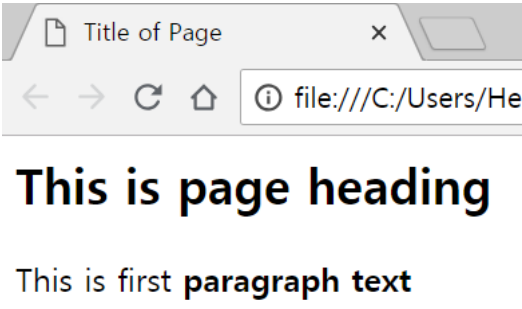


[2] HTML 이해하기

논과 밭에서 감자를 직접 캔 후, 흙을 제거하고 껍질을 벗겨야 비로소 우리가 먹을 수 있는 식재료의 형태가 됩니다. 크롤링을 통해 데이터를 얻을 경우 웹페이지에 존재하는 데이터를 일괄적으로 받으므로 투자에 불필요한 데이터도 같이 받아질 때가 있습니다. 따라서 크롤링 작업 이후에는 필요한 데이터만을 찾아낸 후 분석하기 편한 형태로 처리하는 '데이터 클렌징' 작업이 필수입니다.

우리가 사용하는 웹페이지는 대부분 하이퍼텍스트 마크업언어(Hyper Text Markup Language, 즉 HTML)을 이용하여 만들어집니다. 따라서 해당 언어의 구조를 간략하게 이해한다면 크롤링 이후 원하는 데이터만을 찾는 데 매우 도움이 됩니다. 아래 표는 HTML 표현 및 웹페이지에 표시되는 내용¹의 비교입니다.

[표] HTML과 웹페이지 표시 내용의 비교

HTML	<pre><html> <head> <title>Title of Page</title> </head> <body> <h2>This is page heading</h2> <p>This is first paragraph text</p> </body> </html></pre>
웹페이지 표시 내용	


HTML 언어에서 <>로 표시된 부분은 태그입니다. 먼저 <head>는 제목 부분을, <title>은 제목 내용을 나타내는 태그이며, <title>뒤에 적힌 **Title of Page** 글자가 웹페이지의 상단에 그대로 위치

함이 확인됩니다. 태그를 마칠 때는 태그명 앞에 슬래쉬(</태그명>)를 입력하여 줍니다.

<body>는 웹페이지에 나타나는 부분인 본문을 의미합니다. 먼저 <h2>에서 h는 제목의 역할을 하며, 2는 우선순위 및 크기를 나타냅니다. <p>는 새로운 본문이 시작되는 곳을 의미합니다. 은 굵은 글씨를 의미하며, 해당 태그로 둘러싸인 'paragraph text' 라는 단어가 웹페이지에서는 굵게 표현됨이 확인됩니다.

태그의 특징 중 하나는 속성^{attribute}을 추가할 수 있으며, 이를 통해 해당 내용을 어떻게 보여줄지 설정할 수 있습니다. 아래 표는 <p> 태그에 style 속성을 추가한 표현 및 웹페이지 표시내용의 비교입니다.

[표] style 속성과 웹페이지 표시 내용의 비교

HTML	<pre><p style="color: red; font-size: 11;"> 1. Red 11 </p> <p style="color: blue; font-size: 13;"> 2. Blue 13 </p> <p style="color: yellow; font-size: 15;"> 3. Yellow & Black 15 </p></pre>
웹페이지 표시 내용	

style 속성은 글자의 크기, 모양 등을 표현하는데 사용됩니다. color는 글자의 색을, font-size는 글자의 크기를 나타내며, 각각 설정값에 따라 글자의 색 및 크기가 변경됨이 확인됩니다. 또한, background-color는 글자의 배경색을 나타내며, 태그를 통해 적용되었음이 확인됩니다.

크롤링을 통한 데이터 수집에 있어 실무적으로 가장 알아야 하는 속성은 id와 class 입니다. 두 속성은 일종의 책갈피 역할을 합니다. 아래 표는 div 태그에 id 속성과 class 속성을 추가한 표현 및 웹페이지 표시내용의 비교입니다.

[표] id와 class 속성의 웹페이지 표시 내용의 비교

HTML	<div> <div #="" >id="" css<="" div="" div><="" for="" id="div1" uses=""> <div> <div >class="" .="" class="div2" css<="" div="" div><="" for="" uses=""> </div> </div></div></div>
웹페이지 표시 내용	<div> <div>id.html</div> <div> <div>←</div> <div>→</div> <div>↺</div> <div>🏠</div> <div>🔍 file</div> </div> <div> <div>ID uses # for CSS</div> <div>Class uses . for CSS</div> </div> </div>

첫 번째 줄에는 “div1”의 id 속성을 지정해 주었으며, 두 번째 줄에는 “div2”의 class 속성을 지정해 주었습니다. 웹페이지에 표시되는 내용을 봤을 때는 아무런 역할도 하지 않는 것처럼 보입니다. 둘 간의 차이는 개발자도구 화면²을 통해 확인할 수 있습니다.

