

Jin-Soo Kim  
(jinsoo.kim@snu.ac.kr)

Systems Software &  
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

*Python for Data Analytics*

# Text Processing



# Outline

- Introduction to Text Processing
- HTML
- BeautifulSoup

# Introduction to Text Processing

# Text as Data

## ■ Documents

- Articles, books and novels
- E-mails, web pages, blogs
- Tags, comments
- Computer programs, logs

## ■ Collections of documents

- Messages (e-mail, blogs, tags, comments)
- Social networks (personal profiles)
- Academic collaborations (publications)



# Why Analyze Text?

- Understanding the documents
  - Get the "gist" (요점, 요지) of a document
- Grouping the documents
  - Cluster documents for overview or classification
- Comparison of documents
  - Compare document collections, or inspect evolution of collection over time
- Correlation of documents
  - Compare patterns in text to those in other data, e.g., correlate with social networks

# What is Natural Language Processing?

- The study of human languages and how they can be represented computationally and analyzed and generated algorithmically

The cat is on the mat.  $\rightarrow$  on(mat, cat)

on(mat, cat)  $\rightarrow$  The cat is on the mat

- Building computational models of natural language comprehension and production
- Other names: Computational linguistics  
Human Language Technology  
Natural Language Engineering  
Speech and Text Processing  
(long time ago, Information Retrieval)

# NLP is becoming Popular (I)

- IBM Watson won Jeopardy in 2011

William Wilkinson's "An Account of the Principalities of Wallachia and Moldavia" inspired this author's most famous novel



***Bram Stoker***

- Information extraction

volunteer orientation    Inbox x

Bill S. Brown    10:35 AM (8 minutes ago) ☆

to me ▾

Hi John,

Thanks for signing up to be a volunteer tutor! To get started, you need to attend one of our volunteer orientations. We have a session tomorrow at 3pm, but if that doesn't work, our other upcoming sessions are:

6pm on Friday or  
3pm next Tuesday

volunteer orientation    Tue, May 7, 2013

Tue, May 7, 2013 ▾    8am  
Jogging time

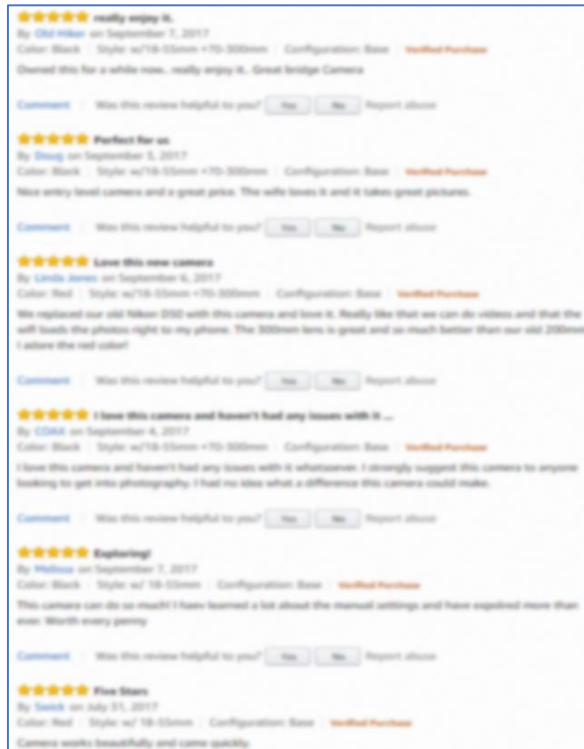
3:00pm ▾    3pm  
volunteer orientation

7pm  
Comedy show

Add to Calendar

# NLP is becoming Popular (2)

## Information extraction & sentiment analysis



### Attributes:

zoom

affordability

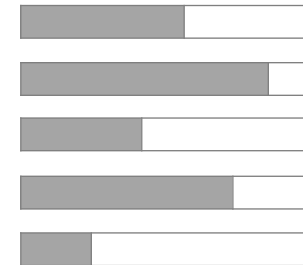
size and weight

flash

ease of use

pos

neg



### Size and weight:

- Nice and compact to carry! → pos
- Since the camera is small and light, I won't need to carry around those heavy, bulky professional cameras either! → pos
- The camera feels flimsy, is plastic and very light in weight you have to be very delicate in the handling of this camera → neg



# NLP is becoming Popular (3)

## ■ Machine translation

- Fully automatic

Input    我这心里七上八下的

Output    마음이 조마조마해

- Helping human translators

Enter Source Text:

تعرض الرئيس اللبناني اميل لحود ل# حملة عنيفة في مجلس النواب الذي انعقد امس في جلسة تشريعية علنية تحولت الي " محاكمة " ل# رئيس الجمهورية علي موقفه من المحكمة الدولية و " الملاحظات " التي ادلي بها +ها حول هذا الموضوع .

