



Overview

Web Handling

최성철 교수
Director of TEAMLAB

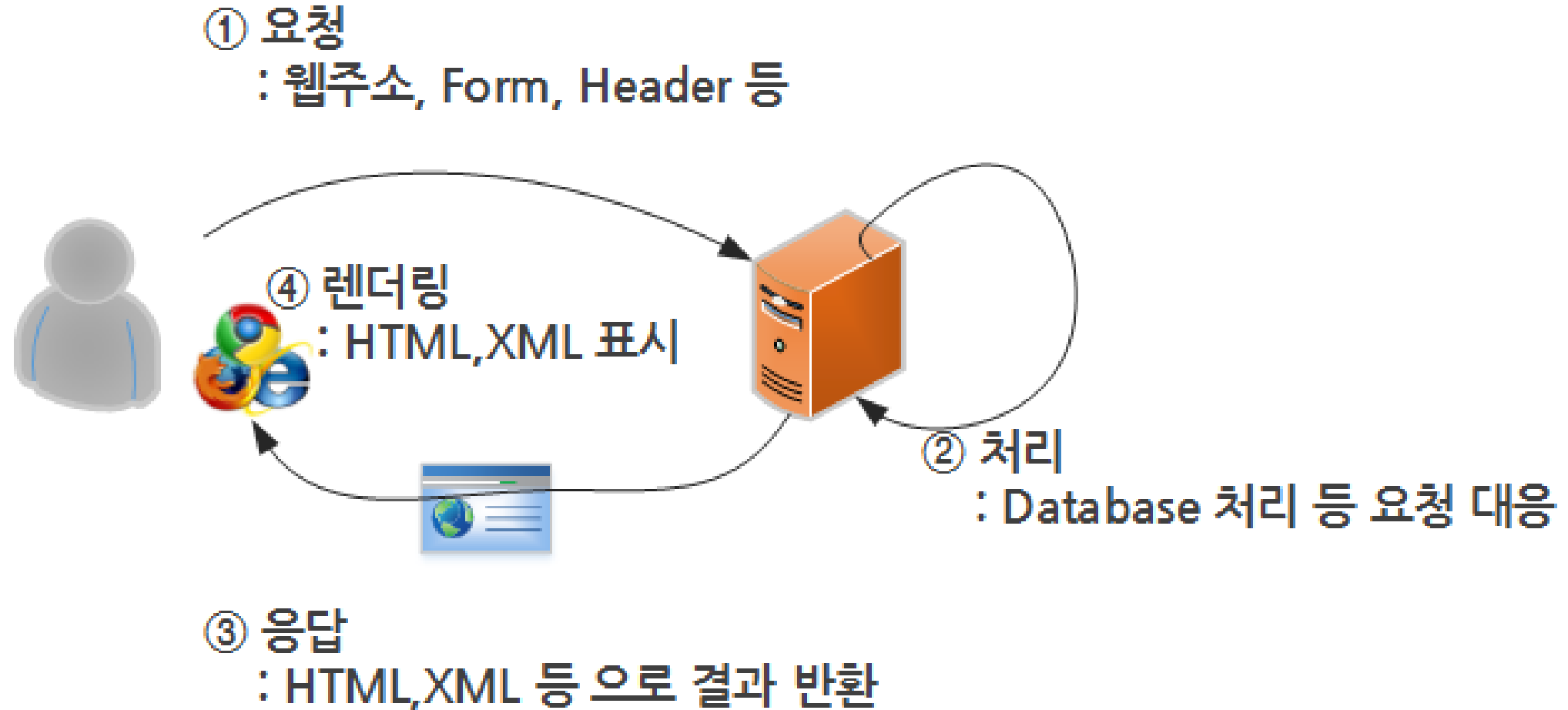
**하루 중 가장 많이
시간을 보내는 곳**

하루 중 가장 많이
시간을 보내는 곳
인터넷 = 웹

Web – 우리가 늘 쓰는 그 것

- **World Wide Web(WWW)**, 줄여서 **웹**이라고 부름
- 우리가 늘 쓰는 **인터넷 공간의 정식 명칭**
- 팀 버너스리에 의해 1989년 처음 제안되었으며,
원래는 **물리학자들간 정보 교환**을 위해 사용됨
- 데이터 송수신을 위한 **HTTP 프로토콜** 사용,
데이터를 표시하기 위해 **HTML 형식**을 사용

Web은 어떻게 동작하는가?



