



# 외부연동 데이터 수집 (웹 크롤링 활용 데이터 수집) with Selenium

김효관 |

# 외부연동 데이터 수집

## 단원 개요

### [단원명]

외부연동 데이터 수집

### [단원 소개]

- 데이터 분석 시 회사 내부에서 접근할 수 있는 데이터 (파일, 데이터베이스) 만으로는 한계가 있다. 외부에서 활용가능한 포털 내 데이터 나 공개한 데이터를 수집하는 방법을 익히고 다양한 각도로 분석 모델을 만들어가는 방법을 학습한다.

### [교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
웹 크롤링 활용 데이터 수집	<ul style="list-style-type: none"><li>- 외부 데이터 수집의 필요성을 이해한다.</li><li>- 웹 포털 내 존재하는 데이터 수집 방법을 실습한다.</li><li>- 브라우저 자동화 라이브러리를 활용한 데이터 수집 자동화 방법을 실습한다.</li></ul>
공공데이터 포털 데이터 수집	<ul style="list-style-type: none"><li>- 공공데이터 포털을 이해한다.</li><li>- 공개된 파일 형태의 자료 수집 방법을 실습한다.</li><li>- API 형태로 공개한 JSON 포맷 데이터 수집 방법을 실습한다.</li><li>- API 형태로 공개한 XML 포맷 데이터 수집 방법을 실습한다.</li></ul>

## 단원 개요

### 웹 크롤링



	0	1
0	더마시나 무항생제 구운계란60구, 1개, 2,100g	10,300
1	오복유통 HACCP인증 구운계란 2판60구, 60구, 2판	13,900
2	잡나무촌 무염훈제계란, 30개입, 1.2kg(한판)	6,200
3	꾼란 맥반석 구운계란 30구 1판, 30개입, 1.2kg	7,900
4	맛군 축축 톡톡 구운 계란, 30알, 1박스	7,900
5	감동란 간이베어 있는 축축한 반숙계란, 50g, 30개입	16,900
6	[계란사랑] 맥반석 구운계란 구운란 60구 (2판), 2700g	11,900
7	진주형 오마이 포켓 메주리알 5p, 25g, 10개입	9,440

### 공공데이터 수집

연도	월	전국 PIR	서울 PIR	부산 PIR	대구 PIR	인천 PIR	광주 PIR	대전 PIR	울산 PIR	대기오염 나쁨 위치		관측소위치	
										0	중구	서울특별시 중구 덕수궁길 15시청서소문별관 3동	
0	2004	3	4.21	4.89	3.95	3.73	4.65	2.81	4.68	2.66	1	청계천로	서울 중구 청계천로 184(청계천4가사거리 남강빌딩 앞)
1	2004	4	4.39	5.59	3.91	3.88	4.59	2.92	3.83	2.74	2	용산구	서울 용산구 한남대로 136서울특별시중부기술교육원
2	2004	5	4.19	5.14	4.90	3.83	4.78	3.41	4.19	2.93	3	강변북로	서울 성동구 강변북로 257한강사업본부 옆
3	2004	6	4.09	4.38	4.20	3.77	4.30	2.83	4.19	2.81	4	홍릉로	서울 동대문구 홍릉로 1(청량리전철역 사거리 SC제일은행 앞)

교육목표: 웹 데이터를 수집하는 방법을 익힌다.

# CONTENTS

1

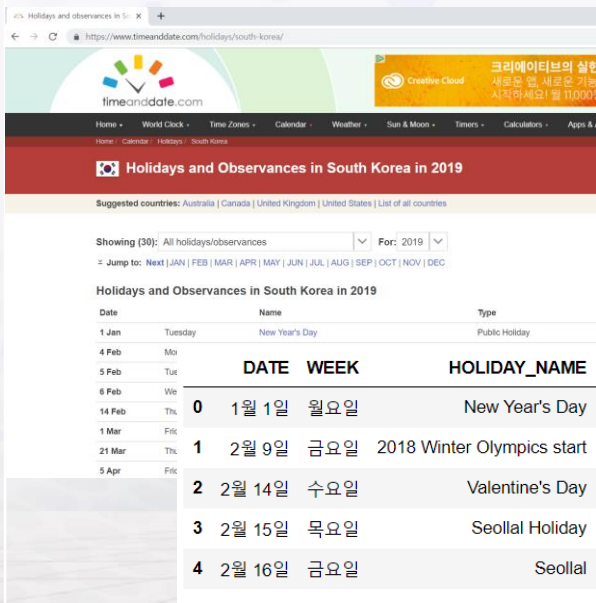
셀레니움 활용 웹브라우저 자동화



# 외부연동 데이터 수집 파트1 (웹 크롤링)

## 모듈 개요

### 웹 크롤링



timeanddate.com

Holidays and Observances in South Korea in 2019

Suggested countries: Australia | Canada | United Kingdom | United States | List of all countries

Showing (30): All holidays/observances For: 2019

Jump to: Next | JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC

Holidays and Observances in South Korea in 2019

Date	Name	Type
1 Jan	Tuesday	New Year's Day
4 Feb	Mo	
5 Feb	Tue	
6 Feb	We	
14 Feb	Th	
1 Mar	Fri	
21 Mar	Th	
5 Apr	Fri	

	DATE	WEEK	HOLIDAY_NAME	HOLIDAY_TYPE
0	1월 1일	월요일	New Year's Day	Public Holiday
1	2월 9일	금요일	2018 Winter Olympics start	Observance
2	2월 14일	수요일	Valentine's Day	Observance
3	2월 15일	목요일	Seollal Holiday	Public Holiday
4	2월 16일	금요일	Seollal	Public Holiday



coupang

계란/알류/가공란 (2,118)

0 1

0	더마시나 무항생제 구운계란60구, 1개, 2,100g	10,300
1	오복유용 HACCP인증 구운계란 2판60구, 60구, 2판	13,900
2	참나무촌 무염혼제계란, 30개입, 1.2kg(한판)	6,200
3	곤란 맥반석 구운계란 30구 1판, 30개입, 1.2kg	7,900
4	맛군 축족 풀깃 구운 계란, 30알, 1박스	7,900
5	감동란 간이버어 있는 축족한 반숙계란, 50g, 30개입	16,900
6	[계란사랑] 맥반석 구운계란 구운란 60구 (2판), 2700g	11,900
7	진주형 오마이 포켓 메추리알 5p, 25g, 10개입	9,440

# 리마인드. 웹 크롤링 이해 및 기본기 살펴보기

# 1. 웹 크롤링 이해 및 기본기 살펴보기

## 모듈 개요

### [과정개요]

웹 크롤링 이해 및 기본기 살펴보기

### [교육목표]

- 웹 크롤링 의 필요성을 이해합니다.
- 웹크롤링을 위한 기본적인 문법 및 데이터프레임 생성방법을 실습을 통해 리마인드 합니다.