Translate   Clear

Enter Translation:

lebanese |

- president
- suffered
- exposed
- president emile
- before
- presented
- offer

Done!

HTML

# Web Pages

- **Hypertext Markup Language (HTML)** is the main language used to define how a Web page should look
- Web pages are created, stored, and sent in HTML encoded form
- A browser converts HTML codes to what we see on the screen
- Features like background color, font, and layout are specified in HTML
  - **HTML tags** are for basic skeletons of documents
  - **CSS rule sets** are for styling and embellishing
- **HTML 5** is the newest and best WWW standard language

# Browsing a Web Page

## ■ <https://cse.snu.ac.kr>

The image shows a web browser displaying the homepage of the Seoul National University Department of Computer Science and Engineering. The page features a header with the university's logo and name, followed by a main content area with several news items and a sidebar with navigation links. A right-click context menu is open over the page, showing options like '메모 추가' (Add Memo), '이 페이지 공유' (Share this page), and '장치에 미디어 캐스트' (Cast media to device). The browser's developer tools are also open, showing the HTML structure of the page, including the DOCTYPE, head section with meta tags, and the body content.

서울대학교 컴퓨터공학부  
Seoul National University  
Dept. of Computer Science and Engineering

2019년 8월 우수학위논문  
수상자 안내

김석준 박사, 미국 시러큐스  
대학교 교수로 임용

바이오 지능 연구실 학생들,  
2019 국제 로보컵 준우승

서울대학교 컴퓨터공학부에서는 매 학기  
졸업생을 대상으로 우수학위논문상을 수  
여합니다.

김석준 박사가 미국의 명문 사립대학인  
시러큐스 대학교의 전기전자컴퓨터공학  
조교수로 임용되었습니다.

2019 국제 로보컵 대회(RoboCup)에서 바  
이오 지능 연구실 학생들이 숙한 연합 로  
봇팀으로 우승을 차지했습니다.

Top Conference List  
컴퓨터공학부 Top Conference List (바로 가기)

학부생 교내장학금 추가 신청  
신청기간: 2020. 1. 13.(월) 09:00 ~ 1. 17.(금) 23:59  
신청방법: 마이스누에서 신청

새소식  
1/6 (월) 창의성은 지루한 기초학문의 고  
12/2 (월) 프로그래밍언어 분야 10년 연  
11/21 (목) 편광기반 위치추적기술로 V  
11/20 (수) SW미래인재 교과과정, 고교  
11/19 (화) 2019년 컴공인의 밤 개최  
11/13 (수) CG 기술로 뇌영상의 한자리의 정형수술 효과... 1/15 (수) 2020 DREAM ON 서울대학교 이공계

학부 소개  
학부 소식  
찾아오는 길  
컴퓨터미래인재양성사업단

교수진  
연구실 목록  
Top Conference List  
서울대학교-SCSC

전기·전자·컴퓨터 분야 미래 7대 기술  
컴퓨터연구소  
해동학술정보실  
Facebook Group

```
1 <!DOCTYPE html>
2 <html lang="ko" dir="ltr">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="shortcut icon" href="https://cse.snu.ac.kr/sites/all/themes/dcse/favicon.ico" type="image/x-icon">
6 <link rel="alternate" type="application/rss+xml" title="공지사항" href="https://cse.snu.ac.kr/departemen...>
7 <link rel="alternate" type="application/rss+xml" title="새소식" href="https://cse.snu.ac.kr/news.xml" />
8 <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
9 <title>서울대학교 컴퓨터공학부</title>
10 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_fv...>
11 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_d21Mou7...>
12 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_tK11-CC...>
13 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_dn8VFLu...>
14 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_dn8VFLu...>
15
16 <!--[if lte IE 8]>
17 <link type="text/css" rel="stylesheet" href="https://cse.snu.ac.kr/sites/default/files/css/css_RjdLjrH...>
18 <![endif]-->
19 <meta name="google-translate-customization" content="6a6d90d7527df5b2-d0c482b2e806f2cb-g4c306c95986...>
20 </head>
21 <body class="html front not-logged-in no-sidebars page-home i18n-ko">
22 <div id="skip-link">
23 <a href="#navigation" class="element-invisible element-focusable">내비게이션으로 건너뛰기</a>
24 </div><!-- // -->
25 <div id="page-wrapper"><div id="page">
26
27 <div id="header">
28 <div id="header-top"><div class="section">
29 <div id="menu-wrapper">
30 <div id="login-block" class="block"><a href="/user/login">로그인</a></div>
31 <div class="region region-secondary">
32 <div id="block-locale-language" class="block block-locale first last odd">
33
34
35 <div class="content">
36 <ul class="language-switcher-locale-url"><li class="ko first active"><a href="/" class="language-li...>
37 <li class="en last"><a href="/en" class="language-link">English</a></li>
38 </ul> </div>
39 </div><!-- /.block -->
40 </div><!-- /.region -->
41 </div><!-- /.region -->
42 </div> <!-- #menu-wrapper -->
43 </div></div> <!-- /.section /.header-top -->
44 </div></div>
```

