

ESTSOFT WASSUP

AI 서비스 기획

맹광국 강사

ggmaeng@gmail.com

데이터 수집

데이터 수집의 필요성

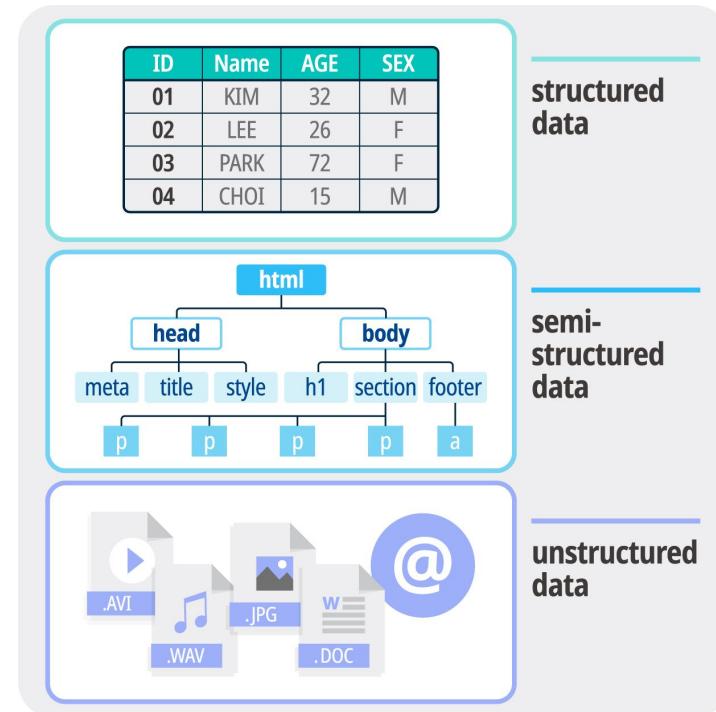
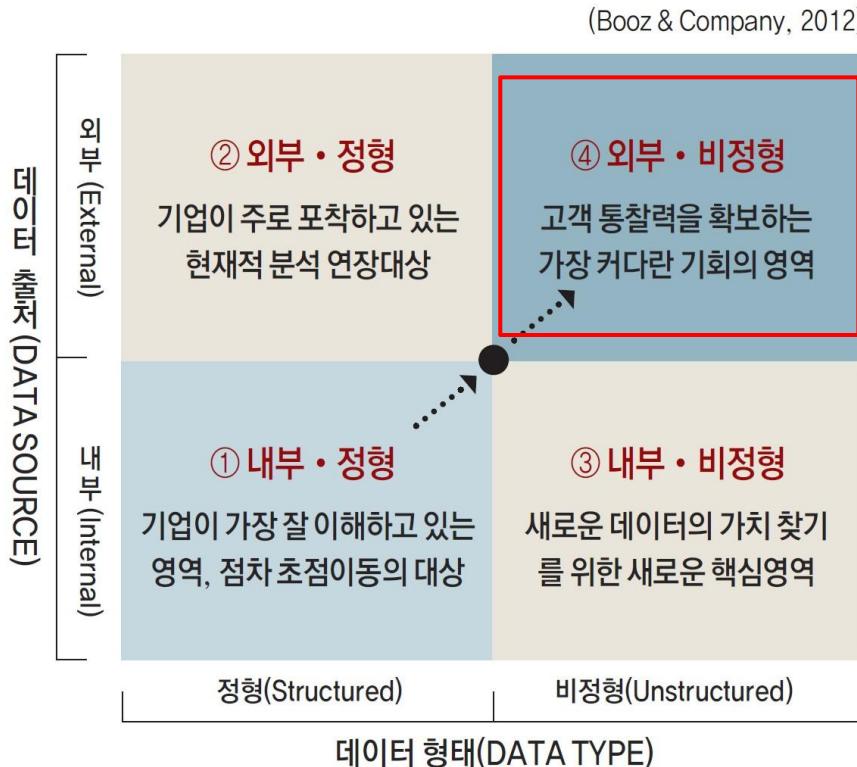
수집 데이터 형태 파악

데이터 형태별 수집 난이도/가치

수집 데이터의 형태에 따른 수집 방법 분류

데이터 수집의 필요성

EST



수집 데이터 형태 파악

EST

정형 데이터

특징

정형 데이터(Structured Data)는 관계형 데이터베이스 시스템의 테이블과 같이 고정된 컬럼에 저장되는 데이터와 파일, 그리고 지정된 행과 열에 의해 데이터의 속성이 구별되는 스프레드시트 형태의 데이터도 있을 수 있다. 관계형 데이터베이스 시스템의 정형 데이터를 비정형 데이터(Unstructured Data)와 비교할 때 가장 큰 차이점은 데이터의 스키마를 지원하는 것이다.

데이터 탐색

스키마에 의해 정의된 컬럼			
column1	column2	column3	column4
data	data	data	data
data	data	data	data
data	data	data	data
data	data	data	data

컬럼에 의해 정의된 데이터

[그림 I-1-1] 정형 데이터의 구조

형태

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

정형 데이터의 경우, 스케마 구조를 가지고 있기 때문에 데이터를 탐색하는 과정이 테이블 탐색, 컬럼 구조 탐색, 로우 탐색 순으로 정형화되어 있다.

예) SELECT COLUMN1, COLUMN2... FROM TABLE WHERE CONDITION

정형 데이터의 예

RDBMS의 테이블들(단일 테이블 혹은 조인한 테이블 포함)
스프레드시트

반정형 데이터(Semi-Structured Data)

특징

정형 데이터는 데이터의 스키마 정보를 관리하는 DBMS와 데이터 내용이 저장되는 데이터 저장소로 구분되지만, 반정형 데이터는 데이터 내부에 정형데이터의 스키마에 해당되는 메타데이터를 갖고 있으며. 일반적으로 파일 형태로 저장된다.

데이터 탐색	형태
<pre>[{"Sepal.Length": 6.8, "Sepal.Width": 3.2, "Petal.Length": 5.9, "Petal.Width": 2.3, "Species": "virginica"}, {"Sepal.Length": 6.7, "Sepal.Width": 3.3, "Petal.Length": 5.7, "Petal.Width": 2.5, "Species": "virginica"}]</pre>	<pre>[{"Sepal.Length": 5.1, "Sepal.Width": 3.5, "Petal.Length": 1.4, "Petal.Width": 0.2, "Species": "setosa"}, {"Sepal.Length": 4.9, "Sepal.Width": 3, "Petal.Length": 1.4, "Petal.Width": 0.2, "Species": "setosa"}, {"Sepal.Length": 4.7, "Sepal.Width": 3.2, "Petal.Length": 1.3, "Petal.Width": 0.2, "Species": "setosa"}, {"Sepal.Length": 4.6, "Sepal.Width": 3.1, "Petal.Length": 1.5, "Petal.Width": 0.2, "Species": "setosa"}]</pre>
	<p>반정형 데이터의 예</p> <p>URL 형태로 존재 - HTML 오픈 API 형태로 제공 - XML, JSON 로그형태 - 웹로그, IOT에서 제공하는 센서 데이터</p>

반정형 데이터의 경우 데이터 내부에 데이터 구조에 대한 메타정보를 갖고 있기 때문에 어떤 형태를 가진 데이터인지를 파악하는 것이 필요하다. 데이터 내부에 있는 규칙성을 파악해 데이터를 파싱할 수 있는 파싱 규칙을 적용한다.

비정형 데이터

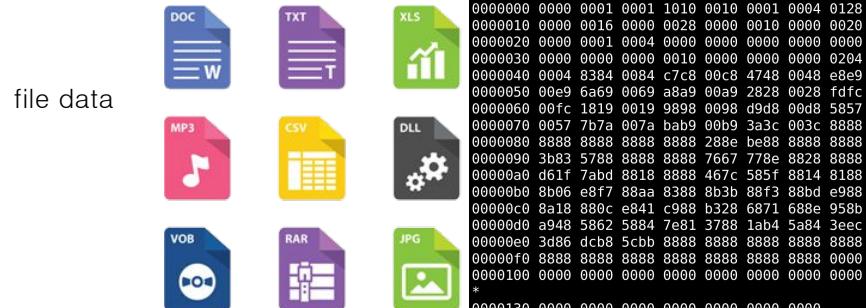
특징

비정형 데이터(Unstructured-Data)는 데이터 세트가 아닌 하나의 데이터가 수집 데이터로 객체화돼 있다. 언어 분석이 가능한 텍스트 데이터나 이미지, 동영상 같은 멀티미디어 데이터가 대표적인 비정형 데이터다. 웹에 존재하는 데이터의 경우 html 형태로 존재하여 반정형 데이터로 구분할 수도 있지만, 특정한 경우 텍스트 마이닝을 통해 데이터를 수집하는 경우도 존재하므로 명확한 구분은 어렵다.

데이터 탐색

이진 파일 형태: 동영상, 이미지

스크립트 파일 형태: 소셜 데이터의 텍스트



file data

바이너리 데이터

```
0000000 0000 0001 0001 1010 0010 0001 0004 0128  
0000010 0000 0016 0000 0028 0000 0010 0000 0020  
0000020 0000 0001 0004 0000 0000 0000 0000 0000  
0000030 0000 0000 0000 0010 0000 0000 0000 0204  
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9  
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc  
0000060 00fc 1819 0019 9998 0098 d9d8 00d8 5857  
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888  
0000080 8888 8888 8888 8888 288e be88 8888 8888  
0000090 3b83 5788 8888 8888 7667 778e 8828 8888  
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188  
00000b0 8b06 e9f7 88aa 8388 8b3b 88f3 88bd e988  
00000c0 8a18 880c e841 c988 b328 6871 688e 958b  
00000d0 a948 5862 5884 7e81 3788 1ab4 5848 3eec  
00000e0 3d86 dcbb 8888 8888 8888 8888 8888 8888  
00000f0 8888 8888 8888 8888 8888 8888 8888 0000  
0000100 0000 0000 0000 0000 0000 0000 0000 0000  
*  
0000130 0000 0000 0000 0000 0000 0000 0000 0000  
000013e
```

이진 파일 형태의 데이터일 때, 데이터를 탐색하는 방법은 데이터의 종류별로 응용소프트웨어를 이용하여 탐색한다.

예) 동영상: 동영상 플레이어 (스크립트 파일 형태일 경우 데이터를 파싱해 처리)

데이터 형태별 수집 난이도/가치

EST

형태	특징	< 데이터 형태별 수집 난이도 비교 >	난이도
정형 데이터	내부 시스템인 경우가 대부분이라 수집이 쉽다. 파일 형태의 스프레드시트라도 내부에 형식을 가지고 있어 처리가 쉬운 편이다.		하
반정형 데이터	보통 API 형태로 제공되기 때문에 데이터 처리 기술이 요구 된다.		중
비정형 데이터	텍스트 마이닝 혹은 파일일 경우 파일을 데이터 형태로 파싱해야 하기 때문에 수집 데이터 처리가 어렵다.		상

형태	특징	< 데이터 형태별 아키텍처 구성 난이도 비교 >	난이도
정형 데이터	CRUD가 일어나는 일반적인 아키텍처 구조로 이루어져 있다.		하
반정형 데이터	데이터의 메타구조를 해석해 정형 데이터 형태로 바꿀 수 있는 아키텍처 구조를 수정해야 한다.		중
비정형 데이터	텍스트나 파일을 파싱해 메타구조를 갖는 데이터의 셋 형태로 바꾸고 정형 데이터 형태의 구조로 만들 수 있도록 아키텍처 구조를 수정해야 한다.		상

형태	특징	< 데이터 형태별 잠재가치 비교 >	잠재가치
정형 데이터	내부 데이터의 특성상 현실적 가치의 한계상 활용측면에서 잠재적 가치는 상대적으로 낮다.		보통
반정형 데이터	데이터의 제공자가 선별해 제공하는 데이터로 잠재적 가치는 정형 데이터 보다 높다.		높음
비정형 데이터	수집주체에 의해 데이터에 대한 분석이 선행되었기 때문에 목적론적 데이터 특징이 가장 잘 나타나는 데이터이다. 그렇기 때문에 일단 수집이 가능하면 수집주체에게는 가장 높은 잠재적 가치를 제공한다.		매우높음

수집 데이터의 형태에 따른 수집 방법

분류

EST

수집 데이터 결정 → 수집 방법 결정 → 다양한 수집기술을 선택해 적용

데이터 유형	데이터 종류	수집
정형 데이터	RDB, 스프레드 시트	ETL, FTP, Open API
반정형 데이터	HTML, XML, JSON, 웹문서, 웹로그, 센서 데이터	Crawling, RSS, Open API, FTP
비정형 데이터	소셜 데이터, 문서, 이미지, 오디오, 비디오, IoT	Crawling, RSS, Open API, FTP, Streaming

- Download : txt, csv
- RSS : Feed. TCP/IP 프로토콜을 이용하는 인터넷 서버로부터 파일을 송수신.
- Open API : Application. Web 운영 주체가 공개하는 API.
- Crawling : 인터넷 긁어오기. 외부 데이터에 대한 HTTP 수집 (SNS, News, Website..)
- RSS (Rich Site Summary) : 뉴스 구독 등. Web 기반 최신 데이터 공유를 위한 XML 기반 콘텐츠 공유 (배급 프로토콜).
- Streaming : 음성/오디오/비디오 실시간 수집
- Aggregator : Log 데이터 수집 (ChuKwa, Flume, Scribe 등). RDB 관계형 DB로부터 NoSQL, Hadoop 등에 저장 처리 (Scoop, Direct JDBC/ODBC).

웹 서비스의 이해

웹 어플리케이션 아키텍처

웹서비스 프로세스

WWW(World Wide Web)

HTTP와 HTTPS, DNS

Web Server 와 WAS

월드 와이드 웹

文 A 143개 언어 ▾

문서 토론

읽기 편집 역사 보기 도구 ▾

위키백과, 우리 모두의 백과사전.

☞ 이 문서는 인터넷의 정보 공간에 대해 설명하고 있습니다. 다른 뜻에 대해서는 웹, WWW (동음이의) 문서를 참고하십시오.

☞ 웹 브라우저에 대해서는 월드와이드웹 문서를 참고하십시오.

월드 와이드 웹(World Wide Web, WWW, W3)은 인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간을 말한다. 간단히 웹(the Web)이라 부르는 경우가 많다. 이 용어는 인터넷과 동의어로 쓰이는 경우가 많으나 엄격히 말해 서로 다른 개념이다. 웹은 전자 메일과 같이 인터넷 상에서 동작하는 하나의 서비스일 뿐이다. 그러나 1993년 이래로 웹은 인터넷 구조의 절대적 위치를 차지하고 있다.

인터넷에서 HTTP 프로토콜, 하이퍼텍스트, HTML 형식 등을 사용하여 그림과 문자를 교환하는 전송방식을 말하기도 한다.

기본 개념 [편집]

인터넷상의 정보를 하이퍼텍스트 방식과 멀티미디어 환경에서 검색할 수 있게 해주는 정보검색 시스템이다. 하이퍼텍스트 형식으로 표현된 인터넷상의 다양한 정보를 효과적으로 검색하는 시스템으로 전 세계적으로 가장 널리 보급되어 있다.^[1]

하이퍼텍스트는 웹 브라우저라 불리는 프로그램을 통해 웹 서버에서 "문서"나 웹 페이지 등의 정보 조각을 읽어들여 컴퓨터 모니터에 출력하는 형태로 보이게 된다. 그리고 나서 사용자는 각 페이지에 있는 하이퍼링크를 따라 다른 문서로 이동하거나, 그 페이지를 서비스하고 있는 서버로 일련의 정보를 보낼 수도 있다. 하이퍼링크를 따라 이동하는 행위를 흔히 웹 서핑(web surfing, 문화어: 망유람^[2]) 또는 웹 브라우징이라 한다. 그리고 관련된 내용들이 모여있는 웹 페이지들의 집합을 웹 사이트라 한다.

영어 단어 월드와이드(worldwide)는 보통 공백이나 하이픈 없이 한 단어로 쓰이지만, 월드 와이드 웹(World Wide Web)과 그 약어인 WWW는 공식적인 영어 낱말로 사용되고 있다.

월드 와이드 웹은 다음의 세 가지 기능으로 요약할 수 있겠다. 첫 번째 통일된 웹 자원의 위치 지정 방법 예를 들면 URL. 두 번째 웹의 자원 이름에 접근하는 프로토콜(protocol) 예를 들면 HTTP. 세 번째 자원들 사이를 쉽게 항해 할 수 있는 언어 예를 들면 HTML.^[3]



로베르 카이오가 디자인한 역사적 월드 와이드 웹 로고

HTML

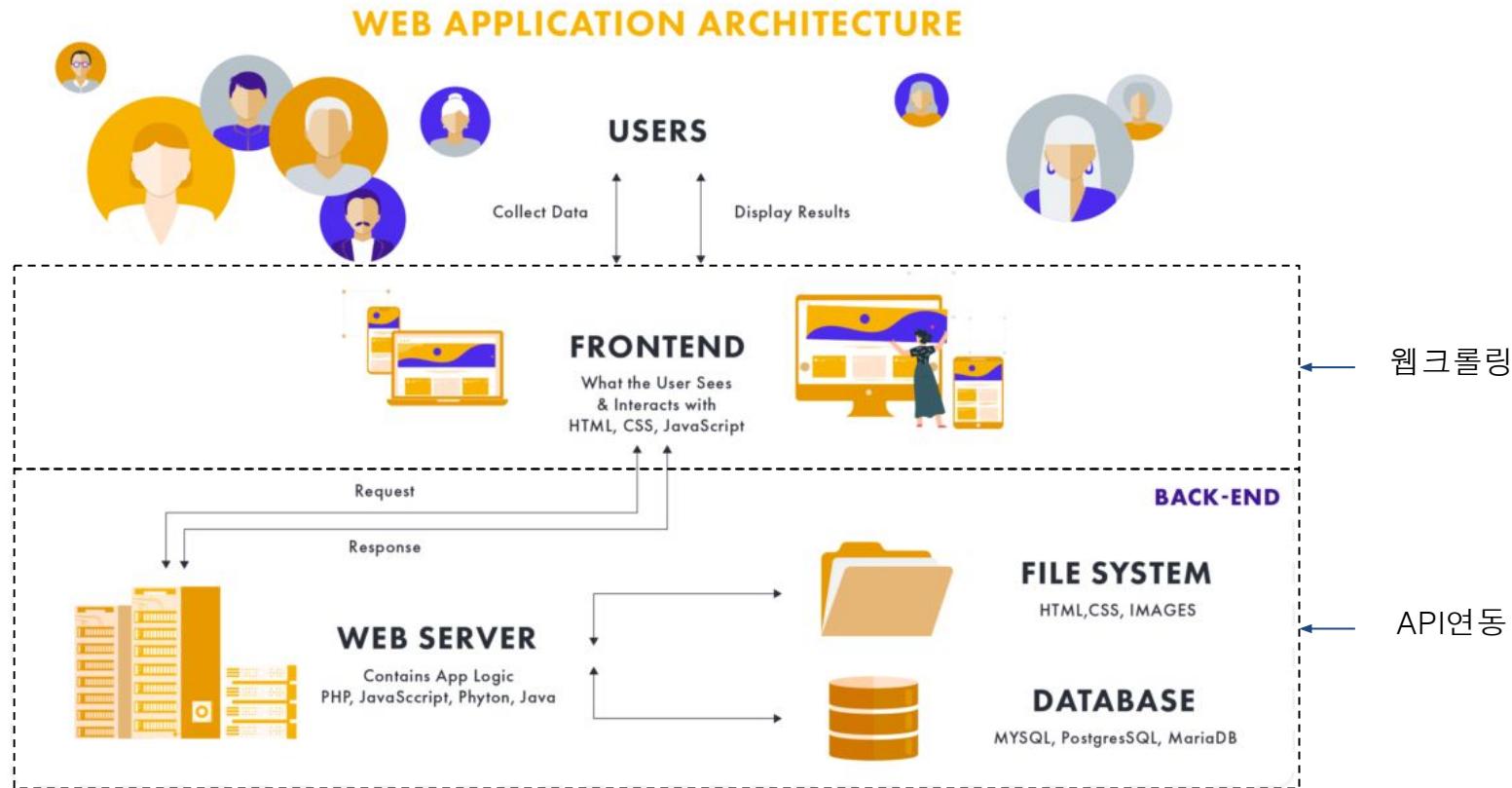
DHTML · HTML5 (오디오 · 캔버스 · 비디오) · XHTML (레이아웃 · 모바일 프로파일 · C-HTML) · HTML 요소 (span과 div) · HTML 속성 · HTML 프레임 · HTML 편집기 · 문자 인코딩 (유니코드) · 언어 코드 · 문서 객체 모델 · 브라우저 오브젝트 모델 · 스타일 시트 (CSS) · 폰트 패밀리 · 웹 색상 · 자바스크립트 (WebGL · WebGL · W3C (웹리더이터) · WHATWG · 큐크 모드 · 웹 스토리지 · 랜더링 엔진

비교

문서 마크업 언어 · HTML 지원 · XHTML (1.1)

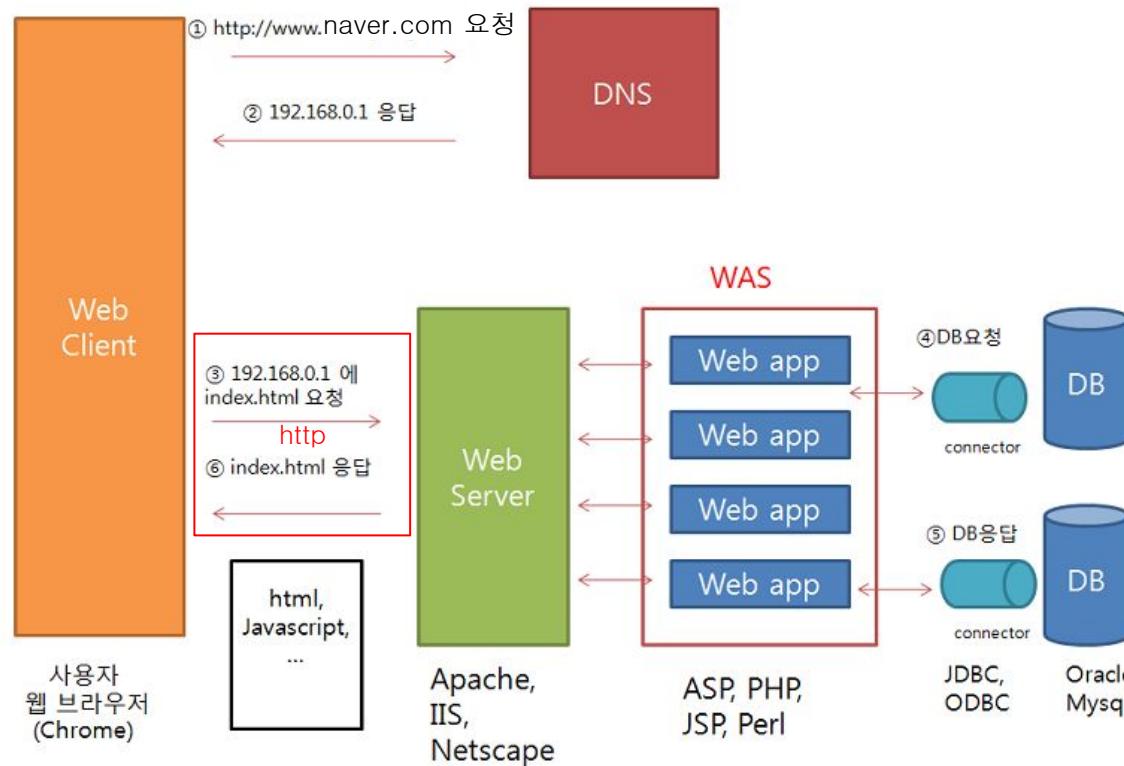
Web Application Architecture

EST



Web Service Process

EST



도메인 네임 시스템

文 A 76개 언어 ▾

문서 토론

읽기 편집 역사 보기 도구 ▾

위키백과, 우리 모두의 백과사전.

도메인 네임 시스템(Domain Name System, DNS)은 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있도록 하기 위해 개발되었다. 특정 컴퓨터(또는 네트워크로 연결된 임의의 장치)의 주소를 찾기 위해, 사람이 이해하기 쉬운 도메인 이름을 숫자로 된 식별 번호(IP 주소)로 변환해 준다. 도메인 네임 시스템은 흔히 "전화번호부"에 비유된다. 인터넷 도메인 주소 체계로서 TCP/IP의 응용에서, www.example.com과 같은 주 컴퓨터의 도메인 이름을 192.168.1.0과 같은 IP 주소로 변환하고 라우팅 정보를 제공하는 분산형 데이터베이스 시스템이다.

