웹 크롤링 - selenium 문법예시

작성일시	@2023년 3월 1일 오후 10:45
강의 번호	Python
유형	
자료	
복습	☐
학습 소스 출처 1	
학습 소스 출처 2	
날짜	

가져온 요소 조작하기

browser.find_element 메소드를 이용해 가져온 요소를 해당 브라우저 상에서 조작할 수 있는 명령어 들이 있다.

```
#Keys 라이브러리를 이용하면 키워드를 통해 원하는 키보드 입력을 사용할 수 있다.
from selenium.webdriver.common.keys import Keys

# 검색창 같은 Element안에 검색어등을 입력하거나
# 회원가입 및 로그인 시 개인정보를 입력가능
원하는Element.send_keys("입력할 텍스트")

# 해당요소에 키보드 입력을 지원하는 명령어
# 대표적으로 검색창에서 엔터를 누르는 명령등 사용가능
원하는Element.send_keys(Keys.키워드)
```

위의 Keys 라이브러리의 명령어는 아래와 같은 예시가 있다.

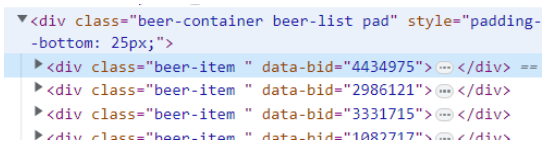
```
class Keys(object):
    """
    Set of special keys codes.
    """

    NULL = '\ue000'
    CANCEL = '\ue001'  # ^break
    HELP = '\ue002'
    BACKSPACE = '\ue003'
    BACK_SPACE = BACKSPACE
    TAB = '\ue004'
    CLEAR = '\ue005'
    RETURN = '\ue006'
    ENTER = '\ue007'
    SHIFT = '\ue008'
    LEFT_SHIFT = SHIFT
    CONTROL = '\ue009'
    LEFT_CONTROL = CONTROL
    ALT = '\ue00a'
    LEFT_ALT = ALT
    PAUSE = '\ue00b'
    ESCAPE = '\ue00c'
    SPACE = '\ue00d'
    PAGE_UP = '\ue00e'
```

```
    PAGE_DOWN = '\ue00f'
    END = '\ue010'
    HOME = '\ue011'
    LEFT = '\ue012'
    ARROW_LEFT = LEFT
    UP = '\ue013'
    ARROW_UP = UP
    RIGHT = '\ue014'
    ARROW_RIGHT = RIGHT
    DOWN = '\ue015'
    ARROW_DOWN = DOWN
    INSERT = '\ue016'
    DELETE = '\ue017'
    SEMICOLON = '\ue018'
    EQUALS = '\ue019'

    NUMPAD0 = '\ue01a'  # number pad keys
    NUMPAD1 = '\ue01b'
    NUMPAD2 = '\ue01c'
    NUMPAD3 = '\ue01d'
    NUMPAD4 = '\ue01e'
    NUMPAD5 = '\ue01f'
    NUMPAD6 = '\ue020'
    NUMPAD7 = '\ue021'
    NUMPAD8 = '\ue022'
    NUMPAD9 = '\ue023'
    MULTIPLY = '\ue024'
    ADD = '\ue025'
    SEPARATOR = '\ue026'
    SUBTRACT = '\ue027'
    DECIMAL = '\ue028'
    DIVIDE = '\ue029'

    F1 = '\ue031'  # function  keys
    F2 = '\ue032'
    F3 = '\ue033'
    F4 = '\ue034'
    F5 = '\ue035'
    F6 = '\ue036'
    F7 = '\ue037'
    F8 = '\ue038'
    F9 = '\ue039'
    F10 = '\ue03a'
    F11 = '\ue03b'
    F12 = '\ue03c'

    META = '\ue03d'
    COMMAND = '\ue03d'
```

라이브러리에서 사용하는 오브젝트 값으로 키워드값을 이용해 키보드 입력을 사용할 수 있다.

찾으려는 요소가 해당 페이지에 있는지 확인하는 코드

untapped 에서 크롤링할때 , 모든 나라의 모든 맥주스타일을 순회할것이다.



