

타 대학 이용자를 위한 Customizing Guide

제작 및 문의: 송종현 (서울대 기계 17)

<http://github.com/yc-song>

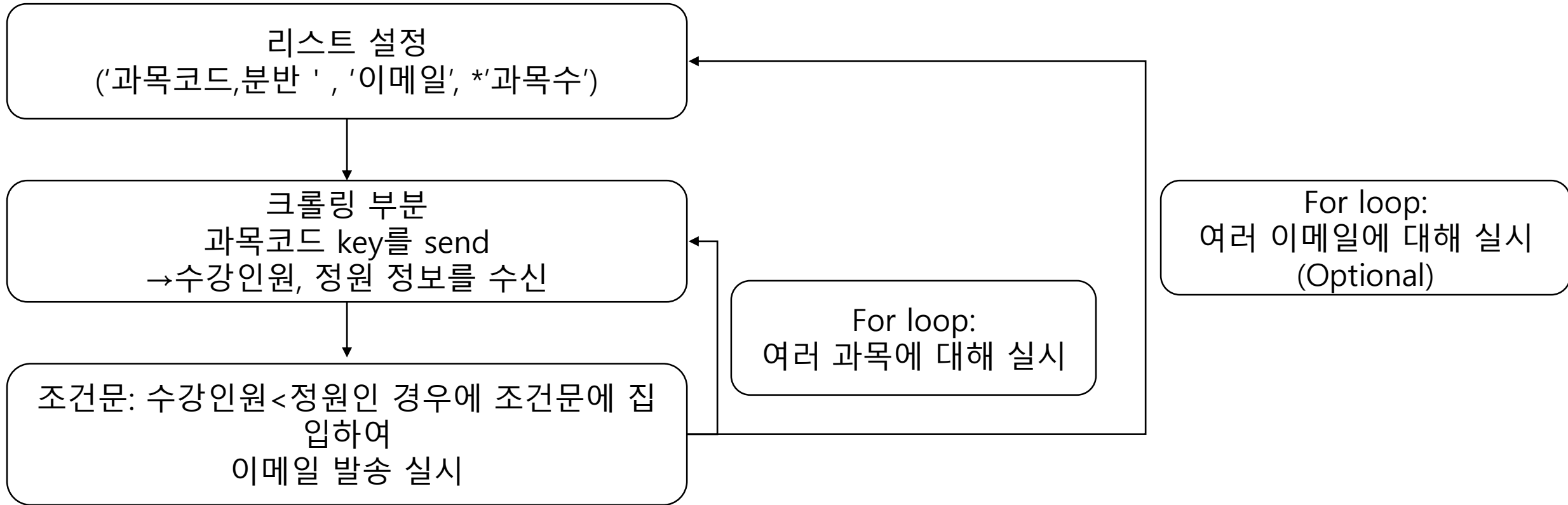
hyeongoon11@snu.ac.kr

기본 사양

- Python 3.8 이상
 - 필요 패키지: 우측 그림 참조
 - .exe, .apk등으로 배포 계획 없음
- : 소스 코드를 건드려서 다양한 상황에 맞추기가 어렵기 때문

Package	Version	Latest version
altgraph	0.17	0.17
beautifulsoup4	4.9.1	4.9.1
bs4	0.0.1	0.0.1
future	0.18.2	0.18.2
pefile	2019.4.18	2019.4.18
pip	20.2.2	20.2.2
pyinstaller	4.0	4.0
pyinstaller-hooks-contrib	2020.7	2020.7
pywin32-ctypes	0.2.0	0.2.0
selenium	3.141.0	3.141.0
setuptools	49.6.0	49.6.0
soupsieve	2.0.1	2.0.1
urllib3	1.25.10	1.25.10

알고리즘 설명



각 부 설명 (1)

```
6 import selenium
7 from selenium import webdriver
8 from bs4 import BeautifulSoup
9 from email.mime.text import MIMEText
10 import smtplib
11 import threading
12 import time
13 from selenium.webdriver.chrome.options import Options
```

설명

필요한 Pacakage를 import하는 부분

각 부 설명 (2)

```
16 def thread_run():
17     snipping_list = [[['M2794.001200', '001'], ['031.033', '001', '002', '003'], ['043.071', '003'], ['270.549', '001']],
18                       ['hyeongoon11@snu.ac.kr', 7], [['M2794.001200', '002']], ['hyeongoon@gmail.com', 12]]
```

설명

함수를 정의하고, 3차원 리스트를 받아오는 부분으로, [[[과목코드,과목분반(들)]...],이메일,강의 수(뒤에 설명 有)] 순으로 지정

각자 상황에 맞게 상기 리스트를 조정하면 된다.