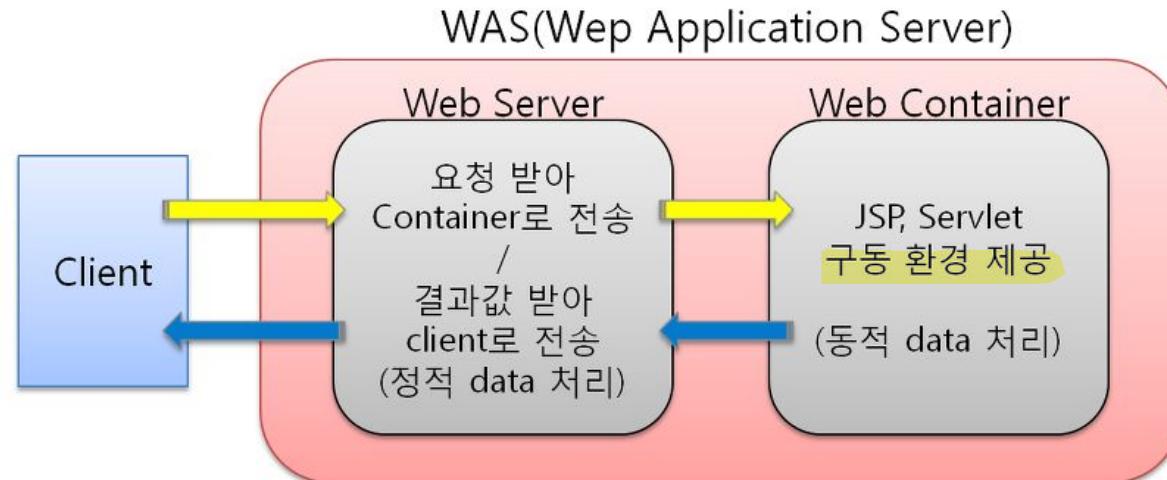
인터넷은 2개의 주요 이름공간을 관리하는데, 하나는 도메인 네임 계층^[1], 다른 하나는 인터넷 프로토콜(IP) 주소 공간이다.^[2] 도메인 네임 시스템은 도메인 네임 계층을 관리하며 해당 네임 계층과 주소 공간 간의 변환 서비스를 제공한다. 인터넷 네임 서버와 통신 프로토콜은 도메인 네임 시스템을 구현한다.^[3] DNS 네임 서버는 도메인을 위한 DNS 레코드를 저장하는 서버이다. DNS 네임 서버는 데이터베이스에 대한 쿼리의 응답 정보와 함께 응답한다.

계층별 OSI 모형

7. 응용 계층 [펼치기]
6. 표현 계층 [펼치기]
5. 세션 계층 [펼치기]
4. 전송 계층 [펼치기]
3. 네트워크 계층 [펼치기]
2. 데이터 링크 계층 [펼치기]
1. 물리 계층 [펼치기]

V • T • E

- Web Server 는 클라이언트가 웹 브라우저에게 서버에 페이지 요청을 하면 웹 서버에서 요청을 받아 정적 페이지(.html .jpeg .css 등) 을 제공하는 서버입니다.
- WAS(Web Application Server) 는 html 만으로 할 수 없는 데이터베이스 조회나 다양한 로직처리 같은 동적인 컨텐츠를 제공 하기 위해 만들어진 애플리케이션 서버입니다. WAS는 웹서버와 웹 컨테이너의 결합으로 다양한 기능을 컨테이너에 구현하여 다양한 역할을 수행할 수 있습니다. 클라이언트의 요청이 있을 때 내부의 프로그램을 통해 결과를 만들어 내고 이것을 다시 클라이언트에 전달해주는 역할을 하는 것이 바로 웹 컨테이너 입니다.



웹 크롤링의 이해

웹 크롤링 vs 파싱 vs 스크래핑

크롤링은 불법인가?

로봇 배제 표준(Robots.txt)

크롤링(crawling)

크롤링이란 단어는 웹 크롤러(crawler)라는 단어에서 시작한 말이다.

크롤러란 조직적, 자동화된 방법으로 월드와이드 웹을 탐색하는 컴퓨터 프로그램이다.(출처: 위키백과)

크롤링은 크롤러가 하는 작업을 부르는 말로, 여러 인터넷 사이트의 페이지(문서, html 등)를 수집해서 분류하는 것이다.

대체로 찾아낸 데이터를 저장한 후 쉽게 찾을 수 있게 인덱싱한다.

파싱(parsing)

파싱이란 어떤 페이지(문서, html 등)에서 내가 원하는 데이터를 특정 패턴이나 순서로 추출하여 정보를 가공하는 것이다.

위 문장만 보면 굉장히 간단해 보이지만 컴퓨터 과학적 정의를 보면 파싱이란 일련의 문자열을 의미있는 토큰(token)으로 분해

하고 이들로 이루어진 파스 트리(parse tree)를 만드는 과정을 말한다.(출처: 위키백과)

인터프리터나 컴파일러의 구성 요소 가운데 하나로, 입력 토큰에 내제된 자료 구조를 빌드하고 문법을 검사하는 역할을 한다.

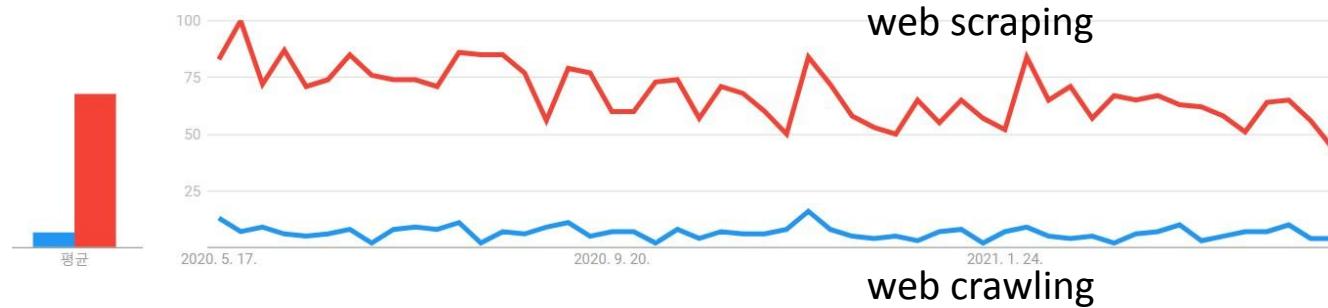
스크래핑(scraping)

스크래핑이란 HTTP를 통해 웹 사이트의 내용을 긁어다 원하는 형태로 가공하는 것이다.

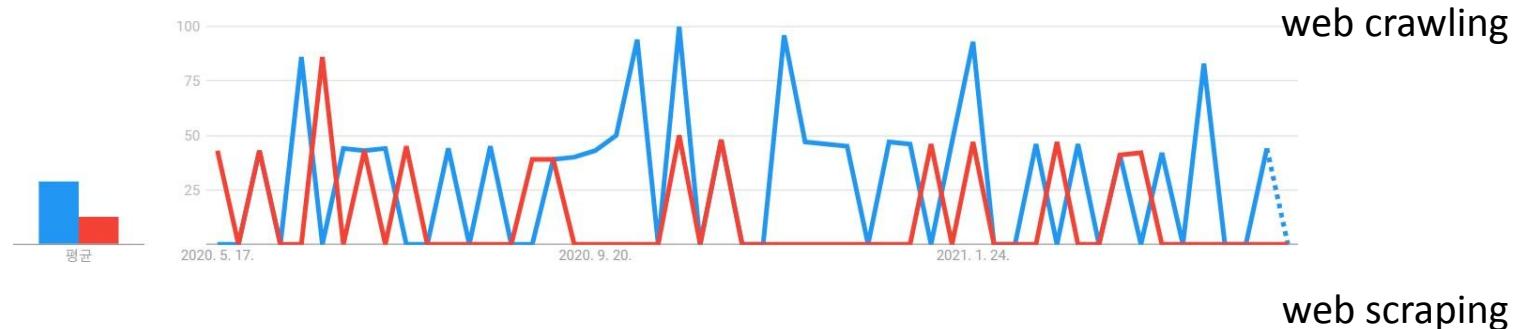
쉽게 말해 웹 사이트의 데이터를 수집하는 모든 작업을 뜻한다.

크롤링도 일종의 스크래핑 기술이라고 할 수 있다.

- Google Trend : 전세계



- Google Trend : 대한민국



크롤링은 불법인가?

EST

- 기술은 합법이나, 이용의 문제 : 칼 자체는 합법이나, 이용의 문제
- 사람인 vs. 잡코리아 판례 : 경쟁사의 정보를 자사의 이익을 위해 사용
- 관련 법률 : 저작권법, 부정경쟁방지법 등
- 저작권법 예외 사항 : 제 30조 개인소장용도(공유X), 제 94조 교육, 학술, 연구의 비상업적 목적
- 기타 예외 : 제목과 요약문. URL을 통해 원 저작권자의 사이트로 이동하는 것(bigKinds)
- 확인사항 : domain.com/robots.txt
 - <https://www.naver.com/robots.txt>, www.google.com/robots.txt
- 매너사항 : 요청과 요청 사이에 time.sleep을 이용하여 대기시간 주기(대개 6초 이상 권장)

로봇 배제 표준

文 A 26개 언어 ▾

문서 토론

읽기 편집 역사 보기 도구 ▾

위키백과, 우리 모두의 백과사전.

"Robots.txt"는 이 문서를 가리킵니다. 위키백과의 Robots.txt의 파일을 보실려면, 미디어위키:Robots.txt와 ko.wikipedia.org/robots.txt를 참조하시길 바랍니다.

로봇 배제 표준(robots exclusion standard), 로봇 배제 프로토콜(robots exclusion protocol)은 웹 사이트에 로봇이 접근하는 것을 방지하기 위한 규약으로, 일반적으로 접근 제한에 대한 설명을 robots.txt에 기술한다.

이 규약은 1994년 6월에 처음 만들어졌고, 아직 이 규약에 대한 RFC는 없다.

이 규약은 권고안이며, 로봇이 robots.txt 파일을 읽고 접근을 중지하는 것을 목적으로 한다. 따라서, 접근 방지 설정을 하였다고 해도, 다른 사람들이 그 파일에 접근할 수 있다. robots.txt 파일은 항상 사이트의 루트 디렉토리에 위치해야 한다.^[1]

(예시) 모든 로봇에게
문서 접근 허락

```
User-agent: *\nAllow: /
```

(예시) 모든 로봇 차단

```
User-agent: *\nDisallow: /
```

(예시) 모든 로봇에 특정
디렉토리 접근 차단

```
User-agent: *\nDisallow: /cgi-bin/\nDisallow: /tmp/\nDisallow: /junk/
```

다양하게 조합하여 사용

User-agent: googlebot\nDisallow: /private/	# googlebot 로봇만 적용 # 이 디렉토리를 접근 차단한다.
User-agent: googlebot-news\nDisallow: /	# googlebot-news 로봇만 적용 # 모든 디렉토리를 접근 차단한다.
User-agent: *\nDisallow: /something/	# 모든 로봇 적용 # 이 디렉토리를 접근 차단한다.

HTML의 이해

HTML과 HTML5, MarKUp

HTML 기본 구조, 요소 구조

HTML List, Table

BlocK과 Inline

CSS와 Javascript

Web Site Example

EST

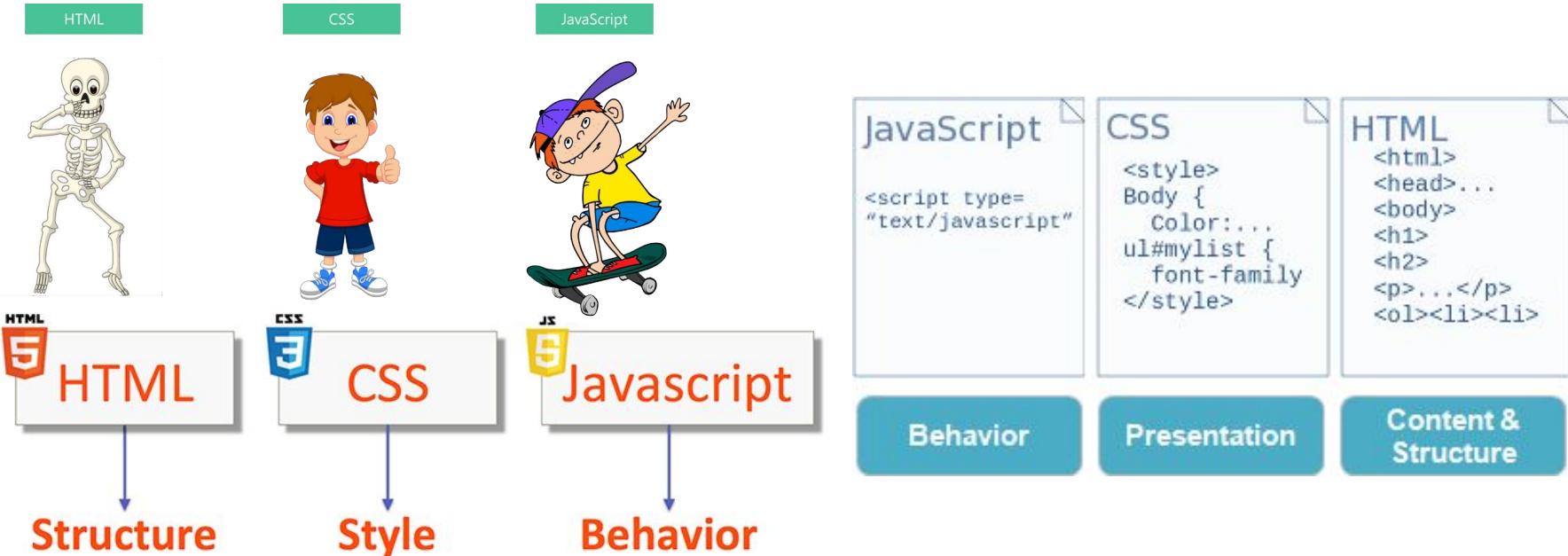
웹 브라우저 개발자 도구(F12)

The screenshot shows a news website with a banner at the top advertising Starbucks Americano coffee. The banner text reads: "만기일 상관없이 내 차 보험료 확인하면 스타벅스 아메리카노 2잔". Below the banner, there's a navigation bar with links like 메일, 카페, 블로그, 쇼핑, 뉴스, 증권, 부동산, and 지도. The main content area features a headline about a news event and a list of news sources at the bottom.

The screenshot shows the Chrome DevTools Element tab with the element tree open. The selected element is a banner ad with the ID 'ad_timeboard_tgtREC'. The right panel displays the Styles tab, showing the CSS rules applied to the element, including its width (830px), aspect ratio (auto 830 / 130), and height (130px). A detailed breakdown of the element's bounding box is shown in the bottom right corner, illustrating the margin, border, padding, and width properties.

HTML과 CSS, Javascript

EST



HTML

문서 토론

위키백과, 우리 모두의 백과사전.

하이퍼 텍스트 마크업 언어(영어: Hyper Text Markup Language, HTML, 문화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지 표시를 위해 개발된 지배적인 마크업 언어다. 또한, HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것 뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하여 대화형 양식을 생성하는 데 사용될 수 있다. HTML은 웹 페이지 콘텐츠 안의 꺼ళ 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성 한다. HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트, 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다. HTML과 CSS 표준의 공동 책임자인 W3C는 명확하고 표상적인 마크업을 위하여 CSS의 사용을 권장한다.^[1]

HTML5

문서 토론

위키백과, 우리 모두의 백과사전.



HTML5는 HTML의 완전한 5번째 버전으로 월드 와이드 웹 (World Wide Web)의 핵심 마크업 언어이다. 2004년 7월 Web Hypertext Application Technology Working Group(WWHATWG)에서 웹 애플리케이션 1.0이라는 이름으로 세부 명세 작업을 시작하였다.

HTML5는 HTML 4.01, XHTML 1.0, DOM 레벨 2 HTML에 대한 차기 표준 제안이다. 비디오, 오디오 등 다양한 부가기능과 최신 멀티미디어 콘텐츠를 액티브X 없이 브라우저에서 쉽게 볼 수 있게 하는 것을 목적으로 한다.

W3C는 2014년 10월 28일 HTML5 표준안을 확정했다고 발표했다.

이후 2016년 11월 1일 HTML5의 마이너 업데이트 버전인 HTML5.1 표준안을 확정, 2017년 12월 14일 HTML5.2 표준안을 확정했다. HTML5.3 표준안은 현재 작업 초안 단계로 진행 중이다.

마크업 [편집]

HTML 마크업은 HTML 요소(엘리먼트, Elements)와 그들의 속성(Attribute)과 문자 기반 데이터 형태와 문자 참조와 엔티티 참조를 포함하는 몇 가지 핵심 구성 요소로 이루어져 있다. 또 다른 중요한 구성 요소로는 문서 형식 정의(DTD, Document Type Definition)를 명시하는 문서 형식 선언(document type declaration)이다. 차기 HTML 5에서는 DTD를 지정하지 않아도 되고 오직 레이아웃 모드로 지정된다 [4] 2.

Hello world 프로그램은 프로그래밍 언어와 스크립트 언어 그리고 마크업 언어를 비교하기 위해 사용되는 일반적인 컴퓨터 프로그램이다. 그리고 HTML에서의 Hello world 프로그램은 단 9줄에 불과하다:

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2  <html>
3    <head>
4      <title>Example</title>
5      <link href="screen.css" rel="sty
6    </head>
7    <body>
8      <h1>
9        <a href="/">Header</a>
10     </h1>
11     <ul id="nav">
12       <li>
13         <a href="one/">One</a>
14       </li>
15       <li>
16         <a href="two/">Two</a>
17       </li>
```

HTML 4 구문 강조

<!DOCTYPE>

<!DOCTYPE> : 현재 문서가 HTML 문서 타입을 명시한다. (HTML5 문서 타입은 <!DOCTYPE html> 이다.)

<html>

<html> : HTML 문서의 루트(root) 요소를 정의한다.

<head>

<head> : HTML 문서의 메타데이터(metadata)를 정의한다.

<title>페이지 타이틀</title>

- 메타데이터(metadata)란 HTML 문서에 대한 정보(data)로 웹 브라우저에는 직접적으로 표현되지 않는 정보를 의미한다.
- 이러한 메타데이터는 <title>, <style>, <meta>, <link>, <script>, <base> 태그 등을 이용하여 표현할 수 있다.

</head>

<title> : HTML 문서의 제목(title)을 정의하며, 다음과 같은 용도로 사용된다.

<h1>여기는 제목입니다.</h1>

- 웹 브라우저의 툴바(toolbar)에 표시된다.
- 웹 브라우저의 즐겨찾기(favorites)에 추가할 때 즐겨찾기의 제목이 된다.
- 검색 엔진의 결과 페이지에 제목으로 표시된다.

<p>여기는 문장입니다.</p>

<body> : 웹 브라우저를 통해 보이는 내용(content) 부분이다.

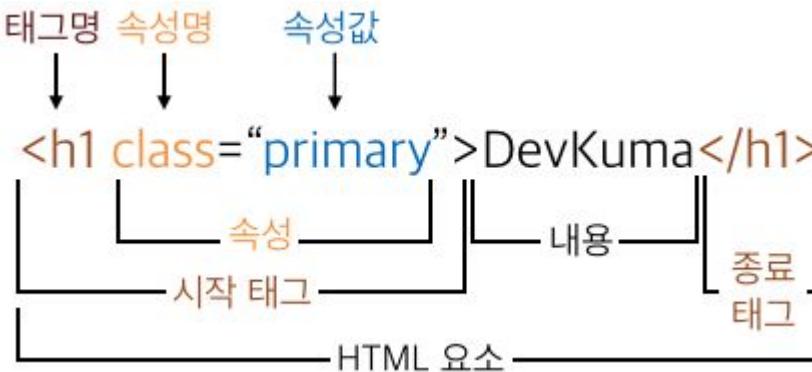
</body>

<h1>~<h6> : 제목(heading)을 나타낸다.

</html>

<p> : 단락(paragraph)을 나타낸다.

- HTML 요소(element)는 여러 속성을 가질 수 있으며, 속성(attribute)은 해당 요소에 대한 추가적인 정보이다.
- HTML 요소는 시작 태그로 시작해서 종료 태그로 끝난다. <태그명 속성명 = “속성값”> 내용 </태그명>
- class : 1페이지에 여러 번 사용 > 수집시 결과가 1개이상 존재 > 목록(list)형 결과
- id : 1페이지에 1번만 사용(Unique) > 수집시 결과가 1개만 존재



요소(Element)

HTML에서 시작 태그와 종료 태그로 이루어진 모든 명령어들을 의미한다.

태그(Tag)

요소(Elements)의 일부로 시작 태그와 종료 태그 두 종류가 있다.
시작 태그는 요소를 시작하며, 종료태그는 요소를 끝내는 기능을 가지고 있다. 일부 태그 중에는 종료 태그가 없는 것도 있다.

요소와 태그의 개념이 뚜렷이 구별되지 않고 혼용되는 경우가 많은데 HTML에서는 굳이 요소라는 말을 쓰지 않아도 되지만 CSS나 Javascript에서는 요소라는 말이 아주 중요한 용어가 된다.

속성(Attribute)

요소의 시작 태그 안에서 사용되는 것으로 좀 더 구체화된 명령어 체계를 의미한다.
속성은 HTML 요소 중에서도 언제나 시작 태그 내에서만 정의되며, 속성 이름과 속성 값(value)으로 표현된다.

리스트(list)란 여러 요소들을 일렬로 나열한 목록이나 명단을 의미한다.

HTML에서는 여러 요소들을 일렬로 나열한 목록이나 명단을 표현 할 수 있는 태그를 제공하고 있다.

1. 순서가 있는 목록(ordered list) : code → display

```
<ol>
  <li>HTML</li>
  <li>Java</li>
  <li>C++</li>
</ol>
```



1. HTML
2. Java
3. C++

2. 순서가 없는 목록(unordered list)

```
<ul>
  <li>HTML</li>
  <li>Java</li>
  <li>C++</li>
</ul>
```



- HTML
- Java
- C++

테이블(table)이란 여러 종류의 데이터(data)를 보기 좋게 정리하여 보여주는 표를 의미한다.

HTML에서는 `<table>` 태그를 사용하여 이러한 테이블을 작성할 수 있다.

`<table>` 태그는 다음과 같은 태그들로 구성된다.

1. `<tr>` 태그는 테이블에서 열을 구분해 준다.
2. `<th>` 태그는 각 열의 제목을 나타내며, 모든 내용은 자동으로 굵은 글씨에 가운데 정렬이 된다.
3. `<td>` 태그는 테이블의 열을 각각의 셀(cell)로 나누어 준다.

```
<table>
  <tr>
    <th>분류</th>
    <th>항목</th>
  </tr>
  <tr>
    <td>과일</td>
    <td>사과</td>
  </tr>
  <tr>
    <td>채소</td>
    <td>당근</td>
  </tr>
</table>
```

분류	항목
과일	사과
채소	당근

HTML 블럭(block)과 인라인.inline)

HTML의 모든 요소는 해당 요소가 웹 브라우저에 어떻게 보이는가를 결정짓는 `display` 속성을 가진다.

대부분의 HTML 요소는 기본적으로 `display` 속성값으로 다음 두 가지 값 중 하나를 가지게 된다.

블록(block) 속성을 갖는 요소

`display` 속성값이 블록(block)인 요소는 언제나 새로운 줄(line)로 바뀌며, 해당 줄의 모든 너비를 차지한다.

`display` 속성값이 블록(block)인 대표적인 요소는 아래와 같다.

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- ``
- ``
- `<form>`

블록 속성의 특징은 아래와 같다.

- 블록 속성을 가지고 있는 태그는 기본적으로 너비 100%(`width: 100%`) 속성을 가지고 있다. 화면의 가로 폭을 100%로 완전히 차지하기 때문에, 다음 요소가 양 옆으로 붙을 공간이 없어서 자연히 줄 넘김이 되는 것이다.
- 또한, 인라인 요소와 다르게 `margin`, `width`, `height` 속성을 정의하면 모두 적용된다. 모양새를 쉽게 제어할 수 있는 속성 때문에 대부분 블록 속성을 가진 태그를 화면 구성이나 레이아웃에 사용한다.

블록 요소 (block elements)

block
block
block
block

```
<div style="background-color: lightgrey; color:green;">  
  <h1>오늘의 명언</h1>  
  <p>오늘 내가 죽어도 세상은 바뀌지 않는다. 하지만 내가 살아 있는 한 세상은 바뀐다.</p>  
</div>
```

오늘의 명언

오늘 내가 죽어도 세상은 바뀌지 않는다. 하지만 내가 살아 있는 한 세상은 바뀐다.

인라인(inline) 속성을 갖는 요소

`display` 속성값이 인라인(inline)인 요소는 새로운 줄(line)로 바꾸지 않고 다른 요소와 같이 표시된다.

또한, 요소의 너비도 해당 라인 전체가 아닌 해당 HTML 요소의 내용(content)만큼만 차지한다.

`display` 속성값이 인라인(inline)인 대표적인 요소이다.

- ``
- `<a>`
- ``
- ``

인라인 속성의 특징은 아래와 같다.

- 상, 하단 외부 여백(`margin-top`, `margin-bottom`) 속성을 정의해도 적용되지 않는다. 인라인 요소의 상, 하 여백은 `margin` 속성이 아니라 `line-height` 속성에 의해 발생한다.
- 너비(`width`)와 높이(`height`) 속성이 적용되지 않는다. 인라인 요소의 너비와 높이는 태그가 품고 있는 내부 요소의 부피에 맞춰진다.
- 인라인 속성을 가진 태그끼리 연속으로 사용되는 경우에는 최소한의 간격을 유지하기 위해서 좌, 우에 약 5px 가량의 외부 여백(`margin`)이 자동으로 발생한다.

인라인 요소 (inline elements)

inline inline inline inline inline

나는 당신을 `사랑`합니다.

나는 당신을 사랑합니다.

CSS

문서 토론

위키백과, 우리 모두의 백과사전.

 다른 뜻에 대해서는 CSS (동음이의) 문서를 참고하십시오.

종속형 시트 또는 캐스케이딩 스타일 시트(영어: Cascading Style Sheet)는 마크업 언어가 실제 표시되는 방법을 기술하는 스타일 언어(영어: Style sheet language 스타일 시트 랭귀지^[1])로^[1], HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이고, 레이아웃과 스타일을 정의할 때의 자유도가 높다. 기본 파일명^[2]은 style.css이다.

마크업 언어(ex: HTML)가 웹사이트의 몸체를 담당한다면 CSS는 옷과 액세서리처럼 꾸미는 역할을 담당한다고 할 수 있다. 즉, HTML 구조는 그대로 두고 CSS 파일만 변경해도 전혀 다른 웹사이트처럼 꾸밀 수 있다.

현재 개발 중인 CSS3의 경우 그림자 효과, 그라데이션, 변형 등 그래픽 편집 프로그램으로 제작한 이미지를 대체할 수 있는 기능이 추가되었다. 또한 다양한 애니메이션 기능이 추가되어 어도비 플래시를 어느 정도 대체하고 있다.