### [교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
웹 크롤링 이해 및 기본기 살펴보기	<ul style="list-style-type: none"><li>- 웹 크롤링이 인공지능 영역에 왜 필요한지 이해합니다.</li><li>- 기본적인 문법 및 데이터프레임 생성 방법을 실습합니다.</li></ul>

# 1. 웹 크롤링 이해 및 기본기 살펴보기

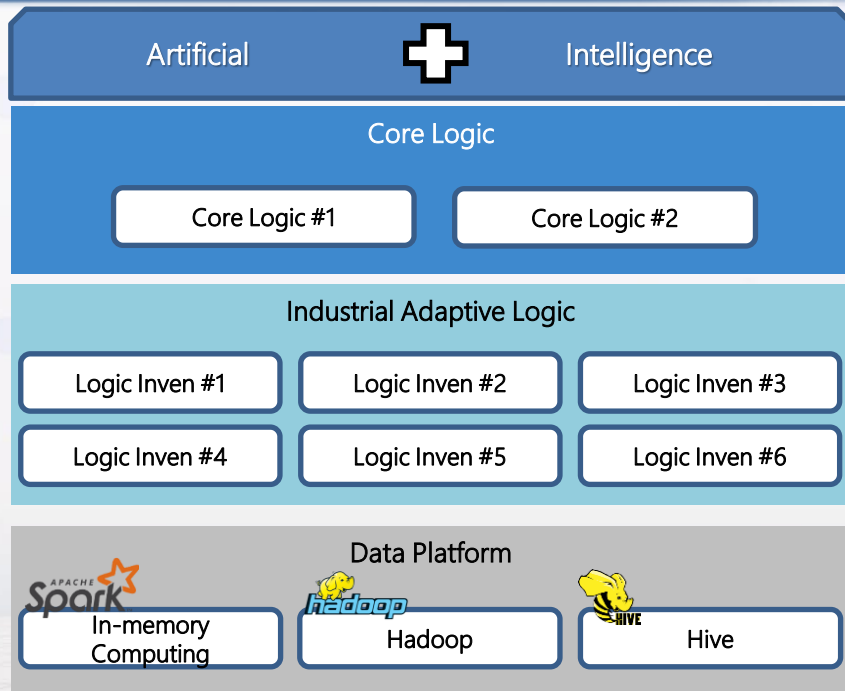
우리 회사에 필요한 가치를 창출하기 위해 구성도를 그린다...

분석 데이터

내부 데이터



이미 사내에서  
접속 가능한  
정제 후 활용 가능 데이터



시각화 / 웹 시연





# 1. 웹 크롤링 이해 및 기본기 살펴보기

시간이 지나자... 조금 한계에 ...



가지고 있는 걸로 분석 해보자..

1일...

2일...

3일...

하아... 가지고 있는 데이터만으로는 다양한 각도로 분석을 할 수 없네.....

# 1. 웹 크롤링 이해 및 기본기 살펴보기

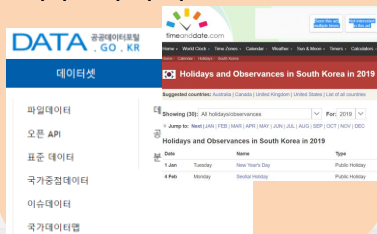
외부에 접속하여 필요한 정보들을 수집해보자!

분석 데이터

내부 데이터



외부 데이터



Analytics



Intelligence

Core Logic

Core Logic #1

Core Logic #2

Industrial Adaptive Logic

Logic Inven #1

Logic Inven #2

Logic Inven #3

Logic Inven #4

Logic Inven #5

Logic Inven #6

Data Platform



In-memory  
Computing



Hadoop



Hive

Cloud Environment

시각화 / 웹 시연



4차산업혁명 단계별로

www.youtube.com/hkcode

# 1. 웹 크롤링 이해 및 기본기 살펴보기

데이터가 많아지니 분석해볼만한게 많아지네!

## 웹 상에 공개된 데이터 스크랩 후 활용

접속 사이트	제공항목(데이터셋)
국가통계포털	코스닥지수
국가통계포털	코스닥 150 지수
국가통계포털	코스닥 주가이익비율 (PER)
국가통계포털	코스닥 주가순자산비율 (PBR)
국가통계포털	코스닥 배당수익률
국가통계포털	코스닥 산업별 투자지표
공공데이터포털	한국감정원 오피스텔 동향조사 현황
공공데이터포털	한국감정원 부동산 매매가격지수 현황
공공데이터포털	공간융합정보

## 내부 데이터 정제 후 활용



## 데이터 탐색 후 필요 데이터 직접 스크랩 후 활용



# 파트1. 셀레니움 활용 웹브라우저 자동화

# 5. 셀레니움 활용 웹브라우저 자동화

## 모듈 개요

### [과정개요]

셀레니움 활용 웹브라우저 자동화

### [교육목표]

- 브라우저를 자동으로 동작 시키는 방법을 실습합니다.

### [교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
셀레니움 활용 웹브라우저 자동화	<ul style="list-style-type: none"><li>- 셀레니움 라이브러리 환경 구축방법을 익힙니다.</li><li>- 셀레니움 활용 웹 브라우저 자동화방법을 익힙니다.</li></ul>

# 5. 셀레니움 활용 웹브라우저 자동화

## selenium 라이브러리

1-1

웹 브라우저 테스트 자동화 라이브러리

```
pip install selenium  
pip install webdriver_manager
```



1-2

크롬 등 웹 엔진 드라이버 활용

(크롬드라이버 다운로드 링크)

<https://developer.chrome.com/docs/chromedriver/downloads?hl=ko>

홈 > Docs > ChromeDriver

도움이 되었나요?  

## 다운로드



최신 버전



경고:

- Chrome 버전 115 이상을 사용하는 경우 [Chrome for Testing](#) 사용 가능 여부 대시보드를 참고하세요. 이 페이지에서는 특정 ChromeDriver 버전을 편리하게 다운로드할 수 있는 [JSON 엔드포인트](#)를 제공합니다.
- 더 낮은 버전의 Chrome은 아래에서 지원되는 ChromeDriver 버전을 참고하세요.

## 5. 셀레니움 활용 웹브라우저 자동화 (참조)

### 1. 라이브러리 선언 및 드라이버 설정 (자동)

#### # 라이브러리 선언

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager
```

패키지 설치 `pip install selenium`

```
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
```

#### # 드라이버 위치 설정

```
def setChromeDriver():
    options = Options()
    user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75 Safari/'
    options.add_argument('user-agent=' + user_agent)
    # options.add_argument('--headless') # 웹 브라우저를 시각적으로 띄우지 않는 headless chrome 옵션
    driver = webdriver.Chrome(service=Service(executable_path=ChromeDriverManager().install()), options=options)
    return driver
```

<https://www.selenium.dev/selenium/docs/api/rb/Selenium/WebDriver/Chrome/Options.html>