HTML(Hyper Text Markup Language)

- 웹 상의 정보를 구조적으로 표현하기 위한 언어
- 제목, 단락, 링크 등 요소 표시를 위해 Tag를 사용
- 모든 요소들은 꺾쇠 괄호 안에 둘러 쌓여 있음

`<title> Hello, World </title>` # 제목 요소, 값은 Hello, World

- 모든 HTML은 트리 모양의 포함관계를 가짐
- 일반적으로 웹 페이지의 HTML 소스파일은
컴퓨터가 다운로드 받은 후 웹 브라우저가 해석/표시

HTML(Hyper Text Markup Language)

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

HTML 구조

<html> – <head> – <title>
– <body> – <p>

Element, Attribute Value 이루어짐

<tag attribute1= " att_value1" attribute2=" att_value1 ">
보이는 내용(Value)
</tag>

Source: <http://ko.wikipedia.org/wiki/HTML>

왜 웹을 알아야 하는가?

- 정보의 보고, 많은 데이터들이 웹을 통해 공유됨


환율정보: <http://goo.gl/95q3mz>

상장기업 매출정보: <http://goo.gl/nwi8WE>


미국 특허정보: <http://goo.gl/wrmhWK>


- HTML도 일종의 프로그램, 페이지 생성 규칙이 있음
: 규칙을 분석하여 데이터의 추출이 가능
- 추출된 데이터를 바탕으로 하여 다양한 분석이 가능


[참고] Excel로 웹 데이터 추출하기


- 엑셀 실행 후 [데이터] → [웹] 메뉴 클릭
- 아래 Dialog에서 주소 입력 후  추출 대상 데이터 선택 (테이블 중심)

새 웹 쿼리

주소(D): 이동(G) 

가져오려는 테이블 옆의  를 클릭한 다음 [가져오기]를 클릭합니다(C).

데이터 선택 <http://info.finance.naver.com/marketindex/> 

☒ 

통화명	매매기준율	현찰		송금		환가료율	미화환산율
		사실 때	파실 때	보내실 때	받으실 때		
미국 USD	1,018.00	1,035.81	1,000.19	1,027.90	1,008.10	2.005	1.000
유럽연합 EUR	1,352.51	1,379.42	1,325.60	1,366.03	1,338.99	2.046	1.329
일본 JPY (100엔)	981.30	998.47	964.13	990.91	971.69	2.071	0.964
중국 CNY	165.37	176.94	157.11	167.02	163.72	5.067	0.162
홍콩 HKD	131.35	133.96	128.74	132.66	130.04	2.305	0.129
대만 TWD	33.94	36.65	32.25	0.00	0.00	N/A	0.033
영국 GBP	1,688.45	1,722.05	1,654.85	1,705.33	1,671.57	2.476	1.659

데이터 추출 >

완료



Human knowledge belongs to the world.