# HTML Page Structure

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

# Required Tags

- HTML tags that are required for every Web page:

- `<head>` tag: 문서전체에 적용되는 정보

- `<title>` 문서 제목
- `<script>` 클라이언트측 스크립트 정의
- `<style>` 문서의 style 정보 정의
- `<meta>` HTML 문서에 대한 메타데이터 정의

- `<body>` tag: 문서의 contents 정보

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Title</title>
  </head>
  <body>
    <p>Content</p>
  </body>
</html>
```

# HTML Tags

- Over 118 tags! (including HTML5)
  - Each tag has 0 to 30 attributes
  - <http://www.w3schools.com/html/default.asp>

<!-->	<!DOCTYPE>	<a>	<abbr>	<acronym>	<address>
<applet>	<area>	<article>	<aside>	<audio>	<b>
<base>	<basefont>	<bdi>	<bdo>	<big>	<blockquote>
<body>	 	<button>	<canvas>	<caption>	<center>
<cite>	<code>	<col>	<colgroup>	<datalist>	<dd>
<del>	<details>	<dfn>	<dialog>	<dir>	<div>
<dl>	<dt>	<em>	<embed>	<fieldset>	<figcaption>
...					

# Text Formatting

## Text Formatting

`<h?> ... </h?>`

`<b> ... </b>`

`<i> ... </i>`

`<u> ... </u>`

`<strike> ... </strike>`

`<sup> ... </sup>`

`<sub> ... </sub>`

`<small> ... </small>`

`<tt> ... </tt>`

`<pre> ... </pre>`

`<blockquote> ... </blockquote>`

`<strong> ... </strong>`

`<em> ... </em>`

`<font> ... </font>`

Heading (?= 1 for largest to 6 for smallest, eg h1)

Bold Text

Italic Text

Underline Text

Strikeout

Superscript - Smaller text placed below normal text

Subscript - Smaller text placed below normal text

Small - Fineprint size text

Typewriter Text

Pre-formatted Text

Text Block Quote

Strong - Shown as Bold in most browsers

Emphasis - Shown as Italics in most browsers

Font tag obsolete, use [CSS](#). (\*)



# Section Divisions

## Section Divisions

`<div> ... </div>`

Division or Section of Page Content

`<span> ... </span>`

Section of text within other content

`<p> ... </p>`

Paragraph of Text

`<br>`

Line Break

`<hr>`

Basic Horizontal Line

`<hr>` Tag Attributes:

`size="?"`

Line Thickness in pixels

`width="?"`

Line Width in pixels

`width="??%"`

Line Width as a percentage

`color="#??????"`

Line Colour (\*)

`align="?"`

Horizontal Alignment: `left`, `center`, `right` (\*)

`noshade`

No 3D cut-out

`<nobr> ... </nobr>`

Line Break

# Images and Linking Tags

## Images

```

```

**<img> Tag Attributes:**

```
src="url"  
alt="text"  
align="?"  
width="?"  
height="?"  
border="?"  
vspace="?"  
hspace="?"
```

Basic Image

URL or filename of image (required!)

Alternate Text (required!)

Image alignment within surrounding text (\*)

Image width (in pixels or %)

Image height (in pixels or %)

Border thickness (in pixels) (\*)

Space above and below image (in pixels) (\*)

Space on either side of image (in pixels) (\*)

## Linking Tags

```
<a href="url"> link text </a>
```

**<a> Tag Attributes:**

```
href="url"  
name="?"  
target="?"  
href="url#bookmark"  
href="mailto:email"
```

Basic Link

Location (url) of page to link to.

Name of link (name of anchor, or name of bookmark)

Link target location: `_self`, `_blank`, `_top`, `_parent`.

Link to a bookmark (defined with `name` attribute).

Link which initiates an email (dependant on user's email client).

# Lists

## Lists