```
driver = setChromeDriver()
```

#### # 웹페이지 파싱 될때까지 최대 3초 기다림

```
driver.implicitly_wait(3)
```

크롬드라이버 다운로드 관련글 아래 참고

<https://velog.io/@lcs3947/selenium-driver-%EA%B4%80%EB%A6%AC-%EB%B0%8F-%EC%9C%A0%EC%9A%A9%ED%95%9C-%EA%B8%B0%EB%8A%A5>

4차산업혁명 단계별로 익히는 빅데이터&인공지능(광문각, 김효관 교수)

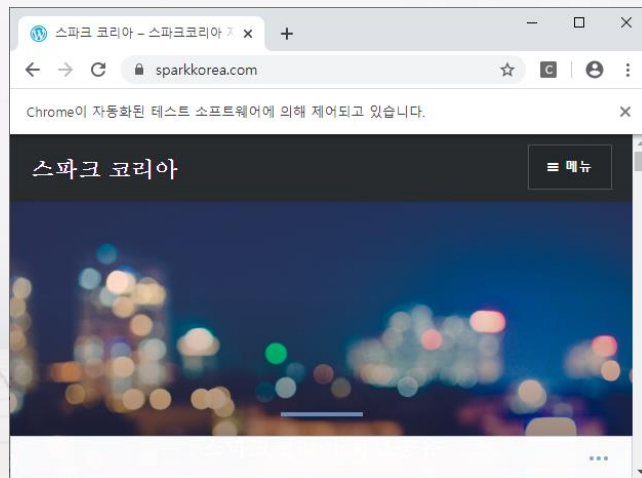
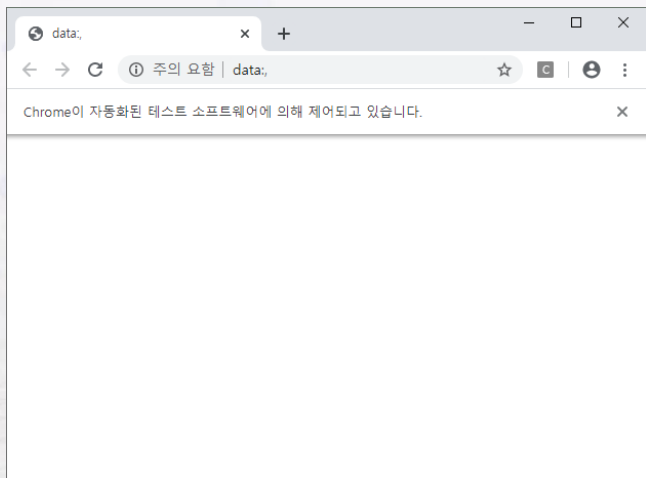
[www.youtube.com/hkcode](http://www.youtube.com/hkcode)

# 5. 셀레니움 활용 웹브라우저 자동화

## 2. 웹페이지 이동

이동하고자 하는 주소!

`driver.get("url주소")`





# 5. 셀레니움 활용 웹브라우저 자동화

## 2. 웹페이지 이동

```
driver.get("url주소")
```

### Example

# URL 정의

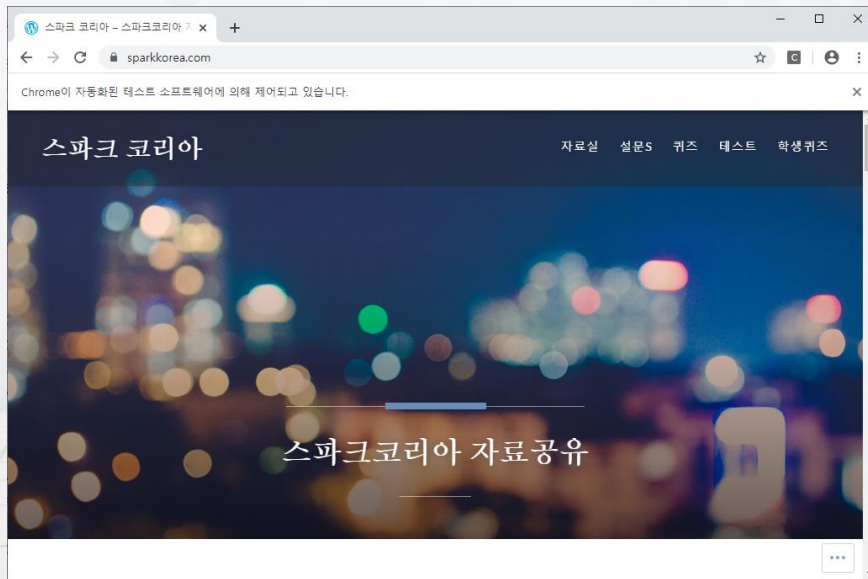
```
baseUrl = "https://sparkkorea.com"
```

# URL 이동

```
driver.get(baseUrl)
```

# 현재 URL 정보

```
driver.current_url
```



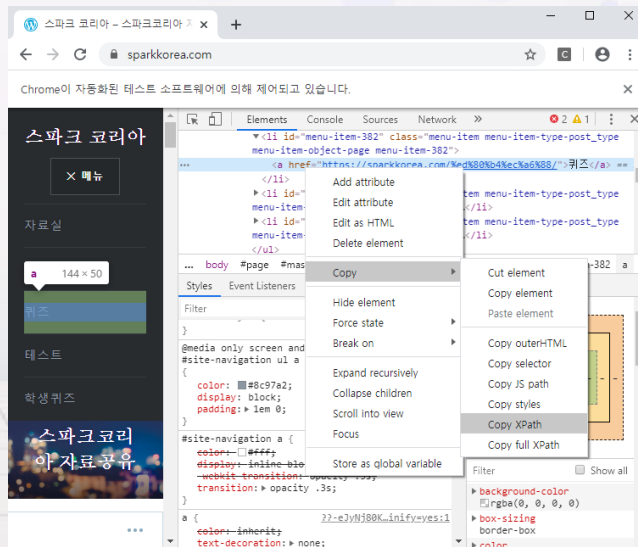
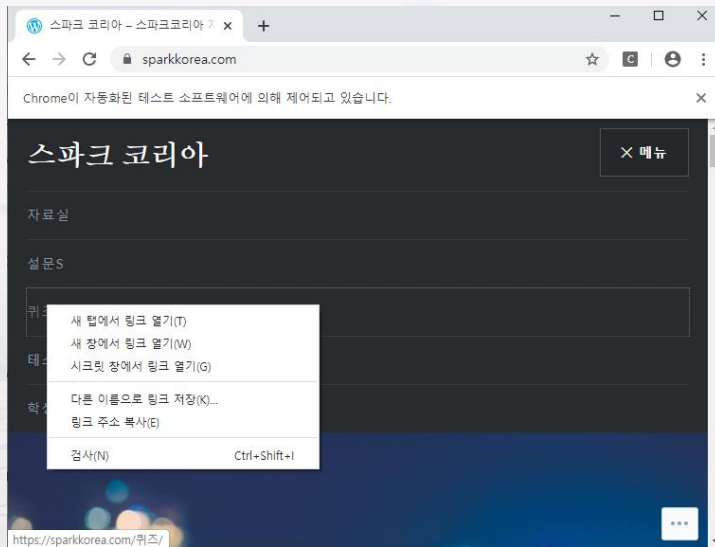
# 5. 셀레니움 활용 웹브라우저 자동화

## 3. 액션대상 요소 탐색

원하는 요소에서 우클릭 후 "검사" 이후 소스코드에서 우클릭 후 Copy XPath (클래스 접근 시 앞에 . 공백.)

### Example

퀴즈 메뉴 버튼 -> xpath = '//\*[@id="menu-item-382"]/a'



## 5. 셀레니움 활용 웹브라우저 자동화

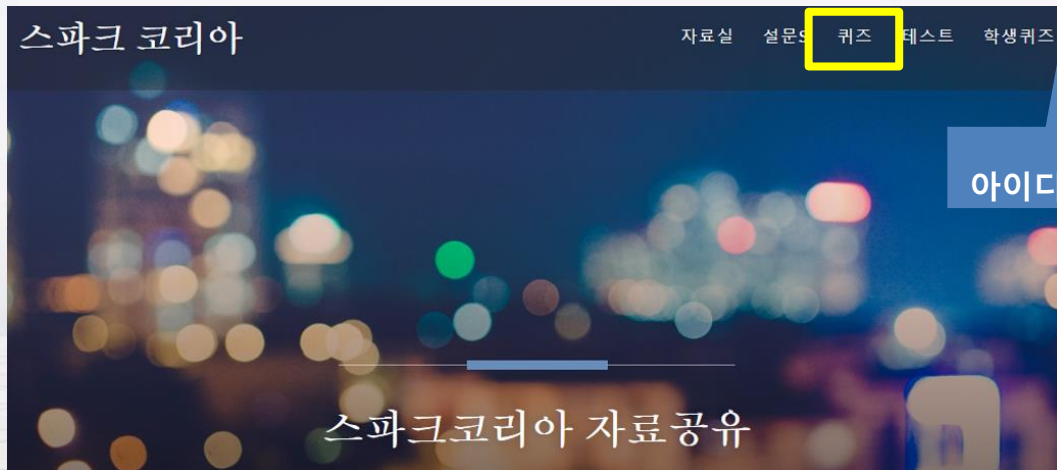
### 4. 액션 적용하기 (클릭)

from selenium.webdriver.common.by import By

요소

액션대상 XPath