[표] id와 class 속성의 개발자도구 화면 비교

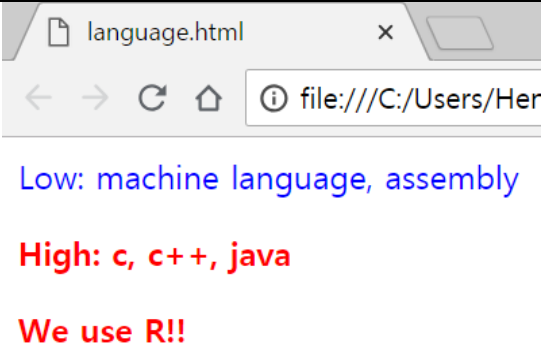
ID	<div> <div>ID uses # for CSS</div> <div>div#div1 602 × 21.33</div> </div> <div> <div>Elements Console Sources Network</div> <div> <div><html></div> <div><head></head></div> <div><body></div> <div> <div>... <div id="div1">ID uses # for CSS</div> == \$0</div> <div><div class="div2">Class uses . for CSS</div></div> </div> <div></body></div> <div></html></div> </div> </div>
Class	<div> <div>ID uses # for CSS</div> <div>Class uses . for CSS</div> <div>div.div2 602 × 21.33</div> </div> <div> <div>Elements Console Sources Network</div> <div> <div><html></div> <div><head></head></div> <div><body></div> <div> <div>... <div id="div1">ID uses # for CSS</div> == \$0</div> <div><div class="div2">Class uses . for CSS</div></div> </div> <div></body></div> <div></html></div> </div> </div>

먼저 우측의 개발자도구 화면에서 Elements 탭을 선택합니다. 첫 번째 줄에 해당하는 id 속성의 “div1” 부분으로 마우스를 움직이면, 좌측의 웹페이지 화면에 해당 태그의 정보인 div#div1이 표시 됩니다. 다음으로 class 속성의 “div2” 부분으로 마우스를 움직이면 이번에는 태그의 정보가 div.div2로 표시됩니다. 즉 id 속성은 ‘태그#속성명’, class 속성은 ‘태그.속성명’으로 인식됩니다.

이 외에도 id와 class 속성의 차이는 다음과 같습니다. id 속성은 문서안에서 한번만 사용 가능하며, 보통 레이아웃에 사용됩니다. 반면에 class 속성은 문서안에서 중복으로 사용 가능 하며 일관성 있는 스타일을 적용시 사용됩니다.

각 클래스 별로 일관된 스타일을 적용하는 방법의 예시입니다.

[표] class 속성별 스타일 적용

HTML	<pre><style type="text/css"> .low-level { color: blue; } .high-level { color: red; font-weight: bold; } </style> <p class="low-level"> Low: machine language, assembly </p> <p class="high-level"> High: c, c++, java </p> <p class="high-level"> We use R!! </p></pre>
웹페이지 표시 내용	 <p>Low: machine language, assembly</p> <p>High: c, c++, java</p> <p>We use R!!</p>

먼저 style 태그를 통해 각 클래스 별 스타일을 정의해줍니다. 이는 프로그래밍에서 함수와 비슷한 역할을 합니다. .low-level, 즉 속성명이 low-level인 클래스는 푸른색을 적용하며, .high-level, 즉 속성명이 high-level인 클래스는 붉은색 및 굵은 표시를 적용합니다.

그 후 첫번째 단락인 **Low: machine language, assembly**은 low-level 클래스가, 두번째와 세번째 단락인 **High: c, c++, java** 및 **We use R!!**은 high-level 클래스로 설정되었습니다. 웹페이지의 표시 내용을 살펴보면, style 태그에서 적용한 것처럼 low-level 클래스에는 푸른 글씨가, high-level 클래스에는 붉은 글씨 및 굵은 표시가 적용되었음이 확인됩니다.

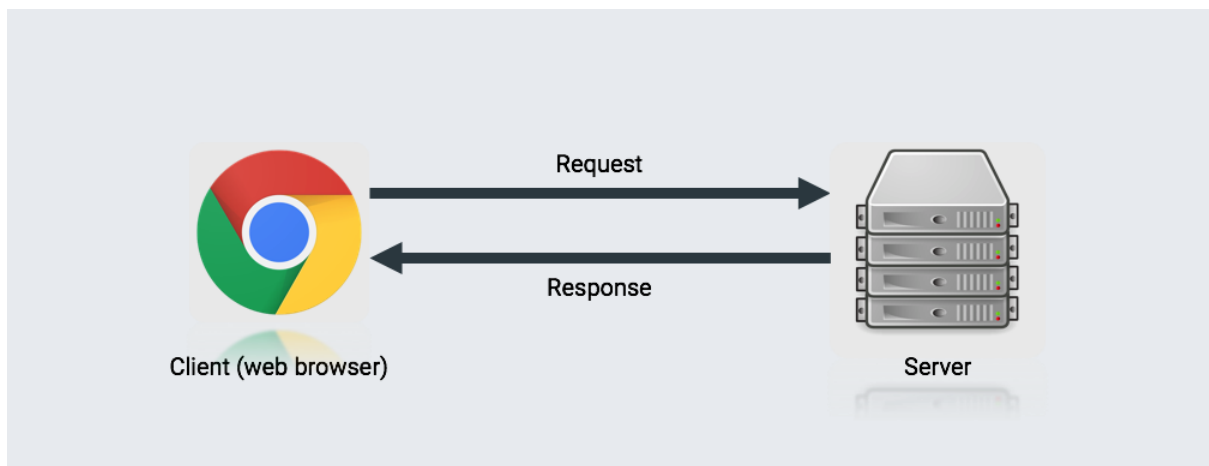
크롤링에 필요한 HTML 관련 지식은 해당 장에서 언급한 정도로도 충분하다고 생각됩니다. 추가적인 정보가 필요하거나 내용이 궁금하신 분들은 아래 사이트를 참고하기 바랍니다.