검색결과가 있는경우

검색결과가 없는 경우

위의 캡처 사진은 untapped사이트에서 맥주 스타일과 국가를 바꿀때 검색결과의 유무에 따른 차이다. `beer-container beer-list pad`인 div의 자식으로 `beer-item`이 있는지 없는지로 구분할 수 있다.

이때 `beer-container beer-list pad` div의 자식으로 `beer-item` 이 있는지 없는지를 확인하는 코드.

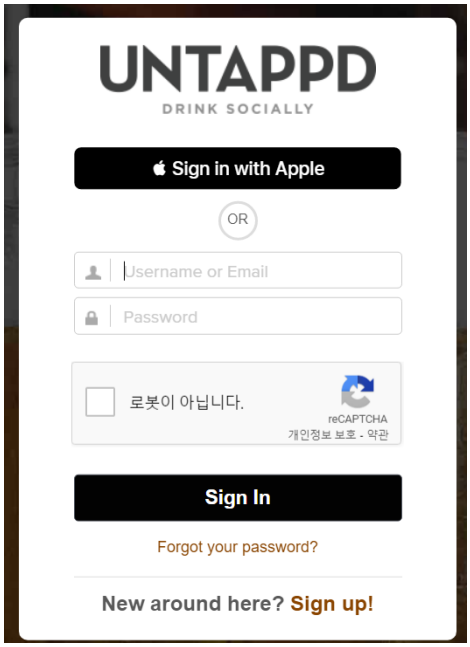
```
# 검색결과 존재하는지 아닌지 판별하는 코드
    try:
        print("검색결과 존재 ",  browser.find_element(By.CSS_SELECTOR,'.beer-container.beer-list.pad>.beer-item'))
    except :
        print("검색결과 없음")
```

`find_element` 메소드를 사용할때 해당 페이지에 조건에 맞는 요소가 없다면 `noSuchElement` exeception을 터트리는 것을 이용한 코드이다.

로그인 해결책 미흡

unttaped 사이트의 경우 사용자가 클라이언트를 사용하더라도 컴퓨터인지 구분하는 reCAPTCHA를 사용해서 해결법을 찾아야했다. 왜냐하면 unttaped사이트에서 모든 리뷰를 확인하려면 로그인한 상태여야 했다.

그렇기 때문에 UserAgent를 사용하거나 Apple 로그인을 해보려 했지만 둘다 코드로 하기엔 배보다 배꼽이 커지는 꼴이여서 포기했다. UserAgent는 똑같이 reCAPTCHA를 사용해야했고 , Apple의 경우 apple아이디의 본인인증을 거쳐야했다.



untapped의 로그인 화면

그렇기 때문에 login 사이트를 띄우고 20초정도 sleep 시켜 수동으로 로그인 하는 방식을 사용했다.

```
# 로그인 화면을 띄우고
browser = webdriver.Chrome()
browser.get("https://untappd.com/login")

# 20초 딜레이를 준 후
```

```
time.sleep(20)

# 크롤링할 주소로 넘어간다.
browser.get("https://untappd.com/beer/top_rated")
```

다른 페이지로 이동시 주의할점

셀레니움의 find_element들은 해당 객체를 저장하는 것이 아니라 , 셀레니움에서 제공하는 오브젝트로, 현재페이지의 매칭된 페어들과의 연결을 제공하는 것이다.

덕분에 셀레니움의 코드로 얼마든지 메소드 및 값을 이용할 수 있지만, 한번 다른페이지로 넘어간다면 해당 오브젝트들의 매칭이 끊길확률이 높으므로 주의해야한다.

```
cur_style = browser.find_element(By.CSS_SELECTOR,p.element)
# 실제로 접속하는 코드
cur_style.click()
# 에러발생!!! 이미 click으로 인해 다른 페이지로 넘어가
# cur_style 의 매칭이 풀렸음
print(cur_style.text)
```

위의 코드에서 보듯이 `NoSuchElement` 에러가 터지므로

```
cur_style = browser.find_element(By.CSS_SELECTOR,p.element)
# 실제로 접속하는 코드
cur_style.click()
cur_style = browser.find_element("다시 찾을 문법", "해당값")
# 새로 찾아서 에러가 발생하지 않음!!
print(cur_style)
```

위의 방식대로 새로 초기화하거나 , 안의 값을 사용하려면 다른 변수에 미리 초기화시켜 두어야 한다.