

# 파이썬X웹크롤링 업무자동화

파이썬을 통한 웹크롤링 강의

- 파이썬을 통해서 웹크롤링을 자유자재로 해보자
- 웹크롤링 뿐만아니라 가져온 데이터를 분석하고 시각화 하자
- 자동화를 통해 업무를 좀 더 편하게 보자

- 유튜브영상 / pdf 파일 제공
- 카카오톡 오픈채팅방으로 수강신청 및 질문답변
- 깃허브/네이버블로그에 pdf파일, 유튜브주소, 소스코드 제공
- 총 4주차로 진행할 예정입니다. ( 상황에 따라 더 빨리 될 수도 있습니다 )
- 자세한 강의소개는 **깃허브(<http://github.com/etilelab>)** 네이버블로그(<http://blog.naver.com/luckperson7>)을 참고해주세요.

- Etilelab 공식 네이버블로그 : <http://blog.naver.com/luckperson7> -> 파이썬x웹크롤링 카테고리 참고
- **Etilelab 깃허브 : <https://github.com/etilelab> -> webcrawling 레파지토리 참고**
- 유튜브 채널 : [https://www.youtube.com/channel/UC8trp6SJGnFGyvC1tT\\_4JLw](https://www.youtube.com/channel/UC8trp6SJGnFGyvC1tT_4JLw)

# 파이썬X웹크롤링 1강

beautifulsoup를 통한 간단한 웹크롤링 실습

- Beautifulsoup, requests 라이브러리의 설치와 import
- 간단한 실습을 통한 웹크롤링 익숙해지기

- pip3 install requests
- pip3 install bs4

```
import requests  
from bs4 import BeautifulSoup
```



# 실습 문제 1

beautifulsoup를 통한 간단한 웹크롤링

- 홈페이지(<http://lambutan.dothome.co.kr>)에 접속
- 표의 데이터들을 크롤링 해보자

HOME

## Jsoup test page with bootstrap

This page is jsoup test page  
Double clicks this page table, you can look professor information

이 표의내용을 크롤링 해오는것이 목표

Search words

#	Professor	Lecture names	other	Grades	Evaluation
1	John	C language	Anything	2	1P
2	Kim	JAVA	Anything	3	2.5P
3	Gun	PHP / python	Anything	1	5P
4	Hun	Soccer	Special	2	5P
5	...	...	...	...	...

If you double clicks this table, you look detail lecture information.

- requests 라이브러리를 이용해 홈페이지에 접속 후 컨텐츠 가져오기

```
r=requests.get("http://lambutan.dothome.co.kr/")  
c=r.content
```