- 웨버 스터디: <http://webberstudy.com/>
- w3schools: <https://www.w3schools.in/html-tutorial/>

[3] GET과 POST 방식 이해하기

우리가 인터넷에 접속하여 서버에 파일을 요청하면, 서버는 이에 해당하는 파일을 우리에게 보내줍니다. 이러한 과정을 사람이 수행하기 편하고 시각적으로 보기 편하도록 만들어진 것이 크롬과 같은 웹브라우저이며, 서버의 주소를 기억하기 쉽게하기 위해 만든 것이 인터넷 주소입니다. 우리가 서버에 데이터를 요청하는 형태는 다양하지만 크롤링에서는 주로 GET과 POST 방식을 사용합니다.

[그림] 클라이언트와 서버 간의 요청/응답 과정



GET 방식

GET 방식은 인터넷 주소를 기준으로, 이에 해당하는 데이터나 파일을 요청하는 것입니다. 주로 클라이언트가 요청하는 쿼리를 앰퍼샌드(&) 혹은 물음표(?) 형식으로 결합하여 서버에 전달됩니다.

한경컨센서스(<http://hkconsensus.hankyung.com/>)에 접속한 후 전체 REPORT를 선택하면, 홈페이지의 주소 뒤에 `/apps.analysis/analysis.list`가 붙으며 이에 해당하는 페이지의 내용을 보여줍니다. 상단의 탭에서 기업을 선택하면, 주소의 끝부분에 `?skinType=business`가 추가되며 이에 해당하는 페이지의 내용을 보여줍니다. 즉, 해당 페이지는 GET 방식을 사용하고 있으며 입력종류는 skinType, 이에 해당하는 기업 탭의 입력값은 business 임을 알 수 있습니다.

① hkconsensus.hankyung.com/apps.analysis/analysis.list?skinType=business

이번에는 파생 탭을 선택하여 봅니다. 역시나 홈페이지 주소가 변경되며 해당 주소에 맞는 내용이 나타납니다. 주소의 끝부분이 **?skinType=derivative** 로 변경되며, 입력 값이 변경됨에 따라 페이지의 내용이 이에 맞게 변하는 모습이 확인됩니다. 여러 다른 탭들을 눌러보면 **?skinType=** 뒷부분의 입력값이 변함에 따라 이에 해당하는 페이지로 내용이 변경됨이 확인됩니다.

다시 기업 탭을 선택한 후, 다음 페이지를 확인하기 위해 하단의 2를 클릭합니다. 기존 주소인 **?skinType=business** 뒤에 추가로 sdate와 edate, 그리고 now_page 쿼리가 추가됨이 확인됩니다. sdate에 검색 기간의 시작시점, edate에 검색 기간의 종료시점, now_page에 원하는 페이지를 수기로 입력해도 이에 해당하는 페이지의 데이터를 보여줍니다. 이처럼 GET 방식으로 데이터를 요청할 경우, 웹 페이지 주소를 수정하여 원하는 종류의 데이터를 받아올 수 있습니다.

① hkconsensus.hankyung.com/apps.analysis/analysis.list?skinType=business&sdate=2018-06-07&edate=2018-07-07&order_type=&now_page=2

한경컨센서스

2018-06-07

~

2018-07-07

전체

기업

기업

검색

종목

전체

기업

산업

시장

파생

경제

상한

하한

기업정보

LIST

20

50

80

작성일	제목	적정가격	투자의견	작성자	제공출처	기업정보	차트	첨부파일
2018-07-06	BGF리테일(282330)점포당 매출 회복에...	240,000	Buy	하나래	한국투자증권			
2018-07-06	휴비츠(065510)안광학 의료가기 전문...	0	nr	김한경	이베스트증권			
2018-07-06	신세계(004170)면세점으로 한 단계 도...	460,000	Buy	하나래	한국투자증권			
2018-07-06	삼성중공업(010140)허반기 개선점 찾...	8,000	Hold	이상우	유진투자증권			
2018-07-06	GS리테일(007070)오피스와 편의점의 ...	57,000	Buy	하나래	한국투자증권			

«

1

2

3

4

5

6

7

8

9

10

»

POST 방식

POST 방식은 사용자가 필요한 값을 추가해서 요청하는 방법입니다. GET 방식과의 차이는 클라이언트가 요청하는 쿼리를 body에 넣어서 전송하므로, 요청 내역을 직접적으로 볼 수 없습니다.

