<파이썬을 활용한 실전 웹 크롤링 - WEEK 5>

* + - * Error, 크롤링 차단을 피하는 방법
* Selenium.common.exceptions
* Robots.txt
* Header & cookie
* API

1. Selenium.common.exceptions

: 브라우저 창을 제어하여 크롤링 할 때, 예상치 못한 에러나 해결이 어려운 에러가 발생할 때, try / exception 이용

1. WebDriverException 에러

* 주로 광고 팝업창에 의해 원하는 내용이 가려져 element를 선택할 수 없을 때 발생하는 에러
* Error message – WebDriveException: Message: unknown error

Element A is not clickable at point ----

* 스크롤 이동을 통해 문제 해결
* 상황 예제, 첨부파일 “not clickable”

1. NoSuchElementException

* 찾고자 하는 element가 없는 경우 발생하는 에러

1. Iframe에 속해 있어 찾을 수 없는 경우

Switch\_to.frame 코드를 이용하여 원하는 frame으로 이동하여 코드 실행

1. 데이터 로딩이 되지 않은 경우

: Time.sleep 코드로 로딩시간을 확보

: 스크롤을 움직여야 하는 경우 스크롤 이동 코드 추가

1. CSS\_Selector, X-path 잘못 입력한 경우

: 속성값 오타, X-path index값 오류. 속성값은 복사해서 이용하고 index값을 재확인

1. 검색 키워드를 변경하거나 크롤링 사이트가 변경되었을 경우

: web페이지 구조다 달라질 수 있고 상황마다 x-path가 달라질 수 있으므로 매번 확인하여 수정하거나 css-selector 사용 추천

* 상황 예제, 첨부파일 blog\_error

1. Robots.txt / 법적문제

* 크롤링에 대한 가이드라인이나 안전장치 없는 상황
* Robots.txt는 크롤링 가능 여부, 범위를 설명하는 페이지
* 주소 검색 창에 url주소/robots.txt 를 입력하면 내용 확인 가능
* Disallow 영역은 크롤링 할 수 없는 사용자 또는 사이트 내에서도 할 수 없는 영역을 명시

Ex) [www.naver.com/blog----](http://www.naver.com/blog-----)가 사이트 주소, disallow 내용에 /blog가 있다면 크롤링 허락x

* 모든 영역이 불가할 때는 ‘/’ 로 표현
* 따라서 최근에는 API를 제공하여 이용량을 제한하고 기준 이상을 이용할 때는 비용 요구
* 문제, 분쟁을 줄이려면 데이터는 요청하거나 사는 것을 추천
* Robots.txt 확인 예제, robot\_txt\_check

1. Header와 cookie

* Header : get/post 방식을 이용하여 정보를 요청할 때, header와 body부분으로 정보가 전달. Header에 담긴 사용자 정보를 통해 운영자가 크롤링 봇인지, 사람인지 판단
* 크롤링을 할 때는 header에 담긴 사용자 정보가 제한적으로 전달되기 때문에 평소와 다르게 화면이 나타남.
* Cookie : 이용자가 웹사이트에 접속할 경우 사용자의 컴퓨터에 남게 되는 기록정보. 같은 사이트를 접속할 때마다 읽히고 새로운 정보로 변경
* 자동 로그인과 같은 경우가 쿠키 기록이 계속 남아 있어 입력절차 없이 로그인 가능함
* 쿠키를 사용하여 자동로그인을 하면 로그인 횟수 제한이나 봇으로 쉽게 차단 되지 않음
* 실습 예제. Cookie\_save, cookie\_use

1. 가짜 에이전트

* 사용자의 정보를 다르게 구성하고 전달하여 봇으로 막히는 일을 줄임.
* 실습예제. Fake\_agent

1. API사용

* 정보 요청과 database를 연결하는 매개체?
* 요청한 정보에 접근할 수 있는 통로??
* 공공데이터 포털 API는 활용가능한 KEY를 받고 사이트에 가이드 참고 data 내용 이해 가능.

Ex) ‘idvid = adongCd’, ‘key = 1168064000’ 각각의 코드, 내용 가이드에 설명

* 국내의 많은 곳이 get방식(URL변경하여 정보 요청, 받음) 이용.
* json형태로 파일 제공.
* Json : [ ] list형과 { } dictionary형이 혼합된 형태. json파일에서 원하는 정보를 찾고자 할 때, 요청을 받은 source파일을 .json으로 저장하여 dict형의 ‘key’값으로 탐색가능.

Ex) response = requests.get(url)

P\_json = response.json()[‘body’][‘items’] 🡪 ‘item’ key값에 저장된 내용 불러오기 가능

* 실습예제, 첨부파일 api\_public, api\_youtube