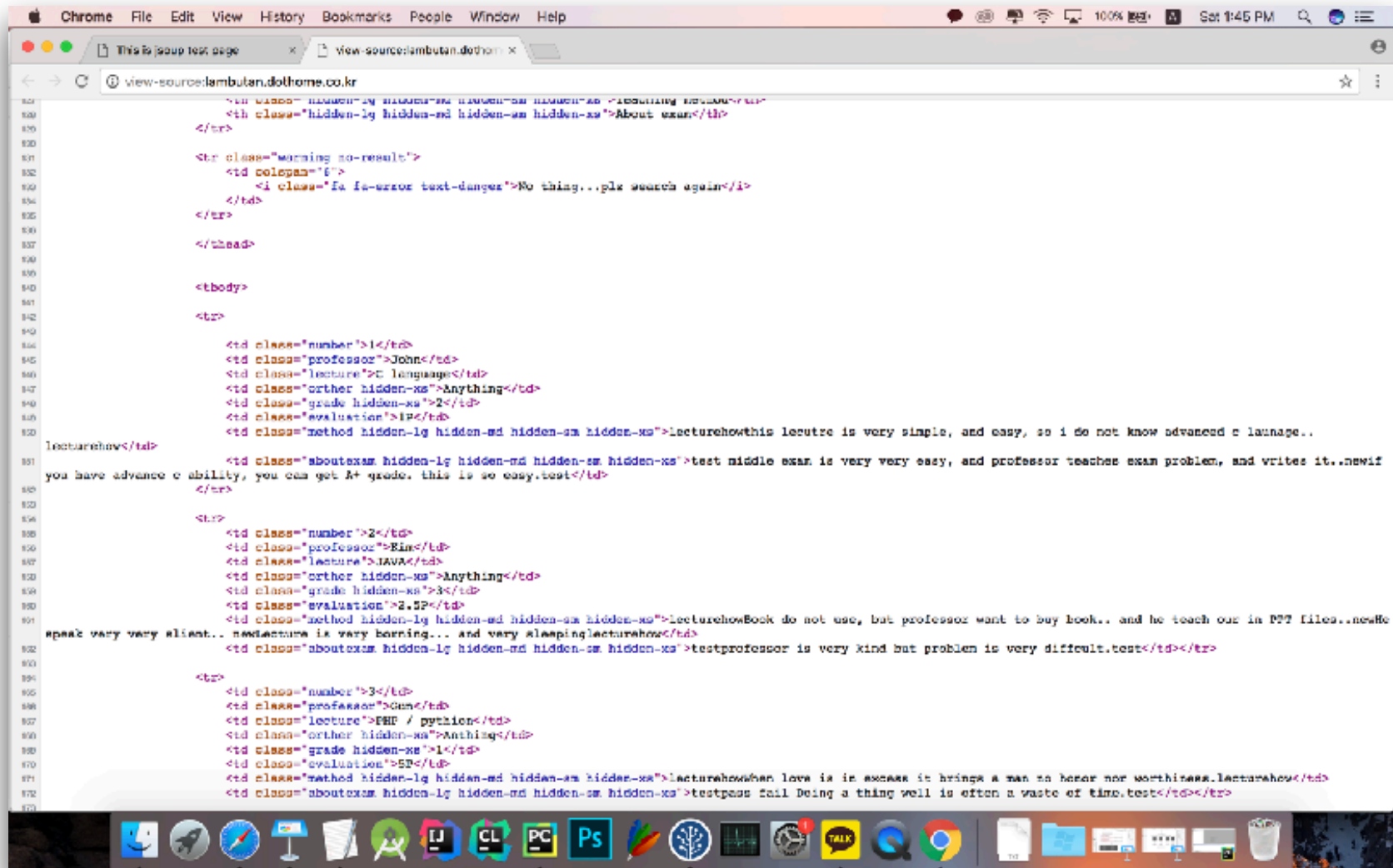


그렇게 가져온 내용은 가독성이 떨어짐

- BeautifulSoup의 html.parser 메소드로 보기 쉽게 정렬하기

```
soup=BeautifulSoup(c,"html.parser")
```