```
driver.find_element(By.XPATH, "XPath").click()
```



Selector 선택 후  
아이디(#) 또는 클래스(.) 로 접근 가능

# 5. 셀레니움 활용 웹브라우저 자동화

## 4. 액션 적용하기 (클릭)

버튼클릭: `driver.find_element(By.XPATH, ' XPATH ').click`

### Example

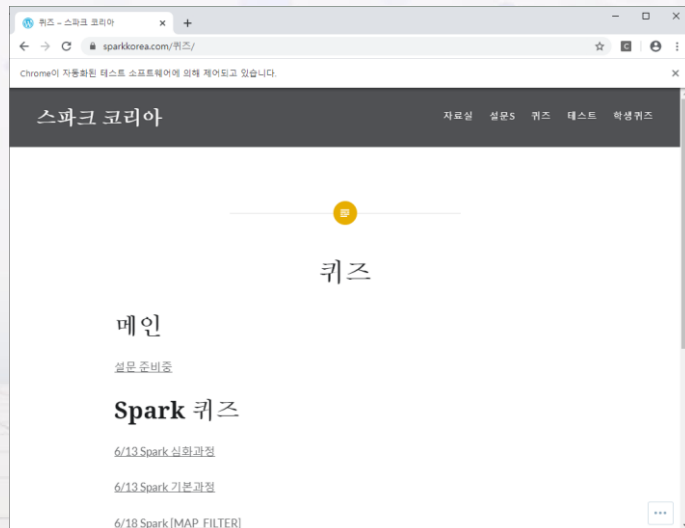
#### # URL 정의

```
sparkUrl = 'https://sparkkorea.com'
```

```
driver.get(sparkUrl)
```

```
quizBtnXpath = '//*[@id="menu-item-382"]/a'  
driver.find_element(By.XPATH, quizBtnXpath).click()
```

```
from selenium.webdriver.common.by import By
```



## 5. 셀레니움 활용 웹브라우저 자동화

### 4. 액션 적용하기 (키 입력)

액션대상 XPath

```
driver.find_element(By.XPATH, "XPath").sendKeys("입력키")
```



```
//*[@id="APjFqb"]
```

```
from selenium.webdriver.common.keys import Keys
```

## 5. 셀레니움 활용 웹브라우저 자동화

### 4. 액션 적용하기 (키 입력)

```
from selenium.webdriver.common.keys import Keys
문자입력: driver.find_element(By.XPATH, ' XPATH ').send_key( 키입력 )
```

#### Example

##### # URL 정의

```
googleUrl = 'https://www.google.co.kr'
```

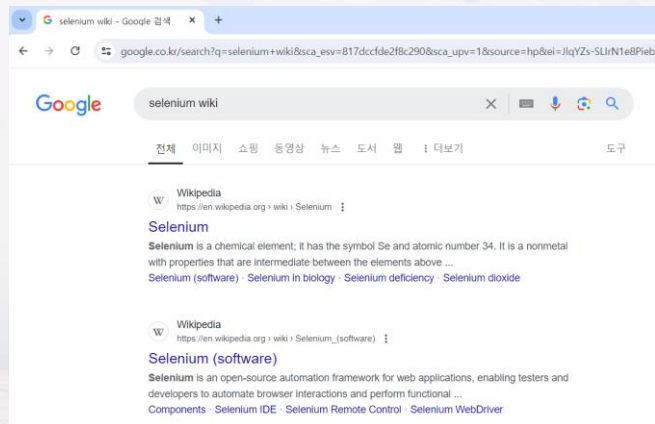
##### # URL 이동

```
driver.get(googleUrl)
```

##### # 요소 탐색

```
searchPath = '//*[@id="APjFqb"]'
driver.find_element(By.XPATH, searchPath).send_keys("selenium wiki")
driver.find_element(By.XPATH, searchPath).send_keys(Keys.ENTER)
```

<https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.keys.html>



## 5. 셀레니움 활용 웹브라우저 자동화

### 5. 웹페이지 소스 가져오기

페이지 소스 가져오기 : driver.page\_source

#### Example

# URL 정의 의 스파크코리아 퀴즈 사이트 이동

sparkUrl = 'https://sparkkorea.com'

driver.get(sparkUrl)

quizBtnXPath = '//\*[@id="menu-item-382"]/a'

driver.find\_element(By.XPATH, quizBtnXPath).click

# 페이지 소스 가져오기 전 1초 대기

import time

time.sleep(1)

# 현재 페이지 소스 가져오기

html = driver.page\_source

# BeautifulSoup로 페이지 소스 파싱

bs = bs4.BeautifulSoup(html,"html.parser")

bs

```
1 # 현재 페이지 소스 가져오기
2 html = driver.page_source
3
4 import bs4
5 # BeautifulSoup로 페이지 소스 파싱
6 bs = bs4.BeautifulSoup(html,"html.parser")
7 bs
```