한국거래소 상장공시시스템(<http://kind.krx.co.kr/>)에 접속하여 전체메뉴보기를 누른 후, 상장법인상세정보 중 상장종목현황을 선택합니다. 웹 페이지 주소가 바뀌며, 상장종목현황이 보여집니다.

[그림] 상장공시시스템의 상장종목현황 메뉴

① kind.krx.co.kr/corpgeneral/listedIssueStatus.do?method=loadInitPage

KIND소개 | ENGL

회사명 또는 종목코드를 입력하세요.

[홈](#) [상장법인상세정보 +](#) [상장종목현황 +](#)

○ 상장종목현황

조회일자

※ 조회 목록의 특정 행을 클릭하면 상세 화면이 출력됩니다.

○ 유가증권시장

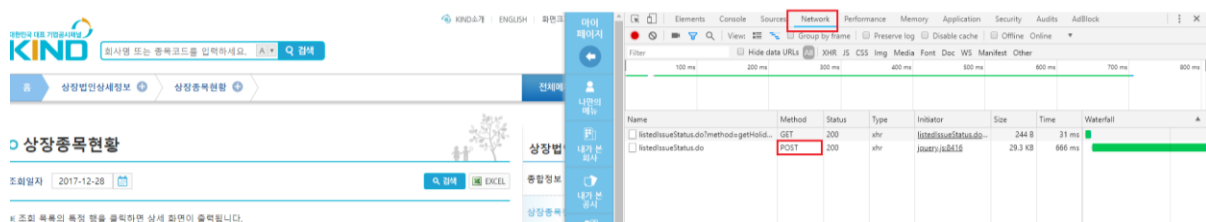
구분	회사수	종목수	상장주식수(천주)	자본금(백만원)	시가총액(백만원)
주권	760	874	50,218,456	106,767,816	1,514,487,938
외국주권	1	1	47,870	-	179,271
투자회사	7	7	742,204	3,250,887	3,876,410
부동산투자회사	5	5	128,743	181,827	386,724
선박투자회사	6	6	62,472	312,358	136,357
소계	779	893	51,199,744	110,512,900	1,519,066,700

이번엔 조회일자를 2017-12-28로 선택한 후, 검색을 눌러보도록 합니다. 페이지의 내용은 선택일 기준으로 변경되었지만, 주소는 변경되지 않고 그대로 남아있습니다. GET 방식에서는 선택항목

에 따라 웹 페이지 주소가 변경되었지만, POST 방식을 사용하여 서버에 데이터를 요청하는 해당 사이트는 그렇지 않음이 확인됩니다.

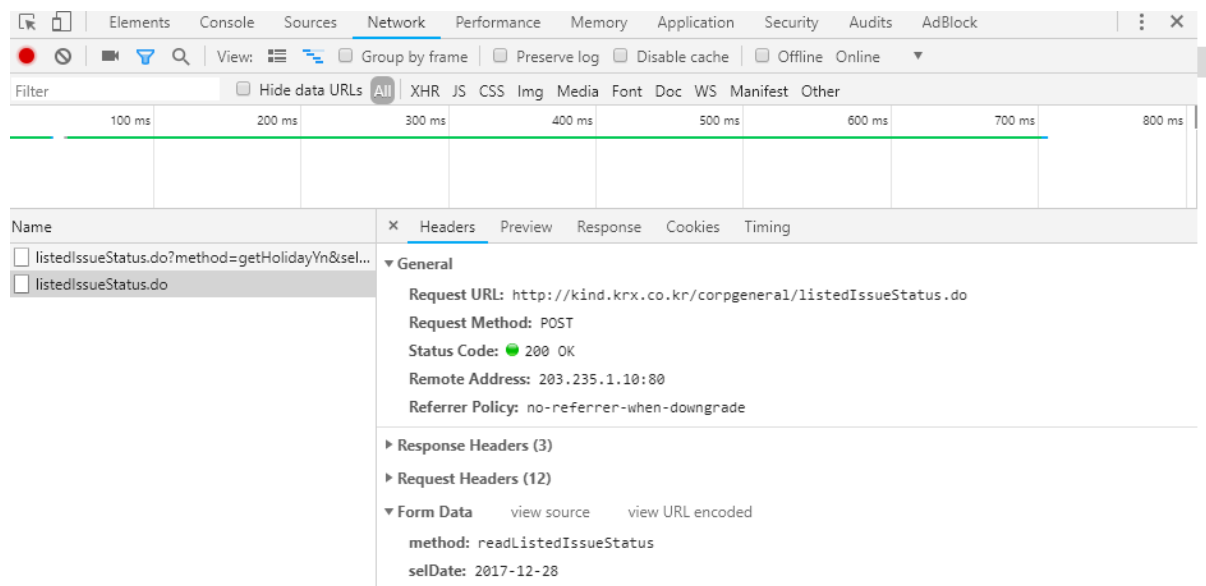
POST 방식의 데이터 요청과정을 살펴보기 위해서는 개발자도구를 이용하면 되며, 개발자도구를 연 상태에서 다시 한번 '검색'을 클릭해 봅니다. 개발자도구의 Network 탭을 클릭하면, '검색'을 클릭함과 함께 브라우저와 서버간의 통신 과정을 살펴볼 수 있습니다. 이 중 listedIssueStatus.do 라는 항목이 POST 형태임을 알 수 있습니다.