```
<ol> ... </ol>
<ul> ... </ul>
<li> ... </li>
<ol type="?">
<ol start="??">
<ul type="?">
<li value="??">
<li type="??">
<dl> ... </dl>
<dt> ... </dt>
<dd> ... </dd>
```

Ordered List

Un-ordered List

List Item (within ordered or unordered)

Ordered list type: **A, a, I, i, 1**

Ordered list starting value

Unordered list bullet type: **disc, circle, square**

List Item Value (changes current and subsequent items)

List Item Type (changes only current item)

Definition List

Term or phrase being defined

Detailed Definition of term

# Tables

## Tables

`<table> ... </table>`

`<table>` Tag Attributes:

`border="?"`

`bordercolor="#??????"`

`cellspacing="?"`

`cellpadding="?"`

`align="??"`

`bgcolor="#??????"`

`width="??"`

`height="??"`

`<tr> ... </tr>`

`<th> ... </th>`

`<td> ... </td>`

`<td>` Tag Attributes:

`colspan="?"`

`rowspan="?"`

`width="??"`

`height="??"`

`bgcolor="#??????"`

`align="??"`

`valign="??"`

`nowrap`

Define a Table

Thickness of outside border

Border [Colour](#)

Space between cells (pixels)

Space between cell wall and content

Horizontal Alignment: [left](#), [center](#), [right](#) (\*)

Background Colour (\*)

Table Width (pixels or %) (\*)

Table Height (pixels or %) (\*)

Table Row within table

Header Cell within table row

Table Cell within table row

Number of columns the cell spans across (cell merge)

Number of row a cell spans across (cell merge)

Cell Width (pixels or %) (\*)

Cell Height (pixels or %) (\*)

Background Colour (\*)

Horizontal Alignment: [left](#), [center](#), [right](#) (\*)

Vertical Alignment: [top](#), [middle](#), [bottom](#) (\*)

Force no line breaks in a particular cell

```
<table border="1">
  <caption>Country Data</caption>
  <tr>
    <th>Country</th>
    <th>Capital</th>
    <th>Language(s)</th>
  </tr>
  <tr>
    <td>Canada</td>
    <td>Ottawa</td>
    <td>English/French</td>
  </tr>
  <tr>
    <td>Iceland</td>
    <td>Reykjavik</td>
    <td>Icelandic</td>
  </tr>
  <tr>
    <td>Norway</td>
    <td>Oslo</td>
    <td>Norwegian</td>
  </tr>
</table>
```



Country	Capital	Language(s)
Canada	Ottawa	English/French
Iceland	Reykjavik	Icelandic
Norway	Oslo	Norwegian

# BeautifulSoup 4

# What is bs4 module?

- "bs4": The most famous python package for parsing HTML and XML documents
  - Since 2004 (original author: Leonard Richardson)
  - Useful for web scraping
- BeautifulSoup class
  - Transforming an HTML document into a complex tree of Python objects
  - A parse tree for parsed pages that can be used to extract data from HTML
  - We can easily navigate & search a parse tree
  - Convert incoming documents to Unicode and outgoing documents to UTF-8

```
>>> from bs4 import BeautifulSoup
```

# Installing Parsers for BeautifulSoup Class

Parser	Typical usage	Notes
Python's 'html.parser'	BeautifulSoup(markup, 'html.parser')	Not as fast as <b>lxml</b>
'lxml' HTML parser	BeautifulSoup(markup, 'lxml')	Very fast \$pip install lxml
'lxml' XML parser	BeautifulSoup(markup, 'lxml-xml')	
'html5lib' parser	BeautifulSoup(markup, 'html5lib')	\$ pip install html5lib

```
from bs4 import BeautifulSoup file

with open('doc.html') as fp:
    soup = BeautifulSoup(fp, 'html.parser')
```

```
from bs4 import BeautifulSoup string

html_doc = "<html><head> .... </html>"
soup = BeautifulSoup(html_doc, 'html.parser')
```

```
import requests network
from bs4 import BeautifulSoup

response = requests.get('http://cse.snu.ac.kr')
soup = BeautifulSoup(response.text, 'html.parser')
```

# BeautifulSoup Class: Attribute List

Functions		
<code>get()</code>	<code>find_all_next()</code>	<code>insert_after()</code>
<code>get_text()</code>	<code>find_next()</code>	<code>clear()</code>
<code>prettify()</code>	<code>find_all_previous()</code>	<code>extract()</code>
<code>find()</code>	<code>find_previous()</code>	<code>decompose()</code>
<code>find_all()</code>	<code>select()</code>	<code>replace_with()</code>
<code>find_parents()</code>	<code>append()</code>	<code>wrap()</code>
<code>find_parent()</code>	<code>new_tag()</code>	<code>unwrap()</code>
<code>find_next_siblings()</code>	<code>insert()</code>	
<code>find_next_sibling()</code>	<code>insert_before()</code>	



# Parsing

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
"""
```

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
soup
```

```
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters; and their names were
<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a> and
<a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>
<p class="story">...</p>
</body></html>
```

# Using Attributes

