

R for Web Crawling

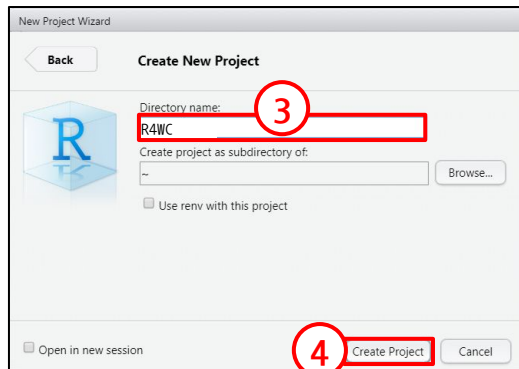
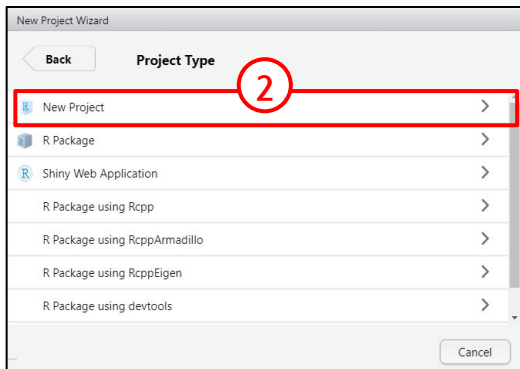
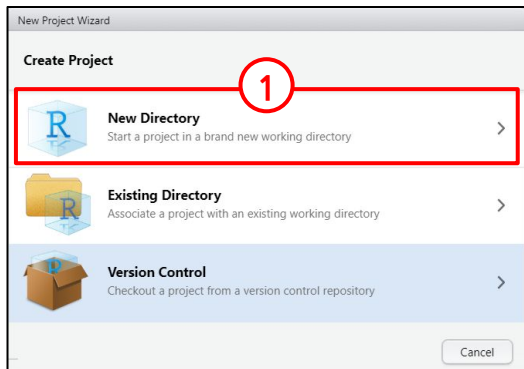
YouTube Version

이번 시간 강의 내용

- 웹 크롤링의 이해
- HTTP 통신 기초
- HTML 요소 기초
- [실습] 네이버 쇼핑 상품 리뷰 수집
- JavaScript 기초
- [실습] 네이버 블로그 본문 수집
- [실습] 네이버 카페 본문 수집

프로젝트 설정

- RStudio의 오른쪽 상단에서 '**Project: (None)**' 메뉴를 클릭하고 '**New Project**'를 선택하면 아래 왼쪽 이미지와 같은 팝업이 뜹니다.
 - New Directory → New Project → 프로젝트명(R4WC) 입력 → Create Project 클릭



실습 데이터셋 내려받기

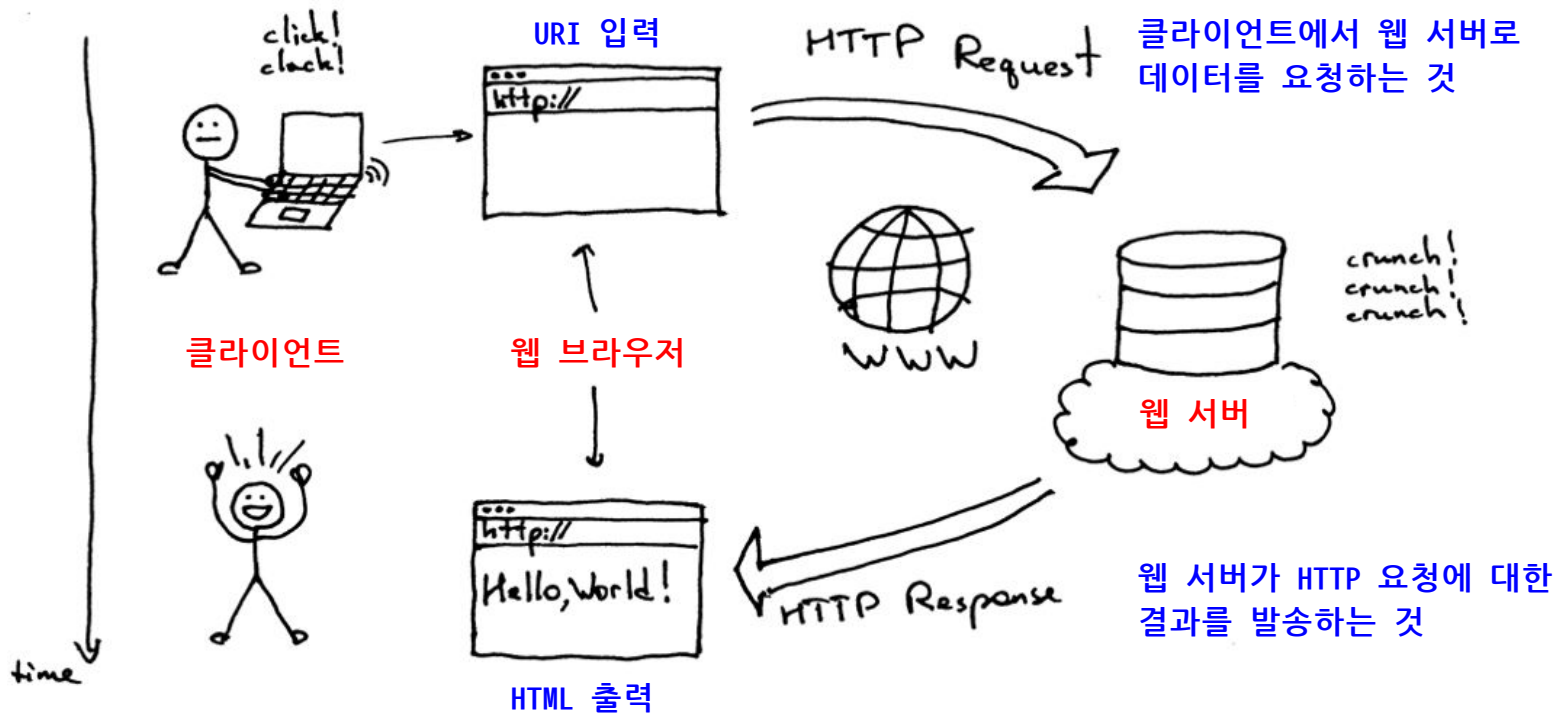
- Windows의 탐색기 또는 MacOS의 파인더를 열면 문서^{Documents} 폴더 안에 새로 생성된 R4TM 폴더를 확인할 수 있습니다.
- 크롬 브라우저를 열고, 아래 URL로 접속하여 과정에 필요한 파일을 내려받습니다.
 - 깃허브 저장소: <https://github.com/HelloDataScience/R4WC>
 - 초록색 'Code' 버튼을 클릭하면 zip 파일을 다운로드^{Downloads} 폴더에 내려받습니다.
- zip 파일을 풀고, 아래 두 가지 폴더를 SKHynix 폴더로 이동시킵니다.
 - code: R 코드가 저장되어 있는 폴더입니다.
 - data: 데이터 파일이 저장되어 있는 폴더입니다.

웹 크롤링의 이해

웹 크롤링과 웹 스크래핑

- 웹 크롤링^{Web Crawling}은 여러 웹 페이지를 탐색하는 작업을 의미하며, 이를 수행하는 프로그램을 웹 크롤러^{Web Crawler}라고 합니다.
 - 웹 페이지에서 일부 내용을 선택하여 수집하는 것은 웹 스크래핑^{Web Scraping}입니다.
- 인터넷 익스플로어 및 크롬 같은 웹 브라우저에서 출력되는 텍스트는 스크래핑할 수 있습니다. 따라서 수집하려는 텍스트를 포함하고 있는 웹 사이트를 발견하는 것이 웹 크롤링의 시작이 됩니다.
- 웹 페이지마다 적용된 방법이 다르므로, 웹 크롤러를 개발할 때에도 웹 페이지에 따라 다른 방법을 적용해야 합니다. 본 강의를 통해 웹 크롤링에 필요한 다양한 방법을 익힐 수 있습니다.

