

Week 9 • 데이터 저널리즘

Social Data Mining 01

Joonhwan Lee

human-computer interaction + design lab.

오늘 다룰 내용

- Crawling from websites

1. Crawling from Websites

웹 데이터 수집

- ♦ RQ: 어떤 사람의 트위터 팔로워 구성을 통해 그 사람의 성향을 유추할 수 있을까?
 - ♦ 예1: A라는 사람의 트위터 팔로워는 모두 500명, 그 중에 30% 정치인, 60%는 연예인 → 연예 정보에 관심이 많은 사람.
 - ♦ 예2: A라는 사람이 팔로우하는 정치인 중, 보수성향 정치인 10%, 진보성향 정치인 90% → 진보적인 성향을 가진 사람.
 - ♦ Q1: 팔로우하는 사람의 속성 (연예인인지, 정치인인지, 보수성향의 정치인인지 등)은 어떻게 수집하나..?

웹 데이터 수집

http://twtkr.com/fpl.php?d=3_1&n=

The screenshot shows the twtkr directory website. The main content area displays a list of users with their profile pictures, names, and follower counts. The users are ranked by follower count, with the top four being:

- 유시민 (@u_simin) with 533,403 followers (#1)
- 정봉주 (@BBK_Sniper) with 390,192 followers (#2)
- 김용민 (@funro) with 378,190 followers (#3)
- 문성근 (민주당,배우) (@actormoon) with 242,760 followers (#4)

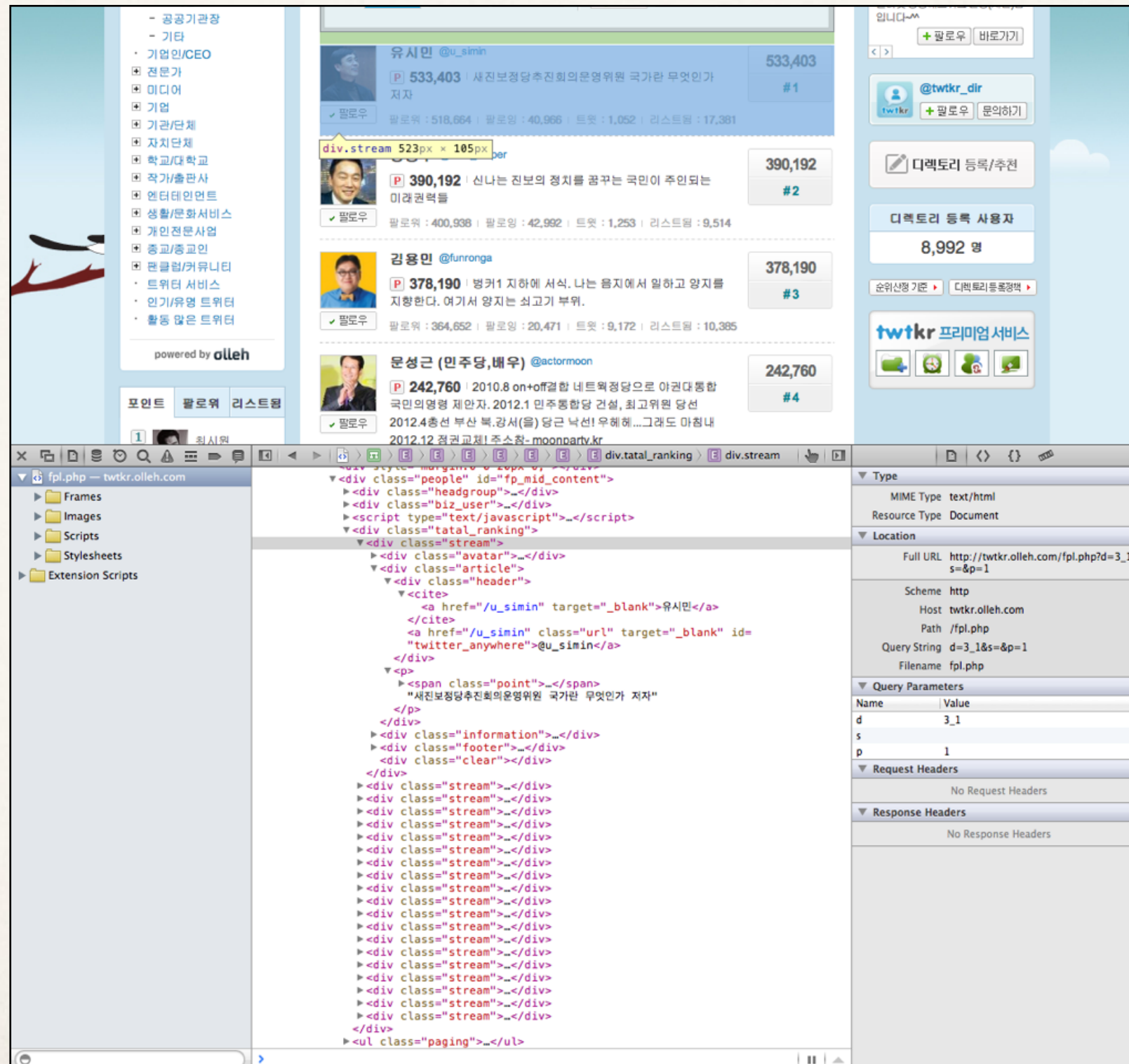
Red arrows point to the profile pictures of 유시민, 정봉주, and 김용민. A red arrow also points to the '팔로워' (Followers) button for 유시민.