HTML Parsing

Web Handling

최성철 교수
Director of TEAMLAB

**Web으로
할 수 있는 것들**

Web 데이터를 다운로드

필요한 정보 저장하기

웹에서 데이터 다운로드

- 웹 상에 있는 파일을 로컬폴더에 저장함
- Built-in 모듈인 urllib의 urlretrieve() 함수 사용

```
import urllib.request    #urllib 모듈 호출
```

```
url = "http://storage.googleapis.com/patents/grant_full_text/2014/ipg140107.zip"
```

```
# 다운로드 URL 주소
```

```
print ("Start Download")
```

```
fname, header = urllib.request.urlretrieve(url, 'ipg140107.zip')
```

```
#urlretrieve 함수 호출 (url 주소, 다운로드 될 파일명), 결과값으로 다운로드된 파일명과 Header 정보를 언패킹
```

```
print ("End Download")
```

**어떨 때
쓸 수 있을까?**

강의 자료 자동 다운로드 하기

<http://web.eecs.umich.edu/~radev/coursera-slides/>

Draft slides for the online course

These slides are released on an as-is basis. The official versions will be posted w

- [01.01.pdf](#)
- [01.02.pdf](#)
- [01.03.pdf](#)
- [01.04.pdf](#)
- [01.05.pdf](#)
- [01.06.pdf](#)
- [01.07.pdf](#)
- [02.01.pdf](#)
- [02.02.pdf](#)
- [02.03.pdf](#)
- [02.04.pdf](#)
- [02.05.pdf](#)
- [02.06.pdf](#)
- [02.07.pdf](#)
- [03.01.pdf](#)
- [03.02.pdf](#)
- [03.03.pdf](#)
- [03.04.pdf](#)
- [03.05.pdf](#)
- [03.06.pdf](#)
- [03.07.pdf](#)
- [04.01.pdf](#)
- [04.02.pdf](#)
- [04.03.pdf](#)
- [04.04.pdf](#)
- [04.05.pdf](#)
- [05.01.pdf](#)
- [05.02.pdf](#)
- [05.03.pdf](#)
- [05.04.pdf](#)
- [05.05.pdf](#)
- [05.06.pdf](#)
- [05.07.pdf](#)
- [05.08.pdf](#)
- [06.01.pdf](#)
- [06.02.pdf](#)
- [06.03.pdf](#)
- [06.04.pdf](#)
- [06.05.pdf](#)
- [06.06.pdf](#)
- [06.07.pdf](#)

```
<LL>
<LI><A HREF=lpintro_co3_01_01_DR_Edit.pdf>01_01.pdf</A>
<LI><A HREF=lpintro_co3_01_02_DR_Edit.pdf>01_02.pdf</A>
<LI><A HREF=lpintro_co3_01_03_DR_Edit.pdf>01_03.pdf</A>
<LI><A HREF=lpintro_co3_01_04_DR_Edit.pdf>01_04.pdf</A>
<LI><A HREF=lpintro_co3_01_05_DR_Edit.pdf>01_05.pdf</A>
<LI><A HREF=lpintro_co3_01_06_DR_Edit.pdf>01_06.pdf</A>
<LI><A HREF=lpintro_co3_01_07_DR_Edit.pdf>01_07.pdf</A>
<LI><A HREF=lpintro_co3_02_01_DR_Edit.pdf>02_01.pdf</A>
<LI><A HREF=lpintro_co3_02_02_DR_Edit.pdf>02_02.pdf</A>
<LI><A HREF=lpintro_co2_02_03_DR_Edit.pdf>02_03.pdf</A>
<LI><A HREF=lpintro_co2_02_04_DR_Edit.pdf>02_04.pdf</A>
<LI><A HREF=lpintro_co2_02_05_DR_Edit.pdf>02_05.pdf</A>
<LI><A HREF=lpintro_co3_02_06_DR_Edit.pdf>02_06.pdf</A>
<LI><A HREF=lpintro_co3_02_07_DR_Edit.pdf>02_07.pdf</A>
<LI><A HREF=lpintro_co2_03_01_DR_Edit.pdf>03_01.pdf</A>
<LI><A HREF=lpintro_co2_03_02_DR_Edit.pdf>03_02.pdf</A>
<LI><A HREF=lpintro_co2_03_03_DR_Edit.pdf>03_03.pdf</A>
<LI><A HREF=lpintro_co2_03_04_DR_Edit.pdf>03_04.pdf</A>
<LI><A HREF=lpintro_co3_03_05_DR_Edit.pdf>03_05.pdf</A>
<LI><A HREF=lpintro_co3_03_06_DR_Edit.pdf>03_06.pdf</A>
<LI><A HREF=lpintro_co3_03_07_DR_Edit.pdf>03_07.pdf</A>
<LI><A HREF=lpintro_co4_04_01_DR_Edit.pdf>04_01.pdf</A>
<LI><A HREF=lpintro_co5_04_02_DR_Edit.pdf>04_02.pdf</A>
<LI><A HREF=lpintro_co5_04_03_DR_Edit.pdf>04_03.pdf</A>
<LI><A HREF=lpintro_co5_04_04_DR_Edit.pdf>04_04.pdf</A>
<LI><A HREF=lpintro_co5_04_05_DR_Edit.pdf>04_05.pdf</A>
<LI><A HREF=lpintro_co1_05_01_DR_Edit.pdf>05_01.pdf</A>
<LI><A HREF=lpintro_co1_05_02_DR_Edit.pdf>05_02.pdf</A>
<LI><A HREF=lpintro_co1_05_03_DR_Edit.pdf>05_03.pdf</A>
<LI><A HREF=lpintro_co1_05_04_DR_Edit.pdf>05_04.pdf</A>
<LI><A HREF=lpintro_co5_05_05_DR_Edit.pdf>05_05.pdf</A>
<LI><A HREF=lpintro_co5_05_06_DR_Edit.pdf>05_06.pdf</A>
<LI><A HREF=lpintro_co5_05_07_DR_Edit.pdf>05_07.pdf</A>
<LI><A HREF=lpintro_co5_05_08_DR_Edit.pdf>05_08.pdf</A>
<LI><A HREF=lpintro_co6_06_01_DR_Edit.pdf>06_01.pdf</A>
<LI><A HREF=lpintro_co6_06_02_DR_Edit.pdf>06_02.pdf</A>
```

