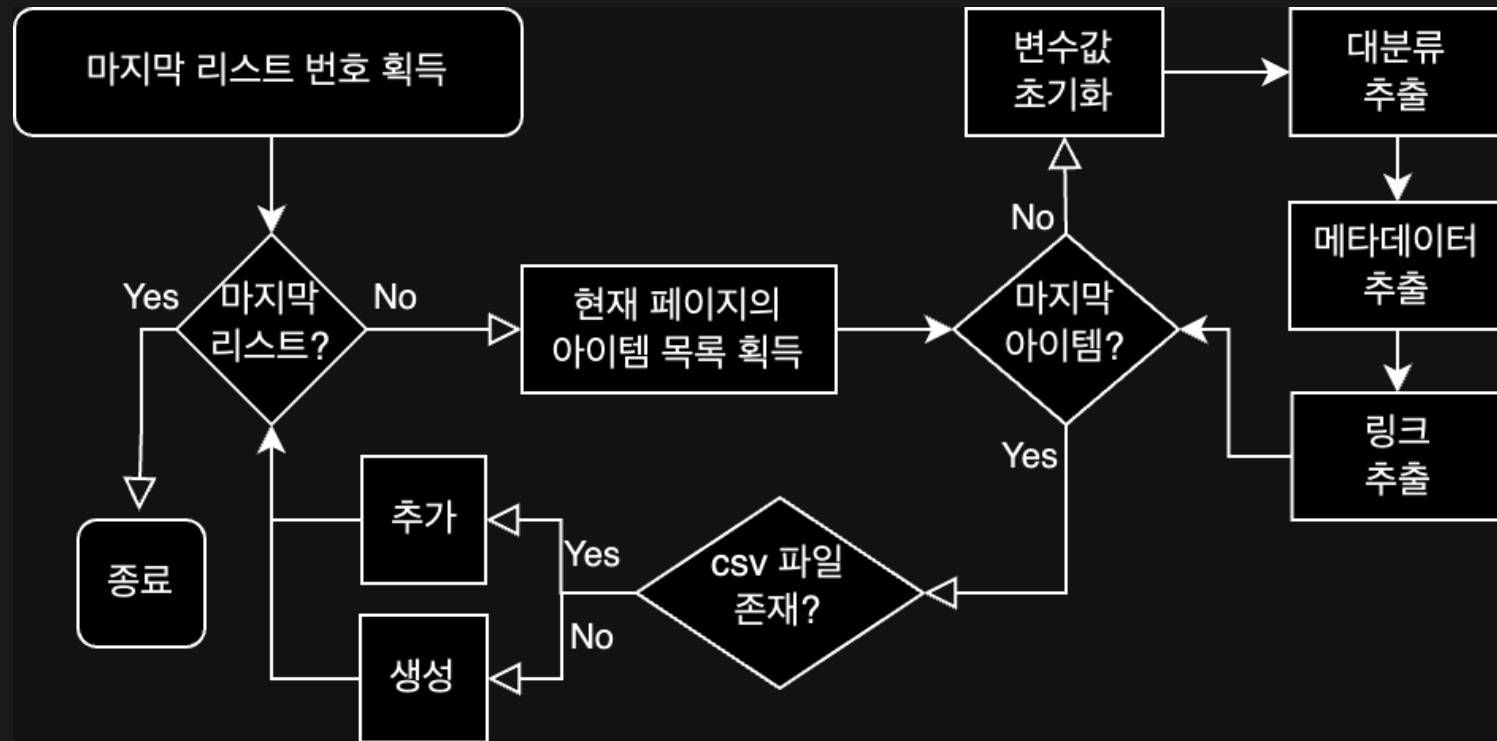


## 패턴 파악 및 결과물 구상

- 게시판 구조를 통해 , 새로 생성할 메타데이터셋 지정
  - 상품 대분류 / 공급자 정보 / 태그 / 등록일 / 파일 / 상품 설명
- 예외 처리
  - 대분류에 따라 , 파일 유무가 결정되는 경우
  - 파일이 있다 하더라도 , 수량이 변동되는 경우
- I/O 속도 : csv 포맷 선정