The left sidebar contains a '순위 디렉토리' (Ranking Directory) with categories such as:

- 전체
- 전체(연예인 제외)
- 연예인(아이돌)
- 연예인
- 스포츠
- 정치인/공직자
- 기업인/CEO
- 전문가
- 미디어
- 기업
- 기관/단체
- 자치단체
- 학교/대학교
- 작가/출판사
- 엔터테인먼트
- 생활/문화서비스
- 개인전문사업
- 종교/종교인
- 팬클럽/커뮤니티
- 트위터 서비스
- 인기/유명 트위터
- 활동 많은 트위터

The right sidebar contains a 'twtkr스폰서' (twtkr Sponsors) section with a list of sponsors and their logos.

http://twtkr.com/fpl.php?d=3_1&n=



웹 데이터 수집

- ◆ 실습: 소스코드 분석

- ◆ 수집하려는 웹 페이지의 소스를 분석하여, 필요한 데이터가 담긴 반복되는 패턴블럭을 찾아낸다.
- ◆ 반복되는 패턴블럭의 계층 구조를 찾아내 각각의 요소를 정리한다.
- ◆ 계층 구조 내에서 필요한 요소를 따로 찾아 정리한다.
- ◆ twtkr_example.html을 열고 주요한 데이터의 반복되는 패턴블럭을 찾고, 내부 데이터를 구조화 하시오.

웹 데이터 수집

◆ 실습

```
<div class="total_ranking">
```

```
<div class="stream">
```

```
<div class="avatar">
```

```
<div class="article">
```

```
<div class="header">
```

```
<cite>
```

```
...
```

```
<div class="stream">
```

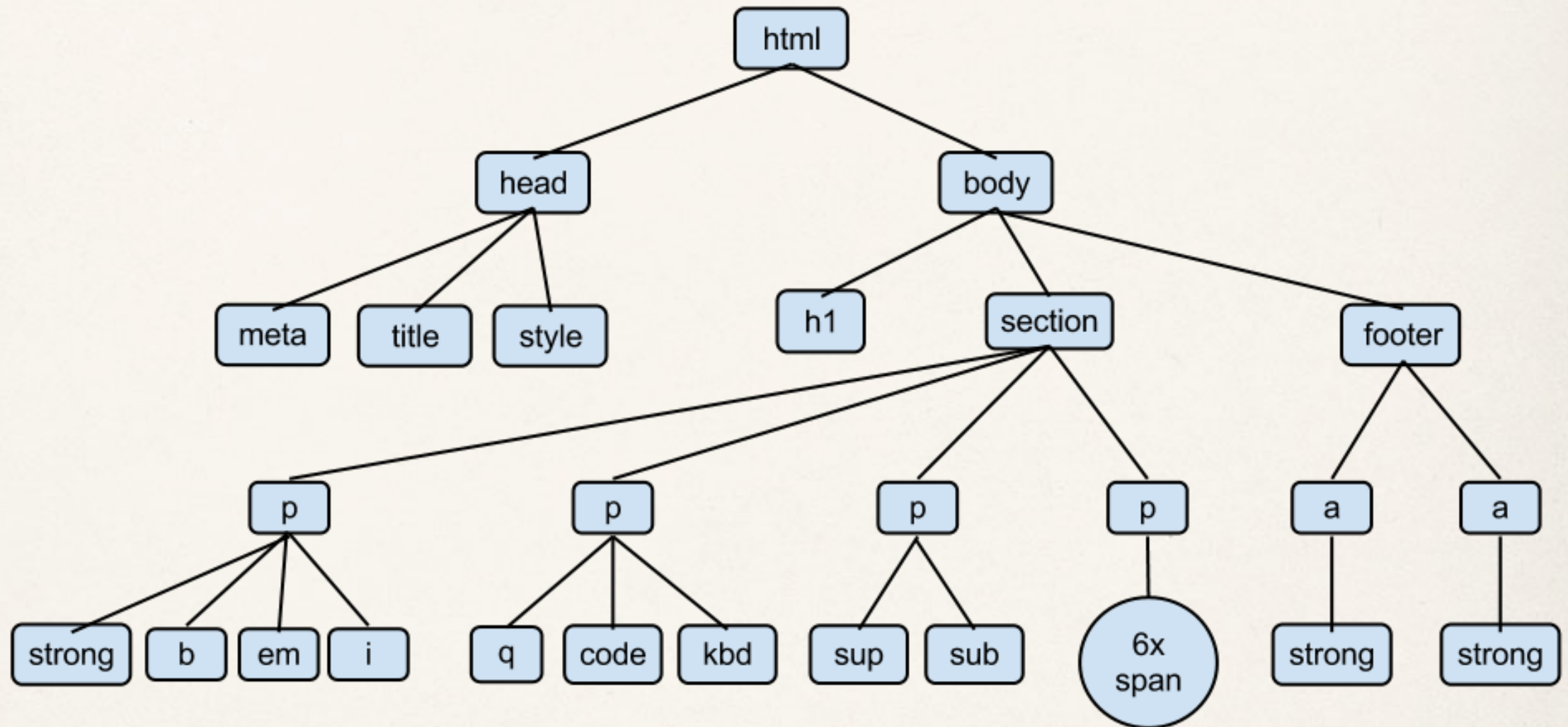
```
<div class="stream">
```

```
...
```