우리가 인터넷에서 정보를 검색하는 방법



웹 크롤링은 인터넷 검색과 유사

HTTP Request

- GET과 POST 방식의 HTTP 통신
- JavaScript 및 Selenium 이용

HTTP Response

- 응답 결과에서 상태코드 및 인코딩 방식 등 확인
- 응답 받은 객체를 텍스트로 출력하여 수집하려는 데이터 포함 여부 확인

수집할 데이터 추출

- 응답 받은 객체를 HTML로 변환
- CSS Selector, XPath를 이용하여 HTML 요소 찾기
- HTML 요소로부터 수집하려는 데이터 추출

데이터 전처리 및 저장

- 텍스트 전처리(결합, 분리, 추출, 대체 등)
- 다양한 형태로 저장(xlsx, csv 등)

http
RSelenium

rvest
jsonlite

dplyr

추가로 알아야 할 사항

크롬 개발자도구

인코딩 및 로케일

다양한 에러 해결법

정규표현식

웹 크롤링 관련 주의사항

- 웹 크롤링 자체는 불법이 아니지만 다른 회사의 영업권과 지식재산권을 침해하는 행위로 판단되면 민사소송에 휘말릴 수 있습니다.(잡코리아, 사람인에 승소)
 - 관련 뉴스: http://it.chosun.com/site/data/html_dir/2017/09/27/2017092785016.html
- 웹 사이트의 메인 페이지에서 robots.txt를 확인하여 웹 크롤링이 허용된 범위를 먼저 파악합니다.
 - 참고: <https://www.naver.com/robots.txt>
- 만약 웹 크롤러로 수집한 데이터로 영리를 추구할 계획이라면 반드시 법률 검토를 진행하시기 바랍니다.

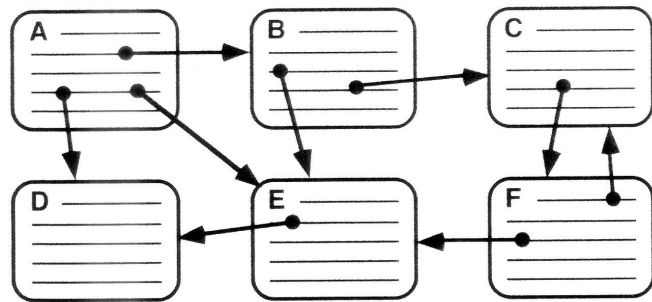
HTTP 통신 기초

HTTPHyperText Transfer Protocol

- HTTP는 초본문 전송 규약으로 번역되는데, 초본문(주로 HTML)은 인터넷에서 주고 받는 문서의 형태를 의미합니다.
- 인터넷에서 데이터를 주고 받는 주체는 클라이언트와 웹 서버입니다.
- 클라이언트가 웹 서버에 데이터를 요청하고, 웹 서버는 요청에 응답합니다.
- HTTP 요청 방식^{Method}은 8가지가 있지만, 그 중 웹 크롤링에서 가장 많이 사용되는 방식은 GET 방식과 POST 방식입니다.
 - 웹 크롤러를 개발할 때, 웹 페이지의 요청 방식을 반드시 확인해야 합니다.
 - 웹 페이지의 요청 방식은 크롬 개발자도구에서 확인할 수 있습니다.

[참고] 초본문^{Hypertext}이란?

- 초본문은 참조^{hyperlink}를 통해 독자가 어떤 지점에서 다른 지점으로 즉시 접근할 수 있는 텍스트를 의미합니다.[위키백과 참조]
 - 일반적으로 문서의 내용을 열람할 때 작성된 순서를 따라가야 합니다.
 - 하지만 초본문은 순서에 상관없이 링크를 통해 내용을 열람할 수 있습니다.
 - 예를 들어, 웹 페이지의 내용을 확인하다가 마우스로 링크를 클릭하면 해당 웹 페이지로 바로 이동할 수 있습니다.



출처: <http://www.isko.org/cyclo/hypertext>

HTTP 요청

- 클라이언트는 HTTP 요청을 실행할 때 웹 서버에 요청 메시지를 제공해야 합니다.
 - 요청 방식에 따라 요청 내용에 차이가 있습니다.
- GET 방식은 요청 라인과 요청 헤더를 포함해야 합니다.
- POST 방식은 메시지 바디를 추가해야 합니다.

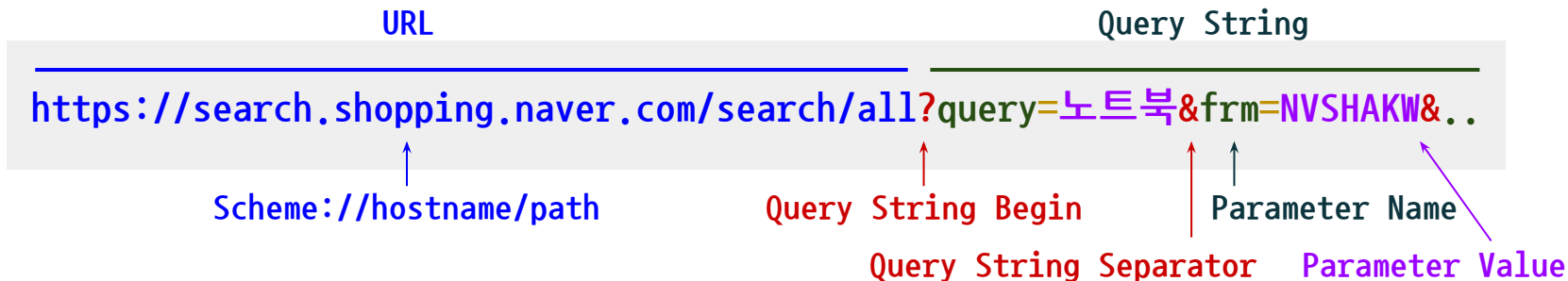
구분	세부 항목
요청 라인	<ul style="list-style-type: none"> 요청 방식(GET or POST) - 경로(URI) HTTP 버전 등
요청 헤더	<ul style="list-style-type: none"> 가능한 콘텐츠 형식(Content-Type) 가능한 인코딩 방식(Character set) 인증 스펙(Authorization) Cookies - Accept - Referer - User-agent
메시지 바디	<ul style="list-style-type: none"> 파라미터 할당(길이 제한 없음)

GET 방식과 POST 방식의 차이

구분	GET 방식	POST 방식
사전적 의미	받다, 얻다, 구하다, 가져오다 등	게시하다, 붙히다 등
요청메시지 확인	웹 브라우저의 주소창에 보이는 URI를 사용할 수 있습니다.	크롬 개발자도구에서 요청 URL과 Body를 찾아야 합니다.
공통	필요시 accept, referer, user-agent 등의 request headers를 추가합니다.	

URL vs URI

- URL은 Uniform Resource Locator의 머리글자로, 리소스가 포함되어 있는 위치를 의미합니다.
- URI는 Uniform Resource Indicator의 머리글자로, 리소스를 식별하는 문자열을 차례대로 배열한 것으로 URL과 Query String을 결합한 형태입니다.
- [예시] 네이버 쇼핑에서 '노트북'으로 검색한 웹 페이지 주소



HTTP 응답

- 웹 서버는 클라이언트의 요청에 대해 헤더와 바디로 구성된 응답 메시지를 발송합니다.
- 응답 헤더에는 HTTP 버전, 상태코드, 요청 일시, 콘텐츠 형태, 인코딩 방식, 크기 등이 포함되어 있습니다.
- 바디에는 HTML 본문이 포함됩니다.
 - 상태코드에 따라 HTML에 포함되는 내용이 달라집니다!