- 오른쪽마우스 -> 소스코드 보기 클릭



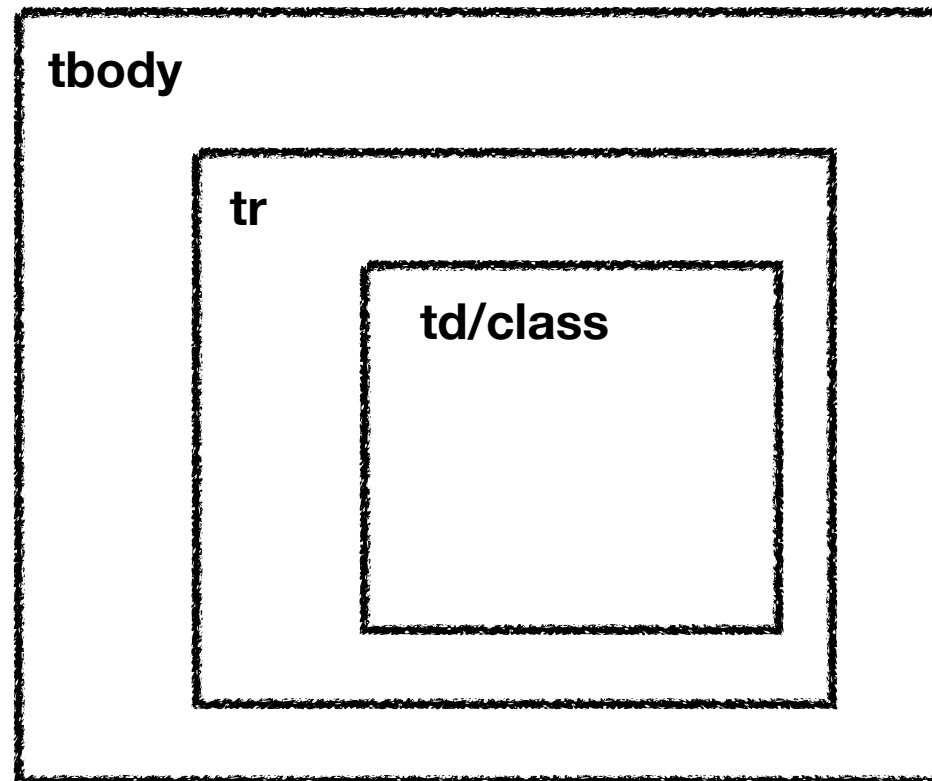
- 우리가 원하는 데이터들이 어떤 태그로 구성되어있는가 확인 !

```

<tbody>
<tr>
  <td class="number">1</td>
  <td class="professor">John</td>
  <td class="lecture">C language</td>
  <td class="orther hidden-xs">Anything</td>
  <td class="grade hidden-xs">2</td>
  <td class="evaluation">1P</td>
  <td class="method hidden-lg hidden-md hidden-sm hidden-xs">lecturehowthis lecutre is very simple, and easy, so i do not know advanced c launage.. lect
  <td class="aboutexam hidden-lg hidden-md hidden-sm hidden-xs">test middle exam is very very easy, and professor teaches exam problem, and writes it..
</td>
</tr>
<tr>
  <td class="number">2</td>
  <td class="professor">Kim</td>
  <td class="lecture">JAVA</td>
  <td class="orther hidden-xs">Anything</td>
  <td class="grade hidden-xs">3</td>
  <td class="evaluation">2.5P</td>
  <td class="method hidden-lg hidden-md hidden-sm hidden-xs">lecturehowBook do not use, but professor want to buy book.. and he teach our in PPT files..
  <td class="aboutexam hidden-lg hidden-md hidden-sm hidden-xs">testprofessor is very kind but problem is very difficult.test</td></tr>
<tr>
  <td class="number">3</td>
  <td class="professor">Gun</td>
  <td class="lecture">PHP / pythion</td>
  <td class="orther hidden-xs">Anthing</td>
  <td class="grade hidden-xs">1</td>
  <td class="evaluation">5P</td>
  <td class="method hidden-lg hidden-md hidden-sm hidden-xs">lecturehowWhen love is in excess it brings a man no honor nor worthiness.lecturehow</td>
  <td class="aboutexam hidden-lg hidden-md hidden-sm hidden-xs">testpass fail Doing a thing well is often a waste of time.test</td></tr>
<tr>
  <td class="number">4</td>

```

- 표는 tbody라는 태그로, 각 행은 tr이라는 태그로, 각 행의 열은 td라는 태그로 구성되어있음



- 우리가 가져올 데이터들의 구성 태그



첫째로, **tbody** 태그를 찾아야한다. 표 내용을 가져오는것이 목표이기 때문 !

- find 메소드로 태그를 찾자
- 특정 태그를 찾는 메소드
- 특정 태그가 하나만 존재할 때, 하나만 가져오고 싶을때 사용
- 특정 태그가 여러개면, 첫번째 태그를 가져옴

```
all=soup.find("tbody")
```

```
<tbody>
<tr>
<td class="number">1</td>
<td class="professor">John</td>
<td class="lecture">C language</td>
<td class="orther hidden-xs">Anything</td>
<td class="grade hidden-xs">2</td>
<td class="evaluation">1P</td>
<td class="method hidden-lg hidden-md hidden-sm hidden-xs">lecturehowth:
<td class="aboutexam hidden-lg hidden-md hidden-sm hidden-xs">test midd
```

둘째, tr 태그를 찾아야한다. 표 내용을 가져왔으면, 표의 행을 가져와야 하기 때문 !