BeautifulSoup을 이용한 웹페이지 수집 및 분석

- ✦ 웹 문서로 부터 특정한 데이터를 추출하기 위해서는 HTML 문서를 읽고 구조를 해석할 수 있는 소프트웨어가 필요.
- ✦ BeautifulSoup은 HTML, XML 등을 읽고 해석할 수 있는 소프트웨어 (parser)
 - ✦ BS4는 문서를 파싱한 후 DOM Tree 를 만든다.
- ✦ BeautifulSoup 설치
 - ✦ `pip install beautifulsoup4`

HTML Document 와 DOM Tree



HTML Document 와 DOM Tree

The Document

```
<html>
<body>
<h1>Title</h1>
<p>A <em>word</em></p>
</body>
</html>
```

The DOM Tree

```
DOCUMENT
├── ELEMENT: html
│   ├── TEXT: '\n'
│   ├── ELEMENT: body
│   │   ├── TEXT: '\n'
│   │   ├── ELEMENT: h1
│   │   │   └── TEXT: 'Title'
│   │   ├── TEXT: '\n'
│   │   ├── ELEMENT: p
│   │   │   ├── TEXT: 'A'
│   │   │   └── ELEMENT: em
│   │   │       └── TEXT: word
│   └── TEXT: '\n'
└── TEXT: '\n'
```

BS4를 이용한 HTML Parsing

✦ BeautifulSoup의 사용

```
> from bs4 import BeautifulSoup
> html_doc = "<html><body><h1>Mr. Belvedere Fan
Club</h1></body></html>"

> soup = BeautifulSoup(html_doc, "html.parser")
> soup
=> <html><body><h1>Mr. Belvedere Fan Club</h1></
body></html>

> print(soup.prettify())

> heading = soup.find_all("h1")
=> [<h1>Mr. Belvedere Fan Club</h1>]

> heading[0].get_text()
=> 'Mr. Belvedere Fan Club'
```

BS4를 이용한 HTML Parsing

♦ find_all 의 사용법

- ♦ `find_all("h1")`

- ♦ `<h1>~</h1>` 태그 안의 내용

- ♦ `find_all("div")`

- ♦ `<div>~</div>` 태그 안의 내용

- ♦ `find_all("div", class_="footer")`

- ♦ `<div class="footer">~</div>` 태그 안의 내용

- ♦ `find_all("div", id="footer")`

- ♦ `<div id="nav">~</div>` 태그 안의 내용

- ♦ `divs = soup.find_all("div", class_="header")`

- `for div in divs:`

- `if div.a["href"] == "twitter_anywhere":`

- ♦ `<div class="header">~</div>` 태그 안의 내용

BS4를 이용한 HTML Parsing

♦ find_all의 사용법

- ♦ find_all이 반환하는 값은 array (한 페이지에 같은 요소가 여럿 있을 것을 가정하므로...)
- ♦ 따라서 find_all이 수집한 데이터를 처리하기 위해서는 for-loop 등의 iterator 를 사용한다.

```
id_list = []  
divs = soup.find_all("div", class_="header")  
for div in divs:  
    if div.a["href"] == "twitter_anywhere":  
        id_list.append(div.a.text)
```

twitter 아이디와 사용자 이름 수집

- ✦ twtkr_example.html 파일을 읽어 트위터 아이디와 사용자 이름을 수집해 보자. 수집된 id 에서 @ 기호를 삭제하여 출력한다.

- ✦ 예: u_simin, 유시민

- ✦ (참고) HTML 파일 불러오는 방법

```
with open("data/twtkr_example.html") as  
file:
```

```
    html_doc = file.read()
```

웹에서 직접 데이터 수집

- ✦ 항상 저장된 페이지에서 파일을 수집할 수 없음.
- ✦ 실시간으로 웹페이지에 접속해서 저장된 페이지를 수집해야 함.
- ✦ 인터넷에 접속하여 페이지의 소스코드를 받아 처리하기 위해서는 다음과 같은 명령어를 사용.
 - ✦

```
import urllib.request  
with urllib.request.urlopen("http://twtkr.com/fpl.php?d=3&n=20") as url:  
    doc = url.read()
```

Questions?