```
<html lang="ko-KR"><head><script async="" src="https://graph.facebook.com/?callback=WPCOMSharj
acebook_count&ids=https%3A%2F%2Fsparkkorea.com%2F%25ed%2580%25b4%25ec%25a6%2588%2F&_j
8"></script>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<link href="http://gmpg.org/xfn/11" rel="profile"/>
<link href="https://sparkkorea.com/xmlrpc.php" rel="pingback"/>
<title>퀴즈 - 스파크 코리아</title>
<meta content="QRZVgHsoL23RuwmgmXNuXf0-mUaxCkdvhP2rRvLT270" name="google-site-verification"/>
<!-- Async WordPress.com Remote Login -->
```

## 5. 셀레니움 활용 웹브라우저 자동화

### 6. 페이지정보 스크랩 및 저장

페이지 소스 가져오기 : `driver.page_source`

Example

다음장 퀴즈



[selenium 활용] sparkkorea.com 사이트내  
퀴즈 페이지 에서  
스파크 퀴즈 퀴즈이름 및 링크정보를 스크랩 후  
finalResult 변수에 저장하세요

	spark퀴즈 타이틀	spark퀴즈 링크
0	6/13 Spark 심화과정	<a href="https://forms.gle/Fw49w9GhWQChDcZm7">https://forms.gle/Fw49w9GhWQChDcZm7</a>
1	6/13 Spark 기본과정	<a href="https://forms.gle/G4TcXm3fKuHLHA6D6">https://forms.gle/G4TcXm3fKuHLHA6D6</a>
2	6/18 Spark [MAP_FILTER]	<a href="https://forms.gle/M8gr1kC2ubA3UDVp8">https://forms.gle/M8gr1kC2ubA3UDVp8</a>
3	6/18 Spark GroupBy 심화	<a href="https://forms.gle/h8w5mZ4MNaPLCPbi6">https://forms.gle/h8w5mZ4MNaPLCPbi6</a>
4	6/25 Spark RDD 실전 분석	<a href="https://forms.gle/q5yL6QHfueDLM5w27">https://forms.gle/q5yL6QHfueDLM5w27</a>
5	6/27 Spark RDD 실전 분석2	<a href="https://forms.gle/Gxb4y6LfVYiaLu4M7">https://forms.gle/Gxb4y6LfVYiaLu4M7</a>

## 5. 셀레니움 활용 웹브라우저 자동화

### 6. 데이터 저장

# csv 파일로 저장

```
finalResult.to_csv("./link_scraping_result.csv", encoding="ms949", index=False)
```

	A	B
1	spark퀴즈 타이틀	spark퀴즈 링크
2	6/13 Spark 심화과정	<a href="https://forms.gle/Fw49w9GhWQChDcZm7">https://forms.gle/Fw49w9GhWQChDcZm7</a>
3	6/13 Spark 기본과정	<a href="https://forms.gle/G4TcXm3fKuHLHA6D6">https://forms.gle/G4TcXm3fKuHLHA6D6</a>
4	6/18 Spark [MAP_FILTER]	<a href="https://forms.gle/M8gr1kC2ubA3UDVp8">https://forms.gle/M8gr1kC2ubA3UDVp8</a>
5	6/18 Spark GroupBy 심화	<a href="https://forms.gle/h8w5mZ4MNaPLCPbi6">https://forms.gle/h8w5mZ4MNaPLCPbi6</a>
6	6/25 Spark RDD 실전 분석	<a href="https://forms.gle/q5yL6QHfueDLM5w27">https://forms.gle/q5yL6QHfueDLM5w27</a>
7	6/27 Spark RDD 실전 분석2	<a href="https://forms.gle/Gxb4y6LfVYiaLu4M7">https://forms.gle/Gxb4y6LfVYiaLu4M7</a>

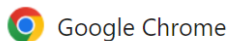
# 참고. 크롬드라이버 다운로드

# 참고. 셀레니움 환경 구축 (크롬드라이버 윈도우)

## 셀레니움 환경 구축 (윈도우 환경)

설정 -> Chrome 정보 (앞에 3자리만 확인!)

Chrome 정보



업데이트가 거의 완료되었습니다. 업데이트를 마치려면 Chrome을 다시 실행하세요.



버전 126.0.6478.127(공식 빌드) (64비트)

Chrome 도움말 보기

문제 신고

개인정보처리방침

### Chrome for Testing availability

This page lists the latest available cross-platform Chrome for Testing versions and assets per Chrome release channel. Consult our [FAQ](#) and [resources](#) if you're looking to build automated scripts based on Chrome for Testing release data.

Last updated: 2024-02-27T13:00:34.100Z

Channel	Version	Revision	Status
<a href="#">Stable</a>	126.0.6478.182	r1300313	✓
<a href="#">Beta</a>	127.0.6533.57	r1313161	✓

### Stable

Version: 126.0.6478.182 (r1300313)

Binary	Platform	URL
chrome	linux64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/linux64/chrome-linux64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/linux64/chrome-linux64.zip</a>
chrome	mac-arm64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-arm64/chrome-mac-arm64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-arm64/chrome-mac-arm64.zip</a>
chrome	mac-x64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-x64/chrome-mac-x64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-x64/chrome-mac-x64.zip</a>
chrome	win32	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win32/chrome-win32.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win32/chrome-win32.zip</a>
chrome	win64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win64/chrome-win64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win64/chrome-win64.zip</a>
chromedriver	linux64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/linux64/chromedriver-linux64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/linux64/chromedriver-linux64.zip</a>
chromedriver	mac-arm64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-arm64/chromedriver-mac-arm64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-arm64/chromedriver-mac-arm64.zip</a>
chromedriver	mac-x64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-x64/chromedriver-mac-x64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/mac-x64/chromedriver-mac-x64.zip</a>
chromedriver	win32	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win32/chromedriver-win32.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win32/chromedriver-win32.zip</a>
chromedriver	win64	<a href="https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win64/chromedriver-win64.zip">https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.182/win64/chromedriver-win64.zip</a>

# 참고. 셀레니움 환경 구축 (크롬드라이버 리눅스)

## 셀레니움 환경 구축 (리눅스 환경)

### 1 자바 설치

### 2 브라우저 설치 (크롬)

curl <https://intoli.com/install-google-chrome.sh> | bash  
sudo yum install google-chrome-stable

### 3 크롬 드라이버 설치

wget “크롬드라이버 위치”  
yum install -y unzip zip  
unzip chromedriver\_linux64.zip

chmod 775 chromedriver  
mv chromedriver /usr/local/bin/