상태코드	내용
1XX	정보 교환
2XX	데이터 전송 성공 or 수락됨
3XX	방향 바꿈(server-side redirect)
4XX	클라이언트 오류 (주소 오류, 권한 없음, 접근 금지 등)
5XX	서버 오류 (올바른 요청을 처리할 수 없음 등)

httr 패키지 소개

- httr은 HTTP 요청 및 응답에 관한 작업에 사용되는 패키지입니다.

```
> library(httr)
```

- 주요 함수는 다음과 같습니다.
 - HTTP 요청: GET(), POST()
 - HTTP 요청 헤더: user_agent(), add_headers(), set_cookies()
 - HTTP 응답 헤더: status_code(), content(), headers(), cookies()

HTML 요소 기초

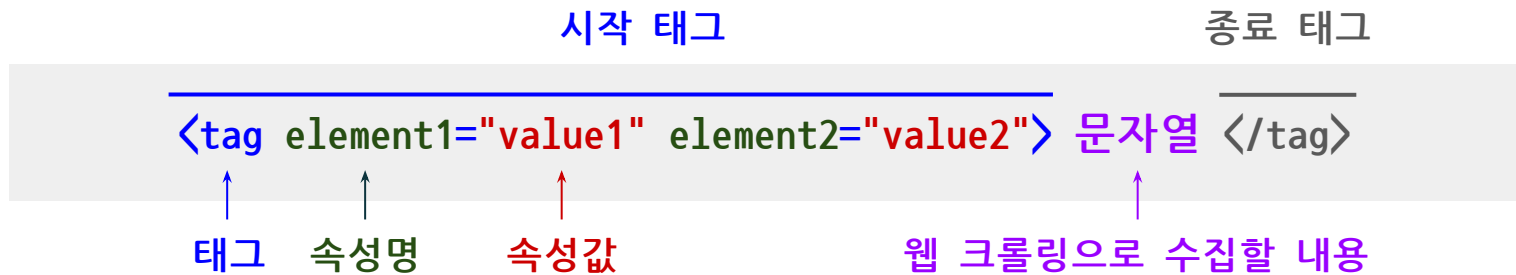
HTML HyperText Markup Language

- HTML은 웹 페이지의 제목, 단락, 목록 등 문서 구조를 나타내는 마크업 언어입니다.
- HTML은 꺾쇠 괄호 < > 안에 태그를 포함하는 HTML 요소^{element} 형태로 작성됩니다.
- 요즘 제작되는 웹 페이지에는 HTML 콘텐츠의 스타일을 담당하는 CSS와 웹 브라우저의 동적 제어를 담당하는 JavaScript가 함께 사용되어 사용자와 상호작용할 수 있도록 구현됩니다.



HTML 요소

- HTML 요소는 HTML 문서나 웹 페이지를 구성하는 개별 항목을 의미합니다.
- HTML 요소는 시작 태그와 종료 태그로 작성되며, 그 사이에 텍스트가 포함됩니다.
 - 태그는 꺾쇠 괄호로 감싸줘어야 합니다.
 - 시작 태그에 속성명과 속성값이 포함되고, 종료 태그 앞에 슬래쉬(/)가 추가됩니다.
- 웹 크롤링은 수집하려는 문자열을 포함하는 HTML 요소를 찾는 것이 필수입니다.



HTML 주요 태그 소개 #1

태그	기능	태그	기능
body	HTML의 본문 포함	a	하이퍼링크 연결(href 파라미터 사용)
h1 ~ h6	제목 설정(숫자가 클수록 크기가 큼)	img	이미지 삽입(src 파라미터 사용)
hr / br	hr(수평줄 삽입), br(줄바꿈)	table	HTML 문서 내 표 추가
p	단락(paragraph) 구분	tr	표의 각 행(row) 설정
div	공간(division) 분할	th	표의 각 컬럼명(header) 설정
span	같은 라인에서 공간 분할(줄바꿈X)	td	표의 각 행에 포함되는 데이터(data)
li	목록(list) 생성	input	입력값을 받을 때 사용(로그인 등)
ol / ul	ol(순서형 목록), ul(순서 없는 목록)	iframe	웹 페이지에 다른 페이지 넣기

HTML 주요 태그 소개 #2

태그	기능	태그	기능
b	글자 굵게	del	삭제된 글자 표시로 가운데 선 추가
strong	글자 굵게(의미론적 강조)	ins	삽입된 글자 표시로 밑줄 추가
i	글자 기울이기	sub	아랫첨자로 표시
em	글자 기울이기(의미론적 강조)	sup	윗첨자로 표시
mark	글자에 배경색 추가	q	인용된(quoted) 글자로 따옴표 추가
small	글자 작게	blockquote	인용된 글자 덩어리를 들여쓰기로 표시
big	글자 크게	address	연락처 정보 표시로 글자 기울이기

CSS Selector과 XPath 비교

- CSS Selector는 HTML의 디자인을 담당합니다.
 - HTML 요소에 포함된 Selector를 참조하여 웹 브라우저에 출력되는 모습(예를 들어 폰트, 컬러, 크기, 굵기 등)을 변경합니다.
- XPath는 XML 문서의 특정 부분을 찾을 때 사용하는 언어입니다.
 - 계층 구조를 갖는 XML 문서에서 노드와 속성을 탐색할 때 사용됩니다.
- 위 두 가지는 HTML 요소를 지정할 때 (우열 없이) 사용됩니다.
 - 다만 Selenium에서는 Xpath가 사용될 때 더 빠르다는 의견이 있습니다.
 - 하지만 가독성 측면에서는 CSS Selector이 더 나을 수 있습니다.

CSS Selector와 XPath 표기법[매우 중요!]

CSS Selector		XPath	
표현식	의미	표현식	의미
태그명	태그명이 같은 모든 태그를 선택	노드명	노드명이 같은 모든 노드를 선택
>	앞 태그의 직계 자손 태그만 선택	/	루트노드부터 탐색. 직계로 연결
#	속성명이 id인 태그를 선택	//	위치와 상관없이 지정된 노드부터 탐색
.	속성명이 class인 태그를 선택	.	현재노드 선택('..'는 부모노드 선택)
[]	속성을 지정할 때 사용	@	속성노드 선택
:nth-child	n번째 태그를 선택	[]	속성 지정 및 n번째 노드를 선택

CSS Selector와 XPath 표기법[매우 중요!]

목표	CSS Selector	XPath
모든 요소	*	//*
p 태그를 포함하는 모든 요소	p	//p
p 태그의 모든 자식 요소(p는 제외)	p > *	//p/*
id가 "foo"인 모든 요소	#foo	//*[@id="foo"]
class가 "foo"인 모든 요소	.foo	//*[@class="foo"]
title="header" 속성을 포함하는 모든 요소	*[title="header"]	//*[@title="header"]
li 태그 중 n번째인 요소	li:nth-child(n)	//li[n]
ul 자식 태그인 li 태그 중 n번째인 요소	ul > li:nth-child(n)	//ul/li[n]

rvest 패키지 소개

- rvest는 웹 페이지로부터 데이터를 수집할 때 사용하는 패키지입니다.

```
> library(rvest)
```

- 주요 함수는 다음과 같습니다.
 - 응답 객체를 HTML로 변환: `read_html()`
 - HTML 요소 선택: `html_node()`, `html_nodes()`
 - HTML 요소의 데이터 추출: `html_text()`, `html_table()`
 - HTML 요소의 속성 추출: `html_attr()`, `html_attrs()`, `html_name()`