[그림] 크롬 개발자도구의 Network 화면



해당 메뉴를 클릭하면 통신 과정을 좀 더 자세히 알 수 있습니다. 가장 하단의 Form Data에 서버에 데이터를 요청하는 내역이 있습니다. method에는 readListIssueStatus, selDate에는 2017-12-28라는 값이 있습니다.

[그림] POST 방식의 서버 요청 내역



이번에는 조회일자를 하루 전인 2017-12-27로 선택하여 검색을 누릅니다. 새로운 데이터 요청이 있음에 따라 개발자도구 화면에 이에 해당하는 내용들이 추가되며, 가장 하단에 이번에도 listedIssueStatus.do 메뉴가 생성됩니다. 해당 메뉴를 선택해보면, method는 기존과 동일하지만 selDate가 2017-12-27로 변경되었습니다. 즉 POST 방식은 요청하는 데이터에 대한 쿼리가 body를 통해 전송되며, 이를 웹 브라우저를 통해 확인할 수는 없습니다.

[그림] 요청내역 변경에 따른 Form Data의 변경

The screenshot displays a web application interface on the left and a Chrome DevTools network panel on the right. The web application shows a table titled '유가증권시장' (KOSPI) with columns for '구분' (Category), '회사수' (Number of Companies), '종목수' (Number of Stocks), '상장주식수(전주)' (Listed Stocks (Previous Week)), '자본금(백만원)' (Capital (100 million KRW)), and '시가총액(백만원)' (Market Capitalization (100 million KRW)). The table contains data for '주권' (Equity), '외국주권' (Foreign Equity), and '투자회사' (Investment Company). The developer tool shows a POST request to 'http://kind.krx.co.kr/corpgeneral/listedIssueStatus.do' with a status code of 200 OK. The 'Form Data' tab is selected, showing a single parameter: 'selDate: 2017-12-27'.

구분	회사수	종목수	상장주식수(전주)	자본금(백만원)	시가총액(백만원)
주권	756	869	41,558,748	104,157,806	1,580,688,466
외국주권	1	1	47,870	-	225,944
투자회사	7	7	742,204	3,250,887	3,647,544

※ 조회 목록의 특정 항목을 클릭하면 상세 화면이 출력됩니다.

유가증권시장

개발자도구: 요청내역

Request URL: http://kind.krx.co.kr/corpgeneral/listedIssueStatus.do
 Request Method: POST
 Status Code: 200 OK
 Remote Address: 203.235.1.10:80
 Referrer Policy: no-referrer-when-downgrade

Form Data

method: read:listedIssueStatus
 selDate: 2017-12-27

[4] 크롤링 예제

크롤링의 일반적인 과정은 http 패키지의 GET() 혹은 POST() 함수를 이용하여 데이터를 다운로드 받은 후, rvest 패키지의 함수들을 이용하여 원하는 데이터를 찾아내는 과정으로 이루어집니다. 해당 장에서는 GET 방식의 예제로 인기도서 순위 중 책 제목만을 추출하는 과정을, POST 방식의 예제로 상장공시시스템의 상장종목현황 테이블을 가져오는 과정을 살펴보도록 하겠습니다.

GET 방식을 이용하여 인기도서 제목 추출하기

먼저 온라인서점인 예스24(<http://www.yes24.com/>)에 접속하여 'R 프로그래밍'을 검색합니다. 검색 후 웹주소가 아래와 같이 변경되며, 페이지가 변경됩니다.

```
http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&Wcode=001_005&query=R+%C7%C1%B7%CE%B1%D7%B7%A1%B9%D6
```

주소에 ?와 &가 결합되어 있는 점으로 보아 GET 방식으로 데이터를 요청하는 것으로 짐작되며, 마지막 query= 항목 뒷부분이 우리가 검색한 검색어임을 유추할 수 있습니다.