```
rw-rw-rw- | root root 12405080 Jan 6 20:30 chromedriver
-rwxrwxrwx | root root 6121623 Jan 6 20:30 chromedriver_linux64.zip
-rw-r--r-- | root root 72668416 Jan 16 03:26 google-chrome-stable_current_x86_64.rpm
[root@a2663278b6e5 down]# chown root:root /home/down/chromedriver
[root@a2663278b6e5 down]# ll
total 89064
-rwxrwxrwx | root root 12405080 Jan 6 20:30 chromedriver
-rwxrwxrwx | root root 6121623 Jan 6 20:30 chromedriver_linux64.zip
-rw-r--r-- | root root 72668416 Jan 16 03:26 google-chrome-stable_current_x86_64.rpm
[root@a2663278b6e5 down]#
```

## 5. 셀레니움 활용 웹브라우저 자동화 (참조)

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
driver = webdriver.Chrome('d:/chromedriver/chromedriver.exe')
driver.get('https://www.google.ca/imghp?hl=en&tab=ri&authuser=0&ogbl')
```

```
searchPath = '//*[@id="sbtc"]/div/div[2]/input'
driver.find_element_by_xpath(searchPath).send_keys("cat")
driver.find_element_by_xpath(searchPath).send_keys(Keys.ENTER)
```

```
# 더 이상 스크롤 되지 않을때까지 반복해서 내림
last_height = driver.execute_script('return document.body.scrollHeight')
```

```
while True:
    driver.execute_script('window.scrollTo(0,document.body.scrollHeight)')
    time.sleep(2)
    new_height = driver.execute_script('return document.body.scrollHeight')
    try:
        # 마지막줄 show me more 버튼의 xpath
        driver.find_element_by_xpath('//*[@id="isImp"]/div/div/div/div/div[4]/div[2]/input').click()
        time.sleep(2)
    except:
        pass
    if new_height == last_height:
        break
    last_height = new_height
```

## 5. 셀레니움 활용 웹브라우저 자동화 (참조)

```
from selenium import webdriver
```

```
options = webdriver.ChromeOptions()
options.add_argument('--headless')
# options.add_argument('window-size=1200x600')
options.add_argument('--no-sandbox')
# options.add_argument('--disable-dev-shm-usage')
prefs = {
    "download.default_directory": "/home/down",
    "download.prompt_for_download": False,
    "download.directory_upgrade": True
}
```

```
options.add_experimental_option('prefs', prefs)
#chrome드라이버가 PATH 환경변수 설정이 되어있지 않다면 executable_path 옵션으로 chromedriver 위치 지정
driver = webdriver.Chrome(chrome_options=options, executable_path="/usr/local/bin/chromedriver")
```

```
url = "http://google.com"
```

```
driver.get(url)
driver.save_screenshot("google.png")
driver.quit()
```

# 참고. 액션 연속 적용하기



## 5. 셀레니움 활용 웹브라우저 자동화

### 참고. 액션 취하기 (고급)

액션 취하기 (연속) : ActionChains

\* 단 일부 사이트에서는 차단 됨! 웹브라우저 자동화는 별도확인

### Example

```
from selenium.webdriver.common.action_chains import ActionChains
coupangUrl = 'http://www.coupang.com'
driver.get(coupangUrl)
```

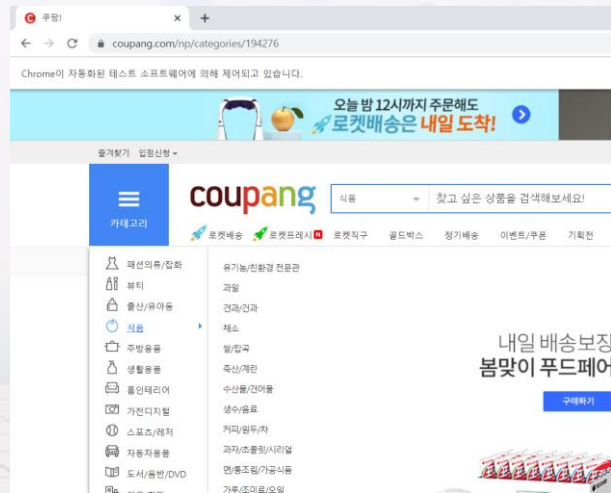
# 카테고리 메뉴 (카테고리 -> 식품)

```
mainMenu = '//*[@id="header"]/div'
subMenu = '//*[@id="gnbAnalytics"]/ul[1]/li[4]/a'
```

```
main = driver.find_element(By.XPATH, mainMenu)
sub = driver.find_element(By.XPATH, subMenu)
```

# 메인 이동 후 클릭

```
ActionChains(driver).move_to_element(main).click(sub).perform()
```



4차산업혁명 단계별로 익히는 빅데이터&인공지능(광문각, 김효관 교수)

<http://allselenium.info/python-selenium-all-mouse-actions-using-actionchains> de

## 5. 셀레니움 활용 웹브라우저 자동화

### 참고. 액션 취하기 (고급)

요소 존재여부 확인 후 action 취하기

액션대상 XPath

#### Example

<https://selenium-python.readthedocs.io/ Waits.html>

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

```
sparkUrl = "https://www.sparkkorea.com"
driver.get(sparkUrl)
quizMenu = '//*[@id="menu-item-382"]/a'
```

```
# 기다렸다가 클릭 (try ~ except 도 적용)
element = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located( (By.XPATH, quizMenu) ) )
driver.find_element_by_xpath(quizMenu).click()
```

# 참조. 메일전송 자동화

# 참조 - 메일전송 자동화

## 모듈 개요

### [과정개요]

메일전송 자동화

### [교육목표]

- 수집한 결과 자동 메일 전송 구축

### [교육대상]

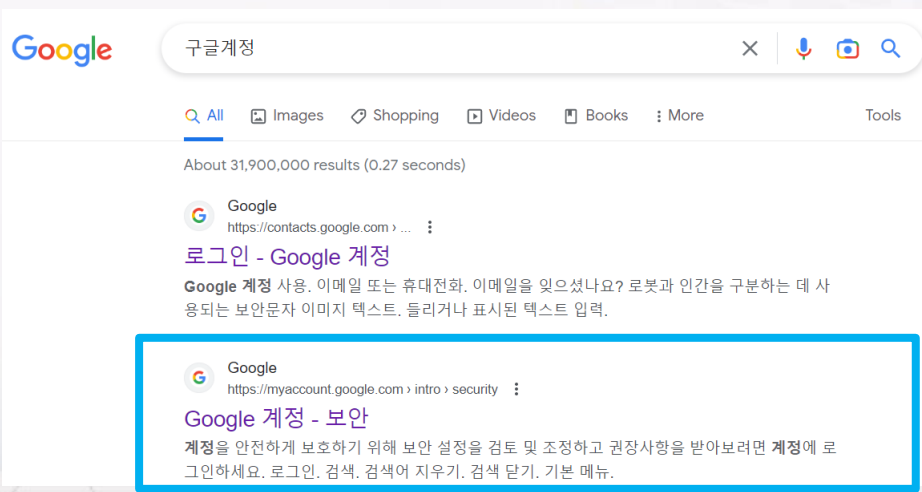
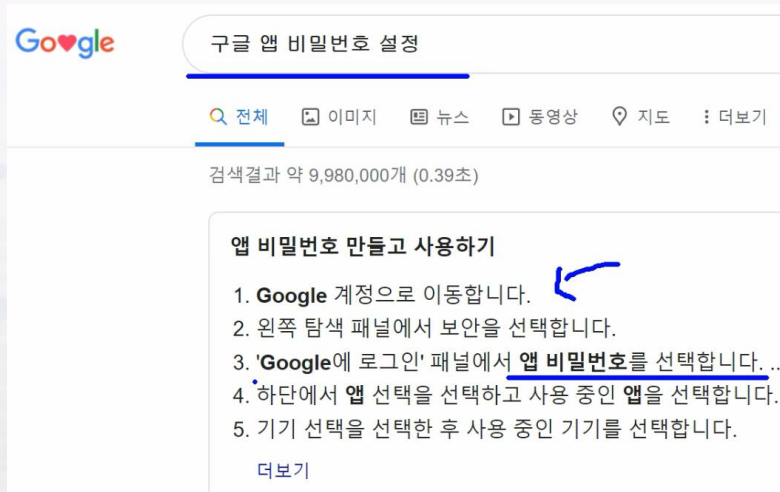
- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
메일전송 자동화	<ul style="list-style-type: none"><li>- SMTPLIB 라이브러리를 활용하여 파이썬에서 메일을 전송하는 방법을 실습합니다. (SMTP: Simple Mail Transfer Protocol)</li></ul>