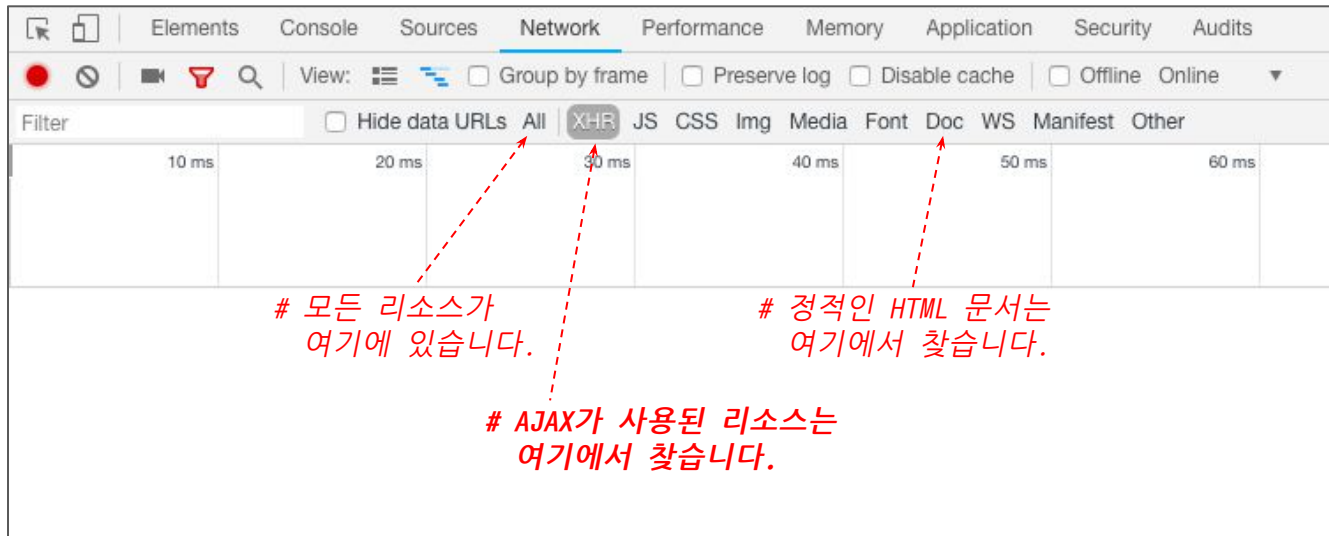
크롬 개발자도구 사용법

크롬 개발자도구를 사용하는 이유

- 웹 크롤링 과정에서 크롬 개발자도구를 사용하여 두 가지를 해결할 수 있습니다.
 - HTTP 요청 과정에서 웹 서버로부터 받은 리소스를 찾습니다.[Network]
 - 수집할 텍스트를 담고 있는 HTML 요소를 찾습니다.[Elements]
- 크롬 개발자도구를 여는 방법은 다음과 같이 두 가지가 있습니다.
 - 크롬 브라우저 오른쪽 상단에 있는 3점 메뉴(:) -> 도구 더보기(More Tools) -> 개발자 도구(Developer Tools)를 차례로 선택합니다.
 - 크롬 브라우저 안에서 오른쪽 마우스를 클릭하면 팝업 메뉴가 열리며, 팝업 메뉴 중에서 검사(Inspect)를 선택합니다.

통신 리소스 찾기: Network 탭

- 크롬 개발자도구 네트워크 탭에서 HTTP 통신에 사용된 리소스를 종류별로 확인할 수 있습니다.



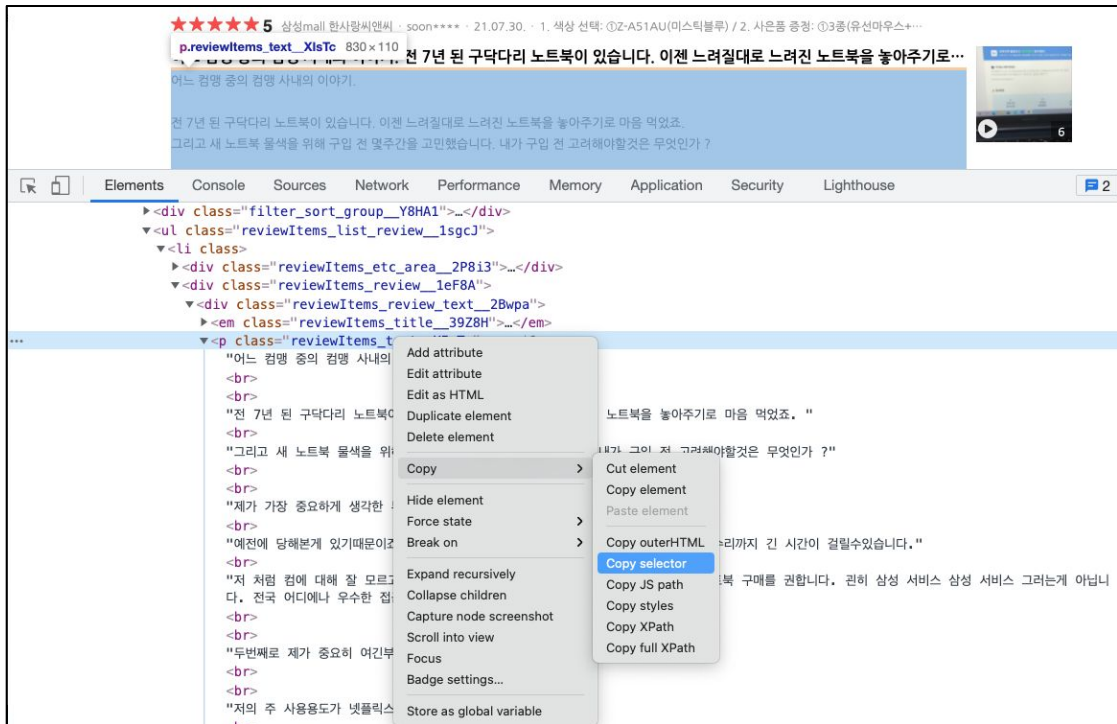
HTML 요소 찾기: Element 탭

- 크롬 개발자도구에서 **Elements** 탭을 선택하고 빨간 점선으로 표시된 버튼을 한 번 클릭하면 해당 버튼이 파란색으로 바뀝니다.
- 이 상태로 마우스를 웹 페이지 위로 이동하면 마우스가 가리키는 내용을 포함하는 HTML 요소가 파란색 블록으로 표시되며, 마우스를 클릭하면 고정시킵니다.



HTML 요소 선택: CSS Select 또는 XPath

- HTML 요소 위에서 마우스 오른쪽 버튼을 클릭합니다.
- 메뉴에서 **Copy**를 선택하면 하위 메뉴가 열립니다.
- Copy selector**를 선택하면 HTML 요소가 복사됩니다.
 - Copy XPath**도 가능합니다.
- RStudio로 이동하여 복사한 값을 붙여 넣습니다.