자동화 하기 위해선

HTML Parsing

HTML Parsing

- 웹으로 부터 데이터를 추출해 내는 행위
- 대부분의 웹은 사용자 요구에 따라 동적으로 생성됨

페이지를 생성하는 파일 (프로그램)

<http://finance.naver.com/item/main.nhn?code=005930>

본 프로그램에서는 상장코드에 따라 다른 정보를 생성함
(GET 방식)

해당 프로그램의 변수
code: 변수명, 005930: 값

- HTML Parsing을 위해서는 **HTML 생성 규칙 파악**
- HTML은 Tree 구조 → **구조 파악 필요**

HTML 규칙 파악하기 (1/2)

<http://finance.naver.com/item/main.nhn?code=005930>

삼성전자 005930 코스피 2016.11.04 기준(장마감) 실시간 기업개요	
1,627,000 전일대비 ▲11,000 +0.68%	전일 1,616,000 고가 1,634,000 (상한가 2,100,000) 거래량 141,995
	시가 1,605,000 저가 1,605,000 (하한가 1,132,000) 거래대금 230,766 백만

- HTML 열어서 분석 하기
(브라우저에서 오른쪽 마우스 클릭 후 소스보기 클릭)
- HTML 파일에서 유일하게 위 데이터를 나타낼 수 있는 패턴을 찾아야 함

HTML 규칙 파악하기 (2/2)

```
<dl class="blind">                                <dl class="blind"> ~ </dl>
  <dt>종목 시세 정보</dt>                          사이 데이터 존재
  <dd>2016년 11월 04일 16시 10분 기준 장마감</dd>
  <dd>종목명 삼성전자</dd>
  <dd>종목코드 005930 코스피</dd>
  <dd>현재가 1,627,000 전일대비 상승 11,000 플러스 0.68 퍼센트</dd>
  <dd>전일가 1,616,000</dd>
  <dd>시가 1,605,000</dd>
  <dd>고가 1,634,000</dd>
  <dd>상한가 2,100,000</dd>
  <dd>저가 1,605,000</dd>
  <dd>하한가 1,132,000</dd>
  <dd>거래량 141,995</dd>
  <dd>거래대금 230,766백만</dd>
</dl>
```