주의: 이 코드로 메일을 보내달라고 요청한 사람이 여러 명이라서 부득이 다차원 리스트를 사용하였다. 혼자 사용하는 경우에도 리스트의 차원은 건드리지 않는 것을 권장 (뒤에 for문이 다 망가짐)

각 부 설명 (3)

```
19     for jh in range(len(snipping_list)):  
20         ratio = []  
21         # 크롬 옵션 설정  
22         options = webdriver.ChromeOptions()  
23         chrome_options = Options()  
24         chrome_options.add_argument("--headless")  
25         driver = webdriver.Chrome('./webdriver/chrome/chromedriver.exe', options=chrome_options)  
26         # driver = webdriver.Chrome('./webdriver/chrome/chromedriver.exe')
```

설명

3차원 리스트 상에서 첫번째 리스트의 반복문이다. (소스코드의 리스트를 예로 들면, [['M2794.001200', '001'], ['031.033', '001', '002', '003'], ['043.071', '003'], ['270.549', '001']], 'hyeongoon11@snu.ac.kr', 7]에 해당하는 리스트를 처음으로 불러온다.)

각자 크롬 버전에 맞는 웹드라이버를 다운받아 파이선 파일과 같은 경로에 지정한 후, 사용하면 된다.

* 만약 크롬 드라이버가 실제로 작동하는 모습을 보고 싶다면, 22~25줄을 각주처리하고 26줄의 각주를 해제하면 된다.

각 부 설명 (4)

```
29     # 접속. 수강편람
30     html = driver.page_source
31     soup = BeautifulSoup(html, 'html.parser')
32     driver.implicitly_wait(3)
33     for t in range(len(snipping_list[jh][0])):
34         driver.get('https://sugang.snu.ac.kr/sugang/cc/cc100.action')
35         driver.find_element_by_xpath('//*[@id="srchSbjtCd"]').send_keys(snipping_list[jh][0][t][0])
36         driver.find_element_by_xpath('//*[@id="cond00"]/a[4]').click()
```

설명

수강신청 사이트에 접속하여, 원하는 과목 화면을 조회하는 코드이다.
34번째 줄은 각자 학교의 수강편람 사이트로 링크를 변경하면 된다.

각 부 설명 (5)

```
36     for j in range(1, snipping_list[jh][2]): # 유효한 정보가 있 는 셀을 받아옴 (학교 수신 사이트가 좀 구조가 이상함)
37         try:
38             print(int(driver.find_element_by_xpath(
39                 '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[16]'.format(j)).text.strip()[
40                 0:3]), int(
41                 driver.find_element_by_xpath(
42                     '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[17]'.format(
43                         j)).text.strip()))
44             ratio.append(j)
45         except selenium.common.exceptions.NoSuchElementException:
46             pass
```

설명

이 부분은 서울대 수강신청 사이트의 거지같은 구조로 인하여 생겨났으며, 학교 수강편람 사이트의 구조가 정상적이라면 적용하지 않아도 되는 예외처리문이다. 자세한 설명은 다음 장표 참고

* find_element_by_xpath괄호 안의 부분은 각 사이트 상황에 맞게 변경하여 사용하면 된다. (td[16]은 정원, td[17]은 수강신청 인원이다.

(이어서)

교과목 번호	강좌 번호	교과목명 (부제명)	학점- 강의- 실습	수업교시	수업 형태	강의실 (동-호) (#연건, *평창)	주담당 교수	강의 계획서	정원 (재학생)	수강 신청 인원	비고
√2794.001 200	001	동역학	3-3-0	월(11:00~12:15)	이론	301-105	강연준	Y	90 (90)	90	1~2일: 기계/기 주전공(자유, 다 포함) 2학년 초: 3일: 기계/기계학 전공(자유, 다 포함) 초수강생4 계/기계항공 공 (자유, 다전공포 채학년5일: 타
				수(11:00~12:15)	이론	301-105					
√2794.001 200	002	동역학	3-3-0	월(11:00~12:15)	이론	301-204	이동준	Y	90 (90)	76	◎1~2일: 기계/ 공 주전공(자유 공포함) 2학년 : 생3일: 기계/기: 주전공(자유, 다 포함) 초수강생4 계/기계항공 공 (자유, 다전공포
				수(11:00~12:15)	이론	301-204					

왼쪽 수강신청 사이트의 구조를 보면,
001 강좌의 수강신청인원 XPATH는
`//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[1]/td[17]`이고,
002 강좌는
`//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[3]/td[17]`으로 되어있어,
아무 생각 없이 for문을 1,2,3... 순서로 작동하면 에러가 발생
한다.
따라서 유효한 행을 ratio 리스트에 저장해 사용하였으며, 학교
사이트가 1,2,3... 순서로 정렬되어 있다면 빠른 코드 실행을
위해 생략해도 되는 코드다.