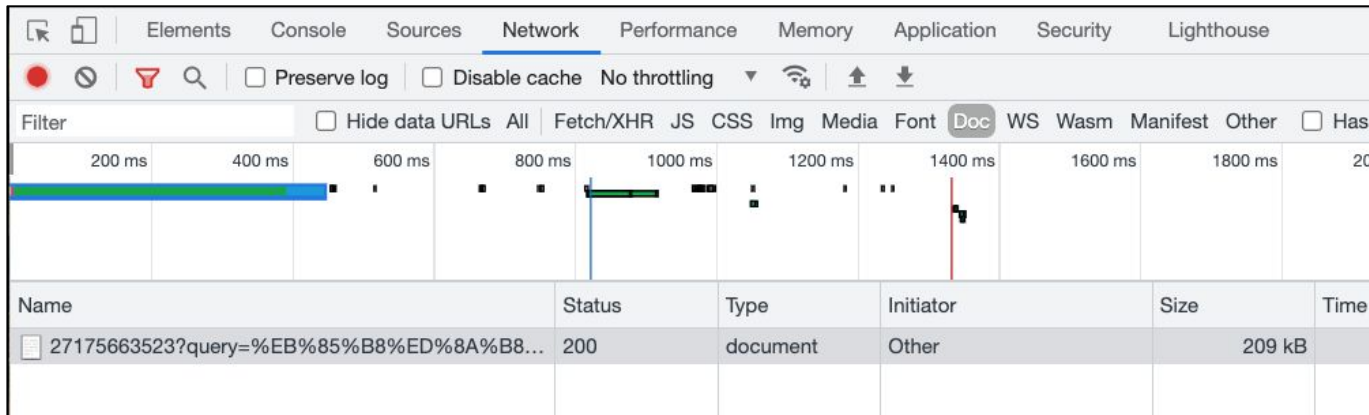
[실습] 네이버 쇼핑 상품 리뷰 수집

네이버 쇼핑 수집 과정

- 네이버 쇼핑(<https://shopping.naver.com/>)에서 아래 과정을 실행합니다.
 - 관심 있는 검색어로 조회한 다음, 마음에 드는 상품을 클릭합니다.
 - 웹 페이지 중간에서 **쇼핑몰리뷰** 버튼을 클릭합니다.
 - 크롬 개발자도구를 열고, Network 탭의 **Doc**로 이동합니다.
 - 웹 페이지를 새로고침 합니다.
 - '숫자'로 시작하는 리소스를 클릭하고, **Preview**에서 리뷰 미리보기를 찾습니다.
 - **Headers**로 이동하여 HTTP 요청 방식(GET), URL 및 Query String을 찾습니다.
 - 2페이지부터는 JavaScript가 사용되므로, 이번 예제에서는 생략합니다.

네이버 쇼핑 리뷰 관련 리소스 찾기

- 네이버 쇼핑에서 크롬 개발자도구를 열고 Network 탭의 Doc로 이동하여 쇼핑 리뷰 관련 통신 방식을 찾습니다.



'숫자'로 시작하는 리소스 하나만 포함되어 있습니다.

네이버 쇼핑 리뷰 관련 리소스 찾기(계속)

- 리소스를 클릭하여 **Preview**에서 리뷰 미리보기를 확인합니다.

The screenshot shows the Chrome DevTools Network tab. The 'Preview' sub-tab is selected, displaying a preview of a product review. The review is for a Samsung A51AU (Mystic Blue) smartphone, dated 2021.07.30. The review text is in Korean and mentions a 7-year-old Samsung laptop. The interface includes a timeline at the top showing the request duration, and various tabs for Headers, Preview, Response, Initiator, Timing, and Cookies.

Network tab interface showing a resource preview for a Samsung product review. The preview content includes a 5-star rating, the product name (삼성mall 한사람씨앤씨 · soon****), the date (21.07.30), and the review text (어느 컴맹 중의 컴맹 사내의 이야기. 전 7년 된 구닥다리 노트북이 있습니다. 이젠 느려질...).

- **Headers**로 이동하면 HTTP 요청 관련 파라미터를 찾을 수 있습니다.

요청 URL은 Request URL에서 물음표 앞부분까지 복사하면 됩니다.

Query String은 마우스를 맨 아래로 이동하면 잘 정리되어 있습니다.

JavaScript를 활용한 웹 크롤링

JavaScript의 개요

- JavaScript는 객체 기반의 스크립트 언어입니다.
 - 스크립트 언어는 간단한 작업을 반복적으로 실행하는데 적합한 프로그래밍 언어입니다.
 - 한 줄씩 바로 실행이 되는 R과 Python도 스크립트 언어입니다
 - 스크립트 언어는 인터프리터가 필요합니다. C는 컴파일러가 필요한 컴파일 언어입니다.
 - JavaScript는 Google의 V8 엔진이 인터프리터의 역할을 합니다.
- JavaScript는 HTML 및 CSS와 함께 사용됩니다.
 - HTML은 웹 페이지의 전체 틀을 잡고, CSS는 개별 요소의 스타일(디자인)을 말합니다.
 - JavaScript는 사용자와의 상호작용을 통해 웹 페이지에서 보여주는 콘텐츠를 동적으로 제어합니다.

AJAX와 XHR

- AJAX는 JavaScript 라이브러리 중 하나이며, Asynchronous Javascript And XML의 머리글자입니다.
 - 웹 브라우저를 새로고침하면 현재 웹 페이지가 새로 열립니다.
 - 이렇듯 웹 페이지의 전체 요소가 한꺼번에 바뀌는 것을 동기 방식이라고 합니다.
- AJAX는 웹 서버와 통신할 때 전체 웹 페이지를 새로고침하지 않고 특정한 부분에 사용되는 데이터만 웹 서버로부터 내려받으므로 비동기 방식이라고 합니다.
- AJAX는 HTTP 요청 대신 XHR^{XML Http Request} 객체를 사용합니다.
- AJAX는 웹 서버와의 통신을 통해 XML 및 JSON 데이터를 주고 받습니다.

XML

- XML은 인터넷으로 연결된 시스템끼리 데이터를 주고 받기 위해 만들어졌습니다.
 - XML은 Extensible Markup Language의 줄임말로 마크업 언어입니다.
 - XML은 사람과 컴퓨터가 인식할 수 있도록 유니코드를 사용합니다.
 - 많은 API에서 XML로 텍스트 데이터를 주고 받습니다.
- XML은 HTML과 비슷합니다.
 - HTML처럼 시작 노드와 종료 노드 사이에 텍스트가 포함됩니다.
 - HTML과 다른 점은 노드가 특정 기능 없이 컬럼명처럼 사용된다는 점입니다.
 - 따라서 XML의 노드는 대소문자를 구분합니다.

JSON

- JSON은 JavaScript로 데이터를 주고 받을 때 사용되는 교환 형식입니다.
 - JSON은 JavaScript Object Notation의 줄임말입니다.
 - JSON도 사람과 컴퓨터가 인식하기 쉽고 용량도 작다는 장점이 있습니다.
 - 따라서 API에서 XML을 대체하는 데이터 교환 형식으로 많이 사용됩니다.
- JSON은 다음과 같습니다.
 - 중괄호 {} 안에 Key:Value로 결합된 원소가 콤마로 연결됩니다.
 - Python의 딕셔너리 자료구조와 같습니다.
 - Value에 포함되는 값이 여러 개이면 대괄호 안에 콤마로 연결됩니다.

XML과 JSON 데이터 처리

- XML은 HTML과 동일한 방법으로 처리할 수 있습니다.
 - rvest 패키지 대신 xml2 패키지의 함수가 사용됩니다.
 - [주의] XML Node로 사용된 문자열은 반드시 대소문자를 구분해야 합니다.

```
> res %>% read_xml() %>% xml_nodes('xml node') %>% xml_text()
```

- JSON은 res에 포함된 문자열을 처리합니다.
 - jsonlite 패키지의 fromJSON() 함수는 JSON 문자열로 리스트를 생성합니다.
 - [주의] JSON 문자열이 중괄호로 시작하거나 끝나도록 처리해야 하는 경우가 있습니다.

```
> res %>% content(as = 'text') %>% fromJSON()
```