```
/* A reference to a type */
span.ts span.type-ref {
    color: #rgb(175, 0, 219) !important;
}

/* Signature details */
div.signature > table {
    border-collapse: collapse;
    border: thin #darkgray solid;
    width: 60%;
}
```

자바스크립트란?

자바스크립트(JavaScript)는 객체(object) 기반의 스크립트 언어이다.

HTML로는 웹의 내용을 작성하고, CSS로는 웹을 디자인하며, 자바스크립트로는 웹의 동작을 구현할 수 있다.

자바스크립트는 주로 웹 브라우저에서 사용되나, Node.js와 같은 프레임워크를 사용하면 서버 측 프로그래밍에서도 사용할 수 있다.

컴퓨터나 스마트폰 등에 포함된 대부분의 웹 브라우저에는 자바스크립트 인터프리터가 내장되어 있다.

```
15
16 const LOCALE = globalThis.navigator.language
17
18 const div = document.body.appendChild(document.createElement('div'))
19 const list = div.appendChild(document.createElement('ol'))
20
21 const dayNames = new Map()
22
23 for (let i = 0; i < 7; ++i) {
24   const d = Temporal.PlainDate.from({
25     year: Temporal.Now.plainDateISO().year,
26     month: 1,
27     day: i + 1,
28   })
29
30   dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
31 }
32
33 for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
34   list.appendChild(Object.assign(
35     document.createElement('li'),
36     { textContent: dayNames.get(num) },
37   ))
38 }
39
```

XPath(XML Path Language)는 W3C의 표준으로 확장 생성 언어 문서의 구조를 통해 경로 위에 지정한 구문을 사용하여 항목을 배치하고 처리하는 방법을 기술하는 언어이다. XML 표현보다 더 쉽고 약어로 되어 있으며, XSL 변환(XSLT)과 XML 지시자 언어(XPointer)에 쓰이는 언어이다. XPath는 XML 문서의 노드를 정의하기 위하여 경로식을 사용하며, 수학 함수와 기타 확장 가능한 표현들이 있다.

아래의 XPath 식은

/wiKimedia/projects/project/@name

모든 project 요소의 name 속성을 선택하고, 아래의 XPath 식은

/wiKimedia/projects/project/editions/edition[@language="English"]/text()

모든 영문 WiKimedia 프로젝트의 주소(language 속성이 English인 모든 edition 요소의 문자열)를 선택하고, 아래의 XPath 식은

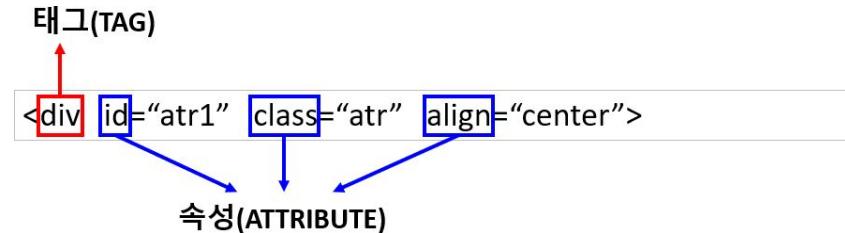
/wiKimedia/projects/project[@name="WiKipedia"]/editions/edition/text()

모든 위키백과의 주소(WiKipedia의 이름 특성을 가진 project 요소 아래에 존재하는 모든 edition 요소의 문자열)를 선택한다.

XML 예제 문서

```
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
  <projects>
    <project name="Wikimedia" launch="2001-01-05">
      <editions>
        <edition language="English">en.wikipedia.org</edition>
        <edition language="German">de.wikipedia.org</edition>
        <edition language="French">fr.wikipedia.org</edition>
        <edition language="Polish">pl.wikipedia.org</edition>
      </editions>
    </project>
    <project name="Wiktionary" launch="2002-12-12">
      <editions>
        <edition language="English">en.wiktionary.org</edition>
        <edition language="French">fr.wiktionary.org</edition>
        <edition language="Vietnamese">vi.wiktionary.org</edition>
        <edition language="Turkish">tr.wiktionary.org</edition>
      </editions>
    </project>
  </projects>
</wikimedia>
```

//div[@class="hdline_article_tit"]/a/text()



- // : 문서 전체에서 찾기
- / : 아래 단계로 내려가기
- .// : 문서 부분의 전체에서 찾기
- div : Tag(div) and Attribute(전체)
- div[@속성=“속성값”] : Tag(div) and Attribute(속성=“속성값”)
- *[@속성=“속성값”] : Tag(전체) and Attribute(속성=“속성값”)
- a/text() : Tag a 밑에 있는 텍스트
- a/@href : Tag a에 있는 Attribute가 href인 속성값
- a/descendant-or-self::text() : Tag a 밑에 있는 모든 텍스트
- a/descendant-or-self::text()[not(ancestor::script)] : Tag a 밑에 있는 모든 텍스트 중 Tag script 제외
- a/descendant-or-self::text()[not(ancestor::script or ancestor::style)] :
Tag a 밑에 있는 모든 텍스트 중 Tag script와 style 제외

DOM객체와 BeautifulSoup

웹 스크래핑 프로세스

HTML 구문 분석(parser)의 필요성

문서 객체 모델(DOM)

HTML Parser 라이브러리(BS4)

1. 웹 접속 및 HTML코드 (str) 추출

- a. request library : 정적페이지(html구조)에 활용
- b. selenium library : 동적페이지(javascript 구조)에 활용

2. 계층 구조(DOM)로 변환 : DOM이란? ?

- a. BS4 library : beautifulsoup(소스코드, 파서-해석기)

3. 검색후 추출

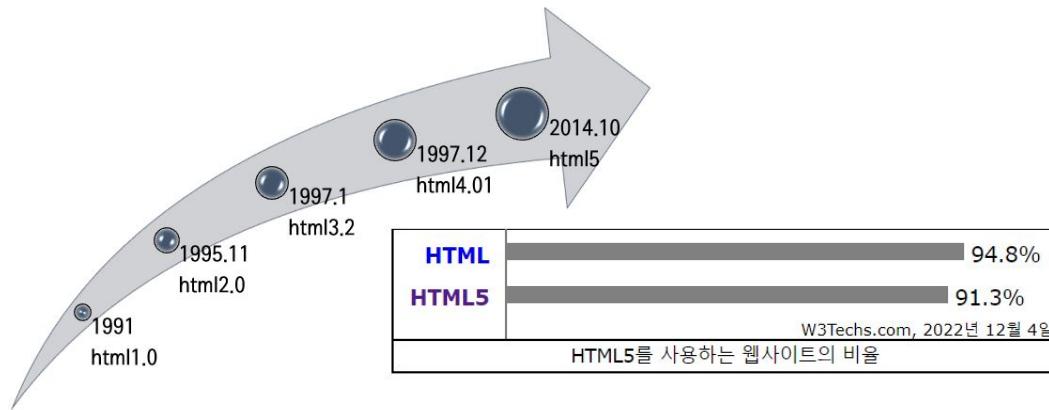
- a. find와 select의 차이, select는 css 구분자 사용, 관계연결가능
- b. find와 select와 조건 방식이 다름
- c. soup.find / .find_all('태그', id='orange', class_='fruit') class는 예약어이기 때문
- d. soup.select /select_one ('태그', '#orange', '.fruit')
- e. find 와 select_one은 단일 태그값 만 반환
- f. find_all과 select는 모든 tag를 list로 반환

4. 가공

HTML 구문 분석(parser)의 필요성

EST

- 전세계의 HTML 도큐먼트들은 생각보다 정확하지 않다.
 - HTML5 웹 표준이 자리 잡기 시작한 것은 얼마되지 않았다.
 - 아주 오래된 HTML 버전의 웹페이지들과 문법 오류의 존재
 - 다양한 디바이스 환경과 운영체제의 등장
 - 다양한 브라우저 랜더링 엔진
- 이에 DOM구조를 바탕으로 도큐먼트를 이해하고 데이터를 추출할 수 있는 구문 분석기가 필요하다.

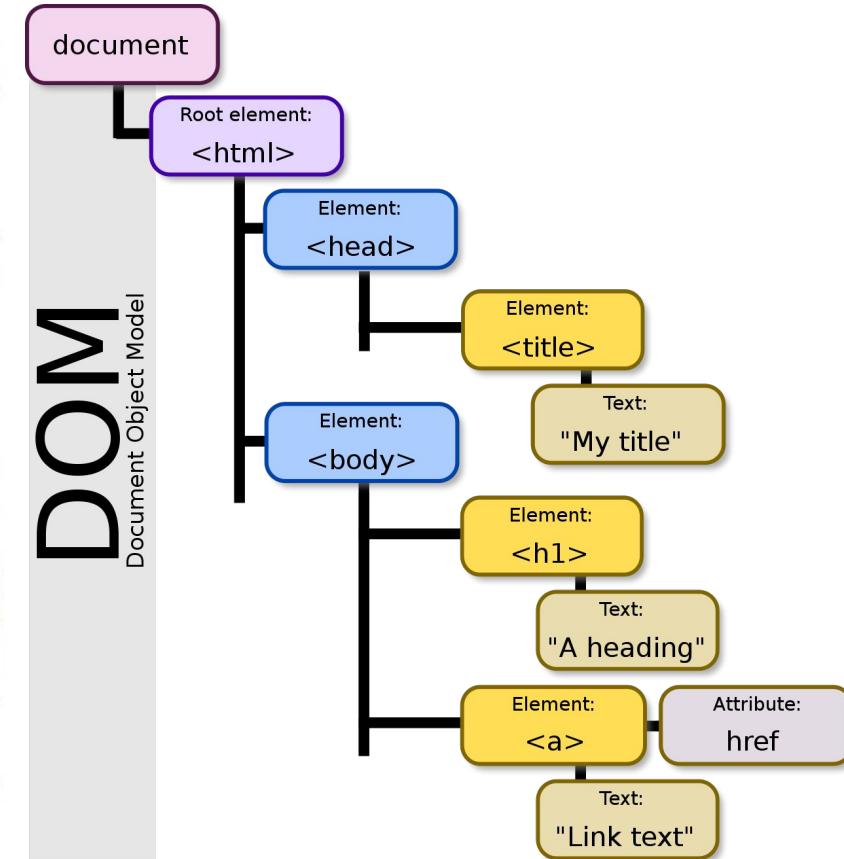


문서 객체 모델(DOM)

문서 객체 모델(영어: Document Object Model 도큐먼트 오브젝트 모델 [1], DOM)은 XML, HTML 문서의 각 항목을 계층으로 표현하여 생성, 변형, 삭제할 수 있도록 드는 인터페이스이다. W3C의 표준이다. W3C의 표준화한 API들의 기반이 된다.

DOM은 HTML 문서의 요소를 제어하기 위해 웹 브라우저에서 처음 지원되었다. DOM은 동적으로 문서의 내용, 구조, 스타일에 접근하고 변경하는 수단이었다. 브라우저 사이에 DOM 구현이 호환되지 않음에 따라, W3C에서 DOM 표준 규격을 작성하게 되었다.

DOM은 문서의 기반이 되는 데이터 구조에 제한을 두지 않는다. 잘 구조화된 문서는 DOM을 사용하여 트리 구조를 얻어낼 수 있다. 대부분의 XML 해석기와 XSL 처리기는 트리 구조의 이용에 대응해 개발되었다. 이 같은 구현에서는 문서의 전체 내용이 해석되어 메모리 저장되어야 한다. 때문에 DOM은 문서 요소가 임의적으로 접근되고 변경할 수 있어야 하는 응용 프로그램에 가장 적합하다. 한 번 해석 시 단 한 번의 선택적 읽기/쓰기가 이루어지는 XML 기반 응용 프로그램에서, DOM은 메모리에 상당한 부하를 가져온다. 이 경우처럼 속도와 효율적인 메모리 소비가 중요한 상황일 경우 SAX 모델이 장점을 가진다.



<https://www.crummy.com/software/BeautifulSoup/bs4/doc.Ko/>

뷰티풀수프 문서

한글판 johnsonj 2012.11.08 [원문](#) [위치](#)

뷰티풀수프는 HTML과 XML 파일로부터 데이터를 뽑아내기 위한 파이썬 라이브러리이다. 여러분이 선호하는 해석기와 함께 사용하여 일반적인 방식으로 해석 트리를 향해, 검색, 변경할 수 있다. 주로 프로그래머의 수고를 덜어준다.

이 지도서에서는 뷰티풀수프 4의 중요한 특징들을 예제와 함께 모두 보여준다. 이 라이브러리가 어느 곳에 유용한지, 어떻게 작동하는지, 또 어떻게 사용하는지, 어떻게 원하는대로 바꿀 수 있는지, 예상을 빗나갔을 때 어떻게 해야 하는지를 보여준다.

이 문서의 예제들은 파이썬 2.7과 Python 3.2에서 똑같이 작동한다.

혹시 [뷰티풀수프 3](#)에 관한 문서를 찾고 계신다면 뷰티풀수프 3는 더 이상 개발되지 않는다는 사실을 꼭 아셔야겠다. 새로 프로젝트를 시작한다면 뷰티풀수프 4를 적극 추천한다. 뷰티풀수프 3와 뷰티풀수프 4의 차이점은 [BS4 코드 이식하기](#)를 참조하자.



HTML Parser Library(BS4)

EST

뷰티풀수프는 파이썬 표준 라이브러리에 포함된 HTML 해석기를 지원하지만, 또 수 많은 3rd party 파이썬 해석기도 지원한다.

가능하다면, 속도를 위해 lxml을 설치해 사용하시기를 권장한다. 2.7.3 이전의 파이썬2, 또는 3.2.2 이전의 파이썬 3 버전을 사용한다면, lxml을 사용하는 것이 필수이다. 그렇지 않고 구형 버전의 파이썬 내장 HTML 해석기 html5lib는 별로 좋지 않다.

문서가 유효하지 않을 경우 해석기마다 다른 뷰티풀수프 트리를 생산한다는 사실을 주목하자.

자세한 것은 [해석기들 사이의 차이점들을 살펴보자](#).

해석기	전형적 사용방법	장점	단점
파이썬의 html.parser	<code>BeautifulSoup(markup, "html.parser")</code>	<ul style="list-style-type: none">각종 기능 완비적절한 속도관대함 (파이썬 2.7.3과 3.2에서.)	<ul style="list-style-type: none">별로 관대하지 않음 (파이썬 2.7.3이나 3.2.2 이전 버전에서)
lxml의 HTML 해석기	<code>BeautifulSoup(markup, "lxml")</code>	<ul style="list-style-type: none">아주 빠름관대함	<ul style="list-style-type: none">외부 C 라이브러리 의존
lxml의 XML 해석기	<code>BeautifulSoup(markup, ["lxml", "xml"])</code> <code>BeautifulSoup(markup, "xml")</code>	<ul style="list-style-type: none">아주 빠름유일하게 XML 해석기 지원	<ul style="list-style-type: none">외부 C 라이브러리 의존
html5lib	<code>BeautifulSoup(markup, html5lib)</code>	<ul style="list-style-type: none">아주 관대함웹 브라우저의 방식으로 페이지를 해석함유효한 HTML5를 생성함	<ul style="list-style-type: none">아주 느림외부 파이썬 라이브러리 의존파이썬 2 전용

1 BeautifulSoup

```
pip install bs4
```

BeautifulSoup <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

1.1 Parser Library

```
pip install lxml
```

```
pip install html5lib
```

```
from bs4 import BeautifulSoup
import lxml
```

아나콘다 사용자는 기본 라이브러리로 내장됨(설치 불필요)

```
# sample
html = '''
<div id=prices >
  <table >
    <tr class=bb>
      <th class="bb lm lft">Date
      <th class="rgt bb">Open
      <th class="rgt bb">High
      <th class="rgt bb">Low
      <th class="rgt bb">Close
      <th class="rgt bb rm">Volume
    <tr>
      <td class="lm">Feb 28, 2014
      <td class="rgt">100.71
      <td class="rgt">100.71
      <td class="rgt">100.71
      <td class="rgt">100.71
      <td class="rgt rm">0
    </table>
    ...
```

무엇을 써야 될까? 정답은 없다. 결과를

비교해보자

```
#xml의 HTML파서  
BeautifulSoup(html, 'xml')
```

executed in 15ms, finished 22:10:44 2023-10-15

```
<html><body><div id="prices">  
  <table>  
    <tr class="bb">  
      <th class="bb lm lft">Date  
        </th><th class="rgt bb">Open  
        </th><th class="rgt bb">High  
        </th><th class="rgt bb">Low  
        </th><th class="rgt bb">Close  
        </th><th class="rgt bb rm">Volume  
      </th></tr><tr>  
    <td class="lm">Feb 28, 2014  
      </td><td class="rgt">100.71  
      </td><td class="rgt">100.71  
      </td><td class="rgt">100.71  
      </td><td class="rgt">100.71  
      </td><td class="rgt rm">0  
    </td></tr></table>  
</div></body></html>
```

```
#파이썬 표준 html파서  
BeautifulSoup(html, 'html.parser')
```

```
<div id="prices">  
  <table>  
    <tr class="bb">  
      <th class="bb lm lft">Date  
        <th class="rgt bb">Open  
        <th class="rgt bb">High  
        <th class="rgt bb">Low  
        <th class="rgt bb">Close  
        <th class="rgt bb rm">Volume  
      </th>  
    <td class="lm">Feb 28, 2014  
      <td class="rgt">100.71  
      <td class="rgt">100.71  
      <td class="rgt">100.71  
      <td class="rgt">100.71  
      <td class="rgt rm">0  
    </td></td></td></td></td></tr></table>  
</div>
```

1.3.1 메서드를 사용한 요소 탐색

1.3.1.1 find

```
| soup.find/find_all(태그, id=orange, class_= 'fruit')  
| # class는 예약어이기 때문에 class_으로 매개변수명을 구  
| 분함
```

- `soup.find(name = "요소명")` : 첫번째 요소 찾기
- `soup.find(attrs = {"속성": "값"})`
- `soup.find(속성 = "값")` # 키워드 가변 인수
- `soup.find(string = "텍스트")`
- `soup.find(recursive = "True")` # 기본값, 후손 요소 전체에서 검색
- `soup.find(reculsive = "False")` # 직계 자식 내에서만 검색
- `soup.find_all()` : 모든 요소 찾기
 - `name, attrs, string, **kwargs, recursive, limit(final_all에서만 가능, limit=2는 2개만 찾기)`

1.3.1.2 select

- `find`와 `select`의 차이, `select`는 CSS선택자 사용, 관계 연결 가능
- `find`와 `select`와 조건 방식이 다름
- `find` 와 `select_one`은 단일 태그값 만 반환
- `find_all`과 `select`는 모든 tag를 list로 반환

```
| soup.select/select_one(태그, '#orange', '.fruit')
```

- `soup.select()` : 모든 요소 찾기 --> CSS 선택자 * `soup.select_one()` : 첫번째
요소 찾기

1.3.2 요소의 속성 및 텍스트 반환

`e.name` : 요소의 이름 얻기

`e['속성명']` : 요소의 속성 얻기

`e.attrs` : 요소의 속성 목록(dict)

`e.string` : 요소의 텍스트

`e.strings` : 자식과 후손 요소 텍스트 목록(iter)

`e.text` : 자식과 후손 요소의 텍스트를 문자열로 얻기

1.3.3 트리 구조를 활용한 요소 탐색

- `e.요소명` : e 하위의 첫번째 요소
- `e.parent` : 부모 요소
- `e.parents` : 모든 조상 요소(iter)
- `e.contents` : 모든 자식 요소(list)
- `e.children` : 모든 자식 요소(iter)
- `e.descendants` : 모든 후손 요소(iter)
- `e.previous_sibling` : 바로 앞의 형제 요소
- `e.previous_siblings` : 모든 앞의 형제 요소(iter)
- `e.next_sibling` : 바로 뒤의 형제 요소
- `e.next_siblings` : 모든 뒤의 형제 요소(iter)
- `e.previous_element` : 바로 앞의 요소
- `e.previous_elements` : 모든 앞에 있는 요소(iter)
- `e.next_element` : 바로 뒤의 요소
- `e.next_elements` : 모든 뒤에 있는 요소(iter)

정적 웹페이지(Static)와 동적 웹페이지(Dynamic)

http와 https

request와 response

get과 post

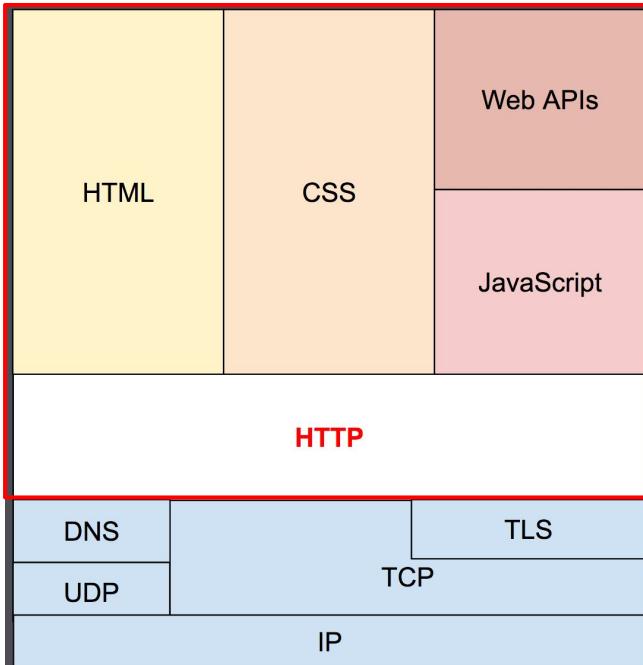
requests라이브러리

selenium 라이브러리

HTTP와 HTTPS

EST

Web



HTTP(HyperText Transfer Protocol, 문화어: 초본문전송규약, 하이퍼본문전송규약)는 W3 상에서 정보를 주고받을 수 있는 프로토콜이다. 주로 HTML 문서를 주고받는 데에 쓰인다. 주로 TCP를 사용하고 HTTP/3부터는 UDP를 사용하며, 80번 포트를 사용한다. 1996년 버전 1.0, 그리고 1999년 1.1이 각각 발표되었다.

HTTP는 클라이언트와 서버 사이에 이루어지는 요청/응답(request/response) 프로토콜이다. 예를 들면, 클라이언트인 웹 브라우저가 HTTP를 통하여 서버로부터 웹페이지(HTML)나 그림 정보를 요청하면, 서버는 이 요청에 응답하여 필요한 정보를 해당 사용자에게 전달하게 된다. 이 정보가 모니터와 같은 출력 장치를 통해 사용자에게 나타나는 것이다.

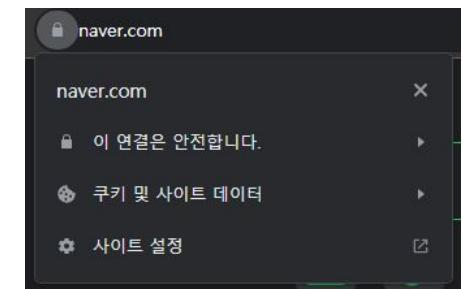
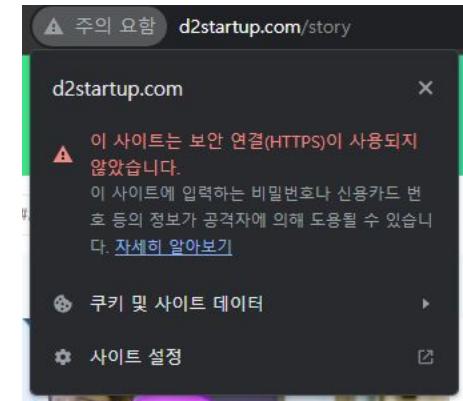
HTTP를 통해 전달되는 자료는 http://로 시작하는 URL(인터넷 주소)로 조회할 수 있다.

HTTPS(HyperText Transfer Protocol over Secure Socket Layer, HTTP over TLS,^{[1][2]} HTTP over SSL,^[3] HTTP Secure^{[4][5]})는 월드 와이드 웹통신 프로토콜인 HTTP의 보안이 강화된 버전이다. HTTPS는 통신의 인증과 암호화를 위해 넷스케이프 커뮤니케이션즈 코퍼레이션이 개발한 넷스케이프 웹 프로토콜이며, 전자 상거래에서 널리 쓰인다.

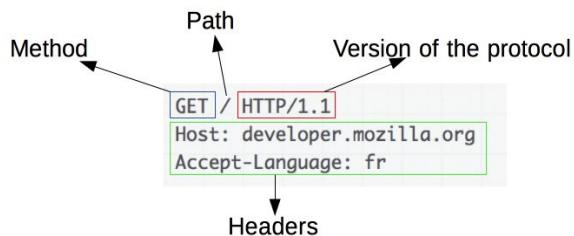
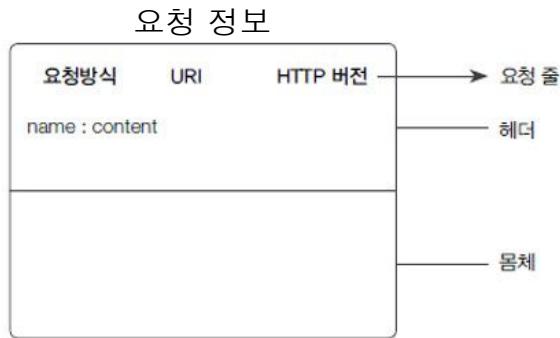
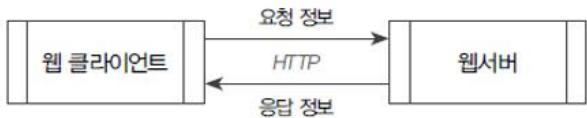
HTTPS는 소켓 통신에서 일반 텍스트를 이용하는 대신에, SSL이나 TLS 프로토콜을 통해 세션 데이터를 암호화한다. 따라서 데이터의 적절한 보호를 보장한다. HTTPS의 기본 TCP/IP 포트는 443이다.

보호의 수준은 웹 브라우저에서의 구현 정확도와 서버 소프트웨어, 지원하는 암호화 알고리즘에 달려있다.

HTTPS를 사용하는 웹페이지의 URI는 'http://대신 'https://'로 시작한다.



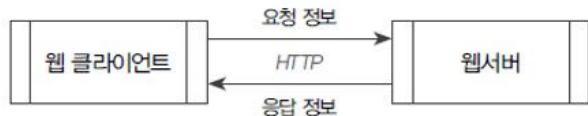
HTTP Protocol – Request



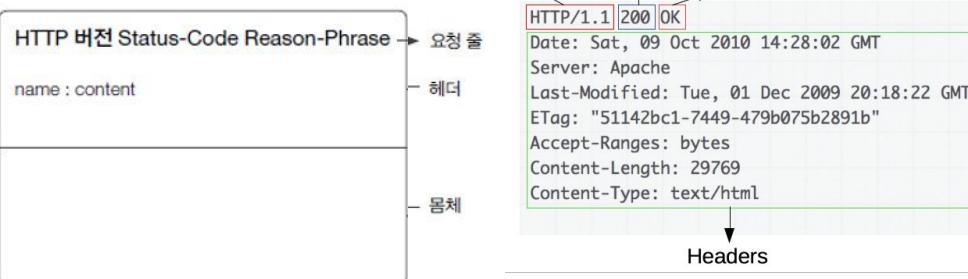
A screenshot of a browser's developer tools Network tab, showing an HTTP request for the URL <https://www.naver.com/>. The request method is GET, and the status code is 200 OK. The response headers include Cache-Control, Content-Encoding, Content-Type, Date, Pragma, Referrer-Policy, Server, Strict-Transport-Security, X-Frame-Options, and X-Xss-Protection. The request headers shown are:

Header	Value
:authority	www.naver.com
:method	GET
:path	/
:scheme	https
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br
Accept-Language	ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7,ru;q=0.6,vi;q=0.5
Cache-Control	max-age=0
Cookie	NNB=7SORXJW6OUVWK; tooltip_setting_close=1; NSCS=
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36

HTTP Protocol – Response



응답 정보



Status Code

- 1xx(정보) : 요청을 받았으며 프로세스를 계속 진행합니다.
- 2xx(성공) : 요청을 성공적으로 받았으며 인식했고 수용하였습니다.
- 3xx(리다이렉션) : 요청 완료를 위해 추가 작업 조치가 필요합니다.
- 4xx(클라이언트 오류) : 요청의 문법이 잘못되었거나 요청을 처리할 수 없습니다.
- 5xx(서버 오류) : 서버가 명백히 유효한 요청에 대한 충족을 실패했습니다.

The screenshot shows the 'Headers' tab of the browser developer tools. The 'General' section displays the following information:

Request URL:	https://www.naver.com/
Request Method:	GET
Status Code:	200 OK
Remote Address:	223.130.195.95:443
Referrer Policy:	strict-origin-when-cross-origin

The 'Response Headers' section is highlighted with a red box and contains the following entries:

Cache-Control:	no-cache, no-store, must-revalidate
Content-Encoding:	gzip
Content-Type:	text/html; charset=UTF-8
Date:	Tue, 17 Oct 2023 06:54:02 GMT
Pragma:	no-cache
Referrer-Policy:	unsafe-url
Server:	NWS
Strict-Transport-Security:	max-age=63072000; includeSubdomains
X-Frame-Options:	DENY
X-Xss-Protection:	1; mode=block

The screenshot shows the 'Response' tab of the browser developer tools. The response body is displayed as HTML code:

