학습 정리

팀	빅나물	구성원	이지현, 조영현

일정	발제자	주제
9일차(6/7)	조영현	BeautifulSoup 사용법 및 간단 웹 파싱 기초(복습)

주요 내용 요약

- **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.

- **urlretrieve**
  파싱이 필요없는 데이터 한 번에 다운로드 받는 경우 좋음.

- **urlopen**
  중간 작업이 필요한 경우는 urlopen이 좋음.

- **open, write, close**

- **with**

  close는 with로 대체할 수 있다.

- **urlopen**

- **urlencode**

- **웹 파싱**: 궁극적으로 어떤 html파일(태그, 요소, 속성 등으로 구성) 잘 분석해서 내가 원하는 데이터의 위치를 찾는 것.

  ex) 이미지 경로, 동영상 경로, 문자, 숫자 등

- **BeautifulSoup이용하면 손쉽게 파싱 가능.**

  - anaconda prompt에서 beautifulSoup 설치

- **urljoin**

- **""" """**
  줄 바꿈이 포함되어 있는 문자열

- **beautifulsoup 기초**

  - **prettify()** : html 자동 들여쓰기해서 출력.

  - **soup.html.body.h1**: soup 객체 html태그 안의 body 태그 안의 h1태그 선택

  - **h1.string**: 문자열만 출력

  - p 태그는 여러 개 있으므로 soup.html.body.p의 경우 첫번째 노드만 가져옴.

    따라서, next_sibling을 이용해 다음 p태그에 접근.

    `"""  """`의 경우, 각 줄 마지막에 \n이 포함되어 있기 때문에 next_sibling을 하면 \n가 선택됨.

    따라서, next_sibling.next_sibling을 해주어야 다음 태그가 출력됨. 물론 한 줄 쓰기 경우엔 그럴 필요 없음.

  - previous_sibling의 경우도 next_sibling과 마찬가지이나 이전 태그로 이동.

- **태그 선택자 이용해 한번에 가져오기**

  - .find_all("a"): 모든 a태그 가져옴.

  - .find_all("a",string="daum"): a 태그 중 string이 조건에 맞는 것만 가져옴.

  - .find("a"): a태그 상위 1개만 가져옴.

  - .find_all("a",limit=3): "a" 태그 중 상위 3개만 가져옴.

  - .find_all(string=["naver","google"]): string에 해당하는 string만 가져옴. 보통 쓸일 x.

  - type이 resultset의 경우 for문 이용해서 출력해야함.

  - .attrs[key 값]: key 속성의 value 출력

    href=""에서 href가 key ""가 value

- css 선택자 이용해서 조건에 맞는 element 가져오기

  - .select("div#main > h1"): div태그에서 id가 main인 것의 하위의 h1

  - print('h1',h1.string)
    에러남. 왜냐하면 h1의 type이 list이기 때문에 바로 리스트의 속성에 접근 불가능.

    따라서 하나라도 반복문을 돌려야 함.

  - 하지만 번거로움 피하기 위해 .select_one 사용하면 됨. 하나인 경우!

- 정규 표현식 활용
  li=soup.find_all(href=re.compile(r"^https://"))

  # .prettify() 이쁘게 보여주기 위해

- **3Vs**

  Volume: 데이터의 양

  Variety: 데이터의 다양성

  Velocity: 데이터의 속도

  - 브라우저를 이용해서 접속한 경우, user-agent 값이 브라우저로 request로 날아감.

    네이버에서 이 값을 보고 response를 해줄 때(브라우저에서 최종적으로 rendering될 때),

jquery라던지 그 프레임워크에 따라 화면이 만들어짐.

but, 실습 시, 브라우저를 사용한 것이 아니라 http 통신으로 값을 날렸기 때문에 태그 요소와 브라우저에서 보는 것이 다를 수 있다.

src값이 아니라 data-source값이 이미지 소스 값이 들어 있음!

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