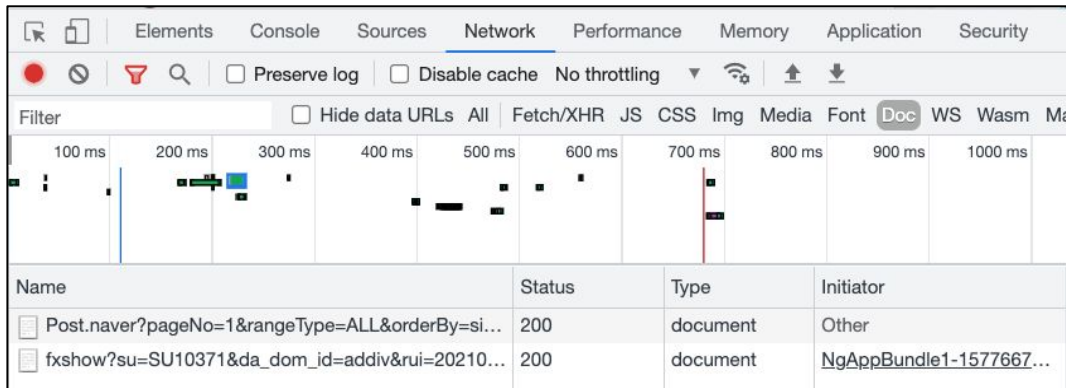
[실습] 네이버 블로그 본문 수집

네이버 블로그 수집 과정

- 네이버 블로그(<https://section.blog.naver.com/>)에서 아래 과정을 실행합니다.
 - 관심 있는 검색어로 조회합니다.
 - 기간 전체를 클릭하고, 빈 칸에 원하는 날짜를 **yyyy-mm-dd** 형태로 입력합니다.
 - 정렬 방식은 **정확도**를 선택합니다.
 - 크롬 개발자도구를 열고, Network 탭의 **Fetch/XHR**로 이동합니다.
 - 하단의 페이지 네비게이션에서 2를 클릭합니다.(최대 572 페이지까지 제공합니다.)
 - '**SearchList**'로 시작하는 리소스를 클릭하고, **Preview**에서 JSON 데이터를 찾습니다.
 - **Headers**로 이동하여 HTTP 요청 방식(GET), URL 및 Query String을 찾습니다.

네이버 블로그 검색 관련 리소스 찾기

- 크롬 개발자도구 Network 탭의 Doc로 이동하여 블로그 검색 결과 관련 통신 방식을 찾습니다.



The screenshot shows the Chrome DevTools Network tab with the 'Doc' filter selected. The timeline shows two document resources loaded. The table below is a representation of the data shown in the table at the bottom of the screenshot.

Name	Status	Type	Initiator
Post.naver?pageNo=1&rangeType=ALL&orderBy=si...	200	document	Other
fxshow?su=SU10371&da_dom_id=addiv&rui=20210...	200	document	NgAppBundle1-1577667...

각 리소스를 클릭하면 오른쪽에 상세 내용이 열립니다.

상세 내용의 메뉴 중에서 **Preview**를 선택하면 이 리소스가 웹 브라우저에 보여주는 내용을 확인할 수 있습니다.

2개의 리소스는 블로그 검색 결과와는 아무런 관련이 없습니다.

네이버 블로그 검색 관련 리소스 찾기(계속)

- Fetch/XHR로 이동하여 리소스를 하나씩 클릭하여 Preview를 확인합니다.

The screenshot shows the Chrome DevTools Network tab with the 'Preview' sub-tab active. The selected resource is 'SearchList.naver?cou...'. The JSON preview shows a search result with a list of 7 blogs and a total count of 3316362.

```

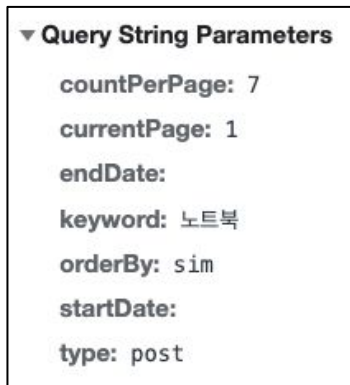
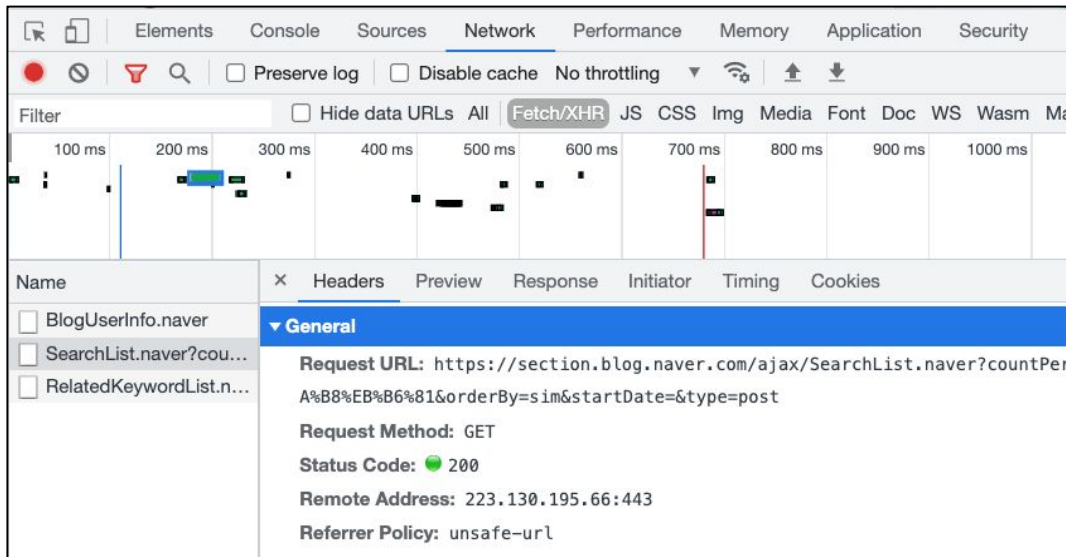
{
  "result": {
    "searchDisplayInfo": {
      "authUrlType": "LOGIN",
      ...
    },
    "isAdultUser": false,
    "pagePerCount": 7,
    "searchDisplayInfo": {
      "authUrlType": "LOGIN",
      ...
    },
    "searchList": [
      {
        "blogId": "hupers",
        "logNo": 222454703800,
        "gdid": "90000003_0000000000"
      },
      {
        "blogId": "rlacofls63",
        "logNo": 222407875539,
        "gdid": "90000003_00000000"
      },
      {
        "blogId": "hogenggggy",
        "logNo": 222425519782,
        "gdid": "90000003_00000000"
      },
      {
        "blogId": "dbsgns2011",
        "logNo": 222408804509,
        "gdid": "90000003_00000000"
      },
      {
        "blogId": "lucky_box7",
        "logNo": 222453254657,
        "gdid": "90000003_00000000"
      },
      {
        "blogId": "mansaa",
        "logNo": 222451058770,
        "gdid": "90000003_0000000000"
      },
      {
        "blogId": "yohhhhj",
        "logNo": 222438788324,
        "gdid": "90000003_0000000000"
      }
    ],
    "totalCount": 3316362
  }
}

```

'searchList'로 시작하는 리소스에
검색 결과가 포함되어 있습니다.

네이버 블로그 검색 관련 요청 파라미터 찾기

- Headers로 이동하면 HTTP 요청 관련 파라미터를 찾을 수 있습니다.



요청 URL은 Request URL에서 물음표 앞부분까지 복사하면 됩니다.

Query String은 마우스를 맨 아래로 이동하면 잘 정리되어 있습니다.

네이버 블로그 요청 헤더 추가

- 지금까지 HTTP 요청을 실행할 때, 요청 헤더를 추가하지 않았지만 실제로는 요청 헤더를 추가해야 HTTP 요청이 제대로 실행되는 경우가 있습니다.
 - 네이버 일부 서비스는 요청 헤더 중 Accept, Referer, User-agent를 요구합니다.
- **Accept**는 웹 서버에 요청하는 Content-Type을 설정합니다.(html, json 등)
- **Referer**는 웹 페이지 간 이동할 때 남는 흔적을 의미합니다.
 - 하이퍼링크를 통해 어떤 한 사이트에서 다른 사이트로 이동한다고 했을 때, 새로 열린 사이트에 이전 사이트의 URI를 전송하는데 이것이 리퍼러입니다.
- **User-agent**는 클라이언트의 운영체제와 브라우저의 종류 등을 담은 문자열입니다.
 - 웹 서버는 User-agent를 통해 클라이언트의 정보를 확인합니다.

네이버 블로그 요청 헤더 찾기

- Request headers에서 accept 및 referer를 탐색합니다.