```
<!doctype html>
<html lang="ko" class="fzoom">
  <head>
    <meta charset="utf-8">
    <meta name="Referrer" content="origin">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=1190">
    <title>NAVER</title>
    <meta name="apple-mobile-web-app-title" content="NAVER">
    <meta name="robots" content="index,nofollow"/>
    <meta name="description" content="네이버 메인어플리케이션">
    <meta property="og:title" content="네이버">
```

사용자 에이전트

文 A 18개 언어 ▾

문서 **로튼**

읽기 편집 역사 보기 도구 ▾

위키백과, 우리 모두의 백과사전.

사용자 에이전트(使用者—, User agent)는 사용자를 대신하여 일을 수행하는 소프트웨어 에이전트이다. 예를 들어, 이메일 리더에서는 메일 사용자 에이전트이고, 사용자 에이전트를 뜻하는 용어인 세션 개시 프로토콜에서는 통신 세션 양쪽 끝을 말한다.^[1]

사용자 에이전트 식별 [편집]

소프트웨어 에이전트가 네트워크 프로토콜 안에서 동작할 때, 문자적 식별 문자열을 피어(peer)에 제출함으로써 종종 자기 자신과 애플리케이션 유형, 운영 체제, 소프트웨어 업체, 소프트웨어 리비전을 식별한다. HTTP^[2], SIP^[1], NNTP^[3] 프로토콜에서 이러한 식별 정보는 *User-Agent*라는 헤더 필드를 통해 전달된다. 웹 크롤러와 같은 봇은 종종 URL이나 이메일 주소를 포함하기도 하며 이로 말미암아 웹마스터가 봇의 운영자와 연락을 취할 수 있다.

HTTP에서의 사용 [편집]

인간이 조작하는 웹 브라우저 형식 [편집]

맥 OS 15.6, 사파리 605.1.15 버전의 예시.

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.6  
Safari/605.1.15
```

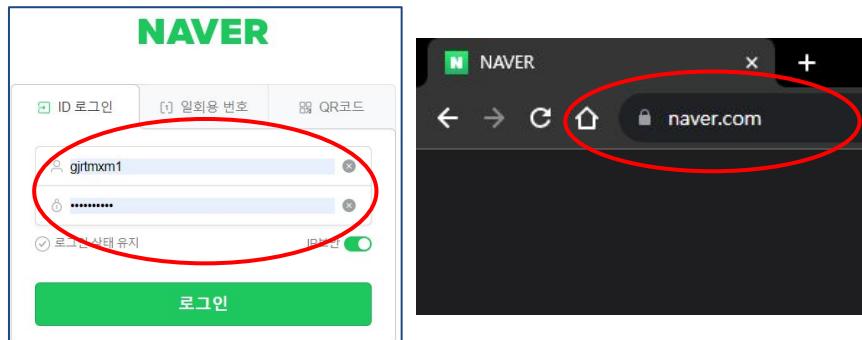
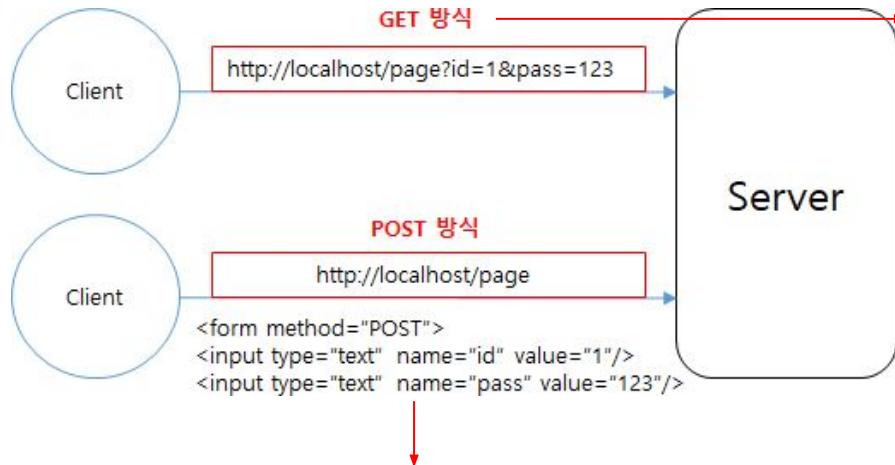
자동화된 에이전트(봇)의 형식 [편집]

구글봇의 예시.

```
Googlebot/2.1 (+http://www.google.com/bot.html)
```

GET과 POST

EST



https://search.naver.com/search.naver
?where=nexearch
&sm=top_hty &fbm=1
&ie=utf8
&query=%EB%B9%84%ED%8A%B8%EC%BD%94%EC%9D%B8

url escape code
비트코인
유니코드 문자의
이스케이프 코드

URL Parameter



POSTMAN

EST

포스트맨은 개발자들이
API를 디자인하고 빌드하고
테스트하고 반복하기 위한
API 플랫폼



The screenshot shows the Postman application interface. At the top, there's a navigation bar with Home, Workspaces, Explore, a search bar, and user authentication options (Sign In, Create Account). Below the navigation is a history panel titled 'History' with a 'New' button and an 'Import' button. A specific request is selected: a GET request to <http://openapi.molit.go.kr>. The request details show the method (GET), URL (http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSObjSvc/getRTMSDataSvcsAptTradeDev?LA...), and a 'Send' button. The 'Params' tab is active, displaying query parameters: LAWD_CD (1110), DEAL_YMD (201512), and serviceKey (QjITnTxSg5%2Bzh%2BWR8hYLMstCDRuf1REcb5E59648Wy7...). Below the parameters is a table for body parameters. The 'Body' tab is selected, showing the XML response received:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <response>
3   <header>
4     <resultCode>99</resultCode>
5     <resultMsg>LIMITED NUMBER OF SERVICE REQUESTS EXCEEDS ERROR.</resultMsg>
6   </header>
7 </response>
```

At the bottom, there are tabs for Body, Cookies (1), Headers (7), Test Results, and a status bar indicating 200 OK, 67 ms, 420 B, and Save Response. The bottom right corner shows a page number 52.

The screenshot shows the Postman application interface. At the top, there is a navigation bar with 'Home', 'Workspaces', 'Explore', a search bar 'Search Postman', and account options 'Sign In' and 'Create Account'. Below the navigation bar, the main workspace displays a history of requests and a detailed view of a current request.

History: Shows a list of recent requests, including a GET request to 'https://search.naver.com/search.naver?query=인공지능' and another to 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPacka'.

Current Request: A GET request to 'https://search.naver.com/search.naver?query=인공지능'. The 'Params' tab is selected, showing a single parameter 'query' with the value '인공지능'. Other tabs include 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. A 'Cookies' tab is also present.

Response Body: The response body is displayed in 'Pretty' format, showing the HTML structure of the search results page. The page includes a doctype declaration, an HTML head with meta charset and referer tags, and a body containing a head section with various meta tags including og:title, og:image, og:description, and og:meta tags.

Status Bar: The status bar at the bottom indicates a 200 OK response with 441 ms latency and 691.1 KB size. It also shows 'Save Response' and a search icon.

Bottom Status: A message at the bottom states 'Console' and 'Not connected to a Postman account'.

Requests 라이브러리

EST

requests는 python 사용자들을 위해 만들어진 간단한 Python용 HTTP 라이브러리이며, 간단하게는 HTTP, HTTPS 웹 사이트에 요청하기 위해 자주 사용되는 모듈 중 하나

파이썬은 기본 라이브러리로 urllib 제공



urllib 보다 간결한 코드로 다양한 HTTP 요청 가능



JavaScript 처리 시 selenium 사용이 일반적



requests도 JavaScript 처리 가능



크롤링시에 웹요청에 requests 사용이 가장 효율적 (정적 페이지 수집용)

Requests 사용법

EST

1 requests 라이브러리

```
pip install requests
```

```
https://finance.naver.com/
```

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

```
PATH = 'https://finance.naver.com/'  
resp = requests.get('https://finance.naver.com/')
```

```
resp
```

```
<Response [200]>
```

```
resp.text
```

```
'<html lang="ko">#\n  <head> #\n    <title>네이버  
    nt-Type" content="text/html; charset=utf-8" /  
    type" content="text/javascript" /> #\n    <meta h  
    ="text/css" /> #\n    <meta name="apple-mobile-w  
    #\n    <meta property="og:title" content="네이버  
    content="https://ssl.pstatic.net/static/m/sto  
    meta property="og:url" content="https://finan  
    description" content="국내 해외 증시 지수, 시  
    #\n    <meta property="og:type" content="article  
    bnailUrl" content="" /> #\n    <meta property="o  
    /> #\n    <meta property="og:article:author:url"  
    #\n    <link rel="stylesheet" type="text/css" hr  
    tic.pc/20230808201105/css/finance_header.css"  
    t/css" href="https://ssl.pstatic.net/imgstock  
    s" /> #\n    <link rel="stylesheet" type="text/c  
    ck/static.pc/20230808201105/css/newstock3.css  
    href="https://ssl.pstatic.net/imgstock3.css"
```

그런데 왜 이건 안되지?

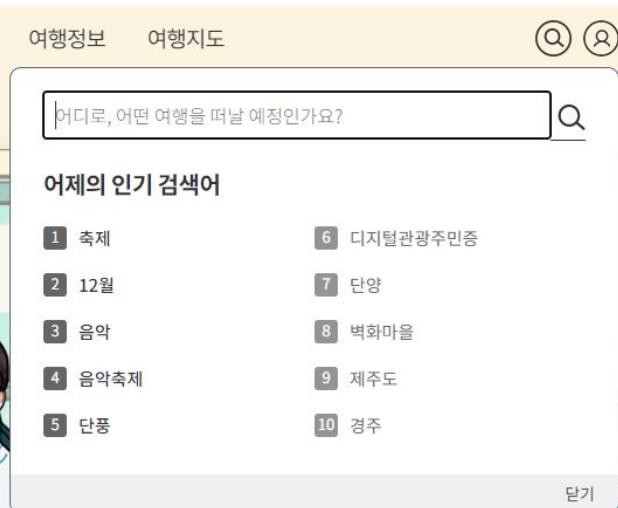
EST

▼ 1 셀레니엄이 필요한 이유

<https://korean.visitkorea.or.kr/>

```
n [3]: import requests  
from bs4 import BeautifulSoup  
  
n [4]: response = requests.get('https://korean.visitkorea.or.kr/')  
html_src = response.text  
  
n [5]: soup = BeautifulSoup(html_src, 'html.parser')  
  