GET 방식의 페이지 정보를 크롤링하는 방법은 다음과 같습니다.

```
library(rvest)
library(httr)

url =
"http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%EC%B2%3F&query=R+%C7%C1%B7%CE%B1%D7%B7%A1%B9%D6"
data = GET(url)
```

먼저 크롤링의 필수 패키지인 rvest와 httr 패키지를 열어줍니다. 웹페이지 주소를 url 변수에

지정해준 후, http의 GET() 함수를 이용하면 GET 방식으로 해당 페이지의 데이터를 다운로드 받게 됩니다. 데이터가 저장된 data 변수를 확인해보면 다음과 같습니다.

```
> data
Response
[http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=
%C0%FC%EC%B2%3F&query=R+%C7%C1%B7%CE%B1%D7%B7%A1%B9%D6]
  Date: 2018-07-20 08:04
  Status: 200
  Content-Type: text/html; charset=ks_c_5601-1987
  Size: 343 kB

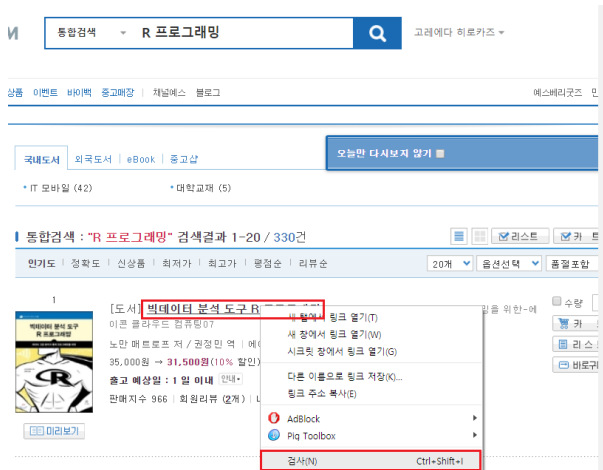
<!DOCTYPE html>
<html lang="ko">

<head>
<meta http-equiv="X-UA-Compatible" content="IE=Edge" />
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr" />
<meta name="viewport" content="width=1170" />
...
```

Response는 데이터 요청에 내역이며, <!DOCTYPE html> 하단의 내용은 해당 페이지의 HTML 정보입니다. Status가 200 값을 나타낼 경우 데이터 요청에 대한 응답이 성공적으로 진행되었음을 의미합니다.

받아진 데이터 중 책 제목만을 추출하기 위해서는 해당 항목의 node가 무엇인지 알 필요가 있으며, 개발자도구를 이용하면 쉽게 찾을 수 있습니다. 먼저 웹페이지에서 최종 목적 데이터인 책 제목에 마우스를 올린 후 우클릭을 하여 검사 항목을 누르면, 개발자도구 화면에서 해당 부분에 대한 HTML 부분이 보여집니다.

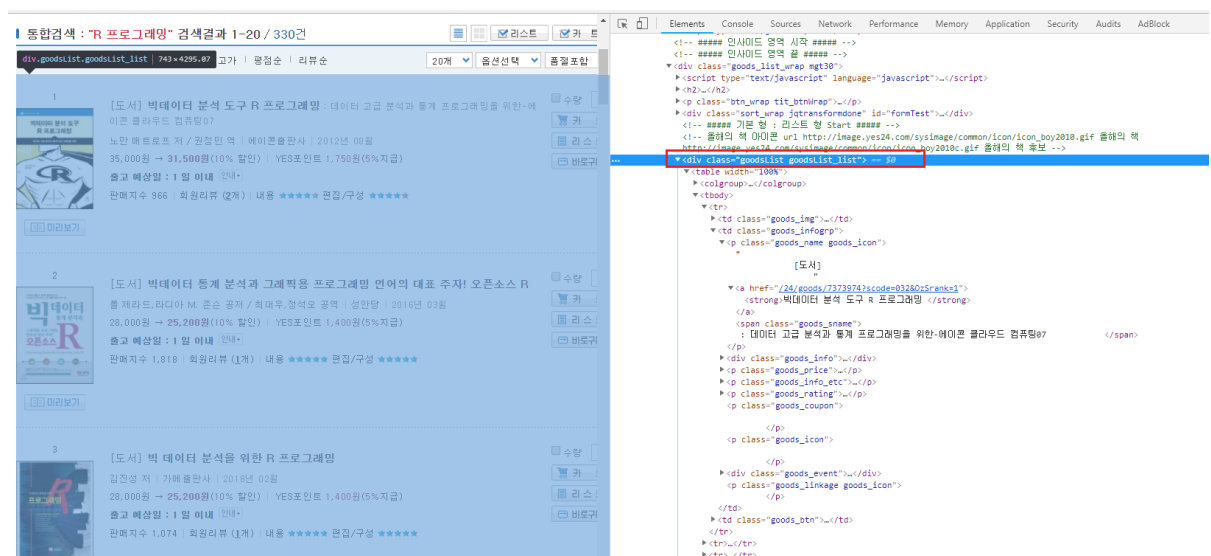
[그림] 인기도서 순위 중 책제목의 HTML 항목 찾기



검사 항목을 누를 경우 개발자도구 화면이 열리며, 책제목이 있는 부분에 회색 음영처리가 되며, 책제목은 태그 가운데 있음이 확인됩니다. HTML 부분을 마우스로 움직여보면, 해당 부분이 해당하는 내용이 웹페이지에 음영처리가 됩니다.