- find\_all 메소드로 태그를 찾자

- 특정 태그를 찾는 메소드(여러개)

- 특정 태그가 여러개 존재할때 사용

- 특정 태그가 여러개면, 리스트처럼 저장함 (all[0], all[1] ...)

```
<tr class="">a</tr>
<tr class="">b</tr>
<tr class="">c</tr>
```

all2[0]  
all2[1]  
all2[2]

tr로 구성된 모든 태그를 찾아 저장하라

```
all2=all.find_all("tr", {"class":""})
```

```

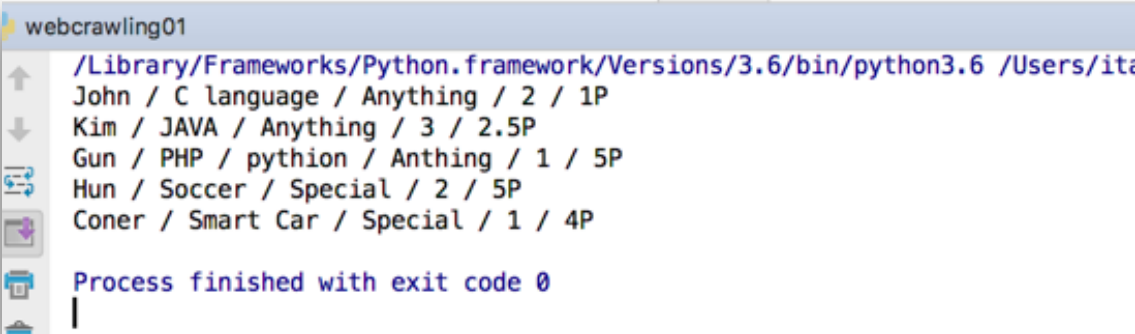
<tr>
<td class="number">1</td>
<td class="professor">John</td>
<td class="lecture">C language</td>
<td class="orther hidden-xs">Anything</td>
<td class="grade hidden-xs">2</td>
<td class="evaluation">1P</td>
<td class="method hidden-lg hidden-md hidden-sm hidden-xs">lecturehowthis lecutre is very simple,
<td class="aboutexam hidden-lg hidden-md hidden-sm hidden-xs">test middle exan is very very easy,
</tr>

```

all2[0] 를 출력한 결과

셋째, td 태그(class="") 를 찾아야한다. 각 행에서 우리가 원하는 데이터를 추출해내야 하기 때문  
우리가 원하는 데이터는 tbody(표)->tr(행)->td(열)에 존재

```
for item in all2:
    professor=item.find("td",{"class":"professor"}).text
    lectureName=item.find("td",{"class":"lecture"}).text
    orther=item.find("td",{"class":"orther"}).text
    grade = item.find("td", {"class": "grade"}).text
    evaluation=item.find("td", {"class": "evaluation"}).text
    print(professor + " / " + lectureName + " / " + orther + " / " + grade + " / " +
evaluation)
```



```
webcrawling01
/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6 /Users/ita
John / C language / Anything / 2 / 1P
Kim / JAVA / Anything / 3 / 2.5P
Gun / PHP / pythion / Anthing / 1 / 5P
Hun / Soccer / Special / 2 / 5P
Coner / Smart Car / Special / 1 / 4P

Process finished with exit code 0
|
```

출력결과

```

import requests
from bs4 import BeautifulSoup

r=requests.get("http://lambutan.dothome.co.kr/") # 홈페이지 접속
c=r.content # content(내용) 받아옴
soup=BeautifulSoup(c,"html.parser") # BeautifulSoup을 사용할수 있게 만들어 줌

all=soup.find("tbody") # tbody 라는 태그를 찾아 all이라는 변수에 저장
all2=all.find_all("tr",{"class":""}) # 각 행(tr태그이면서 class는 공백인)을 all2에 저장

for item in all2: # 각 행을 for 문으로 돌면서
    professor=item.find("td",{"class":"professor"}).text # td 라는 태그 class 는 professor(교수)를 찾는다
    lectureName=item.find("td",{"class":"lecture"}).text # td 라는 태그 class 는 lecture(강의)를 찾는다
    orther=item.find("td",{"class":"orther"}).text # 이하 같음
    grade = item.find("td", {"class": "grade"}).text
    evaluation=item.find("td", {"class": "evaluation"}).text
    print(professor + " / " + lectureName + " / " + orther + " / " + grade + " / " + evaluation) # 출력

```

<https://github.com/etilelab/webcrawling/blob/master/webcrawling01.py>