n [6]: soup.select('#inp_search') # 동적페이지에서 불가, 반드시 액션(click 등)이 있어야 함  
[]
```

[https://Korean.visitKorea.or.Kr/main/main.
do](https://Korean.visitKorea.or.Kr/main/main.do)



The screenshot shows a travel website's search interface. At the top, there are tabs for '여행정보' (Travel Information) and '여행지도' (Travel Map). Below the search bar, there is a section titled '어제의 인기 검색어' (Popular Searches Yesterday) with a list of 10 items:

1 축제	6 디지털관광주민증
2 12월	7 단양
3 음악	8 벽화마을
4 음악축제	9 제주도
5 단풍	10 경주

At the bottom right of the search interface is a '닫기' (Close) button.

정적 웹페이지(Static)

EST

정적 웹 페이지란 서버(Web Server)에 미리 저장된 파일이 그대로 전달되는 웹 페이지를 말합니다.

즉, 제가 특정 웹페이지의 url 주소만 주소창에 입력하면 웹 브라우저로 HTML 정보를 마음대로 가져다 쓸 수 있는 것입니다.

동적 웹 페이지와 가장 큰 차이점은 '**url 주소 외에는 아무 것도 필요없다**' 는 점입니다.

만약 마우스 휠을 스크롤 다운 했는데, url에 변화는 없고 페이지에 내용이 추가된다면 그 페이지는 동적 웹 페이지입니다.

정적 웹 페이지 (Static Web Page)

- 웹 서버에 이미 저장된 파일(HTML 파일, 이미지, JavaScript 파일 등)을 클라이언트에게 전송하는 웹 페이지다.
- 사용자는 서버에 저장된 데이터가 변경되지 않는 한 고정된 웹 페이지를 계속 보게 된다.
- 따라서 모든 사용자는 같은 결과의 웹 페이지를 서버에 요청하고 응답 받게 된다.

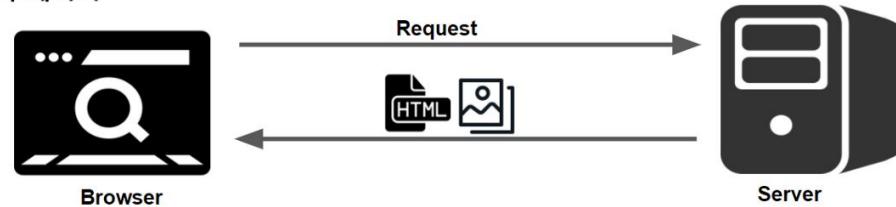
• 장점

- 다른 처리 없이 요청에 대한 파일만 전송하기 때문에 빠르다.
- 단순한 문서로 웹 서버를 구축하므로 호스팅 서버에 연결하는 비용이 적다.

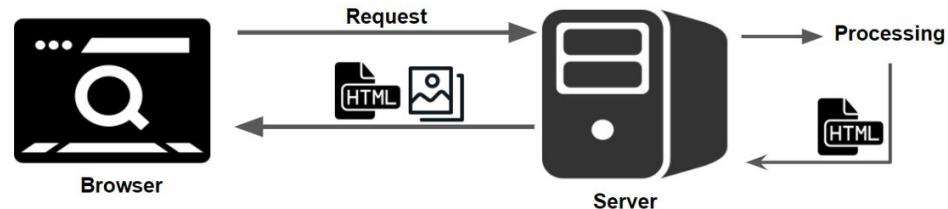
• 단점

- 저장된 정보만 보여주기 때문에 서비스가 한정적이다.
- 추가, 삭제, 수정 등의 작업이 모두 코드를 직접 건드려야 하기 때문에 관리가 힘들다.

정적 페이지



동적 페이지



정적 웹페이지 사례

EST

< 네이버 검색 결과 >

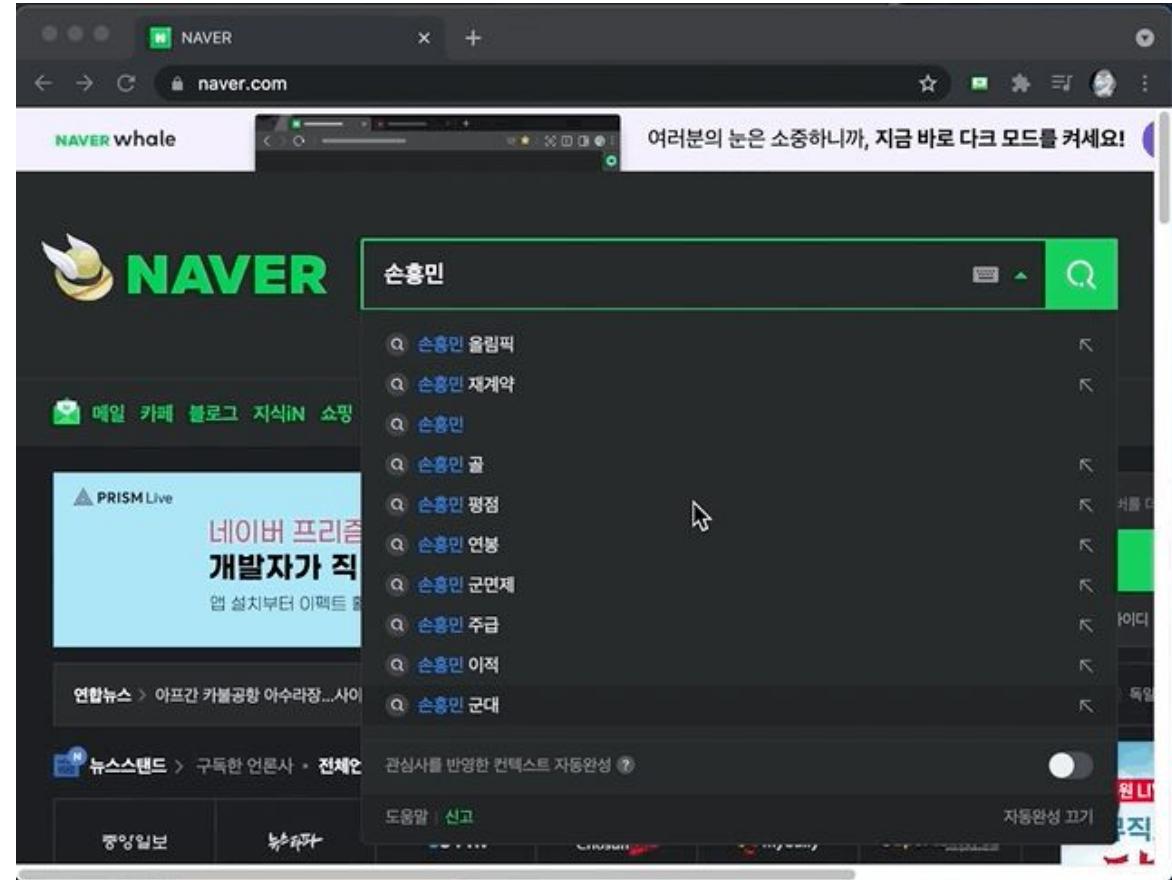
1) 네이버를 열어서 검색창에

'손흥민'을 검색합니다.

2) 검색 결과의 url을 복사하시고

다시 주소창에 해당 url을
입력합니다.

3) 처음 검색결과와 동일한
페이지를 볼 수 있습니다.



동적 웹페이지(Dynamic)

EST

동적 웹페이지란 url만으로는 들어갈 수 없는 웹페이지를 말합니다.

혹시 들어가지더라도 url의 변화가 없는데도 실시간으로 내용이 계속해서 추가되거나 수정된다면 동적 웹 페이지입니다.

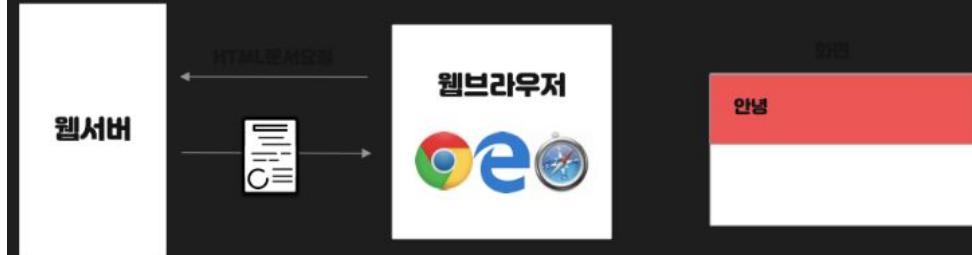
여기서 무언가를 클릭해서 페이지가 변경되는 것은 다른 경우입니다

- 요청에 관하여 사용자는 조건(상황, 시간, 요청 등)에 따라 다른 결과를 받게 된다.
- 동적 웹페이지의 종류(CSR, SSR , MPA, SPA)

1. CSR (Client Side Rendering)



2. SSR (Server Side Rendering)



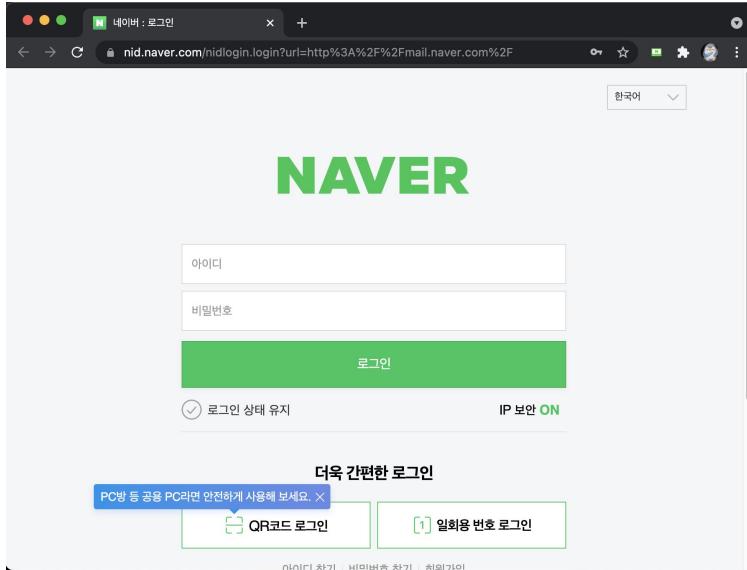
- CSR과 상반되게 서버에서 동적으로 데이터까지 전부 삽입하여 완성된 HTML을 넘겨준다.
- 서버 렌더링은 브라우저에서 응답을 받기 전에 처리되므로 클라이언트에서 데이터를 가져오거나 템플릿 작성에 대한 추가 왕복이 발생하지 않는다. (어쨌든 웹 서버에서 모든 요청이 처리된다.)

- CSR은 데이터가 없는 HTML 문서나 Static 파일만을 처음에 받아와 로드하고, 이후에 데이터를 요청하여 받아오는 방식이다.
- 자바스크립트를 사용하여 브라우저에서 페이지를 직접 렌더링을 진행한다.
- 모든 로직, 데이터 가져오기, 템플릿 및 라우팅 등은 서버가 아닌 클라이언트 측에서 처리한다.

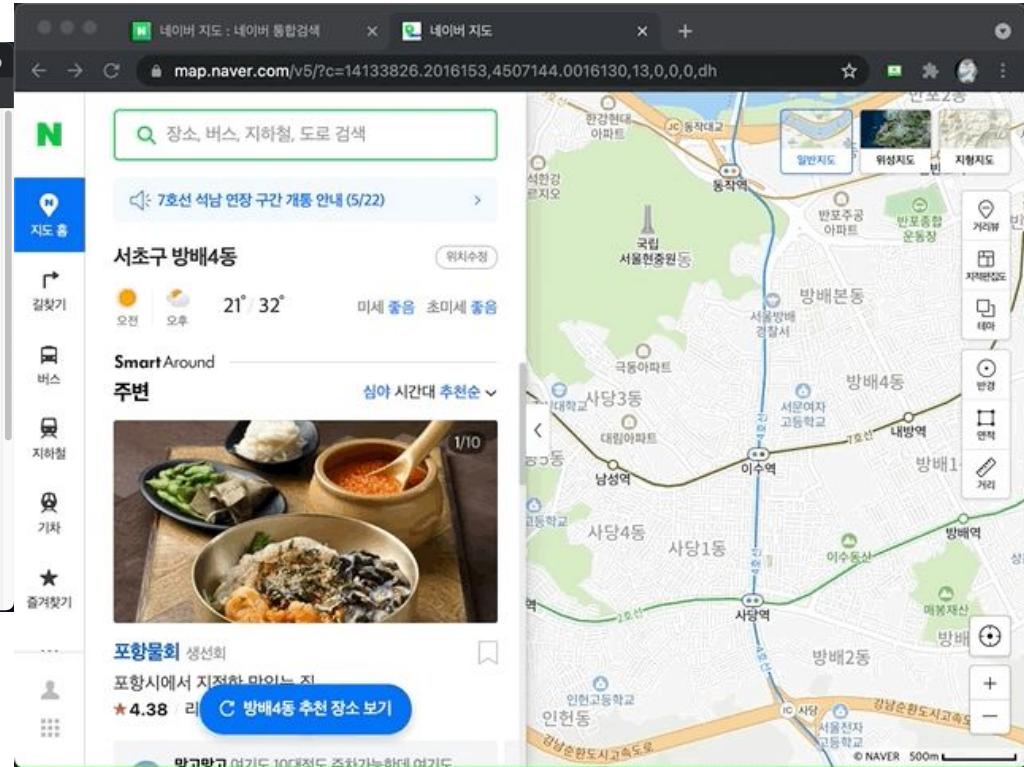
동적 웹페이지 사례

EST

로그인을 해야만 접속 가능한 네이버 메일



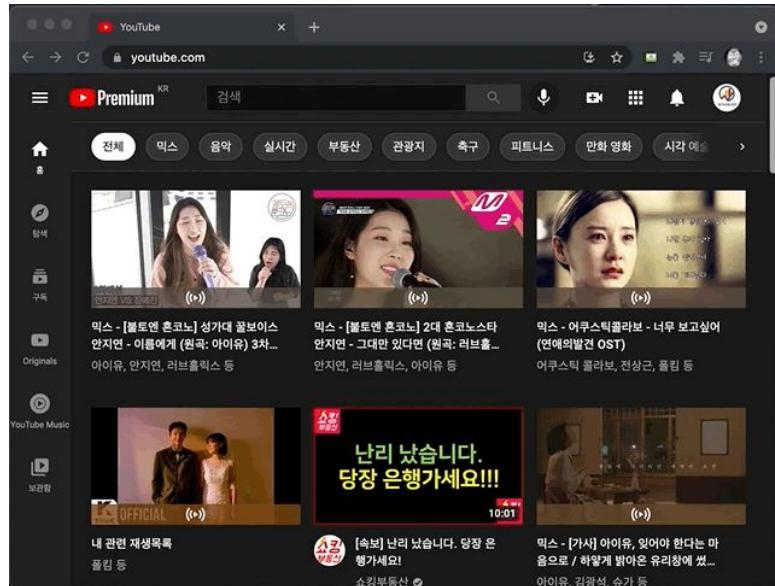
보고 있는 위치에 출력 결과와 url이 계속 변하는 네이버 지도



reference

동적 웹페이지 사례

드래그를 아래로 내리면 계속 새로운 사진과
영상이 나타나는 인스타그램과 유튜브



더보기를 클릭하면 과거 데이터를 볼 수 있는
네이버 주식

m.stock.naver.com/domestic/index/KOSPI/total

① pay 증권 국내증시

코스피 2,460.17

일별 시세 <https://m.stock.naver.com/domestic/index/KOSPI/total>

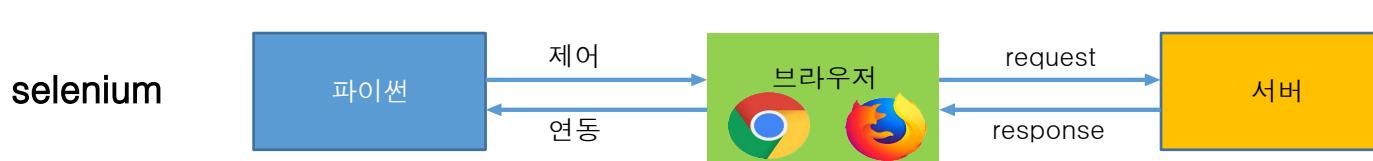
날짜	종가	전일대비	등락률	시가	고가	저가
10.17	2,460.17	▲23.93	+0.98%	2,454.14	2,466.87	2,449.42
10.16	2,436.24	▼19.91	-0.81%	2,442.43	2,453.77	2,422.52
10.13	2,456.15	▼23.67	-0.95%	2,460.85	2,466.62	2,452.83
10.12	2,479.82	▲29.74	+1.21%	2,465.19	2,479.82	2,464.84
10.11	2,450.08	▲47.50	+1.98%	2,436.52	2,463.56	2,436.52
10.10	2,402.58	▼6.15	-0.26%	2,436.58	2,448.24	2,402.44
10.06	2,408.73	▲5.13	+0.21%	2,408.81	2,421.18	2,403.92
10.05	2,403.60	▼2.09	-0.09%	2,423.35	2,426.61	2,402.50
10.04	2,405.69	▼59.38	-2.41%	2,435.78	2,435.78	2,402.84
09.27	2,465.07	▲2.10	+0.09%	2,447.99	2,469.72	2,445.51

더보기 ▼

정적/동적 웹페이지별 수집 방법

EST

	정적 수집	동적 수집
사용 패키지	requests / urllib	selenium
수집 커버리지	정적 웹 페이지	정적/동적 웹 페이지
수집 속도	빠름 (별도 페이지 조작 필요 X)	상대적으로 느림
파싱 패키지	beautifulsoup	beautifulsoup / selenium



코랩에서도 셀레니움을 headless로 사용 가능하지만
UI를 직접 보고 하는게 편하므로 주피터 노트북에서 실습

셀레니움(Selenium)은 웹 애플리케이션 자동화 및 테스트를 위한 포터블 프레임워크이다.

셀레니움은 테스트 스크립트 언어를 학습할 필요 없이 기능 테스트를 만들기 위한 플레이백 도구를 제공한다.

이 테스트들은 현대의 대부분의 웹 브라우저에서 수행이 가능하다. 아파치 2.0 라이선스 오픈 소스이다.

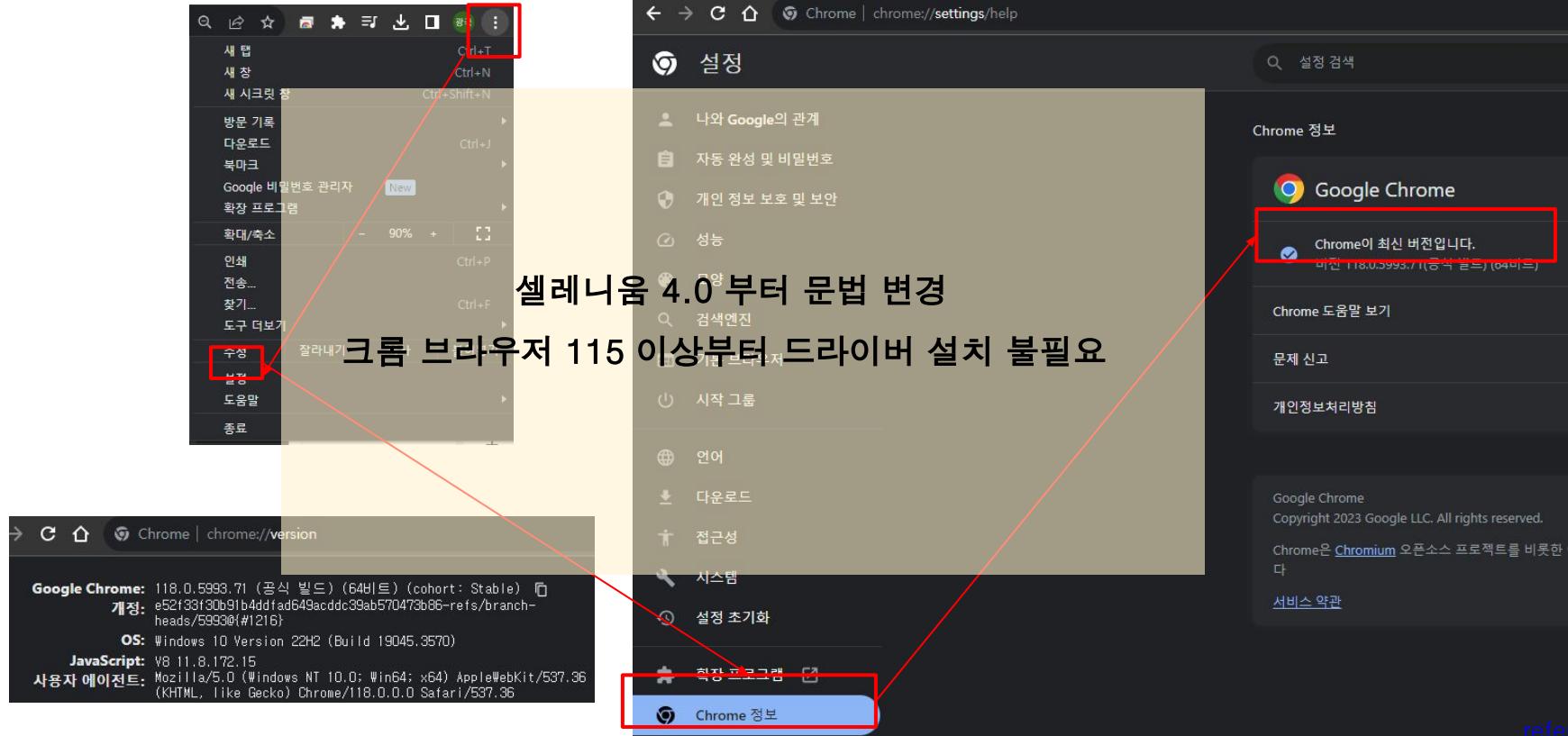
- 셀레니움 라이브러리를 웹 스크래핑에 사용하는 이유.
 1. 자바스크립트가 동적으로 만든 데이터를 크롤링 하기 위해
 2. 사이트의 다양한 HTML 요소에 클릭, 키보드 입력 등 이벤트를 주기 위해
- Selenium을 잘 활용하면, 평소에 반복적으로 하고 있는 웹상의 업무를 자동화할 수도 있습니다.
 1. 자동으로 로그인하기
 2. 메일 보내기 자동화
 3. 블로그 이웃 새글 자동좋아요 누르기
 4. 인스타그램 자동으로 좋아요, 댓글 작성하기
 5. 등등 정말 많은 다양한 일



<https://www.selenium.dev/>
<https://selenium-python.readthedocs.io/>

SELENIUM 사용법(셀레니움3 and 크롬버전114이하) EST

1. 웹 브라우저 버전 확인 : 우상단 설정(점3개) → 설정 메뉴 → 좌하단 크롬 정보 (chrome://version/)



SELENIUM 사용법(셀레니움3 and 크롬버전114이하) EST

2. 웹 드라이버 다운로드 및 압축해제 (<https://chromedriver.chromium.org/downloads>)

- 웹 브라우저를 제어하기 위한 드라이버
- 크롬버전 115까지는 드라이버 최신114버전을 쓰면 됨
- 크롬115버전부터 웹드라이버 버전 통합(<https://googlechromelabs.github.io/chrome-for-testing/>)

Chrome for Testing availability 

Stable

This page lists the latest available cross-platform Chrome for Testing versions and assets per Chrome release channel. Version: 118.0.5993.70 (r1192594)

Consult [our JSON API endpoints](#) if you're looking to build automated scripts based on Chrome for Testing release data

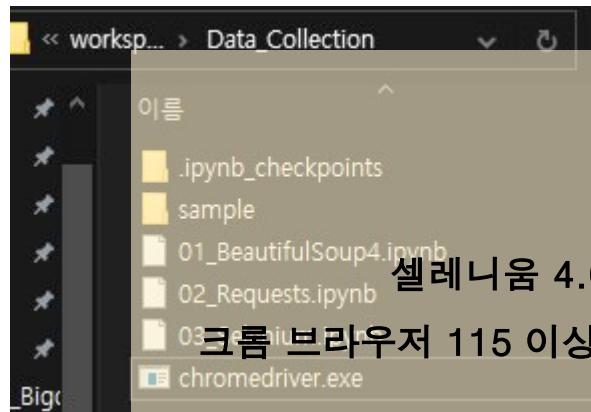
Last updated @ 2023-10-17T12:09:39.293Z

Channel	Version	Revision	Status	Binary	Platform	URL
Stable	118.0.5993.70	r1192594	<input checked="" type="checkbox"/>	chrome	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/linux64/chrome-linux64.zip
Beta	119.0.6045.21	r1204232	<input checked="" type="checkbox"/>	chrome	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/mac-arm64/chrome-mac-arm64.zip
Dev	120.0.6062.2	r1208568	<input checked="" type="checkbox"/>	chrome	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/mac-x64/chrome-mac-x64.zip
Canary	120.0.6071.0	r1210159	<input checked="" type="checkbox"/>	chrome	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win32/chrome-win32.zip
Canary_(upcoming)	120.0.6072.0	r1210586	<input checked="" type="checkbox"/>	chromedriver	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win64/chromedriver-win64.zip
				chromedriver	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/linux64/chromedriver-linux64.zip
				chromedriver	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/mac-x64/chromedriver-mac-x64.zip
				chromedriver	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win32/chromedriver-win32.zip
				chromedriver	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win64/chromedriver-win64.zip
				chrome-headless-shell	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/linux64/chrome-headless-shell-linux64.zip
				chrome-headless-shell	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/mac-arm64/chrome-headless-shell-mac-arm64.zip
				chrome-headless-shell	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/mac-x64/chrome-headless-shell-mac-x64.zip
				chrome-headless-shell	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win32/chrome-headless-shell-win32.zip
				chrome-headless-shell	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.70/win64/chrome-headless-shell-win64.zip

reference

SELENIUM 사용법(셀레니움3 and 크롬버전114이하) EST

3. 워크스페이스 디렉토리에 크롬 드라이버 실행파일을 이동



공식 도큐먼트 참고

<https://www.selenium.dev/>

5. 원하는 태그 찾기

- driver.find_element()
- driver.find_elements()

4. selenium으로 크롬 브라우저창 열기

- driver 경로를 파일경로와 같은 곳에 둘 경우

```
from selenium import webdriver
driver = webdriver.Chrome()
url = 'https://www.google.com'
driver.get(url)
```

- driver 경로를 파일경로와 다른 곳에 둘 경우

```
from selenium import webdriver
driver = webdriver.Chrome('driver의경로')
url = 'https://www.google.com'
driver.get(url)
```

6. 키 입력하기

- 클릭 : .click()
- 키 입력: .send_keys()

전통 로케이터

로케이터	설명
CLASS_NAME	클래스 이름에 검색 값이 포함된 요소를 찾습니다(복합 클래스 이름은 허용되지 않음).
CSS_SELECTOR	CSS 선택기와 일치하는 요소를 찾습니다.
ID	ID 속성이 검색 값과 일치하는 요소를 찾습니다.
NAME	NAME 속성이 검색 값과 일치하는 요소를 찾습니다.
LINK_TEXT	보이는 텍스트가 검색 값과 일치하는 앵커 요소를 찾습니다.
PARTIAL_LINK_TEXT	보이는 텍스트에 검색 값이 포함된 앵커 요소를 찾습니다. 여러 요소가 일치하는 경우 첫 번째 요소만 선택됩니다.
TAG_NAME	태그 이름이 검색 값과 일치하는 요소를 찾습니다.
XPATH	XPath 표현식과 일치하는 요소를 찾습니다.

상대 로케이터

구분	사용예시
Above	email_locator = locate_with(By.TAG_NAME, "input").above({By.ID: "password"})
Below	password_locator = locate_with(By.TAG_NAME, "input").below({By.ID: "email"})
Left of	cancel_locator = locate_with(By.TAG_NAME, "button").to_left_of({By.ID: "submit"})
Right of	submit_locator = locate_with(By.TAG_NAME, "button").to_right_of({By.ID: "cancel"})
Near	email_locator = locate_with(By.TAG_NAME, "input").near({By.ID: "lbl-email"})
로케이터 연 결	submit_locator = locate_with(By.TAG_NAME, "button").below({By.ID: "email"}).to_right_of({By.ID: "cancel"})

요소(element)와 상호작용 하기

기능	설명	사용예시
click	모든 요소에 적용	element.click()
send_keys	텍스트 필드 및 콘텐츠 편집 가능한 요소에만 적용됨.	element.send_keys("webdriver" + Keys.ENTER)
clear	텍스트 필드 및 콘텐츠 편집 가능한 요소에만 적용됨	element.clear()
submit	Form 요소에만 적용됨	element.submit()
select	모든 요소에 적용	-----

SELENIUM 4.0 키보드

EST

- 키 입력: .send_keys()
- 예시 : .send_keys(Keys.RETURN)

명령어	기능
Keys.ENTER , Keys.RETURN	엔터
Keys.SPACE	스페이스바
Keys.ARROW_UP, Keys.ARROW_DOWN Keys.ARROW_LEFT, Keys.ARROW_RIGHT	방향키(상하좌우)
Keys.BACK_SPACE, Keys.DELETE	지우기(백스페이스), 지우기(딜리트)
Keys.CONTROL , Keys.ALT Keys.SHIFT , Keys.TAB	자주 사용하는 기능키 (ctrl, alt, shift, tab)
Keys.PAGE_UP, Keys.PAGE_DOWN	스크롤 업/다운
Keys.F1~9	function key (F1 ~ F9)
Keys.EQUALS/ESCAPE/HOME/INSERT	기타

요소(element) 정보 가져오기

기능	설명	사용예시
크기 및 위치	요소의 치수와 좌표	driver.find_element(By.CSS_SELECTOR, "h1").rect
CSS 값	요소의 스타일 속성	driver.findElement(By.LINK_TEXT, "More information...").value_of_css_property('color')
텍스트 내용	요소의 렌더링된 텍스트	driver.find_element(By.CSS_SELECTOR, "h1").text

수집 자동화

pyinstaller

windows scheduler

PyInstaller란?

EST

PyInstaller란 코딩한 파이썬 프로그램을 파이썬에 대해 전혀 모르는 분들도 사용할 수 있도록 실행 파일(.exe)로 만들어주는 파이썬 패키지입니다.

- 1) Python 3.6이상만 사용 가능
- 2) Windows, Mac OS X 및 GNU / Linux에서 사용가능함
- 3) Windows에서 컴파일 된 실행파일은 Windows에서만 사용가능 (다른 OS의 경우도 마찬가지)
- 4) Windows 8 이상만 지원 / Mac OS X 10.7(Lion) 이상만 지원

<https://pyinstaller.readthedocs.io/en/stable/>

python code

```
from selenium.webdriver.common.keys import Keys

def page_work():
    result = driver.find_elements(By.CSS_SELECTOR, '#search_result .tit>a')
    global contents_no, cnt

    for item in result:
        contents_no += 1

        if contents_no <= cnt :
            print(f'{contents_no} [{item.get_attribute("text")}]')
            item.send_keys(Keys.ENTER) # .click()은 에러 잘남

            time.sleep(2)
            html = driver.page_source
            html_dom = BeautifulSoup(html, 'lxml')
```

python_program.exe

compile & Build



설치 및 사용 방법

EST

- 설치 방법

- 설치 : pip install pyinstaller
- 설치 확인 : pyinstaller --version

- 사용 방법

- cmd 실행
- 파이썬 파일 디렉토리로 이동
- pyinstaller 파이썬파일명.py
 - F, --onefile : 실행파일 1개로 통합 빌드하는 설정
 - 실행파일의 용량이 커지고 실행 속도가 느려지는 단점이 있음
 - w, --windowed, --noconsole : 표준 I/O용 콘솔창 실행 안함
 - icon = ico파일경로 : 실행파일 아이콘 변경
 - 무료 아이콘 사이트 :<https://icon-icons.com/ko/>, <https://www.iconfinder.com/>
 - hidden-import ; pyinstall의 특정모듈 실행오류 대응을 위한 수동 import
 - openpyxl.cell._writer

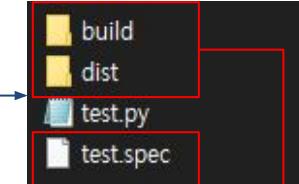
- dist폴더의 파이썬파일명.exe를 실행

```
명령 프롬프트 - pyinstaller test.py -F --icon=bot.ico --... Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\maeng>cd C:\workspace\test\Data_Collection

C:\workspace\test\Data_Collection>pyinstaller test.py -F
--icon=bot.ico --hidden-import openpyxl.cell._writer
4281 INFO: PyInstaller: 6.1.0
4282 INFO: Python: 3.11.5 (conda)
4293 INFO: Platform: Windows-10-10.0.19045-SP0
4294 INFO: wrote C:\workspace\test\Data_Collection\test.s
```

```
517455 WARNING: Execution of 'set_exe_build_timestamp' failed on attempt #1
/ 20: PermissionError(13, 'Permission denied'). Retrying in 0.05 second(s)
525339 INFO: Building EXE from EXE-00.toc completed successfully.
```



python_program.exe

reference

pathlib 오류 해결

```
C:\Users\Min>pyinstaller --version
The 'pathlib' package is an obsolete backport of a standard library package and is incompatible with PyInstaller. Please
remove this package (located in C:\Users\Min\anaconda3\Lib\site-packages) using
    conda remove
then try again.
```

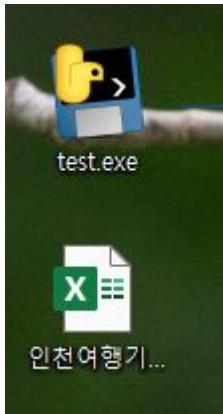
C:\Users\<username>
\\anaconda3\\Lib\\site-packages

pathlib 폴더 삭제

이름	수정한 날짜	유형	크기
pandas	2023-12-14 오후...	파일	폴더
pandas-2.0.3.dist-info	2023-12-14 오후...	파일	폴더
pandocfilters-1.5.0.dist-info	2023-12-14 오후...	파일	폴더
panel	2023-12-14 오후...	파일	폴더
panel-1.2.3.dist-info	2023-12-14 오후...	파일	폴더
param	2023-12-14 오후...	파일	폴더
param-1.13.0.dist-info	2023-12-14 오후...	파일	폴더
paramiko	2023-12-14 오후...	파일	폴더
paramiko-2.8.1.dist-info	2023-12-14 오후...	파일	폴더
parsel	2023-12-14 오후...	파일	폴더
parsel-1.6.0.dist-info	2023-12-14 오후...	파일	폴더
parso	2023-12-14 오후...	파일	폴더
parso-0.8.3.dist-info	2023-12-14 오후...	파일	폴더
partd	2023-12-14 오후...	파일	폴더
partd-1.4.0.dist-info	2023-12-14 오후...	파일	폴더
past	2023-12-14 오후...	파일	폴더
pathlib-1.0.1.dist-info	2023-12-14 오후...	파일	폴더
pathspec	2023-12-14 오후...	파일	폴더

exe 파일 실행 결과

EST