HTML 부분에서 마우스를 위로 올려보면, 우리가 찾는 인기순위 부분은 <div class="goodsList goodsList_list"> 태그에 위치해 있음이 확인됩니다. 개발자도구 화면에서 'goodsList goodsList_list'를 검색하면 통합검색, 중고샵, 리뷰, 기사 및 인터뷰 총 4개 테이블의 클래스가 해당 이름으로 설정되어 있음을 확인할 수 있으며, 그 중 우리가 원하는 인기도서는 첫번째에 위치하고 있습니다.

[그림] 원하는 데이터가 포함된 class 찾기



원하는 데이터가 포함된 class를 찾았다면, 마우스를 아래로 내려가며 데이터의 구조를 파악해 나가면 됩니다. 각 태그들이 웹페이지에서 해당하는 내용은 아래 표와 같습니다.

[표] HTML 태그와 웹페이지 내용의 비교

HTML 태그	웹페이지 내용
<div class="goodsList goodsList_list">	전체 표
<td class="goods_infogrp">	책 관련 정보
<p class="goods_name goods_icon">	책 제목 및 부제목
	책 제목 및 링크
	책 제목

위에서 찾아낸 HTML 구조를 이용하여, 우리가 원하는 책제목을 찾아나가는 코드를 아래와 같습니다.

```
x = read_html(data)
html_node(x, ".goodsList.goodsList_list") %>%
  html_nodes(".goods_infogrp") %>%
  html_nodes(".goods_name.goods_icon") %>%
  html_nodes("a") %>%
  html_text()
```

다운로드 받아진 페이지 정보를 바탕으로, 인기도서 순위 중 책 제목만을 추출하는 과정을 진행하도록 합니다. 먼저 rvest 패키지의 read_html() 함수를 이용하여 HTML 정보만을 추출합니다. 그 후 html_node() 혹은 html_nodes() 함수를 이용하여 원하는 정보를 찾아 내려갑니다.

html_nodes() 함수는 **html_nodes(x, css, xpath)** 형태로 이루어지며, x에 HTML 값을 입력한 후 찾고자 하는 node인 css 혹은 xpath를 입력하면 이에 해당하는 값들을 찾아줍니다. html_nodes() 함수의 경우 node에 해당하는 모든 데이터를 보여주며, html_node() 함수의 경우 이 중 첫번째 데이터만을 보여줍니다.

원하는 css가 태그인 경우 "태그명"을, id를 찾는 경우 "#id"를, class를 찾는 경우 ".class"를 입력하면 됩니다. 또한 class 이름에 공백이 있을 시, 공백 부분을 콤마(,)로 대체하여 주면 됩니다.

개발자도구 화면에서 파악한 HTML 구조와 `html_nodes()` 함수를 이용하여 원하는 데이터를 찾아나면 됩니다. 먼저 해당 페이지에는 총 4개의 `goodsList` `goodsList_list` 클래스가 존재하므로, `html_node()` 함수를 이용하여 첫번째 클래스만을 찾아냅니다. 클래스를 찾는 경우이므로 이름 앞에 콤마(,)를 붙여주며, 공백 부분도 콤마로 대체하여 줍니다. 그 후, 테이블 내에 존재하는 모든 클래스를 찾기 위해 `html_nodes()` 함수를 이용하며, 이들의 연결은 파이프 연산자(`%>%`)를 이용합니다. `html_text()` 함수는 HTML 언어에서 태그 및 속성을 제외하고 텍스트 부분만을 표시해주며, 우리가 원하는 책제목만을 추출하여 줍니다.

위의 코드를 실행하면 다음과 같이 'R 프로그래밍'으로 검색한 페이지의 인기순위 상위 20개 책제목이 추출됩니다.

- [1] "빅데이터 분석 도구 R 프로그래밍 "
- [2] "빅데이터 통계 분석과 그래픽용 프로그래밍 언어의 대표 주자! 오픈소스 R"
- [3] "빅 데이터 분석을 위한 R 프로그래밍"
- [4] "R 병렬 프로그래밍"
- [5] "손에 잡히는 R 프로그래밍"
- [6] "효율적인 R 프로그래밍"
- [7] "손에 잡히는 R 프로그래밍"
- [8] "R 프로그래밍 기초 & 활용"
- [9] "R에서 객체지향 프로그래밍 사용하기"
- [10] "R 프로그래밍"
- [11] "R 프로그래밍 레퍼런스 북"
- [12] "R Shiny 프로그래밍 가이드 "
- [13] "컴퓨터 비전공자를 위한 R언어를 활용한 기초컴퓨터 프로그래밍"
- [14] "R입문 및 기초 프로그래밍"
- [15] "R 프로그래밍"
- [16] "R 통계 프로그래밍 입문 "
- [17] "R을 활용한 통계 프로그래밍 입문"
- [18] "빅데이터 분석을 위한 R 프로그래밍"
- [19] "R 병렬 프로그래밍"
- [20] "알찬 R프로그래밍"

POST 방식을 이용하여 상장종목현황 추출하기

이전의 예제였던 상장공시시스템의 상장종목현황 페이지를 접속합니다. POST 방식의 경우 요청

하는 데이터에 대한 쿼리가 body의 형태를 통해 전송되며, 개발자도구를 통해 해당 쿼리에 대한 내용을 직접 확인할 수 있습니다.