The screenshot shows the Chrome DevTools Network tab. The 'Headers' sub-tab is selected, displaying the 'Request Headers' for a GET request to a Naver blog search endpoint. The headers include authority, method, path, scheme, accept, accept-encoding, accept-language, cookie, and referer.

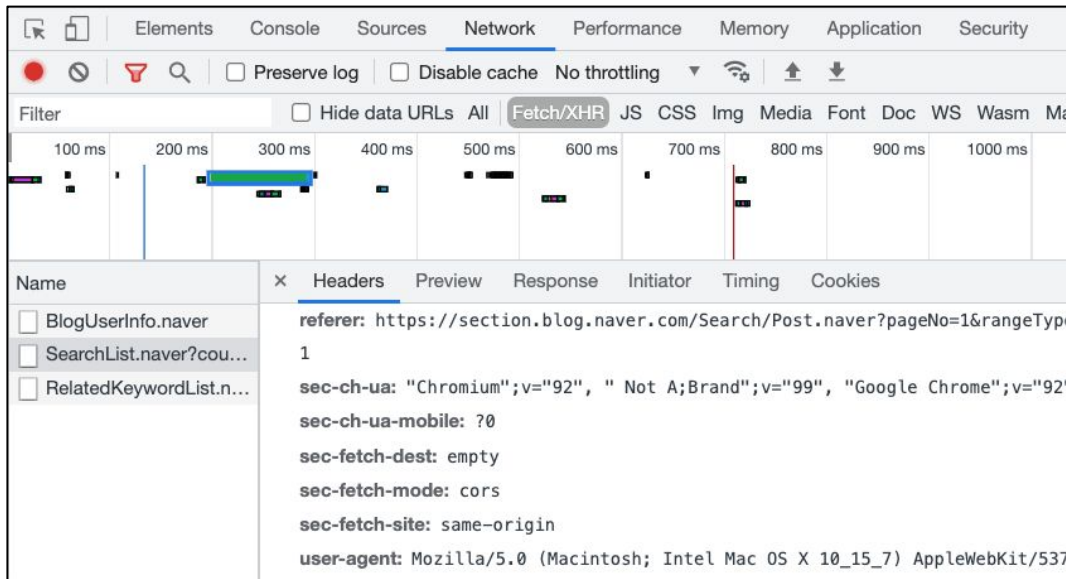
Name	Value
<input type="checkbox"/> BlogUserInfo.naver	
<input type="checkbox"/> SearchList.naver?cou...	
<input type="checkbox"/> RelatedKeywordList.n...	
Request Headers	
:authority:	section.blog.naver.com
:method:	GET
:path:	/ajax/SearchList.naver?countPerPage=7¤tPage=1&endDate=&keyword...
:scheme:	https
accept:	application/json, text/plain, */*
accept-encoding:	gzip, deflate, br
accept-language:	ko,en-US;q=0.9,en;q=0.8
cookie:	NNB=77HRENC5J4EWC; JSESSIONID=518D3EEB5F079F252BD9D78A7523A0B3.jvm
referer:	https://section.blog.naver.com/Search/Post.naver?pageNo=1&rangeType

accept 전체를 복사합니다.

referer에서 물음표 앞까지 복사합니다.

네이버 블로그 요청 헤더 찾기(계속)

- Request headers에서 user-agent를 탐색합니다.



The screenshot shows the Chrome DevTools Network tab. The 'Headers' sub-tab is active. The list of headers includes:

- referer: https://section.blog.naver.com/Search/Post.naver?pageNo=1&rangeType=1
- sec-ch-ua: "Chromium";v="92", " Not A;Brand";v="99", "Google Chrome";v="92"
- sec-ch-ua-mobile: ?0
- sec-fetch-dest: empty
- sec-fetch-mode: cors
- sec-fetch-site: same-origin
- user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537...

user-agent의 일부를 복사합니다.

페이지 네비게이션

- 네이버 블로그는 검색 결과를 페이지당 7개의 게시글을 보여주며, 다음 페이지로 이동하려면 하단에 있는 페이지 이동 버튼을 클릭해야 합니다.



- 위와 같이 전체 게시글의 일부를 여러 페이지에 걸쳐 노출함으로써 사용자가 현재 위치를 파악하는데 도움을 주는 기능을 페이지 네비게이션이라고 합니다.
- 페이지 네비게이션은 JavaScript 라이브러리 중 하나인 jQuery를 활용하는 방식이 많이 사용됩니다.
- 2 페이지를 클릭하고, SearchList로 시작하는 리소스의 요청 파라미터를 확인하면 currentPage가 2로 변경됩니다.

[실습] 네이버 카페 본문 수집

네이버 카페 수집 과정

- 네이버 카페(<https://section.cafe.naver.com/>)에서 아래 과정을 실행합니다.
 - 관심 있는 검색어로 조회하고, 화면 중간에 있는 **전체글 더보기** 버튼을 클릭합니다.
 - 등록기간은 **직접입력**을 선택하고, 원하는 날짜를 **yyyy.mm.dd** 형태로 입력합니다.
 - 영역은 **제목+본문**, 글 분류는 **거래글 제외**, 유사문서 **제외**, 출처 **전체**를 선택합니다.
 - 크롬 개발자도구를 열고, Network 탭의 **Fetch/XHR**로 이동합니다.
 - 하단의 페이지 네비게이션에서 2를 클릭합니다.(최대 100 페이지까지 제공합니다.)
 - **'articles'**로 시작하는 리소스를 클릭하고, **Preview**에서 JSON 데이터를 찾습니다.
 - **Headers**로 이동하여 HTTP 요청 방식(POST), URL 및 Body(Payload)를 찾습니다.

네이버 카페 전체글 검색 화면

NAVER 카페
전체글
노트북
통합검색
로그인

카페홈
주제별
지역별
랭킹
대표카페
내소식
채팅
카페지원센터

노트북에 대한 검색 결과

연관검색어

전체 카페 **전체글** 중고거래

영역
☒ 게시글
☐ 제목

등록기간
☐ 6개월
☐ 1년
☒ 직접입력

2021.10.19 ~
2021.10.20
예) 2020.09.01

글 분류
☐ 전체
☒ 거래글 제외

유사문서
☒ 제외
☐ 포함

출처
☒ 전체
☐ 출처 선택

상세검색

다음 단어 모두 포함
다음 단어 제외
다음 단어중 1개 이상 포함
다음 어절 어구 정확히 일치
적용

전체글 3,364건

정확도순 최신평

노트북 구매예정인데 추천좀부탁드려요

노트북 레노버싱다가 불량품받고 반품후 다시 노트북 구매하려하는데요 전에 삼열열 추천은 받았는데 성능은 별로고 as만줄아서 다른 기기바이트 야수스 msi 보고있는데...

AMD 비헤라 라이젠 컴퓨터 사용자 모임 · 2021.10.19

노트북 이거 괜찮나요

노트북은 처음 사보고 잘몰라서 질문 드립니다. 아래 노트북 가격대는 56 입니다. 좀,메 만 할건데 70이하 선으로 추천해 주실 수 있을까요? 하드보스에서 핵 안걸렸으면...