각 부 설명 (6)

```
47         for j in ratio:
48             if int(driver.find_element_by_xpath(
49                 '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[16]'.format(j)).text.
50                 0:3]) > int(driver.find_element_by_xpath(
51                 '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[17]'.format(
52                     j)).text.strip()) : # 여석검증
53                 for m in range(1, len(snipping_list[jh][0][t])):
54                     if str(snipping_list[jh][0][t][m]) == str(driver.find_element_by_xpath(
55                         '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[8]'.format(j)).text.
56                         strip()):
57                         smtp = smtplib.SMTP('smtp.gmail.com', 587)
58                         smtp.ehlo()
59                         smtp.starttls()
60                         smtp.login('아이디', '비번')
61                         msg = MIMEText('https://sugang.snu.ac.kr/\n\n{} ({} ) 수업의 여석이 발생하였습니다.\n\n
62                         driver.find_element_by_xpath(
63                             '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[17]'.format(
64                                 j)).text.strip(), int(driver.find_element_by_xpath(
65                             '//*[@id="content"]/div/div[3]/div[1]/div[2]/table/tbody/tr[{}]/td[16]'.format(j)).text.
66                             0:3)))
67                         smtp.sendmail('sugang@snu.ac.kr', to_addrs, msg)
```

설명

이 코드의 꽃에 해당하는 부분으로, 수강신청 인원 (td[16])이 정원 (td[17])보다 적으면 이메일을 발송하게 된다. 49~52, 60~66 줄에 해당하는 부분을 커스터마이징하면 된다.

각 부 설명 (7)

```
67         msg['Subject'] = '[긴급] {} ({} ) 수업 여석 발생'.format(snipping_list[jh][0][t][0],snipping_list[jh][0][t][m])
68         msg['To'] = snipping_list[jh][1]
69         | smtp.sendmail('보내는 메일 주소', snipping_list[jh][1], msg.as_string())
70
71         smtp.quit()
72
73         ratio = []
74         threading.Timer(60,thread_run).start()
75
76     thread_run()
```

설명

69줄에 보내고자 하는 메일 주소를 입력하면 된다.

73줄은 이 코드가 반복 작동하도록 하는 부분으로써, 코드 실행이 시작한 시점이 아니라 끝난 시점으로부터 60초 후에 반복 작동한다는 뜻이다. 컴퓨터 사양을 고려하여 반복 시행 시간을 정하되, 너무 잦은 코드의 실행은 (혹시나) 학교 전산 처로부터의 IP 차단가능성이 존재함을 유념하여야 한다.

이메일이 보내지지 않는 경우

Gmail 계정 보안으로 인해 발생했을 가능성이 크다. 이 경우

“<https://support.google.com/accounts/answer/6010255>”을 참고하여 보안 수준이 낮은 앱의 액세스를 허용하여야 한다.

≡ Google 계정 고객센터

🔍 문제를 설명해 주세요.



보안 수준이 낮은 앱 및 Google 계정

앱 또는 사이트가 Google의 [보안 표준](#)을 충족하지 않는 경우 Google은 해당 앱에서 사용자의 계정에 로그인하려는 시도를 차단할 수 있습니다. 보안 수준이 낮은 앱에서는 해커가 사용자의 계정에 쉽게 액세스할 수 있으므로 이러한 앱에서 로그인하지 못하게 차단하면 계정을 안전하게 보호할 수 있습니다.

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 설정된 경우 ▼

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 중지된 경우 ▲

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 중지된 경우 다시 사용 설정할 수 있습니다. 대신 보안 수준이 높은 앱으로 전환하는 것이 좋습니다.

의심스러운 활동

- 📄 비밀번호를 변경하라는 메시지가 표시됨
- 📄 다른 사용자가 내 비밀번호를 변경함
- 📄 계정에서 의심스러운 활동 감지
- 📄 '잘못된 비밀번호' 오류
- 📄 보안 수준이 낮은 앱 및 Google 계정