```
C:\Users\maeng\Desktop\test.exe
검색어:인천
스크래핑 할 건수는 몇건입니까?: 15
DevTools Listening on ws://127.0.0.1:55209/devtools/browser/fe44b8e8-6fb7-43e0-95ce-8c7e18280f89
스크래핑 프로그램 실행
===== 1 페이지 스크래핑 시작 =====
[콘텐츠 1]
서울 근교 여행, 아이와 나들이 떠나는 인천 여행 코스
[콘텐츠 2]
인천 여행, 선선한 저녁에 즐기는 인천 야경 명소 4
[콘텐츠 3]
짜장면이 태어난 차이나타운의 막자골목, 인천 북성동원조자장면거리와 짜장면박물관
[콘텐츠 4]
남만과 그리움을 찾아서, 인천 경인아라뱃길 정서진 드라이브
[콘텐츠 5]
견공의, 견공에 의한, 견공을 위한 놀이 공간 인천대공원 반려견놀이터
[콘텐츠 6]
개항로라 쓰고 뉴트로라 읽는다! 스마트하게 즐기는 인천 Next Level 여행지 8
[콘텐츠 7]
우리나라 최초의 등대, 인천 팔미도등대
[콘텐츠 8]
과거부터 현재까지, 인천으로 떠나는 시간 여행
[콘텐츠 9]
옛 감성 물씬, 인천 구도심(동인천) 테마여행
[콘텐츠 10]
[인천] 도심 속 공원으로 떠나는 비대면 힐링여행 인천 송도 센트럴파크 & 소래습지생태공원
===== 1 페이지 스크래핑 완료 =====
===== 1 페이지 인천여행기사_15건_20231021.xlsx파일 저장 완료 =====
===== 2 페이지 스크래핑 시작 =====
[콘텐츠 11]
시금 기억하는 어제의 풍경, 인천 개항장 거리
[콘텐츠 12]
인천 여행, 아이들과 다녀오기 좋은 당일치기 코스
[콘텐츠 13]
공항철도 타고 한나절 섬 여행, 인천 무의도와 장봉도
[콘텐츠 14]
충황만, 알면 섭섭한 인천 영종도&용유도 로맨틱 여행
[콘텐츠 15]
[인천 당일치기 여행] 부지런히 먹으러 가는 인천 개항장거리 맥루어
===== 2 페이지 스크래핑 완료 =====
===== 2 페이지 인천여행기사_15건_20231021.xlsx파일 저장 완료 =====
스크래핑 프로그램 종료
```

A	B	C
	제목	내용
0	서울 근교 여행, 아이와 이런 분들에게 추천해 드립니다	
1	인천 여행, 선선한 저녁 이런 분들에게 추천해 드립니다	
2	짜장면이 태어난 차이나 인천차이나타운에 있는 북성	
3	남만과 그리움을 찾아서 경복궁 광화문을 기준으로 정	
4	견공의, 견공에 의한, 견공을 위한 놀이 공간 인천대공원	
5	개항로라 쓰고 뉴트로라 읽는다! 스마트하게 즐기는 인천 Next Level 여행지	
6	우리나라 최초의 등대, 인천 팔미도등대	
7	과거부터 현재까지, 인천으로 떠나는 시간 여행	
8	옛 감성 물씬, 인천 구도과거 서민들이 끈질기게 삶을	
9	[인천] 도심 속 공원으로도시는 늘 분주하다. 그래서,	
10	지금 기억하는 어제의 풍물/사진 여행작가 김애진 시	
11	인천 여행, 아이들과 다녀오기 좋은 당일치기 코스	
12	공항철도 타고 한나절 섬 여행, 인천 무의도와 장봉도	
13	충황만, 알면 섭섭한 인천 영종도&용유도 로맨틱 여행	
14	[인천 당일치기 여행] 부지런히 먹으러 가는 인천 개항장거리 맥루어	

실행 속도 향상 팁

EST

--onefile 옵션으로 단 한 개의 실행파일을 생성할 때 편리하긴 하지만 실행이 느리다는 단점이 있다. 컴퓨터 사양에 따라 속도 차이는 있겠지만, 실행을 위해 대략 5초 이상의 시간이 걸린다.

실행 속도를 빠르게 하려면 --onefile 옵션을 생략하고 pyinstaller를 실행하면 된다. 이때는 실행파일 외에 많은 부수적인 파일이 생성되므로 배포 시에 실행 파일 한 개가 아닌 dist 디렉터리에 생성된 모든 파일을 zip 등으로 압축하여 전달하거나 별도의 설치 파일 제작 프로그램 (inno setup 또는 nsis 등)을 이용하여 설치 파일로 만든 다음 제공해야 한다.

NSIS(Nullsoft Scriptable Install System) 예시
스크립트 기반으로 동작하는 윈도우용 설치 시스템(free software)

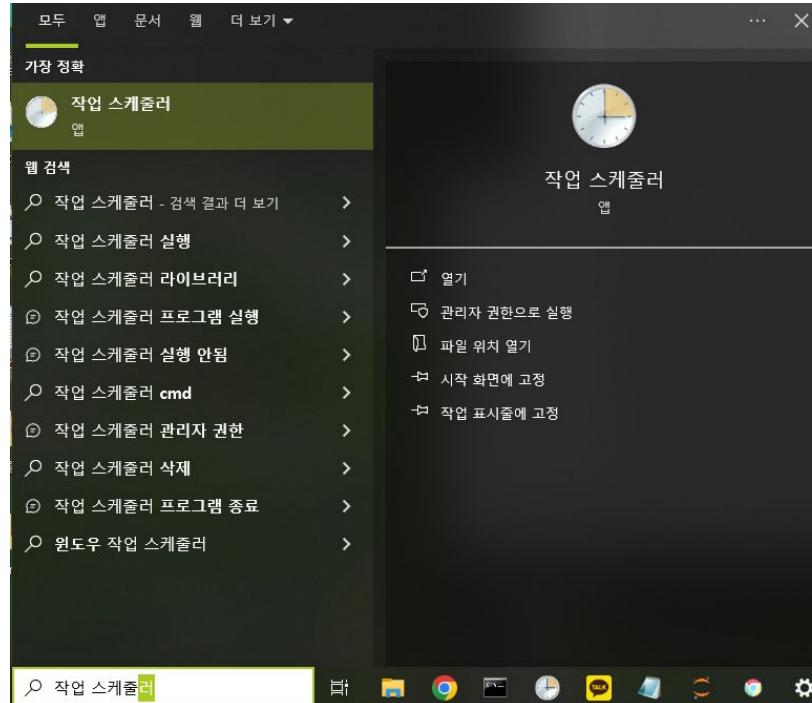


윈도우 작업 스케줄러

EST

반복적인 작업을 일정에 맞추어 실행하는 윈도우 ‘작업 스케줄러’를
사용하여 수집 작업을 규칙적으로 실행하겠습니다.

윈도우 검색에서 ‘스케줄러’입력 후 작업 스케줄러 실행

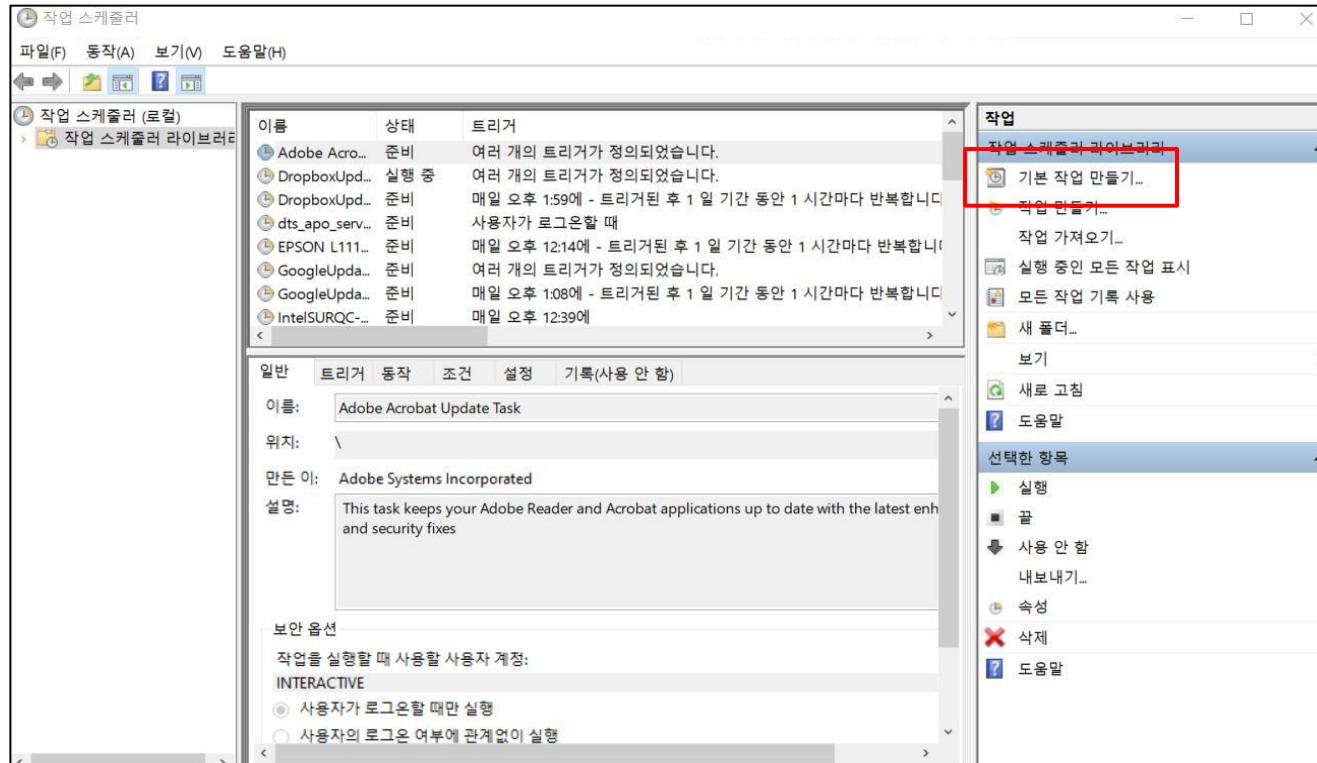


윈도우 작업 스케줄러

EST

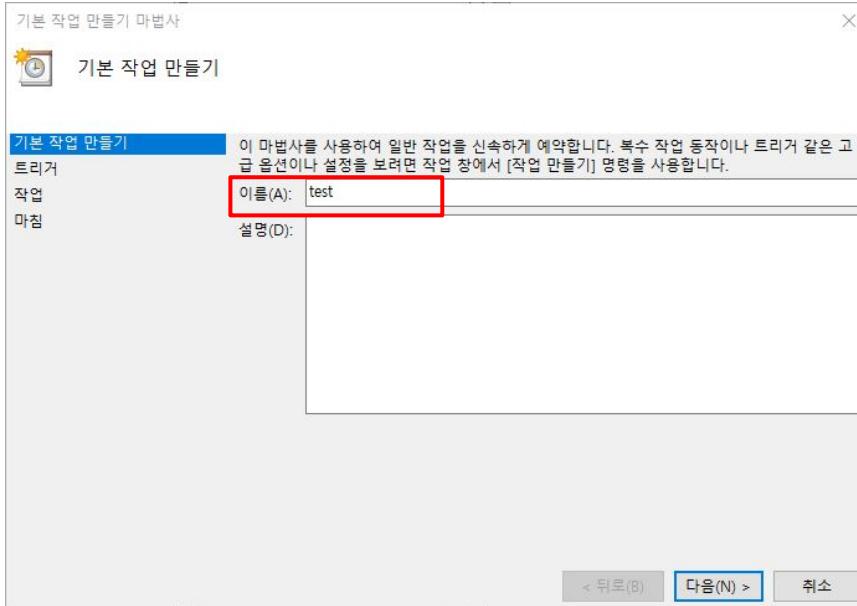
벌써 중요 프로그램들의 update 실행이 등록되어 사용중인 것을 확인할 수 있습니다.

오른쪽 작업 영역에서 ‘기본 작업 만들기’ 클릭



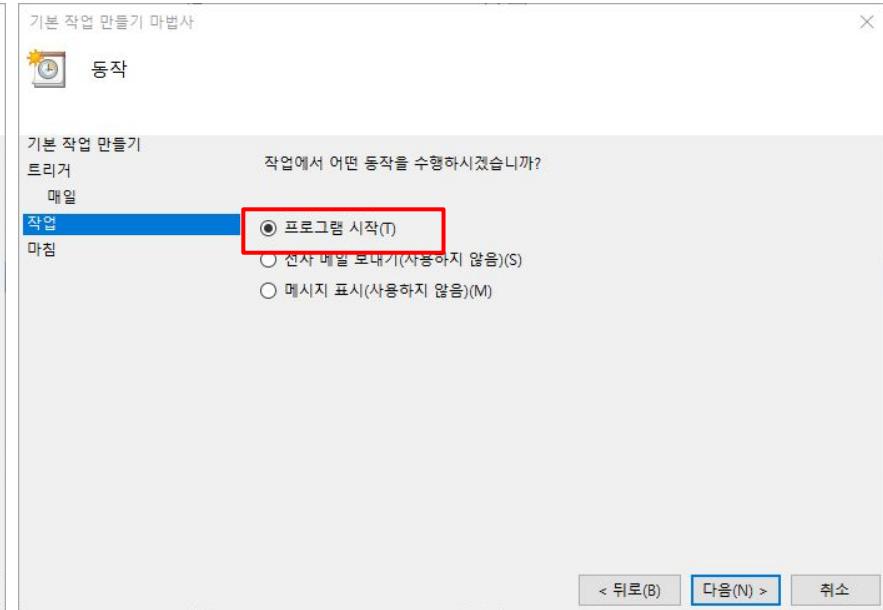
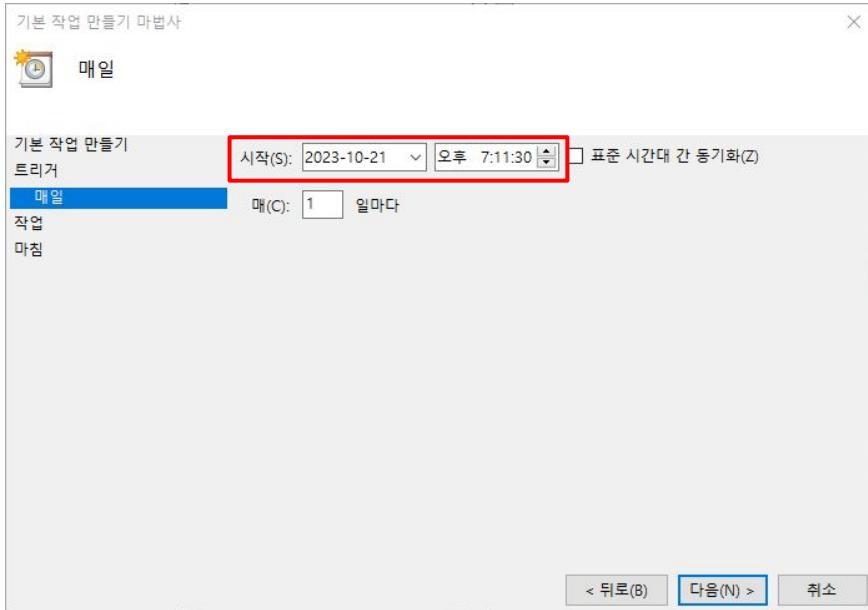
윈도우 작업 스케줄러

EST



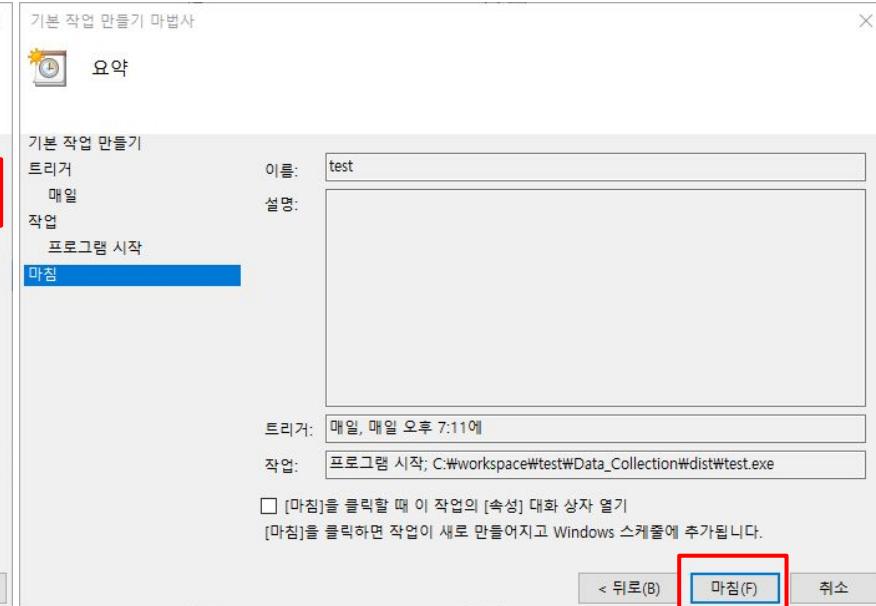
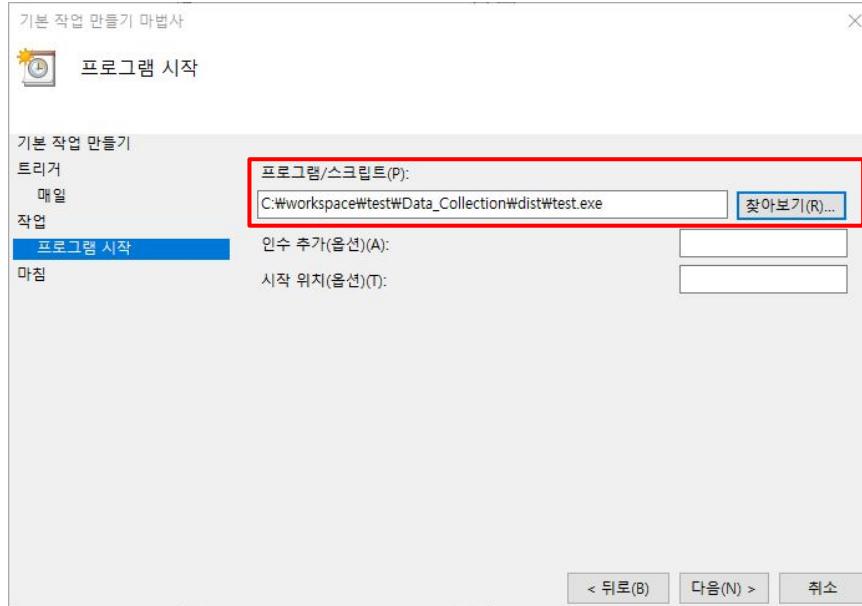
윈도우 작업 스케줄러

EST



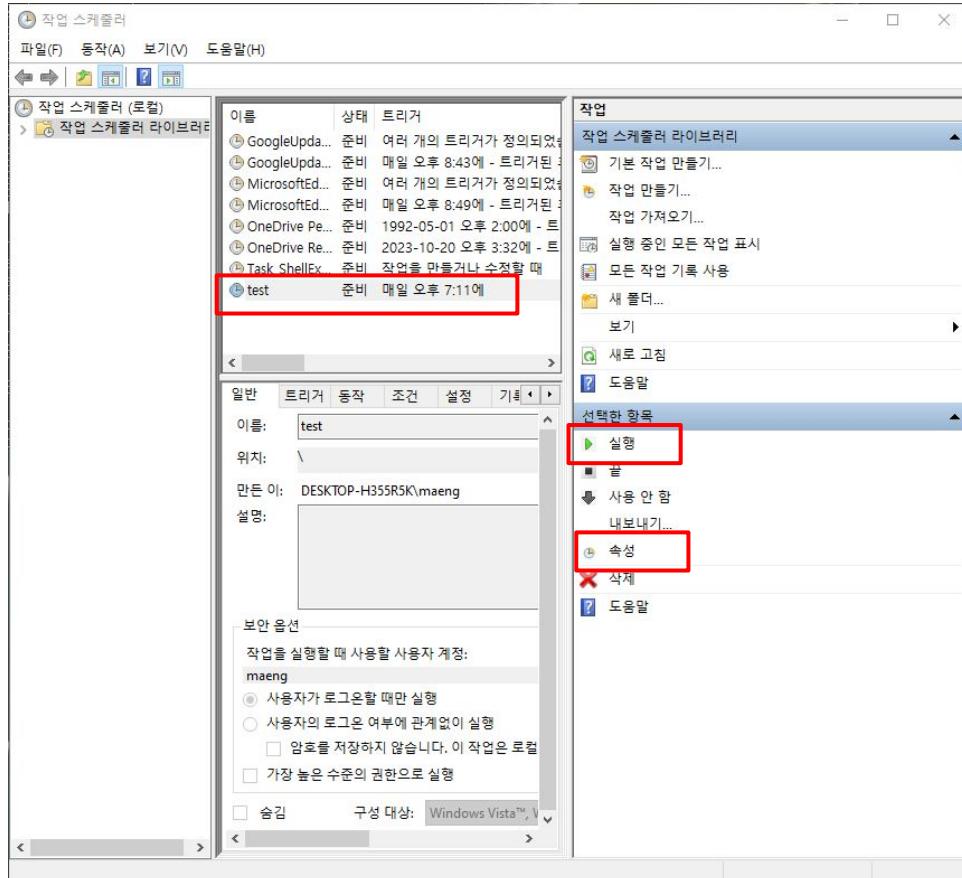
윈도우 작업 스케줄러

EST



윈도우 작업 스케줄러

EST

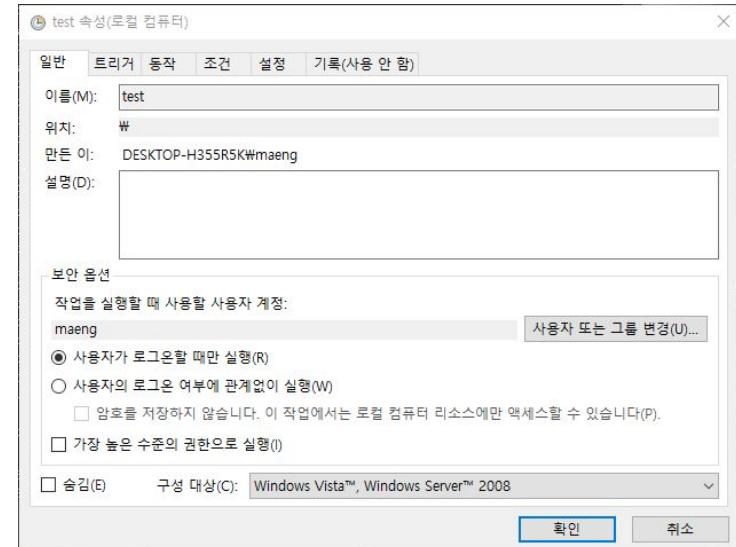


예약된 시간이 되기 전까지 준비상태로 표시

→ 예약 시간이 되면 실행됨

실행 버튼을 통해 테스트 실행해보자

속성에서는 작업 설정을 할 수 있다.



API 연동

공공데이터포털 API

네이버 검색-지식인API

~~오픈AI chatGPT API~~

API (Application Programming Interface)

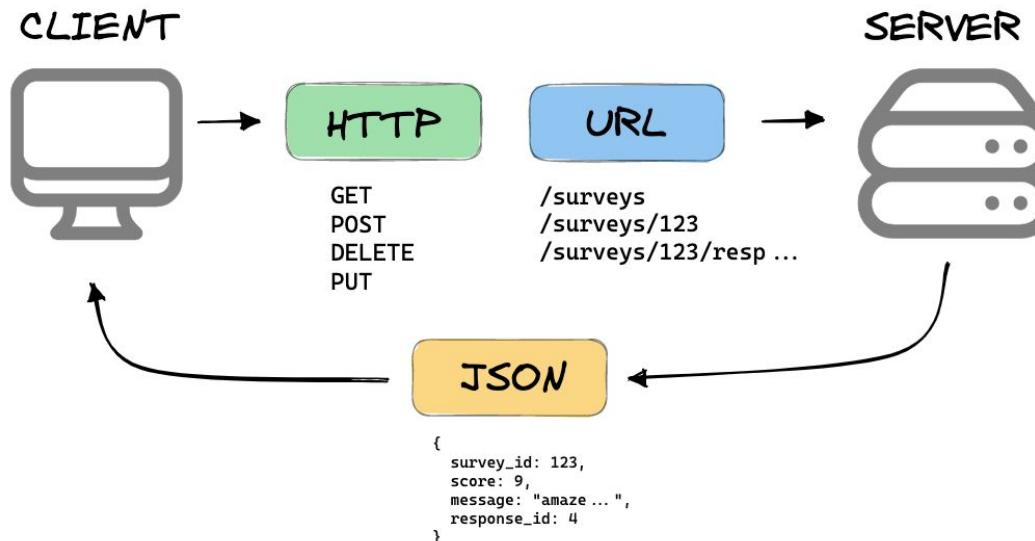
EST

API는 컴퓨터나 컴퓨터 프로그램 사이의 연결이다. 일종의 소프트웨어 인터페이스이며 다른 종류의 소프트웨어에 서비스를 제공한다.

이러한 연결이나 인터페이스를 빌드하거나 사용하는 방법을 기술하는 문서나 표준 API 규격으로 부른다.

이 표준을 충족하는 컴퓨터 시스템은 API가 구현(implement)되었다거나 노출(expose)되었다고 말한다.

컴퓨터와 인간을 연결시키는 사용자 인터페이스와 반대로, API는 컴퓨터나 소프트웨어를 서로 연결한다. 직접 사람(최종 사용자)에 의해 사용되도록 고안된 것이 아니며, 대신 소프트웨어에 이를 통합하고자 하는 컴퓨터 프로그래머가 사용하도록 고안되었다. API는 각기 다른 부분으로 구성되기도 하며 프로그래머가 사용할 수 있는 도구나 서비스의 역할을 한다. 이러한 부분들 중 하나를 사용하는 프로그램이나 프로그래머는 API의 해당 부분을 호출(call)한다고 말한다.



공공데이터포털 openAPI

EST

data.go.kr 회원가입 후 '실거래가'
검색

The screenshot shows the homepage of the data.go.kr portal. At the top, there is a navigation bar with links for '로그인' (Login), '회원가입' (Registration) (which is highlighted with a red box), '사이트맵' (Sitemap), and 'ENGLISH'. Below the navigation bar, there are several menu items: '데이터찾기' (Data Search), '국가데이터맵' (National Data Map), '데이터요청' (Data Request), '데이터활용' (Data Utilization), '정보공유' (Information Sharing), and '이용안내' (Usage Guide). The main content area features a search bar with the text '실거래가' (Real Estate Transaction) and a search button. Below the search bar, a dropdown menu is open, showing the option '4. 범죄' (Crime) under the heading '선택도움말' (Helpful Selection). The background has a blue gradient design.

이 누리집은 대한민국 공식 전자정부 누리집입니다.

로그인 회원가입 사이트맵 ENGLISH

DATA . GO . KR

데이터찾기 국가데이터맵 데이터요청 데이터활용 정보공유 이용안내

실거래가

선택도움말

4. 범죄

선택도움말

콘텐츠 바로가기

오픈API 활용신청

EST

- 검색 결과 중에서 ‘오픈API’ 탭을 클릭
- 국토교통부_아파트매매 실거래자료를 클릭
- 활용 신청 → 활용 목적은 기타(학습)으로 신청

The screenshot shows a search interface with several tabs at the top: '전체(43건)', '파일데이터(23건)', '오픈 API(20건)' (which is highlighted with a red box), and '표준데이터셋(0개)(0건)'. Below the tabs are two dropdown menus: '정확도순' and '10개씩', followed by a blue '정렬' button.

오픈 API (20건)

The screenshot shows a detailed view of an Open API listing. At the top, there are three buttons: '공공행정', '국가행정기관', and '국가중점' (highlighted with a red box). On the right, there is a '미리보기' button. Below the buttons, the title is 'XML 국토교통부_아파트매매 실거래 상세 자료' (highlighted with a red box). A sub-description states: '부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공'. At the bottom, there is information about the provider ('제공기관 국토교통부'), date ('수정일 2023-08-31'), views ('조회수 116500'), applications ('활용신청 17187'), keywords ('키워드 주택,아파트,실거래가'), and a '활용신청' button (highlighted with a red box).

활용 신청 현황

EST

이 누리집은 대한민국 공식 전자정부 누리집입니다.

로그아웃 마이페이지 사이트맵 ENGLISH

DATA .GO .KR 데이터찾기 국가데이터맵 데이터요청 데이터활용 정보공유 이용안내

홈 > 마이페이지 > 데이터 활용 > Open API > 활용신청 현황

마이페이지

활용신청 현황

- 데이터 활용
- Open API
- 활용신청 현황

인증키 발급현황

파일 데이터 >

관심 데이터

데이터 요청 >

나의 문의 >

회원정보 수정 >

공공행정 국토교통부

활용신청 [승인] 국토교통부_아파트매매 실거래 상세 자료

계정 개발 신청일 2021-10-25 만료예정일 2023-10-25

신청 0건	활용 9건	중지 0건
신청중인 단계	승인되어 활용중인 단계	중지신청하여 운영이 중지된 단계
보류 0건	변경신청 0건	
반려 0건		

마이페이지

데이터 활용 >

데이터 요청 >

나의 문의 >

회원정보 수정 >

개발계정 상세보기

기본정보

데이터명	국토교통부_아파트매매 실거래 상세 자료		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 활용신청	처리상태	승인
활용기간	2021-10-25 ~ 2023-10-25		

서비스정보

데이터포맷	XML
End Point	http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTradeDev?_wadl&type=xml

API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.

포털에서 제공되는 **Encoding/Decoding** 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.

* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.

일반 인증키 (Encoding)	QjITnZtxSg%52Bhz%2BWR8hYLMstCDRuf1REcb5E59648Wy77%2B7z8aQBHgv95yOhyoP31mFZWlyiqd2TrMu7HTuw%3D%3D
일반 인증키 (Decoding)	QjITnZtxSg+hz+WR8hYLMstCDRuf1REcb5E59648Wy77+7z8aQBHgv95yOhyoP31mFZWlyiqd2TrMu7HTuw==

openAPI 정보

EST

OpenAPI 정보

 메타데이터 다운로드

데이터 개선요청

분류체계	일반공공행정 - 일반행정	제공기관	국토교통부
관리부서명	거래신고관리부	관리부서 전화번호	053-663-8642
API 유형	REST	데이터포맷	XML
활용신청	17212	키워드	주택,아파트,실거래가
등록	2017-02-06	수정	2023-08-31
비용부과유무	무료	신청가능 트래픽	개발계정 : 1,000 / 운영계정 : 활용사례 래피 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 자동승인		
이용허락범위	이용허락범위 제한 없음		
참고문서	아파트 매매 상세자료 조회 기술문서.hwp		

가. API 서비스 개요

API 서비스 정보	API명(영문)	Apartment Transaction Detailed Data		
	API명(국문)	아파트 매매 신고 상세자료 조회		
	API 설명	지역코드와 기간을 설정하여 해당지역, 해당기간의 아파트 매매 상세자료를 제공하는 아파트 매매 상세자료 조회		
API 서비스 보안적용 기술 수준	서비스 인증/권한	[0] Service Key [] 인증서 (GPKI/NPKI) [] BASID(IP/PW) [] 없음		
	메시지 래밸 암호화	[] 전자서명 [] 암호화 [0] 없음		
	전송 래밸 암호화	[] SSL [0] 없음		
	인터페이스 표준	[] SOAP 1.2 (RPC-Encoded, Document Literal, Document Literal Wrapped) [0] REST (GET) [] RSS 1.0 [] RSS 2.0 [] ATOM 1.0 [] 기타		
	교환 데이터 표준 (증복선택기능)	[0] XML [] JSON [] MINE [] MTOM		
API 서비스 배포정보	서비스 URL	http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSObjSvc/getRTMSDataSvcAptTradeDev		
	서비스 명세 URL (WSDL 또는 WADL)	http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSObjSvc/getRTMSDataSvcAptTradeDev?_wadl&type=xml		
	서비스 버전	1.0		
	서비스 시작일	2016.12.01	서비스 배포일	2016.12.01
	서비스 미력			
	메시지 교환유형	[0] Request-Response [] Publish-Subscribe [] Fire-and-Forgot [] Notification		
	서비스 제공자	(총괄) 박성진 / 국토교통부 / 044-201-3592 (운영) 정가화 / 한국부동산원 / 053-663-8642		
	데이터 경선주기	일 1회		

요청변수(Request Parameter)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	ServiceKey	20	필수	-	공공데이터포털에서 받은 인증키
페이지 번호	pageNo	4	옵션	1	페이지번호
한 페이지 결과 수	numOfRows	4	옵션	10	한 페이지 결과 수
지역코드	LAWD_CD	5	필수	11110	지역코드
계약월	DEAL_YMD	6	필수	201512	계약월

출력결과(Response Element)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	-	-	-	-
결과메시지	resultMsg	샘플코드			
한 페이지 결과 수	numOfRows	Java	Javascript	C#	PHP
페이지 번호	pageNo	curl	Objective-C	Python	
전체 결과 수	totalCount				
거래금액	거래금액	import requests			
건축년도	건축년도	url = 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSObjSvc/getRTMSDataSvcAptTradeDev' params ={'serviceKey' : '서비스키', 'pageNo' : '1', 'numOfRows' : '10', 'LAWD_CD' : '11110', 'DEAL_YMD' : '201512'}			
년	년	response = requests.get(url, params=params) print(response.content)			
도로명	도로명				

요청 메시지 샘플

EST