## 로직 구성



## ( 파이써닉 하지 않은 ) 코드 작성

```
59 driver.get(URL)
60 # last_page_number
61 # = int(re.sub(r'^0-9', '', driver.find_element_by_class_name('current_next').get_a
62 last_page_element = driver.find_element(By.CLASS_NAME, 'current_next')
63 last_page_href = last_page_element.get_attribute('href')
64 last_page_number = int(re.sub(r'^0-9', '', last_page_href))
65
66 print(last_page_number, "&&", type(last_page_number))
67
68 for i in range(0, last_page_number):
69     pageIndex = i + 1
70     driver.get(URL + URL_PAGE + str(pageIndex))
71     html = driver.page_source
72     soup = BeautifulSoup(html, 'html.parser')
73     item_list = soup.find_all('div', 'product_list row mt30 pb20 br_line_bottom')
74     item_count = len(item_list)
75     print('==== 현재 $$ ' + str(pageIndex) + " $$ 페이지 item 갯수 : " + str(item_count) +
76         for j in range(0, item_count):
77             item_index = j + 1
78             # 0. 초기화
```

# if 문, 쿼리문, 리스트 파싱의 연속

```

82 # 1. 대분류 (파일/API/가공서비스) 처리 [+속성]
83 # document.querySelector("#listType > div:nth-child(1) > div.c
84 item_query = '#listType > div:nth-child(' + str(item_index) + '
85 query_section = soup.select(item_query)
86 col_section = query_section[0].text.strip()
87
88 # 2. 분류별 제목, 제공업체명, 등록일 추출
89 # document.querySelector("#listType > div:nth-child(1) > div.c
90 item_query = '#listType > div:nth-child(' + str(item_index) + '
91 if col_section == "가공":
92     # 제목은 일괄처리
93     col_item = "가공서비스 제공업체"
94     # 제공업체명
95     col_company = soup.select(item_query)[0].text.strip()
96     # 등록일
97     # document.querySelector("#listType > div:nth-child(4) > div
98     item_query = '#listType > div:nth-child(' + str(item_index)
99     col_reg_date = soup.select(item_query)[0].text.strip()
100 else:
101     # 제목
102     col_item = soup.select(item_query)[0].text.strip()
103     # 제공업체명
104     # document.querySelector("#listType > div:nth-child(1) > div
105     item_query = '#listType > div:nth-child(' + str(item_index)
106     col_company = getCompany(soup.select(item_query)[0].contents
107     # 등록일
108     # document.querySelector("#listType > div:nth-child(3) > div
109     item_query = '#listType > div:nth-child(' + str(item_index)
110     col_reg_date = soup.select(item_query)[0].text.strip()

```

```

112 # 3. 속성추출 (from. 1번의 query_section)
113 item_query = '#listType > div:nth-child(' + str(item_index) + ' > (
114 query_section = soup.select(item_query)
115 col_attribute = getAttribute(query_section)
116 # 3-1. 속성 리스트 String으로 임시변환
117 col_attribute = ', '.join(col_attribute)
118
119 # 4. 단가표 [+샘플파일] (링크로 제공함)
120 if col_section == "가공":
121     # 단가표
122     item_query = '#listType > div:nth-child(' + str(item_index) + ' > (
123     if (soup.select(item_query)):
124         query_table = soup.select(item_query)[0]['onclick'].split('\')
125         col_table = URL_FILE
126         for i in query_table:
127             if '=' in i:
128                 col_table += i.replace('=', '%3D')
129                 col_table += '&seqno=1'
130     else:
131         col_table = "상세조회참조"
132     # 샘플파일은 없음
133     col_sample = NULL
134 else:
135     # 단가표
136     # document.querySelector("#listType > div:nth-child(4) > div.summ
137     item_query = '#listType > div:nth-child(' + str(item_index) + ' > (
138     if (soup.select(item_query)):
139         query_table = soup.select(item_query)[0]['onclick'].split('\')
140         col_table = URL_FILE
141         for i in query_table:
142             if '=' in i:
143                 col_table += i.replace('=', '%3D')
144                 col_table += '&seqno=1'
145                 break
146     else:
147         col_table = NULL

```

## 쿼리를 위한 별도 함수 작성

```
28 def getAttribute(List):
29     result = []
30     j = 0
31     for i in List:
32         if j == 0:
33             j += 1
34             continue
35         else:
36             j += 1
37         result.append(i.text.strip())
38     return result

20 def getCompany(List):
21     temp_string = []
22     for i in List:
23         temp_string = str(i).strip()
24         if len(temp_string) == 0: continue
25         elif 'span' in temp_string: continue
26         else: return temp_string
```