각 데이터는 <dd> ~ </dd>로 나타나며
데이터 생성순서는 일시, 종목 명 ~ 거래대금 순

HTML Parsing을 위한 두가지 방법

1) 정규식 이용하기

2) 모듈 활용하기



Human knowledge belongs to the world.

Regular Expression

Web Handling

최성철 교수
Director of TEAMLAB

01100
00110

정규식 - Regular Expression

- 정규 표현식, regexp 또는 regex 등으로 불림
- 복잡한 문자열 패턴을 정의하는 문자 표현 공식
- 특정한 규칙을 가진 문자열의 집합을 추출

010-0000-0000 `^\d{3}\-\d{4}\-\d{4}$`

203.252.101.40 `^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$`

정규식 for HTML Parsing

- 주민등록 번호, 전화번호, 도서 ISBN 등 형식이 있는 문자열을 원본 문자열로부터 추출함
- HTML역시 tag를 사용한 일정한 형식이 존재하여 정규식으로 추출이 용이함
- 관련자료: <http://www.nextree.co.kr/p4327/>

정규식 for HTML Parsing

- 문법 자체는 매우 방대, 스스로 찾아서 하는 공부 필요
- 필요한 것들은 인터넷 검색을 통해 찾을 수 있음
- 기본적인 것을 공부 한 후 넓게 적용하는 것이 중요

이메일: `^[a-zA-Z0-9]+@[a-zA-Z0-9]+$ or`

<https://goo.gl/FNTwIO>

`^[_0-9a-zA-Z-]+@[0-9a-zA-Z-]+(.[_0-9a-zA-Z-]+)*$`

휴대폰: `^01(?:0|1|[6-9]) - (?:\Wd{3}|\Wd{4}) - \Wd{4}$`

일반전화: `^\Wd{2,3} - \Wd{3,4} - \Wd{4}$`

주민등록번호: `\Wd{6} \W- [1-4]\Wd{6}`

IP 주소: `([0-9]{1,3}) \W. ([0-9]{1,3}) \W. ([0-9]{1,3}) \W. ([0-9]{1,3})`

해시태그: `#([A-Za-z0-9가-힣]+)`

<https://goo.gl/FNTwIO>

정규식 연습장 활용하기

- 1) 정규식 연습장(<http://www.regexr.com/>) 으로 이동
- 2) 테스트하고 싶은 문서를 Text 란에 삽입
- 3) 정규식을 사용해서 찾아보기

정규식 기본 문법 #1

문자 클래스 `[]`: `[와]` 사이의 문자들과 매치라는 의미

예) `[abc]` ← 해당 글자가 a,b,c중 하나가 있다.

`"a", "before", "deep", "dud", "sunset"`

`"-"`를 사용 범위를 지정할 수 있음

예) `[a-zA-z]` – 알파벳 전체, `[0-9]` – 숫자 전체

<https://wikidocs.net/4308>

정규식 기본 문법 - 메타 문자

정규식 표현을 위해 원래 의미 X, 다른 용도로 사용되는 문자

. ^ \$ * + ? { } [] \ | ()

- . - 줄바꿈 문자인 \n를 제외한 모든 문자와 매치 a[.]b
- * - 앞에 있는 글자를 반복해서 나올 수 있음
tomor*ow tomorrow tomoow tomorrrow
- + - 앞에 있는 글자를 최소 1회 이상 반복

정규식 기본 문법 - 메타 문자

정규식 표현을 위해 원래 의미 X, 다른 용도로 사용되는 문자

. ^ \$ * + ? { } [] \ | ()