```
http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrad  
eDev?LAWD_CD=11110&DEAL_YMD=201512&serviceKey=서비스키
```



```
http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrad  
eDev?LAWD_CD=종로구 지역코드&DEAL_YMD=계약월&serviceKey=인증키
```



```
http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrad  
eDev?LAWD_CD=11110&DEAL_YMD=201512&serviceKey=QjITnZtxSg5%2Bzh%2BWR8hYLMstCDRuf1REcb5E5964  
8Wy77%2B7z8aQBHgv95ylOhyoP31mFZWlyiqd2TrMu7HTuw%3D%3D
```

국토교통부 법정동 코드 : <https://www.data.go.kr/data/15123287/fileData.do>

응답 메시지 샘플

EST

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <header>
        <resultCode>00</resultCode>
        <resultMsg>NORMAL SERVICE.</resultMsg>
    </header>
    <body>
        <items>
            <item>
                <거래금액> 82,500</거래금액>
                <거래유형> </거래유형>
                <건축년도>2008</건축년도>
                <년>2015</년>
                <도로명>사직로 8길</도로명>
                <도로명건물번호코드>00004</도로명건물번호코드>
                <도로명건물번호코드>00000</도로명건물번호코드>
                <도로명시군구코드>11110</도로명시군구코드>
                <도로명일련번호코드>03</도로명일련번호코드>
                <도로명지상자호코드>0</도로명지상자호코드>
                <도로명코드>4100135</도로명코드>
                <등기일자> </등기일자>
                <번정동> 사직동</번정동>
                <번정동분번코드>0009</번정동분번코드>
                <번정동부번코드>0000</번정동부번코드>
                <번정동시군구코드>11110</번정동시군구코드>
                <번정동읍면동코드>11500</번정동읍면동코드>
                <번정동지번코드>1</번정동지번코드>
                <아파트>광화문스페이스본(101동~105동)</아파트>
                <월>12</월>
                <일>10</일>
                <일련번호>11110-2203</일련번호>
                <전용면적>94.51</전용면적>
                <중개사소재지> </중개사소재지>
                <지번>9</지번>
                <지역코드>11110</지역코드>
                <층>11</층>
                <해제사유발생일> </해제사유발생일>
                <해제여부> </해제여부>
            </item>
            <item>
                <거래금액> 60,000</거래금액>
                <거래유형> </거래유형>
                <건축년도>1991</건축년도>
            </item>
        </items>
    </body>
</response>
```

실습) 아파트매매 실거래 자료 수집(API연동 및 추출)

EST

```
import requests
from bs4 import BeautifulSoup

url = 'http://openapi.molit.go.kr:8081/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrade?serviceKey={}&LAMD_CD={}&DEAL_YMD={}'.format('LNuUIfsk5aIHzaYHafhrPhY4ASDZb%2FFn2o0hEktbxtabK5i5zHNyLtv8oNqq5NZi5XX9WvvmuKYdMczeXMhg%3D%3D', '11110', '201512')
res = requests.get(url)
text = res.text
#print(text)

soup = BeautifulSoup(text, 'lxml-xml')

items = soup.select('response > body > items > item')
l = []
for item in items:
    print(item.select_one('거래금액').text.strip(),
          item.select_one('건축년도').text,
          item.select_one('년').text,
          item.select_one('법정동').text.strip(),
          item.select_one('아파트').text,
          item.select_one('층').text,
          item.select_one('일').text,
          item.select_one('전용면적').text,
          item.select_one('지번').text,
          item.select_one('지번').text,
          item.select_one('지역코드').text,
          item.select_one('층').text)
```

실행 결과

82,500	2008	2015	사직동	광화문풀噤스페이스본(101동~105동)	12	10	94.51	9	9	11110	11
60,000	1981	2015	당주동	롯데미도파광화문빌딩	12	22	149.95	145	145	11110	8
130,000	2004	2015	내수동	킹스메니	12	8	194.43	110-15	110-15	11110	6
105,000	2004	2015	내수동	경희궁의아침2단지	12	14	124.17	71	71	11110	8
120,000	2003	2015	내수동	경희궁 파크팰리스	12	24	146.33	95	95	11110	4
17,000	2014	2015	연건동	이화에수를	12	17	16.98	195-10	195-10	11110	8
17,000	2014	2015	연건동	이화에수를	12	18	16.98	195-10	195-10	11110	4
57,000	2006	2015	명륜1가	렉스빌	12	29	106.98	19	19	11110	3
44,000	1995	2015	명륜2가	아남1	12	1	84.8	4	4	11110	18
52,000	1995	2015	명륜2가	아남1	12	10	84.9	4	4	11110	12
49,800	1995	2015	명륜2가	아남1	12	19	84.8	4	4	11110	1
41,000	1999	2015	명륜2가	아남3	12	19	61.13	237	237	11110	7
41,000	1999	2015	장신동	두산	12	2	59.95	232	232	11110	9
39,900	2003	2015	장신동	창신이수	12	3	68.06	23-816	23-816	11110	8
.....

실습) 아파트매매 실거래 자료 수집(결측값 채우기)

EST

```
import requests
from bs4 import BeautifulSoup

def default_text(node, text):
    if node != None:
        return node.text.strip()
    else:
        return text

url = 'http://openapi.molit.go.kr:8081/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSV
c/getRTMSDataSvcAptTrade?serviceKey={}&LAND_CD={}&DEAL_YMD={}'.format('LNUUIfsk5aIHzaYHA
fhrPhy4ASDzb%2FFn2o0hEKtbxtabK5i5zHMyLtvV8oOnqq5NZisXX9WvvmuKYdMczeXMhg%30%3D',
'11110',
'201512')
res = requests.get(url)
text = res.text
#print(text)

soup = BeautifulSoup(text, 'lxml-xml')

items = soup.select('response > body > items > item')
l = []
for item in items:
    print(default_text(item.select_one('거래금액')), ''),
    default_text(item.select_one('건축년도')), '',
    default_text(item.select_one('년')), '',
    default_text(item.select_one('법정동')), '',
    default_text(item.select_one('아파트')), '',
    default_text(item.select_one('월')), '',
    default_text(item.select_one('일')), '',
    default_text(item.select_one('전용면적')), '',
    default_text(item.select_one('지번')), '',
    default_text(item.select_one('지번')), '',
    default_text(item.select_one('지역코드')), '',
    default_text(item.select_one('층')))
```

실행 결과

82,500	2008	2015	사직동	광화문풀림스페이스본(101동~105동)	12	10	94.51	9	9	11110	11
60,000	1981	2015	당주동	롯데미도파광화문빌딩	12	22	149.95	145	145	11110	8
130,000	2004	2015	내수동	킹스매너	12	8	194.43	110-15	110-15	11110	6
105,000	2004	2015	내수동	경희궁의아침2단지	12	14	124.17	71	71	11110	8
120,000	2003	2015	내수동	경희궁 파크팰리스	12	24	146.33	95	95	11110	4
17,000	2014	2015	연건동	이화에수를	12	17	16.98	195-10	195-10	11110	8
17,000	2014	2015	연건동	이화에수를	12	18	16.98	195-10	195-10	11110	4
57,000	2006	2015	명륜1가	렉스빌	12	29	106.98	19	19	11110	3
44,000	1995	2015	명륜2가	아남1	12	1	84.8	4	4	11110	18
52,000	1995	2015	명륜2가	아남1	12	10	84.9	4	4	11110	12
49,800	1995	2015	명륜2가	아남1	12	19	84.8	4	4	11110	1
41,000	1999	2015	명륜2가	아남3	12	19	61.13	237	237	11110	7
41,000	1999	2015	창신동	두산	12	2	59.95	232	232	11110	9
39,900	2003	2015	창신동	창신이수	12	3	68.06	23-816	23-816	11110	8

실습) 아파트매매 실거래 자료 수집(저장)

EST

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

def default_text(node, text):
    if node != None:
        return node.text.strip()
    else:
        return text

url = 'http://openapi.molit.go.kr:8081/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrade?serviceKey={}&LAND_CD={}&DEAL_YMD={}'.format('LNUUIfsk5aIHzaYHAfhrPhy4ASDZb%2FFn2o0hEKtbxtabK5i5zHMyLtvV8o0Nnqg5NZiSXX9WvvmuKYdMczeXMhg%3D%3D', '11110',
'201512')
res = requests.get(url)
text = res.text
#print(text)

soup = BeautifulSoup(text, 'lxml-xml')
```

```
items = soup.select('response > body > items > item')
l = []
for item in items:
    print(default_text(item.select_one('거래금액'), ''),
          default_text(item.select_one('건축년도'), ''),
          default_text(item.select_one('년'), ''),
          default_text(item.select_one('법정동'), ''),
          default_text(item.select_one('아파트'), ''),
          default_text(item.select_one('층'), ''),
          default_text(item.select_one('월'), ''),
          default_text(item.select_one('일'), ''),
          default_text(item.select_one('전용면적'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지역코드'), ''),
          default_text(item.select_one('층'), ''))

l.append([default_text(item.select_one('거래금액'), ''),
          default_text(item.select_one('건축년도'), ''),
          default_text(item.select_one('년'), ''),
          default_text(item.select_one('법정동'), ''),
          default_text(item.select_one('아파트'), ''),
          default_text(item.select_one('층'), ''),
          default_text(item.select_one('월'), ''),
          default_text(item.select_one('일'), ''),
          default_text(item.select_one('전용면적'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지역코드'), ''),
          default_text(item.select_one('층'), '')])

df = pd.DataFrame(l, columns=['거래금액',
                               '건축년도',
                               '년',
                               '법정동',
                               '아파트',
                               '층',
                               '월',
                               '일',
                               '전용면적',
                               '지번',
                               '지번',
                               '지역코드',
                               '층'])

df.to_csv('아파트매매 실거래자료.csv', index=False, encoding='cp949')
```

실습) 아파트매매 실거래 자료 수집(여러달 수집 저장) EST

```
import requests
from bs4 import BeautifulSoup
import datetime
import dateutil
import pandas as pd

def default_text(node, text):
    if node != None:
        return node.text.strip()
    else:
        return text

current_datetime = datetime.datetime(2020, 1, 1)
l = []
for i in range(30):

    date = current_datetime.strftime('%Y%m') #20180220
    #print(i)
    #print(date) #201201

    url = 'http://openapi.molit.go.kr:8081/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTrade?serviceKey={}&LAND_CD={}&DEAL_YMD={}'.format('LNUUIfsk5aIHzaYHAfhrPhY4ASDzb%2FFn2o0hEKtbxtabK5i5zHMyLtvV8oOnqq5NZiSXX9WvvmuKYdMczeXMhg%3D%3D', '11110', date)
    res = requests.get(url)
    text = res.text
    #print(text)

    soup = BeautifulSoup(text, 'lxml-xml')
```

```
items = soup.select('response > body > items > item')
for item in items:
    print(default_text(item.select_one('거래금액')), ''),
    default_text(item.select_one('건축년도'), ''),
    default_text(item.select_one('년'), ''),
    default_text(item.select_one('법정동'), ''),
    default_text(item.select_one('아파트'), ''),
    default_text(item.select_one('풀'), ''),
    default_text(item.select_one('월'), ''),
    default_text(item.select_one('전용면적'), ''),
    default_text(item.select_one('지번'), ''),
    default_text(item.select_one('지번'), ''),
    default_text(item.select_one('지역코드'), ''),
    default_text(item.select_one('층'), ''))

    l.append([default_text(item.select_one('거래금액')), '',
              default_text(item.select_one('건축년도')), '',
              default_text(item.select_one('년')), '',
              default_text(item.select_one('법정동')), '',
              default_text(item.select_one('아파트')), '',
              default_text(item.select_one('풀')), '',
              default_text(item.select_one('월')), '',
              default_text(item.select_one('전용면적')), '',
              default_text(item.select_one('지번')), '',
              default_text(item.select_one('지번'), ''),
              default_text(item.select_one('지역코드'), ''),
              default_text(item.select_one('층'), '')])

    if date >= '202007':
        break;

current_datetime = current_datetime + dateutil.relativedelta.relativedelta(months=1)

df = pd.DataFrame(l, columns=['거래금액',
                               '건축년도',
                               '년',
                               '법정동',
                               '아파트',
                               '풀',
                               '월',
                               '전용면적',
                               '지번',
                               '지번',
                               '지역코드',
                               '층'])

df.to_csv('아파트매매 실거래자료.csv', index=False, encoding='cp949')
```

실습) 아파트매매 실거래 자료 수집(여러지역 저장)

EST

```
import requests
from bs4 import BeautifulSoup
import datetime
import dateutil
import pandas as pd

def default_text(node, text):
    if node != None:
        return node.text.strip()
    else:
        return text

lawd_cds = [11110, 11140, 11170, 11200, 11215, 11230, 11260, 11290, 11305,
           11320, 11350, 11380, 11410, 11440, 11470, 11500, 11530, 11545,
           11560, 11590, 11620, 11650, 11680, 11710, 11740]
l = []
for lawd_cd in lawd_cds:
    current_datetime = datetime.datetime(2020, 1, 1)
    for i in range(30):

        date = current_datetime.strftime('%Y%m') #20180220
        #print(i)
        #print(date) #201201

        url = 'http://openapi.molit.go.kr:8081/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcsAptTrade?serviceKey={}&LAWD_CD={}&DEAL_YMD={}'.format('LNUUIf5k9aIHzaYHafhrPhy4ASDzb%2Fn200hEKTbxtabK5i5zHNyLtVv800nqqSNZiSX9WlvvmuKYdMczeXlhg%30%3D', lawd_cd, date)
        res = requests.get(url)
        text = res.text
        #print(text)

        soup = BeautifulSoup(text, 'lxml-xml')
```

```
items = soup.select('response > body > items > item')
for item in items:
    print(default_text(item.select_one('거래금액'), ''),
          default_text(item.select_one('건축년도'), ''),
          default_text(item.select_one('년'), ''),
          default_text(item.select_one('법정동'), ''),
          default_text(item.select_one('아파트'), ''),
          default_text(item.select_one('풀'), ''),
          default_text(item.select_one('월'), ''),
          default_text(item.select_one('전용면적'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지번'), ''),
          default_text(item.select_one('지역코드'), ''),
          default_text(item.select_one('층'), ''))

    l.append([default_text(item.select_one('거래금액'), ''),
              default_text(item.select_one('건축년도'), ''),
              default_text(item.select_one('년'), ''),
              default_text(item.select_one('법정동'), ''),
              default_text(item.select_one('아파트'), ''),
              default_text(item.select_one('풀'), ''),
              default_text(item.select_one('월'), ''),
              default_text(item.select_one('전용면적'), ''),
              default_text(item.select_one('지번'), ''),
              default_text(item.select_one('지번'), ''),
              default_text(item.select_one('지역코드'), ''),
              default_text(item.select_one('층'), '')])

if date >= '202007':
    break

current_datetime = current_datetime + dateutil.relativedelta.relativedelta(months=1)

df = pd.DataFrame(l, columns=['거래금액',
                               '건축년도',
                               '년',
                               '법정동',
                               '아파트',
                               '풀',
                               '월',
                               '전용면적',
                               '지번',
                               '지번',
                               '지역코드',
                               '층'])