# 참조 - 메일전송 자동화

## 사전준비. GMAIL 구글 앱 비밀번호 16자리 획득

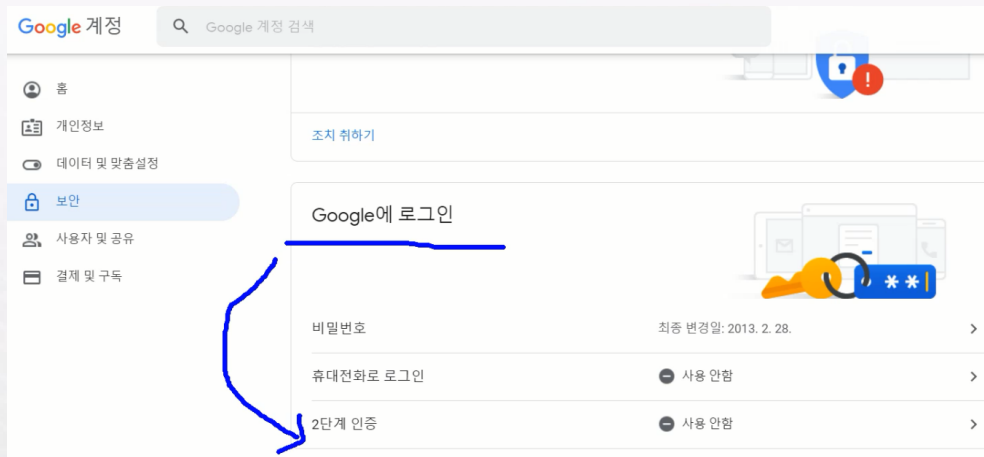
구글계정에서 설정 필요



# 참조 - 메일전송 자동화

## 사전준비. GMAIL 구글 앱 비밀번호 16자리 획득

2단계 인증 후 앱 비밀번호 설정 가능함



### ← 2단계 인증



#### 앱 비밀번호

앱 비밀번호는 권장되지 않으며 대부분의 경우 필요하지 않습니다. 계정을 안전하게 보호하려면 'Google 계정으로 로그인'을 사용하여 앱을 Google 계정에 연결하세요.

#### 앱 비밀번호

비밀번호 2개



#### 계정 도용 방지

누군가 내 비밀번호를 알게 되더라도 내 계정에 로그인할 수 없습니다.

시작하기

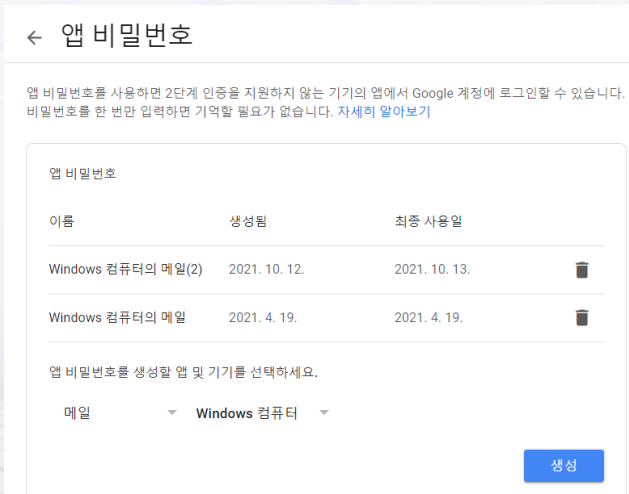
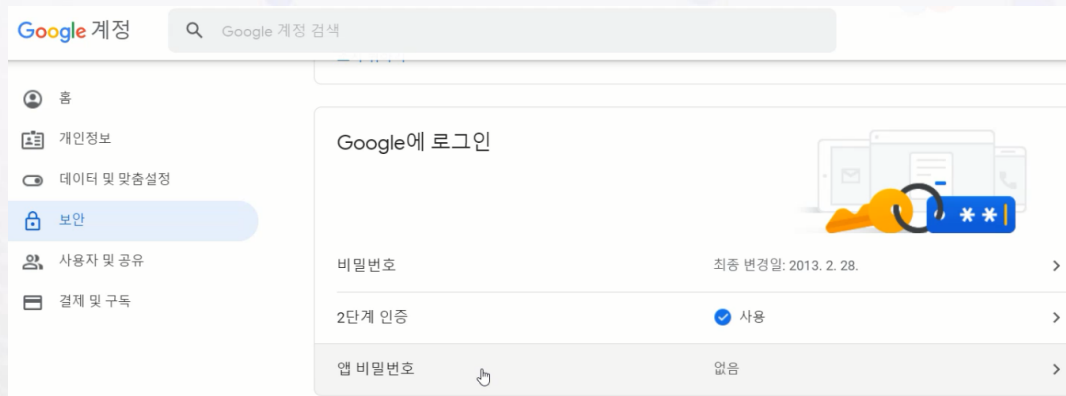
관 교수)

www.youtube.com/hkcode

# 참조 - 메일전송 자동화

## 사전준비. GMAIL 구글 앱 비밀번호 16자리 획득

2단계 인증 설정 완료 후 앱 비밀번호 설정 (구글계정 -> 보안 -> 앱 비밀번호)



# 참조 - 메일전송 자동화

## 사전준비. GMAIL 구글 앱 비밀번호 16자리 획득

16자리 앱 비밀번호 획득 (아래 이미지는 유효하지 않은 번호임)

생성된 앱 비밀번호

Windows 컴퓨터용 앱 비밀번호

cpgf sayr acsc odmw

사용 방법

1. '메일' 앱을 엽니다.
2. '설정' 메뉴를 엽니다.
3. '계정'을 선택한 뒤 내 Google 계정을 선택합니다.
4. 비밀번호를 위에 표시된 16자리[비밀번호로 교체합니다.

일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되도록 적어 놓거나 다른 사용자와 공유하지 마세요.

[자세히 알아보기](#)

확인

Add your Google account

Enter the information below to connect to your Google account.