{m.n} - 반복 횟수를 지정 {1,} , {0,} {1,3}

203.252.101.40 [0-9]{1,3} \d{1,3}

? - 반복 횟수가 1회 01[01]?-[0-9]{4}-[0-9]{4}

| - or (0|1){3} **^** - not

정규식 추출 연습

- ① 정규식 연습장(<http://www.regexr.com/>) 으로 이동
- ② 구글 USPTO Bulk Download 데이터페이지 소스 보기 클릭
- ③ 소스 전체 복사후 정규식 연습장 페이지에 붙여넣기
- ④ 상단 Expression 부분을 수정해가며 “Zip”로 끝나는 파일명만 추출
- ⑤ Expression에 (http)(.)(zip) 를 입력

<http://www.google.com/googlebooks/uspto-patents-grants-text.html>



Human knowledge belongs to the world.

Lab - Regular Expression

Web Handling

최성철 교수
Director of TEAMLAB

01100
00110

정규식 in 파이썬

- re 모듈을 import 하여 사용 : `import re`
- 함수: `search` – 한 개만 찾기, `findall` – 전체 찾기
- 추출된 패턴은 tuple로 반환됨
- 연습 - 특정 페이지에서 ID만 추출하기 <http://goo.gl/U7mSQL>
- ID 패턴: [영문대소문자|숫자] 여러 개, 별표로 끝남
"`([A-Za-z0-9]+W*W*W*)`" 정규식

Code #1

```
import re
import urllib.request

url = "http://goo.gl/U7mSQL"
html = urllib.request.urlopen(url)
html_contents = str(html.read())
id_results = re.findall(r"([A-Za-z0-9]+W*W*W*)", html_contents)
#findall 전체 찾기, 패턴대로 데이터 찾기

for result in id_results:
    print(result)
```

Code #2

```
import urllib.request # urllib 모듈 호출
import re

url = "http://www.google.com/googlebooks/uspto-patents-grants-text.html"
#url 값 입력

html = urllib.request.urlopen(url) # url 열기
html_contents = str(html.read().decode("utf8"))
# html 파일 읽고, 문자열로 변환

url_list = re.findall(r"(http)(.+)(zip)", html_contents)
for url in url_list:
    print("".join(url)) # 출력된 Tuple 형태 데이터 str으로 join
```

Code #3

```
import urllib.request # urllib 모듈 호출
import re
```

```
base_url = "http://web.eecs.umich.edu/~radev/coursera-slides/"
#url 값 입력
html = urllib.request.urlopen(base_url)
html_contents = str(html.read().decode("utf8"))
```

```
url_list = re.findall(r"nlp[0-9a-zA-ZW_.]*W.pdf", html_contents)
for url in url_list:
    file_name = "".join(url)
    full_url = base_url + file_name
    print(full_url)
    fname, header = urllib.request.urlretrieve(full_url, file_name)
    print("End Download")
```

<http://web.eecs.umich.edu/~radev/coursera-slides/>

raft slides for the online course

These slides are released on an as-is basis. The official versions will be posted weekly on the coursera site.

- [01.01.pdf](#)
- [01.02.pdf](#)
- [01.03.pdf](#)
- [01.04.pdf](#)
- [01.05.pdf](#)
- [01.06.pdf](#)
- [01.07.pdf](#)
- [02.01.pdf](#)
- [02.02.pdf](#)
- [02.03.pdf](#)
- [02.04.pdf](#)
- [02.05.pdf](#)
- [02.06.pdf](#)
- [02.07.pdf](#)
- [03.01.pdf](#)
- [03.02.pdf](#)
- [03.03.pdf](#)
- [03.04.pdf](#)
- [03.05.pdf](#)
- [03.06.pdf](#)
- [03.07.pdf](#)
- [04.01.pdf](#)
- [04.02.pdf](#)
- [04.03.pdf](#)
- [04.04.pdf](#)
- [04.05.pdf](#)
- [05.01.pdf](#)
- [05.02.pdf](#)
- [05.03.pdf](#)
- [05.04.pdf](#)
- [05.05.pdf](#)
- [05.06.pdf](#)
- [05.07.pdf](#)
- [05.08.pdf](#)
- [06.01.pdf](#)
- [06.02.pdf](#)
- [06.03.pdf](#)
- [06.04.pdf](#)
- [06.05.pdf](#)
- [06.06.pdf](#)
- [06.07.pdf](#)