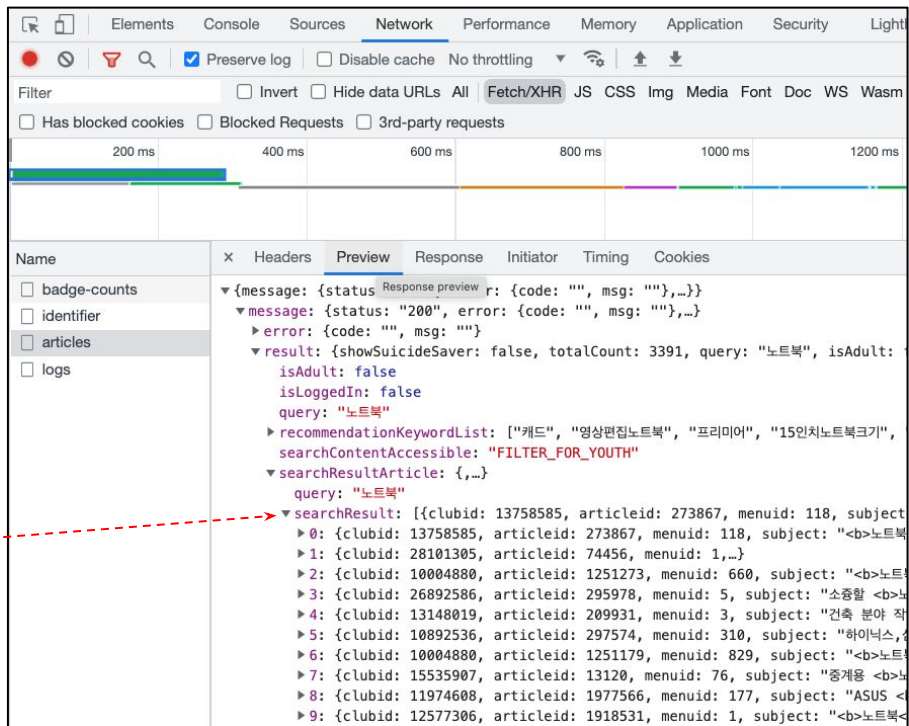
메이플스토리 - 유저 커뮤니티 · 2021.10.19

등록기간은 직접입력을 선택한 다음 시작일자와 종료일자를 입력합니다.

글 분류는 거래글 제외를 선택합니다.

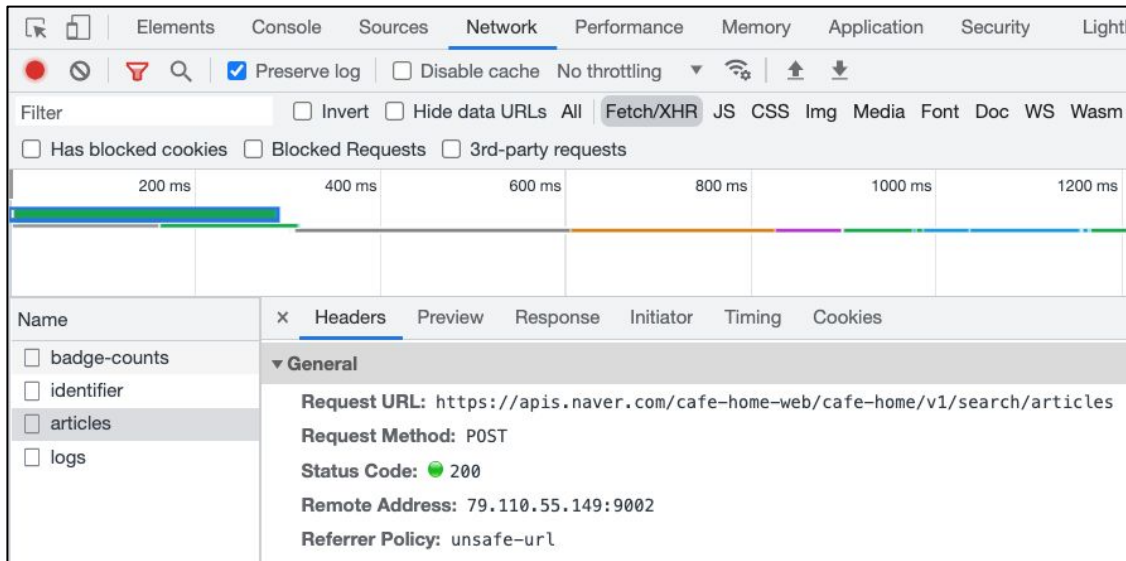
네이버 카페 검색 관련 리소스 찾기

- 크롬 개발자도구 Network 탭의 **Fetch/XHR**로 이동하여 리소스를 하나씩 클릭하여 **Preview**를 확인합니다.
- 'articles'로 시작하는 리소스를 클릭하면 Preview에서 JSON 데이터를 찾을 수 있습니다.
- 'searchResult'에 카페 게시글 관련된 JSON 데이터가 10개 있습니다.



네이버 카페 검색 관련 요청 파라미터 찾기

- Headers로 이동하면 HTTP 요청 관련 파라미터를 찾을 수 있습니다.



네이버 카페 검색 관련 요청 파라미터 찾기(계속)

- POST 방식이므로 하단에 있는 Body 또는 Request Payload를 찾아야 합니다.

The screenshot shows the Chrome DevTools Network tab. The 'Network' panel is active, displaying a list of requests. The first request is selected, and its details are shown in the 'Headers' panel. The 'Request Payload' is expanded, revealing the following JSON data:

```
{
  "query": "노트북",
  "page": 2,
  "sortBy": 0,
  "exceptMarketArticle": 1,
  "period": ["20211019", "20211020"]
}
```

The 'Request Payload' is highlighted in blue, indicating it is the selected view for the request details.

Payload 처리 방법

- POST 방식이 사용된 웹 페이지를 요청할 때, Body가 추가됩니다.
 - GET 방식의 query와 동일하게 리스트로 전달합니다.
 - 하지만 Payload일 때에는 리스트를 JSON 문자열로 변환하고 전달해야 합니다.
 - JSON에서 특정 원소의 값이 2개 이상이면 리스트로 생성합니다.
- 리스트를 JSON으로 변환하려면 toJson() 함수가 필요합니다.

```
> body <- list(a = 1, b = list(2, 3), ...) # 파라미터 b는 원소가 2개이므로 리스트로 생성합니다.
```

```
> body <- toJSON(x = body, auto_unbox = TRUE) # [중요] auto_unbox = TRUE를 추가하면 값을  
대괄호로 감싸지 않습니다.
```

```
> print(x = body)
```

네이버 카페 요청 헤더 찾기

- Request headers에서 content-type, referer 및 user-agent를 탐색합니다.

The screenshot shows the Chrome DevTools Network tab. The 'Network' tab is selected, and the 'Headers' sub-tab is active. The request is a POST to the URL `/cafe-home-web/cafe-home/v1/search/articles`. The 'Request Headers' section is expanded, showing the following headers:

- `:authority:` apis.naver.com
- `:method:` POST
- `:path:` /cafe-home-web/cafe-home/v1/search/articles
- `:scheme:` https
- `accept:` application/json, text/plain, */*
- `accept-encoding:` gzip, deflate, br
- `accept-language:` ko,en-US;q=0.9,en;q=0.8
- `content-length:` 98
- `content-type:` application/json; charset=UTF-8

content-type 전체를 복사합니다.

[참고] 네이버 카페 글쓰기 옵션과 크롤링 가능 여부

- 네이버 카페 편집기 아래에 있는 옵션에 따라 크롤링 가능 여부가 결정됩니다.
 - 기본값은 '멤버공개' & '검색 허용'이므로 **referer를 추가하면 크롤링이 가능합니다.**
 - '검색 허용'을 해제하면 멤버에게만 공개되므로 **referer를 추가해도 크롤링이 안됩니다.**
 - 이와 같은 카페 게시글 본문은 네이버에 로그인한 상태에서 Selenium으로 수집할 수 있습니다.
 - '전체공개'로 변경하면 모든 사람에게 공개되므로 **referer 없이도 크롤링이 가능합니다.**

공개설정

☐ 전체공개
 ☒ 멤버공개

☒ 검색 · 네이버 서비스공개 허용
 ? 검색 및 네이버 서비스를 통해 멤버가 아닌 사람도 글을 볼 수 있습니다.

기능설정

댓글 허용 | 블로그/카페 공유 허용 | 외부 공유 사용 | 마우스 오른쪽버튼 허용 | 동영상 공유 비허용

자동출처 사용 안함 | CCL 사용 안함

변경

End of Document