학습 정리

팀	빅나물	구성원	이지현, 조영현

일정	발제자	주제
4일차(5/30)	이지현	파이썬 기초 스크랩핑

주요 내용 요약

<크롬 개발자 도구>

- DOM 구조 분석(요소검사)

- 선택자 추출

  원하는 요소 copy -> **copy selector**

  ex) #gnbServiceList > ul > li:nth-child(3) > a

  gnbServiceList가 id인 애의 ul 태그의 li태그의 3번째 자식의 a 태그(?)

- Console 도구

  자바스크립트 console에서 바로 실행 가능

- Source-로딩 한 리소스 분석 및 디버깅

  source탭은 일단 패쓰,

- 네트워크 탭 및 기타

  network탭에서 F5 누르면,

  이미지 파일, 동영상 파일, html파일 등 볼 수 있음.

  Preserve log 체크박스 선택 시, 타이밍이나 불러오는 데 걸린 시간들이 새로고침 시에 누적됨!

  capture screenshots 선택 후 새로고침하면, 페이지가 로드되는 각 시간동안 어떤 요소들이 로드되고, 그 시간은 어떤지 나눠서 보여줌. (사이트가 로드되는 과정 순차적으로 볼 수 있음.)

  memory탭에서는 현재 성능에 대한 메모리 누수가 없는지, 병목현상이 없는지, 어떤 부분에서 로드가 오래 걸리는지 확인 가능

  performance 레코드 버튼 누르고 stop하면 현재 한 행동에 대해서 로딩되는 타이밍, 순서를 알려줌. network 탭의 capture screenshots이랑 똑같은 것!

application 탭 현재 구조 볼 수 있고, 쿠키 값이 저장되어 있음.

<파이썬 urllib을 활용해 웹에서 필요한 데이터 추출하기>

- 하고자 하는 것

- html 다운받아 필요한 텍스트, 정보 파싱 -> DB or TXT, 엑셀, JSON파일로 만들어서 다른 server로 전송

- https://docs.python.org

: 버전 선택 후, Library Reference에서 다양한 메소드 사용법 알 수 있음.

- urlretrieve

저장 -> open('r') -> 변수에 할당 -> 파싱 -> 저장.

파싱이 필요없는 데이터 한 번에 다운로드 받는 경우 좋음.

```
imgUrl="https://search.pstatic.net/common/?src=http%3A%2F%2Fcafefiles.naver.net%2F20100113_10%2Fdbwjd177_12633924830421M1mY_jpg%2Fc6f7b8de2_yousongyee_dbwjd177.jpg&type=b360"
 htmlURL="http://google.com"

 savePath1="c:/test1.jpg"
 savePath2="c:/index.html"

 dw.urlretrieve(imgUrl,savePath1)
 dw.urlretrieve(htmlURL,savePath2)
```

- urlopen

urlopen: 변수 할당 -> 파싱 -> 저장(db,...)

중간 작업이 필요한 경우는 urlopen이 좋음.

```
imgUrl="https://search.pstatic.net/common/?src=http%3A%2F%2Fcafefiles.naver.net%2F20100113_10%2Fdbwjd177_12633924830421M1mY_jpg%2Fc6f7b8de2_yousongyee_dbwjd177.jpg&type=b360"
 htmlURL="http://google.com"

 savePath1="c:/test1.jpg"
 savePath2="c:/index.html"

 f=dw.urlopen(imgUrl).read() #이미지 데이터를 f에 할당
 f2=dw.urlopen(htmlURL).read()
```

- open, write, close

```
saveFile1=open(savePath1,'wb') # w:write, r:read, a: add, b:binary
saveFile1.write(f) # f데이터를 쓰겠다.
saveFile1.close()
```

- with
close는 with로 대체할 수 있다.

```
with open(savePath2,'wb') as saveFile2:
    saveFile2.write(f2)
```

- urlopen

```
import urllib.request as req
from urllib.parse import urlparse

url="http://www.naver.com"

mem=req.urlopen(url)

# 자료형 알아보기
print(type(mem))
<class 'http.client.HTTPResponse'>

print(type({}))
<class 'dict'>

print(type([]))
<class 'list'>

print(type(()))
<class 'tuple'>

print("geturl",mem.geturl())
geturl https://www.naver.com/

print("status",mem.status) #200, 404, 403, 500
status 200

print("headers",mem.getheaders())

print("info",mem.info) # headers의 줄바꿈버전
print("code",mem.getcode()) # status와 같음.
code 200
print("read",mem.read(20)) #가져올 만큼만 가져옴. 위에만 필요한 게 있으면 자를 수 있음.
```

```
print("read",mem.read(50).decode
<html lang="ko">
<

print(urlparse("http://www.naver.com?test=test"))
```

참고 사이트
인프런
파이썬 입문 및 웹 크롤링을 활용한 다양한 자동화 어플리케이션 제작하기
https://www.inflearn.com/course/python-%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EC%9B%B9-%EB%8D%B0%EC%9D%B4%ED%84%B0-%ED%81%AC%EB%A1%A4%EB%A7%81/dashboard