```
soup.title
```

```
<title>The Dormouse's story</title>
```

```
soup.title.name
```

```
'title'
```

```
soup.title.string
```

```
"The Dormouse's story"
```

```
soup.title.parent.name
```

```
'head'
```

```
soup.p
```

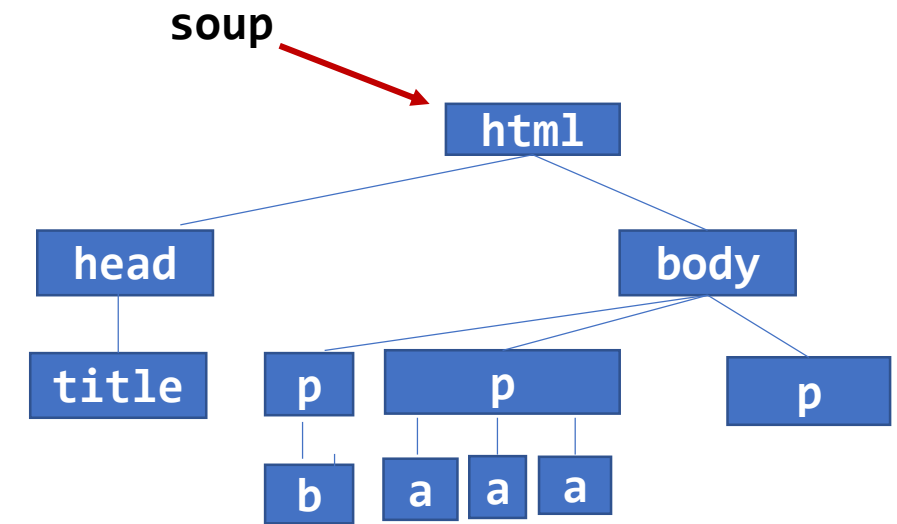
```
<p class="title"><b>The Dormouse's story</b></p>
```

```
soup.p['class']
```

```
['title']
```

```
soup.a
```

```
<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>
```



htmlparser가 만든 parse tree

# Using Functions

```
soup.find_all('a')
```

← Find all hyperlinks

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,  
 <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,  
 <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

```
soup.find(id='link3')
```

← Find all the matching attributes and values

```
<a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>
```

```
for link in soup.find_all('a'):  
    print(link.get('href'))
```

← Extract all the URLs found within a page 'a' tags

```
http://example.com/elsie  
http://example.com/lacie  
http://example.com/tillie
```

```
print(soup.get_text())
```

← Extract all the text from a page

The Dormouse's story

The Dormouse's story

Once upon a time there were three little sisters; and their names were  
Elsie,  
Lacie and  
Tillie;  
and they lived at the bottom of a well.  
...

# find\_all()

- `find_all(name, attrs, recursive, string, limit, ...)`
  - Looking through a tag's descendants and retrieves all descendants that match your filters

```
soup.find_all('title')
```

← Find the tag 'title'

```
[<title>The Dormouse's story</title>]
```

```
soup.find_all(id='link2')
```

← Find the attribute 'id' which has the value 'link2'

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

```
soup.find_all(id=True)
```

← Find the attribute 'id' which has a value

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,  
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,  
<a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

```
soup.find_all('a', class_='sister')
```

← Find a CSS class named 'sister' in the tag 'a'

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,  
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,  
<a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

# find\_all() (cont'd)

```
soup.find_all(string='Elsie')
```

```
['Elsie']
```

← Find the string 'Elsie'

```
soup.find_all('a', string='Lacie')
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

← Find the string 'Lacie' in the tag 'a'

```
soup.find_all('a', limit=2)
```

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,  
  <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

← Just return the maximum 2 results

```
soup.find_all('title', recursive=False)
```

```
[]
```

← Just find among direct children

```
soup.head.find_all('title', recursive=False)
```

```
[<title>The Dormouse's story</title>]
```

← Find the tag in the children of 'head'

```
soup('a')
```

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,  
  <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,  
  <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

← = soup.find\_all('a')

```
<html>  
  <head>  
    <title>  
      ...  
    </title>  
  </head>  
</html>
```

# find()

- Equivalent to `find_all(..., limit=1)`
- `find_all()` returns a list containing the single result, while `find()` just returns the result

```
soup.find('title')
```

```
<title>The Dormouse's story</title>
```

```
soup.find('p')
```

```
<p class="title"><b>The Dormouse's story</b></p>
```

```
soup.find('a')
```

```
<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>
```

# Retrieving Contents

```
soup.find('title').get_text()
```

← Get the text of the tag 'title'

```
"The Dormouse's story"
```

```
soup.find('title').text
```

← .text == .get\_text()

```
"The Dormouse's story"
```

```
for a in soup.find_all('a'):  
    print(a.get('href'))
```

← get the URL in the 'href' attribute

```
http://example.com/elsie  
http://example.com/lacie  
http://example.com/tillie
```