Email address

securesail@gmail.com

Password

\*\*\*\*\*

☐ Include your Google contacts and calendars



# 참조 - 메일전송 자동화

## 1. 메일 전송 라이브러리 선언

```
# SMTP 프로토콜 로드  
import smtplib
```

```
# 이메일을 간단하게 보낼수 있는 라이브러리 로드  
from email.message import EmailMessage
```

# 참조 - 메일전송 자동화

## 2. 메일전송 프로토콜 설정

# GMAIL 메일 설정

```
smtp_gmail = smtplib.SMTP('smtp.gmail.com', 587)
```

# 서버 연결을 설정하는 단계

```
smtp_gmail.ehlo()
```

# 연결을 암호화

```
smtp_gmail.starttls()
```

#로그인

```
userid = "xxxxxxxxx"
```

```
userpw = "xxxxxxxxxxxxxxxxxxx"
```

```
smtp_gmail.login(userid, userpw)
```

보내는 메일에 따라  
smtp.naver.com

```
import getpass  
pw = getpass.getpass()
```

로그인 아이디: gmail 아이디  
패스워드: 앱 비밀번호

# 3. pickle

## 1. 주요 기능

pickle.read : 데이터 읽기  
Pickle.write : 데이터 쓰기

```
data450 = pd.read_csv("../dataset/kopo_450data.csv")
```

```
data450.shape
```

```
(4500000, 16)
```

```
### 피클 파일 저장하기 (바이너리) ###
```

```
with open("data450.pickle", "wb") as fw:  
    pickle.dump(data450, fw)
```

```
### 피클 파일 불러오기 (바이너리) ###
```

```
with open("data450.pickle", "rb") as fr:  
    data = pickle.load(fr)
```

```
data.head(2)
```

	CUSTOMERCODE	STATENAME	ST	GENDER	DOB	GENDER1	EMAIL	FEST_CNT	TOTAL_AMOUNT
0	1503989	State2	2	Male	0	1	1	0	30300

	CUSTOMERCODE	STATENAME	ST	GENDER	DOB	GENDER1	EMAIL	FEST_CNT	TOTAL_AMOUNT
1	8004	9544	0100	0000	0000	008c	1170	616e	
2	6461	732e	636f	7265	2e66	7261	6d65	948c	
3	0944	6174	6146	7261	6d65	9493	9429	8194	
4	7d94	288c	045f	6d67	7294	8c1e	7061	6e64	
5	6173	2e63	6f72	652e	696e	7465	726e	616c	
6	732e	6d61	6e61	6765	7273	948c	0c42	6c6f	
7	636b	4d61	6e61	6765	7294	9394	8c09	6675	
8	6e63	746f	6f6c	7394	8c07	7061	7274	6961	
9	6c94	9394	8c1c	7061	6e64	6173	2e63	6f72	
10	652e	696e	7465	726e	616c	732e	626c	6f63	
11	6b73	948c	096e	6577	5f62	6c6f	636b	9493	
12	9485	9452	9428	680e	297d	948c	046e	6469	
13	6d94	4b02	734e	7494	628c	156e	756d	7079	
14	2e63	6f72	652e	6d75	6c74	6961	7272	6179	
15	948c	0c5f	7265	636f	6e73	7472	7563	7494	
16	9394	8c05	6e75	6d70	7994	8c07	6e64	6172	

# 3. pickle

## 2. 활용

```
import pandas as pd
```

```
data450 = pd.read_csv("../dataset/kopo_450data.csv")
```

```
data450.shape
```

```
### 피클 파일 저장하기 (바이너리) ###
```

```
with open("data450.pickle", "wb") as fw:  
    pickle.dump(data450, fw)
```

```
### 피클 파일 불러오기 (바이너리) ###
```

```
with open("data450.pickle", "rb") as fr:  
    data = pickle.load(fr)
```

```
data.head(2)
```

```
import pandas as pd
```

```
data450 = pd.read_csv("../dataset/kopo_450data.csv")
```

```
data450.shape
```

```
### 피클 파일 저장하기 (바이너리) ###
```

```
with open("data450.pickle", "wb") as fw:  
    pickle.dump(data450, fw)
```

```
### 피클 파일 불러오기 (바이너리) ###
```

```
with open("data450.pickle", "rb") as fr:  
    data = pickle.load(fr)
```

```
data.head(2)
```

	CUSTOMERCODE	STATENAME	ST	GENDER	DOB	GENDER1	EMAIL	FEST_CNT
0	1503989	State2	2	Male	0	1	1	0

# 참조 - 메일전송 자동화

## 3. 수신자 목록 정의 및 불러오기

```
# 저장된 csv 파일 불러오기
# emailist = pd.read_csv("./emailaddress.csv", encoding='ms949')
# emailist
# 이메일 주소정보 리스트 변환
# to = emailist['이메일'].tolist()
# to
```

```
to = ["haiteam@kopo.ac.kr", "haiteam@naver.com"]
```

	A	B
1	이름	이메일
2	김효관	<a href="mailto:haiteam@kopo.ac.kr">haiteam@kopo.ac.kr</a>
3	김효관2	<a href="mailto:haiteam@naver.com">haiteam@naver.com</a>

```
['haiteam@kopo.ac.kr', 'haiteam@naver.com']
```

# 참조 - 메일전송 자동화

## 4. 메일전송

```
msg=EmailMessage()
```

```
# 제목 입력
```

```
msg['Subject']="퀴즈 정보"
```

```
# 내용 입력
```

```
msg.set_content("퀴즈정보")
```

```
# 보내는 사람
```

```
msg['From']='haiteamm@gmail.com'
```

```
# 받는 사람
```

```
msg['To'] = ",".join(to)
```

```
# 첨부파일 #1 추가
```

```
file='0318_오후2.png'
```

```
fp = open(file,'rb')
```

```
file_data=fp.read()
```

```
msg.add_attachment(file_data,  
                    maintype='text',  
                    subtype='plain',  
                    filename=file)
```

```
file2='link_scraping_result.csv'
```

```
fp = open(file,'rb')
```

```
file_data=fp.read()
```

```
msg.add_attachment(file_data,  
                    maintype='text',  
                    subtype='plain',  
                    filename=file2)
```

```
# 메일 전송
```

```
smtp_gmail.send_message(msg)
```

```
smtp_gmail.close()
```

# 참조 - 메일전송 자동화

## 4. 메일전송

### 퀴즈 정보

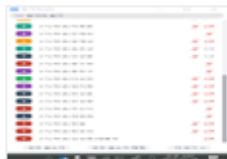


haiteamm@gmail.com

받는사람 : haiteam@kopo.ac.kr , haiteam@naver.com

### 퀴즈정보

첨부파일 2개 모두저장



link\_scraping\_r  
esult.csv

56.6 KB

<https://mailtrap.io/blog/python-send-email-gmail/>



## 5. 셀레니움 활용 웹브라우저 자동화



Selenium  
WebDriver

로그인이 필요한 사이트라면 **se** 활용 로그인 후  
스크랩 대상 페이지로 이동  
(페이지 이동 후 소스획득 후 BeautifulSoup로 !!)



Parse HTML  
BeautifulSoup module

html 원소스를 **bs** 활용 이쁘게 태그만 남기고  
태그내 원하는 자료 획득!  
(Find로 범위를 좁히고 Find\_all로 반복해서 스크랩)



스크랩 자료를 자동으로 메일을 보내보자!

## 2. 핵심정리 및 Q&A

### 기억합시다

1

Selenium을 활용한 자동화 방법을 기억한다.

감사합니다.