정규식 in 파이썬 for html

```
<dl class="blind">
  <dt>종목 시세 정보</dt>
  <dd>2016년 11월 04일 16시 10분 기준 장마감</dd>
  <dd>종목명 삼성전자</dd>
  <dd>종목코드 005930 코스피</dd>
  <dd>현재가 1,627,000 전일대비 상승 11,000 플러스 0.68 퍼센트</dd>
  <dd>전일가 1,616,000</dd>
  <dd>시가 1,605,000</dd>
  <dd>고가 1,634,000</dd>
  <dd>상한가 2,100,000</dd>
  <dd>저가 1,605,000</dd>
  <dd>하한가 1,132,000</dd>
  <dd>거래량 141,995</dd>
  <dd>거래대금 230,766백만</dd>
</dl>
```

이 데이터는 어떻게 뽑을까?

정규식 in 파이썬 for html

① <dl class="blind"> ~~~~ </dl> 에 있는

② <dd> ~~~~ </dd> 정보를 추출하면 됨

```
<dl class="blind">
  <dt>종목 시세 정보</dt>
  <dd>2016년 11월 04일 16시 10분 기준 장마감</dd>
  <dd>종목명 삼성전자</dd>
  <dd>종목코드 005930 코스피</dd>
  <dd>현재가 1,627,000 전일대비 상승 11,000 플러스 0.68 퍼센트</dd>
  <dd>전일가 1,616,000</dd>
  <dd>시가 1,605,000</dd>
  <dd>고가 1,634,000</dd>
  <dd>상한가 2,100,000</dd>
  <dd>저가 1,605,000</dd>
  <dd>하한가 1,132,000</dd>
  <dd>거래량 141,995</dd>
  <dd>거래대금 230,766백만</dd>
</dl>
```


정규식 in 파이썬 for html

① `<dl class="blind"> ~~~~ </dl>`

`(\<dl class=\"blind\"\>)([\>S]+?)(\<\>/dl\>)`

`<dl class`에서 시작해서 / 사이에 아무 글자나 있고 / `</dl>` 로 끝내기

② `<dd> ~~~~ </dd>` 정보를 추출하면 됨

`(\<dd\>)([\>S]+?)(\<\>/dd\>)`

`<dd>` 에서 시작해서 / 사이에 아무 글자나 있고 / `</dl>` 로 끝내기

① 를 먼저 찾고 ① 안에 ②를 차례대로 찾으면 됨

정규식 in 파이썬 for html

```
import urllib.request
import re
```

```
url = "http://finance.naver.com/item/main.nhn?code=005930"
html = urllib.request.urlopen(url)
html_contents = str(html.read().decode("ms949"))
```

```
stock_results = re.findall("(W<dl class=W\"blindW\"W>)([WsWS]+?)(W<W/dlW>)", html_contents)
samsung_stock = stock_results[0] # 두 개 tuple 값중 첫번째 패턴
samsung_index = samsung_stock[1] # 세 개의 tuple 값중 두 번째 값
                                # 하나의 괄호가 tuple index가 됨
index_list= re.findall("(W<ddW>)([WsWS]+?)(W<W/ddW>)", samsung_index)
```

```
for index in index_list:
    print (index[1]) # 세 개의 tuple 값중 두 번째 값
```



Human knowledge belongs to the world.