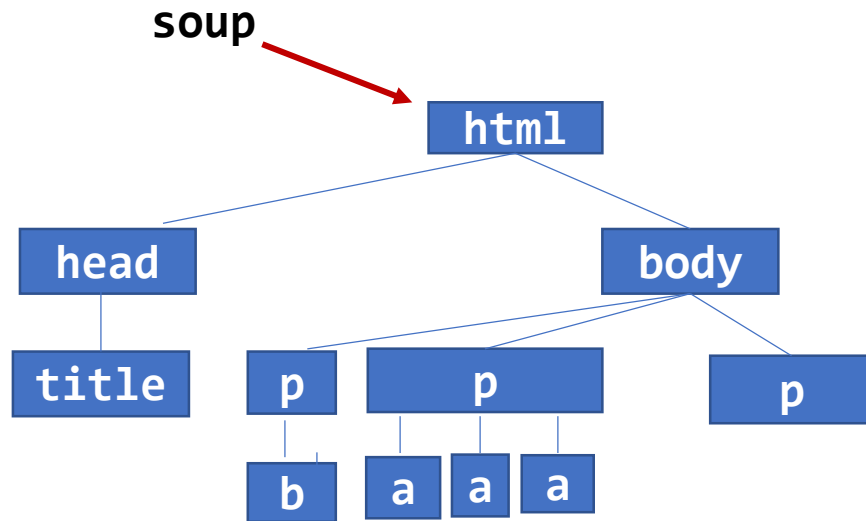
```
for a in soup.find_all('a', id=True):  
    print(a.get('id'))
```

← get the text in the 'id' attribute

```
link1  
link2  
link3
```

# Pretty Printing

## ■ prettify()



htmlparser가 만든 parse tree

```
print(soup.prettify())
```

```
<html>
<head>
  <title>
    The Dormouse's story
  </title>
</head>
<body>
  <p class="title">
    <b>
      The Dormouse's story
    </b>
  </p>
  <p class="story">
    Once upon a time there were three little sisters; and their names were
    <a class="sister" href="http://example.com/elsie" id="link1">
      Elsie
    </a>
    ,
    <a class="sister" href="http://example.com/lacie" id="link2">
      Lacie
    </a>
    and
    <a class="sister" href="http://example.com/tillie" id="link3">
      Tillie
    </a>
    ;
    and they lived at the bottom of a well.
  </p>
  <p class="story">
    ...
  </p>
</body>
</html>
```



# Saving HTML File from a Web Site

## Simple writing

```
import requests

url = 'https://cse.snu.ac.kr'
result = requests.get(url)

fp = open('cse.html', 'w')
fp.write(result.text)
fp.close()
```

## Pretty Printing

```
import requests
from bs4 import BeautifulSoup

url = 'https://cse.snu.ac.kr'
result = requests.get(url)

soup = BeautifulSoup(result.text, 'html.parser')
fp = open('cse.html', 'w')
fp.write(soup.prettify())
fp.close()
```

# Example

- 서울대학교 컴퓨터공학부 사이트에서 연구실 정보가 담긴 page를 scraping 한다.
- <https://cse.snu.ac.kr/research/labs>
- 위 페이지에서 "연구실, 지도교수, 연구실위치, 약자" 정보를 추출한다.
- 301동, 302동에 속한 연구실 목록으로 분류한다.

# Example: Target Page

연구실 목록 | 서울대학 X

https://cse.snu.ac.kr/research/labs

로그인 English

서울대학교 컴퓨터공학부  
Seoul National University  
Dept. of Computer Science and Engineering

소개 구성원 연구 입학 학사 및 교과 온라인서비스 신입교수초빙

홈 > 연구 >

### 연구실 목록

연구실	지도교수	연구실위치	약자
<a href="#">3차원 모델링 및 처리 연구실</a>	김명수	302동 315-1호 <a href="#">(02) 880-1840</a>	3MAP
<a href="#">계산 이론 및 알고리즘 공학 연구실</a>	스리니바사 라오 사티	301동 412호 <a href="#">(02) 880-1805</a>	TCS
<a href="#">데이터 마이닝 연구실</a>	강유	301동 519호 <a href="#">(02) 880-7263</a>	DM
<a href="#">데이터베이스 시스템 연구실</a>	문봉기	301동 418호 / 452-2호 <a href="#">(02) 880-6575</a>	DBS
<a href="#">머신러닝 연구실</a>	송현오	302동 319호	ML
<a href="#">멀티코어 컴퓨팅 연구실</a>	이재진	301동 515호 <a href="#">(02) 880-1837</a>	MCRL
<a href="#">메모리 및 스토리지 구조 연구실</a>	민상렬	301동 552호 / 517호 <a href="#">(02) 880-7296</a>	ARCHI
<a href="#">바이오지능 연구실</a>	장병탁	302동 314-1호 <a href="#">(02) 880-1835</a>	BI
<a href="#">분산시스템 연구실</a>	영허영 영현사	302동 311-2호 / 319호	DCS