## 판다스를 통한 , 테이블 작성 및 파일 입출력

```
181 # 6. 건별 저장
182 data = [col_section, col_company, col_item, col_attribute, col_table, col_sample, col_
183
184 df = pd.DataFrame(data)
185 df = df.transpose()
186 # df.rename(columns=df.iloc[0], inplace=True)
187 # df = df.drop(df.index[0])
188 df.columns = ['분류', '기업명', '상품명', '태그', '가격정책_단가표', '샘플파일', '등록일', '설명']
189 if file_flag == 1 & True != os.path.exists(file_datetime):
190     df.to_csv(file_datetime, index=False, mode='w', encoding='utf-8-sig')
191     file_flag = 0
192     print("====Write OKAY====")
193 else:
194     df.to_csv(file_datetime, index=False, mode='a', header=False, encoding='utf-8-sig')
195     print("====Append OKAY====")
```

## 셀레니움 4 변경사항

```
2 from bs4 import BeautifulSoup
3 from selenium import webdriver
```

```
41 # driver = webdriver.Chrome('.\\chromedriver_win32\\chromedriver.exe')
42 driver = webdriver.Chrome()
```

Finding a multiple elements with `findElements*`

Before

```
driver.findElementsByClassName("className");
driver.findElementsByCssSelector(".className");
driver.findElementsById("elementId");
driver.findElementsByLinkText("linkText");
driver.findElementsByName("elementName");
driver.findElementsByPartialLinkText("partialText");
driver.findElementsByTagName("elementTagName");
driver.findElementsByXPath("xPath");
```

After

```
driver.findElements(By.className("className"));
driver.findElements(By.cssSelector(".className"));
driver.findElements(By.id("elementId"));
driver.findElements(By.linkText("linkText"));
driver.findElements(By.name("elementName"));
driver.findElements(By.partialLinkText("partialText"));
driver.findElements(By.tagName("elementTagName"));
driver.findElements(By.xpath("xPath"));
```



# Web Accessibility (WA/ 웹 접근성)



```
▼<div class="product_list row mt30 pb20 br_line_bottom"> flex == $0
  ▼<div class="thumbnail">
    <!-- BEFORE_WA인증_HSH_20230110 -->
    <!-- AFTER_WA인증_HSH_20230110 -->
    ▶<span>...</span>
  </div>
  ▼<div class="contents">
    <!-- BEFORE_WA인증_HSH_20230110 -->
    <!-- AFTER_WA인증_HSH_20230110 -->
    ▶<div class="mb5">...</div>
    ▶<div class="gr_explanation">...</div>
  </div>
```

```
▼<div class="summary">
  ▼<ul class="summary_list charged_service"> flex
    ▼<li class="summary_item"> flex
      <!-- BEFORE_WA인증_HSH_20230110 -->
      <!-- AFTER_WA인증_HSH_20230110 -->
      ▶<p class="gr_view mr20 " role="img">...</p>
      ▶<p class="gr_good mr-auto" role="img">...</p>
      <a class="btn_interested_product" href="#" onclick style="pos
    </li>
    ▼<li class="summary_item"> flex
      "확장자: "
      <span class="ml5 mr-auto text_green">YLS/YLSX</span>
    <!-- WA인증_HSH_20230111_레이어 팝업 초점 이동 처리 -->
```



## In this session, we see...

---

- How easy to make automation about repeated task with Python!
  - How to planning for solving tasks
  - How to code it
  - How to use it
- Saw why maintenance with new version is important
  - High performance
  - Compatiancy with new machines
  - Governance (WA)
- And OSS
  - Contribution(code, i10n, organizing... etc.)

## Speaker

---



조민성 (Minseong Cho)

[mscho7969@ubuntu.com](mailto:mscho7969@ubuntu.com)

Former DevOps engineer at Metaflyer



Ubuntu Korea Community



[blog.minseong.xyz](http://blog.minseong.xyz)

# Source Code :

[github.com/kkumtree/first\\_web\\_crawler](https://github.com/kkumtree/first_web_crawler)

# UbuCon Korea

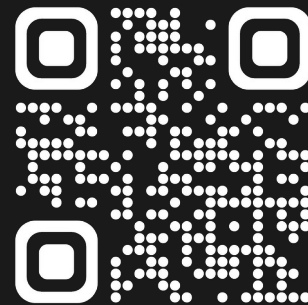
---



# Ubuntu 23.10 Release party in Busan



11 월 18 일 ( 토 )  
토즈 서면점



[discourse.ubuntu-kr.org](https://discourse.ubuntu-kr.org)

[contact@ubuntu-kr.org](mailto:contact@ubuntu-kr.org)