개발자도구를 열고 조회일자를 2017-12-28로 선택하여 검색을 누른 후, listedIssueStatus.do 항목을 살펴보면 Form Data를 통해 서버에 데이터를 요청하는 내역을 확인할 수 있음을 이미 배웠습니다. method에는 readListIssueStatus, selDate에는 2017-12-28라는 값이 있으며, 이를 이용하여 해당 데이터를 다운로드 받는 방법은 다음과 같습니다.

```
url = "http://kind.krx.co.kr/corpgeneral/listedIssueStatus.do?method=loadInitPage"
data = POST(url,
    query=list(
        method = 'readListedIssueStatus',
        selDate = '2017-12-28')
)
```

웹페이지 주소를 url 변수에 지정해준 후, httr의 POST() 함수를 이용하면 POST 방식으로 해당 내역의 데이터를 요청하게 됩니다. 이 중 query 부분에는 Form Data에 있는 쿼리 내용들을 list의 형태로 입력해주면 됩니다. 데이터가 저장된 data 변수를 확인해보면 다음과 같습니다.

```
> data
Response [http://kind.krx.co.kr/corpgeneral/listedIssueStatus.do?method=loadInitPage]
  Date: 2018-07-20 16:16
  Status: 200
  Content-Type: text/html; charset=UTF-8
  Size: 63.7 kB

<!DOCTYPE html>
<html lang="ko">
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="content-language" content="kr" />
...
```

받아진 데이터 중 유가증권시장, 코스닥시장, 코넥스시장에 대한 표를 추출하고자 합니다. 앞서 살펴본 html_nodes 함수를 이용할 수도 있지만, 표 형식의 데이터를 추출할 때는 훨씬 효율적인

방법이 있습니다.

```
data_table = read_html(data) %>%  
  html_table()
```

먼저 read_html() 함수를 이용하여 HTML 정보를 추출한 후, rvest 패키지의 html_table() 함수를 이용합니다. 해당 함수는 HTML에 존재하는 표 형태의 데이터를 찾아내어 데이터프레임 형식으로 읽어옵니다. 위 코드를 실행하면 data_table에 3개의 데이터프레임이 리스트 형태로 저장됩니다. 해당 변수를 확인하면 다음과 같습니다.

```
> data_table
```

```
[[1]]
```

	구분	회사수	종목수	상장주식수(천주)	자본금(백만원)	시가총액(백만원)
1	주권	756	869	41,580,793	104,168,829	1,601,670,789
2	외국주권	1	1	47,870	-	226,423
3	투자회사	7	7	742,204	3,250,887	3,669,787
4	부동산투자회사	4	4	64,487	149,054	120,874
5	선박투자회사	6	6	62,472	312,358	133,040
6	소계	774	887	42,497,825	107,881,138	1,605,820,912
7	신주인수권증권	12	13	279,208	-	259,253
8	신주인수권증서	1	1	4,557	-	962
9	ELW	5	1,930	26,762,414	-	6,211,279
10	ETN	-	184	449,100	-	5,199,361
11	ETF	325	325	1,949,942	-	35,574,913
12	수익증권	12	27	1,440,409	1,551,952	1,463,850

```
[[2]]
```

	구분	회사수	종목수	상장주식수(천주)	자본금(백만원)	시가총액(백만원)
1	주권	1,198	1,201	32,129,584	14,766,746	276,741,261
2	기업인수목적회사	50	50	274,845	27,484	548,523
3	외국주권	15	15	1,221,589	4,914	1,898,334
4	소계	1,263	1,266	33,626,018	14,799,145	279,188,118
5	주식예탁증권(DR)	4	4	132,038	-	3,551,933
6	신주인수권증권	12	12	89,063	-	38,494

[[3]]

구분	회사수	종목수	상장주식수(천주)	자본금(백만원)	시가총액(백만원)
1 주권	154	154	670,598	314,976	4,908,055
2 소계	154	154	670,598	314,976	4,908,055

리스트의 첫번째 항목은 유가증권시장 테이블이, 두번째 항목은 코스닥시장 테이블이, 세번째 항목은 코넥스시장 항목이 저장되어 있음이 확인됩니다. 이처럼 표 형태의 데이터를 추출할 때는 `html_nodes()` 함수를 이용한 후 클렌징 과정을 거치는 것 보다, `html_table()` 함수를 이용하는 것이 훨씬 효율적입니다.

¹ 해당 HTML 코드를 메모장에 입력한 후, `example.html` 파일로 저장하면 웹페이지 형식으로 저장됩니다.

² 크롬 및 인터넷 익스플로러에서 F12키를 누르면 개발자도구 화면이 열리며, 효율성을 위해 크롬을 사용할 것을 권장합니다.