연구 그룹  
연구 센터  
연구실 목록  
CSE Top Conference List

Select Language ▼

# Example: Table in HTML

```
<div class="content">
  <div class="view view-research-labs view-id-research_labs view-display-id-page view-dom-id-3686693d59000d614c0214a4bac5eb34">

    <div class="view-content">
      <table class="views-table cols-4" >
        <thead>
          <tr>
            <th class="views-field views-field-title" >
              연구실      </th>
            <th class="views-field views-field-field-faculty" >
              지도교수      </th>
            <th class="views-field views-field-field-office" >
              연구실위치      </th>
            <th class="views-field views-field-field-abbreviation" >
              약자      </th>
          </tr>
        </thead>
        <tbody>
          <tr class="odd views-row-first">
            <td class="views-field views-field-title" >
              <a href="/lab/3%EC%B0%A8%EC%9B%90-%EB%AA%A8%EB%8D%B8%EB%A7%81-%EB%B0%8F-%EC%B2%98%EB%A6%AC-%EC%97%B0%EA%B5%AC%EC%8B%A4">3차원 모델링 및 처리 연구실</a>      </td>
            <td class="views-field views-field-field-faculty" >
              <a href="/professor/%EA%B9%80%EB%AA%85%EC%88%98">김명수</a>      </td>
            <td class="views-field views-field-field-office" >
              302동 315-1호<br />(02) 880-1840      </td>
            <td class="views-field views-field-field-abbreviation" >
              3MAP      </td>
          </tr>
```

# Example: Extracting Information

```
1 import requests
2 import pandas as pd
3 from bs4 import BeautifulSoup
4
5 def get_cse_labs(url):
6     response = requests.get(url)
7     soup = BeautifulSoup(response.text, 'html.parser')
8     content = soup.find(id='content')
9     headers = [ header.text.strip() for header in content.find_all('th') ]
10    rows = content.find('tbody').find_all('tr')
11    items = [{k:v.get_text().strip() for k, v in zip(headers, row.find_all('td'))}
12              for row in rows]
13    return items
```

**headers** ['연구실', '지도교수', '연구실위치', '약자']

**items** [{ '연구실': '3차원 모델링 및 처리 연구실', '지도교수': '김명수', '연구실위치': '302동 315-1호(02) 880-1840', '약자': '3MAP' }, { '연구실': '계산 이론 및 알고리즘 공학 연구실', '지도교수': '스리니바사 라오 사티', '연구실위치': '301동 412호(02) 880-1805', '약자': 'TCS' }, { '연구실': '데이터 마이닝 연구실', '지도교수': '강유', '연구실위치': '301동 519호(02) 880-7263', '약자': 'DM' }, { '연구실': '데

*(list of dict)*

# Example: To Pandas

```
url = 'https://cse.snu.ac.kr/research/labs'
table = get_cse_labs(url)
df = pd.DataFrame(table, columns=table[0].keys())
df['연구동'] = df['연구실위치'].map(lambda s: s.split()[0])
df1 = df[df['연구동']=='301동']
df2 = df[df['연구동']=='302동']
df
```

	연구실	지도교수	연구실위치	약자	연구동
0	3차원 모델링 및 처리 연구실	김명수	302동 315-1호(02) 880-1840	3MAP	302동
1	계산 이론 및 알고리즘 공학 연구실	스리니바사 라오 사티	301동 412호(02) 880-1805	TCS	301동
2	데이터 마이닝 연구실	강유	301동 519호(02) 880-7263	DM	301동
3	데이터베이스 시스템 연구실	문봉기	301동 418호 / 452-2호(02) 880-6575	DBS	301동
4	머신러닝 연구실	송현오	302동 319호	ML	302동
5	멀티코어 컴퓨팅 연구실	이재진	301동 515호(02) 880-1837	MCRL	301동
6	메모리 및 스토리지 구조 연구실	민상렬	301동 552호 / 517호(02) 880-7296	ARCHI	301동
7	바이오지능 연구실	장병탁	302동 314-1호(02) 880-1835	BI	302동
8	분산시스템 연구실	엄현영, 엄현상	302동 311-2호 / 319호(02) 880-1856	DCS	302동
9	생물정보 및 생명정보 연구실	김선	301동 516호(02) 880-1784	BHI	301동

# Example: Results

df1

	연구실	지도교수	연구실위치	약자	연구동
1	계산 이론 및 알고리즘 공학 연구실	스리니바사 라오 사티	301동 412호(02) 880-1805	TCS	301동
2	데이터 마이닝 연구실	강유	301동 519호(02) 880-7263	DM	301동
3	데이터베이스 시스템 연구실	문봉기	301동 418호 / 452-2호(02) 880-6575	DBS	301동
5	멀티코어 컴퓨팅 연구실	이재진	301동 515호(02) 880-1837	MCRL	301동
6	메모리 및 스토리지 구조 연구실	민상렬	301동 552호 / 517호(02) 880-7296	ARCHI	301동
9	생물정보 및 생명정보 연구실	김선	301동 516호(02) 880-1784	BHI	301동
14	시스템 소프트웨어 및 구조 연구실	김진수	301동 517호(02) 880-7296	CSL	301동
15	실시간 유비쿼터스 시스템 연구실	이창건	301동 415호(02) 880-2562	RUBIS	301동
16	아키텍처 및 코드 최적화 연구실	이재욱	301동 554-1호(02) 880-1836	ARC	301동
17	양자정보 및 양자컴퓨팅 연구실	김태현	301동 416호(02) 880-4165	QUIQCL	301동
20	인간 중심 컴퓨터 시스템 연구실	이영기	301동 416호(02) 880-4165	HCS	301동
21	인터넷 데이터베이스 연구실	김형주	301동 453호(02) 880-1830	IDB	301동
22	인터넷 융합 및 보안 연구실	최양희, 권태경	301동 518호(02) 880-9147	NCSL	301동
24	지능형 데이터 시스템 연구실	이상구	301동 420호(02) 880-1859	IDS	301동
27	컴퓨터 시스템 및 플랫폼 연구실	버나드 에거	301동 419호(02) 880-1819	CSAP	301동
28	컴퓨터이론 및 응용 연구실	박근수	301동 414호(02) 880-1828	CTA	301동
30	통합설계 및 병렬 처리 연구실	하순희	301동 455-1호(02) 880-7292	CAP	301동

df2

	연구실	지도교수	연구실위치	약자	연구동
0	3차원 모델링 및 처리 연구실	김명수	302동 315-1호(02) 880-1840	3MAP	302동
4	머신러닝 연구실	송현오	302동 319호	ML	302동
7	바이오지능 연구실	장병탁	302동 314-1호(02) 880-1835	BI	302동
8	분산시스템 연구실	염현영, 염현상	302동 311-2호 / 319호(02) 880-1856	DCS	302동
10	소셜정보망 연구실	김종권	302동 310-1호(02) 880-1858	SCONE	302동
11	소프트웨어 원리 연구실	허충길	302동 312-2호(02) 880-1865	SF	302동
12	소프트웨어 플랫폼 연구실	전병곤	302동 420호(02) 880-1611	SPL	302동
13	시각 및 학습 연구실	김건희	302동 317호(02) 880-7289	VL	302동
18	운동 연구실	이제희	302동 312-1호(02) 880-1864	MRL	302동
19	이동 컴퓨팅 및 통신 연구실	전화숙	302동 313-1호(02) 880-1841	MCCL	302동
23	임베디드 시스템 연구실	김지홍	302동 315-2호(02) 880-1861	CARES	302동
25	최적화 및 금융공학 연구실	문병로	302동 313-2호(02) 880-1851	OPT	302동
26	컴퓨터 그래픽스 및 이미지 처리 연구실	신영길	302동 320호(02) 880-1860	CGIP	302동
31	프로그래밍 연구실	이광근	302동 312-2호(02) 880-1865	ROPAS	302동
32	휴먼-컴퓨터 인터랙션 연구실	서진욱	302동 314-2호(02) 880-7044	HCI	302동

# Example: Pandas to HTML

```
df1.to_html()
```

```
'<table border="1" class="dataframe">\n  <thead>\n    <tr style="text-align: right;">\n      <th></th>\n      <th>연구실</th>\n      <th>지도교수</th>\n      <th>연구실위치</th>\n      <th>약자</th>\n      <th>연구동</th>\n    </tr>\n  </thead>\n  <tbody>\n    <tr>\n      <th>1</th>\n      <td>계산 이론 및 알고리즘 공학 연구실</td>\n      <td>스리니바사 라오 사티</td>\n      <td>301동 412호(02) 880-1805</td>\n      <td>TCS</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>2</th>\n      <td>데이터 마이닝 연구실</td>\n      <td>강유</td>\n      <td>301동 519호(02) 880-7263</td>\n      <td>DM</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>3</th>\n      <td>데이터베이스 시스템 연구실</td>\n      <td>문봉기</td>\n      <td>301동 418호 / 452-2호(02) 880-6575</td>\n      <td>DBS</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>5</th>\n      <td>멀티코어 컴퓨팅 연구실</td>\n      <td>이재진</td>\n      <td>301동 515호(02) 880-1837</td>\n      <td>MCRL</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>6</th>\n      <td>메모리 및 스토리지 구조 연구실</td>\n      <td>민상렬</td>\n      <td>301동 552호 / 517호(02) 880-7296</td>\n      <td>ARCHI</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>9</th>\n      <td>생물정보 및 생명정보 연구실</td>\n      <td>김선</td>\n      <td>301동 516호(02) 880-1784</td>\n      <td>BHI</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>14</th>\n      <td>시스템 소프트웨어 및 구조 연구실</td>\n      <td>김진수</td>\n      <td>301동 517호(02) 880-7296</td>\n      <td>CSL</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>15</th>\n      <td>실시간 유비쿼터스 시스템 연구실</td>\n      <td>이창건</td>\n      <td>301동 415호(02) 880-2562</td>\n      <td>RUBIS</td>\n      <td>301동</td>\n    </tr>\n    <tr>\n      <th>16</th>\n      <td>아키텍처 및 코드 최적화 연구실</td>\n      <td>이재욱</td>\n      <td>301
```