df.to_csv('아파트매매 실거래자료.csv', index=False, encoding='cp949')
```

네이버 지식인 API

EST

네이버 openapi : 네이버 통합검색

search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=1&ie=utf8&query=네이버+openapi

네이버 openapi

통합 VIEW 이미지 지식IN 인플루언서 동영상 쇼핑 뉴스 어학사전 지도 ...

로그인 공유

developers.naver.com

네이버 개발자센터

API 소개 · 공지사항

네이버 오픈 API들을 활용해 개발자들이 다양한 애플리케이션을 개발할 수 있도록 API 가이드와 SDK를 제공합니다. 제공중인 오픈 API에는 네이버 로그인, 검색, 단축URL, 캡차를 비롯 기계번역, 음성인식, 음성합성 등이 있습니다.

API 공통 가이드 - Open API 가이드 - 네이버 개발자센터

API 공통 가이드 네이버 오픈API는 네이버 플랫폼의 기능을 외부 개발자가 쉽게 이용할 수 있게 웹...

네이버 오픈API 종류 - Open API 가이드

네이버 오픈API 종류 네이버 오픈API는 인증 여부에 따라 로그인 방식 오픈 API와 비로그인 방식...

관련문서 더보기 >

ehpub.co.kr > 30-파이썬을-이용한-naver-open-api-활용하기-시작하기도서-검색

40. 파이썬을 이용한 Naver Open API 활용하기 – 시작하기(도서 ...

네이버 개발자 센터에서 제공하는 Open API는 XML 출력포맷 혹은 JSON 출력포맷으로 결과를 제공하고 있는데 이번 게시글에서는 json 출력포맷을 이용할 것입니다. 요청 URL 주소를 기록해 두세요. OpenAPI 서비스를 사용할 때는 요청 URL과 요청...

코로나19 현황

신속항원검사

당장 떠나고 싶다면, 국내 여행부터!

2023년 첫 해외여행 어디로?

너와 나의 검색 결과가 다른 이유

네이버 도착보장

https://developers.naver.com

네이버 지식인 API

EST

The screenshot shows the NAVER Developers website at developers.naver.com/main/. The 'Products' tab is highlighted with a red box. The main content area displays several API categories:

API 이용 안내	CLOVA	네이버 로그인	파파고	서비스 API
API 소개		네이버 로그인 API	Papago 번역	검색
운영 정책		카페	언어 감지	도움말 API
FAQ		캘린더	한글인명-로마자 변환	캡차
BI 가이드		네이버페이 배송지 정보		네이버 공유하기
이용약관		적용 가이드		모바일 앱 연동
상표 사용 가이드		적용 사례		네이버 오픈메인

A modal window titled "검색 결과" (Search results) is open over the "검색" (Search) button in the Services API section. The modal contains the following text:

Wordpress부터 Pinpoint까지, 네이버 클라우드 플랫폼에 마련된
다양한 오픈소스로 개발 시간을 절약해보세요!

The URL in the browser's address bar is <https://developers.naver.com/products/login/bestpractice/bestpractice.md>.

네이버 openapi : 네이버 통합검색 x | 검색 - SERVICE-API x +
developers.naver.com/products/service-api/search/search.md

NAVER Developers Products Documents Application NAVER D2 Support Forum API 상태 Search Here

데이터랩
검색
단축URL
캡차
네이버 공유하기
모바일앱 연동
네이버 오픈메인

웹, 뉴스, 블로그 등 분야별 네이버 검색 결과를 웹 서비스 또는 모바일 앱에서 바로 보여 줄 수 있습니다. 또한 'OO역 맛집'과 같은 지역 검색을 할 수도 있으며, 부가적으로 성인검색어 판별과 한영키 오타 변환 기능을 이용하실 수 있습니다.

네이버 검색 결과 컨텐츠
웹 서비스 또는 모바일 앱에 네이버 웹문서/블로그/뉴스/책/영화/카페글/지식IN/쇼핑/이미지/백과/전/전문 자료 분야에 대한 검색 결과를 보여 줍니다.

지역 검색
'OO역 맛집', 'OO동 솔직'과 같은 검색 결과를 보여주고 싶을 때 사용하며, 네이버 지역 서비스에 등록된 각 지역별 업체 및 상호 검색결과를 보여줍니다.

검색 부가 기능
검색 부가 기능으로 특정 검색어에 대해 성인검색어 여부를 알려주는 기능과 검색창에 입력된 오타를 바로 잡아주는 오타변환 기능을 제공합니다.



네이버 지식인 API

EST

네이버 openapi : 네이버 통합검색 × 검색 - SERVICE-API × +

developers.naver.com/products/service-api/search/search.md

NAVER Developers Products Documents Application NAVER D2 Support Forum API 상태 Search Here

검색 부가 기능

검색 부가 기능으로 특정 검색어에 대해 성인검색어 여부를 알려주는 기능과 검색창에 입력된 오타를 바로 잡아주는 오타변환 기능을 제공합니다.

* 처리한도 : 25,000/일

데이터랩
검색
단축URL
캡차
네이버 공유하기
모바일앱 연동
네이버 오픈메인

↑

오픈 API 이용 신청 개발 가이드 보기

이용약관 | 개인정보처리방침 | 제휴신청 | 개발자포럼 | 개발자채용
NAVER Copyright © NAVER Corp. All Rights Reserved.

<https://developers.naver.com/apps/#/register?defaultScope=search>

네이버 지식인 API

EST

애플리케이션 이름 ↗ ✓

- 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.
- 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/", "-", "_", 만 입력 가능합니다.

선택하세요. ▾ ✓

사용 API ↗

검색 X

환경 추가 ▾

WEB 설정 X ^

웹 서비스 URL (최대 10개)

비로그인 오픈 API
서비스 환경

http://localhost + ✓

- 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
- http와 https는 구분하지 않습니다.
- www는 빼고 입력해 주세요. 예) http://naver.com
- 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
- 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file:///로컬URI)

NAVER Developers Products **Documents** Application NAVER D2 Support Forum API 상태 Search Here

API 공통 가이드 CLOVA 네이버 로그인 파파고 서비스 API

개요 Papago 번역 데이터랩
검색 가이드 언어 감지 검색
개발 가이드 한글인명-로마자 변환 단축 URL
API 명세 튜토리얼 이미지캡차
튜토리얼 SDK 다운로드 음성캡차
SDK 다운로드 네이버 공유하기
네이버 앱 연동 네이버 오픈메인

블로그
뉴스
책
성인 검색어 판별
백과사전
영화
카페글
지식IN []
지역
오타변환
웹문서
이미지
쇼핑
전문자료

<https://openapi.naver.com/v1/search/kin.json> JSON

프로토콜 [🔗](#)
HTTPS
HTTP 메서드 [🔗](#)
GET

파라미터 [🔗](#)
파라미터를 쿼리 스트링 형식으로 전달합니다.

파라미터	타입	필수 여부	설명
query	String	Y	검색어. UTF-8로 인코딩되어야 합니다.
display	Integer	N	한 번에 표시할 검색 결과 개수(기본값: 10, 최댓값: 100)
start	Integer	N	검색 시작 위치(기본값: 1, 최댓값: 100)

네이버 지식인 API

EST

응답 규칙

요소	타입	설명
rss	-	RSS 컨테이너. RSS 리더기를 사용해 검색 결과를 확인할 수 있습니다.
rss/channel	-	검색 결과를 포함하는 컨테이너. channel 요소의 하위 요소인 title, link, description은 RSS에서 사용하는 정보이며, 검색 결과와는 상관이 없습니다.
rss/channel/lastBuildDate	dateTime	검색 결과를 생성한 시간
rss/channel/total	Integer	총 검색 결과 개수
rss/channel/start	Integer	검색 시작 위치
rss/channel/display	Integer	한 번에 표시할 검색 결과 개수
rss/channel/item	-	개별 검색 결과. JSON 형식의 결과값에서는 items 속성의 JSON 배열로 개별 검색 결과를 반환합니다.
rss/channel/item/title	String	지식IN 질문 제목. 제목에서 검색어와 일치하는 부분은 태그로 감싸져 있습니다.
rss/channel/item/link	String	지식IN 질문의 URL
rss/channel/item/description	String	지식IN 질문 내용을 요약한 패시지 정보. 패시지 정보에서 검색어와 일치하는 부분은 태그로 감싸져 있습니다.

NAVER Developers	Products	Documents	Application	NAVER D2	Support	Forum	API 상태	Search Here	User icon
블로그		SE04	400	Invalid sort value (부적절한 sort 값입니다.)			sort 파라미터의 값에 오타가 있는지 확인합니다.		
뉴스		SE06	400	Malformed encoding (잘못된 형식의 인코딩입니다.)			검색어를 UTF-8로 인코딩합니다.		
책									
성인 검색어 판별		SE05	404	Invalid search api (존재하지 않는 검색 api입니다.)			API 요청 URL에 오타가 있는지 확인합니다.		
백과사전									
영화		SE99	500	System Error (시스템 에러)			서버 내부에 오류가 발생했습니다. "개발자 포럼"에 오류를 신고해 주십시오.		
카페글									
지식iN									
지역									
오타변환									
웹문서									
이미지									
쇼핑									
전문자료									
							403 오류		
							개발자 센터에 등록한 애플리케이션에서 검색 API를 사용하도록 설정하지 않았다면 'API 권한 없음'을 의미하는 403 오류가 발생할 수 있습니다. 403 오류가 발생했다면 네이버 개발자 센터의 Application > 내 애플리케이션 메뉴에서 오류가 발생한 애플리케이션의 API 설정 탭을 클릭한 다음 검색이 선택돼 있는지 확인해 보십시오.		
							참고		
							네이버 오픈API 공통 오류 코드는 " API 공통 가이드 "의 ' 오류 코드 '를 참고하십시오.		
							검색 API 지식iN 검색 구현 예제		
							검색 API로 지식iN 검색 결과를 조회하는 방법은 블로그 검색 결과를 조회하는 방법과 유사합니다. 지식iN 검색 결과 조회를 구현하는 방법은 검색 API 블로그 검색 구현 예제 를 참고합니다.		

[블로그](#)[뉴스](#)[책](#)[성인 검색어 판별](#)[백과사전](#)[영화](#)[카페글](#)[지식iN](#)[지역](#)[오타변환](#)[웹문서](#)[이미지](#)

Python

```
# 네이버 검색 API 예제 - 블로그 검색
import os
import sys
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
encText = urllib.parse.quote("검색할 단어")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # JSON 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # XML 결과
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

파이썬 예제

네이버 지식인 API

EST

NAVER Developers

Products

Documents

Application

NAVER D2

Support

Forum

API 상태

Search Here



Application 목록

Application 등록

Client ID	Application 명	Action
3Ba7woBMp4X0FH6YGvwp	파워비테스트	
AVSJwAUo3JP_Gj9QeErX	교육	
hymiHM8XqY0kadq7wRe1	search_image	

search_image

개요

API 설정

멤버관리

로그인 통계

API 통계

Playground(Beta)

애플리케이션 정보

Client ID

hymiHM8XqY0kadq7wRe1

Client Secret

.....

보기

네이버 지식인 API

EST

```
In [55]: import requests, time, os, json
from html import unescape

executed in 4ms, finished 12:31:22 2023-10-23

In [56]: # input
client_id = 'xVpqDY_a8tcrDFKg7LuZ'
client_secret = 'ND0tbLui7x'

queries = ['전주 여행', '경주 여행']
goal_page = 5

executed in 3ms, finished 12:31:23 2023-10-23

In [57]: # setting
user_agent = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36"

headers = {"User-Agent": user_agent,
           "X-Naver-Client-Id": client_id,
           "X-Naver-Client-Secret": client_secret}

executed in 4ms, finished 12:31:23 2023-10-23

In [58]: file_name = './sample/naver_kin.txt'

with open(file_name, 'w', encoding='utf-8') as f :
    f.write('query#\tno#\ttitle#\tlink#\tdescription#\ttotal_text#\n')

executed in 5ms, finished 12:31:24 2023-10-23
```

네이버 지식인 API

EST

```
In [64]: def get_list(query, page):
    print('='*5, query, page, '='*5)
    url = "https://openapi.naver.com/v1/search/kin.json?display=100&query=" + query + "&start=" + str((page-1)*100+1)
    response = requests.get(url, headers=headers)
    elements = json.loads(response.text)['items']

    for i, elm in enumerate(elements):
        title = elm['title'].replace("<b>", "").replace("</b>", "")
        title = unescape(title) # escape문자를 unescape문자로 변경
        link = elm['link']
        description = elm['description'].replace("<b>", "").replace("</b>", "")
        description = unescape(description)

        print([query, (page*100)+(i+1), title, link, description, title+" "+description])

        with open(file_name, 'a', encoding='utf-8') as f: # overwrite 완료도록 add할 것
            f.write( f'{query}\t{((page*100)+(i+1))}\t{title}\t{link}\t{description}\t{title+')

    return
```

executed in 7ms, finished 12:32:46 2023-10-23

```
In [65]: for query in queries:
    for page in range(goal_page):
        kin_list = get_list(query, page)
        time.sleep(0.5) #웹페이지 크롤링 매너 최소 0.5초
```

executed in 12.4s, finished 12:33:00 2023-10-23

9&dirId=90111&docId=455851613&qb=6rK97K0810yXr02WlQ==&enc=utf8§ion=kin_qna&rank=5

&search_sort=0&spq=0', '경주 여행 가려고 하는데... 11월 초에나 갈 것 같아요. 그쯤 되

면 경주에 가기 좋은 계획은 어떤가요? 그 다음에 경주 여행 계획을 짜고, 나중에 여행 일정을 정하고, 그 이후에는 경주 여행 일정을 정하는 과정입니다.

APPENDIX

OpenAI ChatGPT API

동기와 비동기

OpenAI chatGPT API

EST

chatGPT API 접속

chatgpt api

전체 뉴스 이미지 동영상 도서 더보기 도구

검색결과 약 159,000,000개 (0.24초)

openai.com
https://openai.com › blog › introducing-chatgpt-and-... :

Introducing ChatGPT and Whisper APIs - OpenAI

2023. 3. 1. — ChatGPT and Whisper models are now available on our API, giving developers access to cutting-edge language (not just chat!) and speech-to-text ...

OpenAI Research Product Developers Safety Company

ChatGPT and Whisper models are now available on our API, giving developers access to cutting-edge language (not just chat!) and speech-to-text capabilities. Through a series of system-wide optimizations, we've achieved 90% cost reduction for ChatGPT since December; we're now passing those savings to API users. Developers can now use our open-source Whisper large-v2 model in the API with much faster and cost-effective results. ChatGPT API users can expect continuous model improvements and the option to choose dedicated capacity for deeper control over the models. We've also listened closely to feedback from our developers and refined our API terms of service to better meet their needs.

Get started ↗

Create your account

Please note that phone verification is required for signup. Your number will only be used to verify your identity for security purposes.

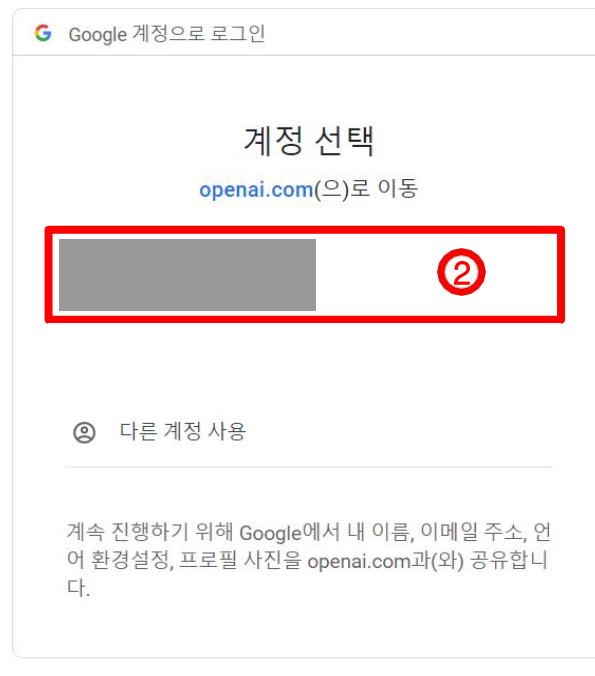
Continue

Already have an account? [Log in](#)

OR

 Continue with Google ①

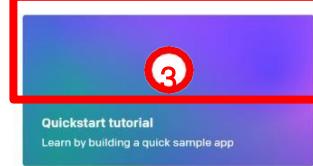
 Continue with Microsoft Account



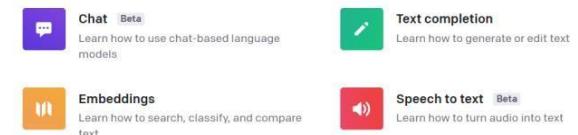
chatGPT API 가입

Welcome to OpenAI

Start with the basics



Build an application



<https://platform.openai.com/docs/guides/api>

To use a GPT model via the OpenAI API, you'll send a request containing the inputs and your API key, and receive a response containing the model's output. Our latest models, `gpt-4` and `gpt-3.5-turbo`, are accessed through the chat completions API endpoint.

	MODEL FAMILIES	API ENDPOINT
Newer models (2023–)	<code>gpt-4</code> , <code>gpt-3.5-turbo</code>	https://api.openai.com/v1/chat/completions
Updated base models (2023)	<code>babbage-002</code> , <code>davinci-002</code>	https://api.openai.com/v1/completions
Legacy models (2020–2022)	<code>text-davinci-003</code> , <code>text-davinci-002</code> , <code>davinci</code> , <code>curie</code> , <code>babbage</code> , <code>ada</code>	https://api.openai.com/v1/completions

You can experiment with GPTs in the [playground](#). If you're not sure which model to use, then use `gpt-3.5-turbo` or `gpt-4`.

<https://openai.com/pricing>

a

다양한 기능과 가격대를 지닌

다양한 모델.

가격은 1,000토큰당

가격입니다.

토큰은 단어 조각으로 생각할 수
있습니다. 여기서 1,000개의

토큰은 약 750개의 단어입니다.

이 단락은 35개의 토큰입니다.

GPT-4

With broad general knowledge and domain expertise, GPT-4 can follow complex instructions in natural language and solve difficult problems with accuracy.

[Learn about GPT-4](#)

Model	Input	Output
8K context	\$0.03 / 1K tokens	\$0.06 / 1K tokens
32K context	\$0.06 / 1K tokens	\$0.12 / 1K tokens

GPT-3.5 Turbo

GPT-3.5 Turbo models are capable and cost-effective.

`gpt-3.5-turbo` is the flagship model of this family and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

[Learn about GPT-3.5 Turbo ↗](#)

Model	Input	Output
4K context	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
16K context	\$0.003 / 1K tokens	\$0.004 / 1K tokens

①  Personal

② View API keys

③ + Create new secret key

④ aiotclass

Cancel Create secret key Done

API key 확인

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

You currently do not have any API keys. Please create one below.

!!! 다시 보여주지
않음

Create new secret key

Please save this secret key somewhere safe and accessible. For security reasons, **you won't be able to view it again** through your OpenAI account. If you lose this secret key, you'll need to generate a new one.

sk-OVpu9zqzRVE2T2WT84zIT3BlbkFJkqbMTMLL2xX7Nxqx19:



Done

OpenAI chatGPT API

EST

!pip install openai

```
# pip install --upgrade openai
!pip install openai

import os
from openai import OpenAI

client = OpenAI(
    api_key= "api key 입력"
)

for m in client.models.list():
    print(m)

Model(id='text-search-babbage-doc-001', created=1651172509, object='model', owned_by='openai-dev')
Model(id='curie-search-query', created=1651172509, object='model', owned_by='openai-dev')
Model(id='text-davinci-003', created=1669599635, object='model', owned_by='openai-internal')
Model(id='text-search-babbage-query-001', created=1651172509, object='model', owned_by='openai-dev')
Model(id='babbage', created=1649358449, object='model', owned_by='openai')
Model(id='babbage-search-query', created=1651172509, object='model', owned_by='openai-dev')
Model(id='text-babbage-001', created=1649364043, object='model', owned_by='openai')
```

OpenAI chatGPT API

EST

```
input_text = "인공지능이 뭐야"

client.chat.completions.create?

response = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "user",
         "content": input_text}
    ]
)

type(response)

openai.types.chat.chat_completion.ChatCompletion
```

```
response.choices[0].message.content
```

'인공지능은 컴퓨터 프로그램이나 시스템으로, 인간의 지능을 모방하거나 따라하는 기능을 가지고 있는 기술이다. 인공지능은 학습과 추론을 통해 문제를 해결하고 결정을 내리는 능력을 갖는다. 이를 위해 대량의 데이터를 분석하고 패턴을 파악하는 기능을 가지고 있으며, 특정한 작업을 수행하기 위한 알고리즘과 모델을 학습하여 문제를 해결한다. 인공지능은 이미지, 음성, 언어, 자율주행 등 다양한 분야에서 응용되고 있으며, 인간의 일상 생활에서도 널리 활용되고 있다.'

OpenAI chatGPT API

EST

```
def chatgpt(input_text):
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": input_text}]
    )
    output = response.choices[0].message.content
    return output
```

```
chatgpt('what is ai')
```

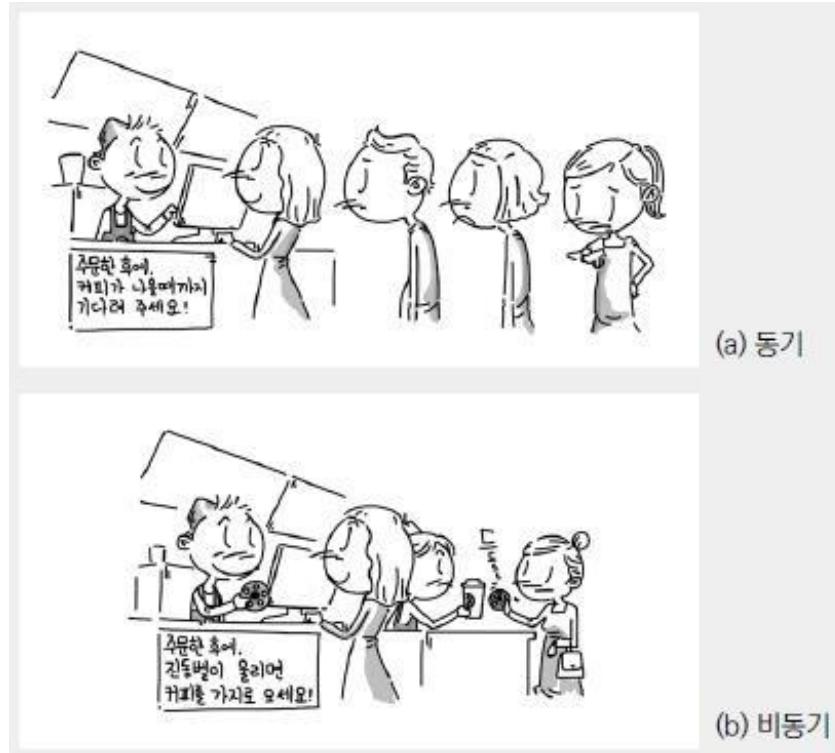
'AI stands for Artificial Intelligence. It refers to the simulation of human intelligence in machines that are programmed to think, learn, and problem-solve like humans. AI encompasses various techniques such as machine learning, natural language processing, computer vision, and robotics. It has applications in a wide range of fields including healthcare, finance, transportation, gaming, and more. AI systems can analyze large amounts of data, recognize patterns, make predictions, and automate tasks, leading to improved efficiency and decision-making.'

동기와 비동기 통신

EST

자바스크립트가 발전을 하면서,
Ajax(비동기 통신) 형태로 서버와
데이터를 주고 받아 화면에 뿌려주는
사이트가 많아 졌습니다.

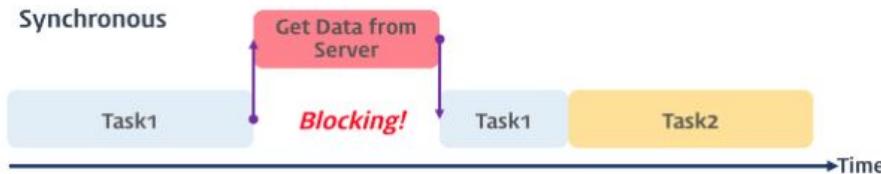
이러한 형식으로 데이터를 주고 받으면
url 변경이나 새로고침 없이 데이터를
가져오게 됩니다.



동기식 처리 모델(Synchronous processing model)은 직렬적으로 태스크(task)를 수행한다.

즉, 태스크는 순차적으로 실행되며 어떤 작업이 수행 중이면 다음 작업은 대기하게 된다.

예를 들어 서버에서 데이터를 가져와서 화면에 표시하는 작업을 수행할 때, 서버에 데이터를 요청하고 데이터가 응답될 때까지 이후 태스크들은 블로킹(blocking, 작업 중단)된다.



다음은 동기식으로 동작하는 코드로, 순차적으로 실행된다.

JAVASCRIPT

```
function func1() {  
    console.log('func1');  
    func2();  
}  
  
function func2() {  
    console.log('func2');  
    func3();  
}  
  
function func3() {  
    console.log('func3');  
}  
  
func1();
```

비동기식 처리 모델

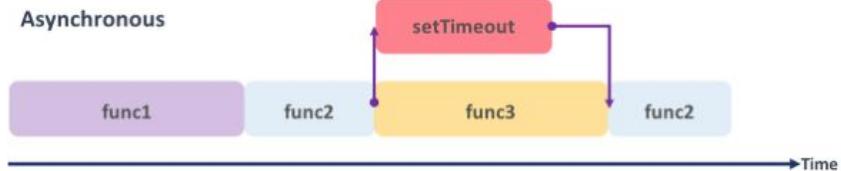
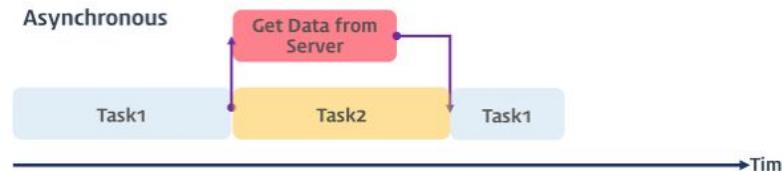
EST

비동기식 처리 모델(Asynchronous processing model 또는 Non-Blocking processing model)은 병렬적으로 태스크를 수행 한다.

즉, 태스크가 종료되지 않은 상태라 하더라도 대기하지 않고 다음 태스크를 실행한다.

예를 들어 서버에서 데이터를 가져와서 화면에 표시하는 태스크를 수행할 때, 서버에 데이터를 요청한 이후 서버로부터 데이터가 응답될 때까지 대기하지 않고(Non-Blocking) 즉시 다음 태스크를 수행한다. 이후 서버로부터 데이터가 응답되면 이벤트가 발생하고 이벤트 핸들러가 데이터를 가지고 수행할 태스크를 계속해 수행한다.

자바스크립트의 대부분의 DOM 이벤트와 Timer 함수(setTimeout, setInterval), Ajax 요청은 비동기식 처리 모델로 동작한다.

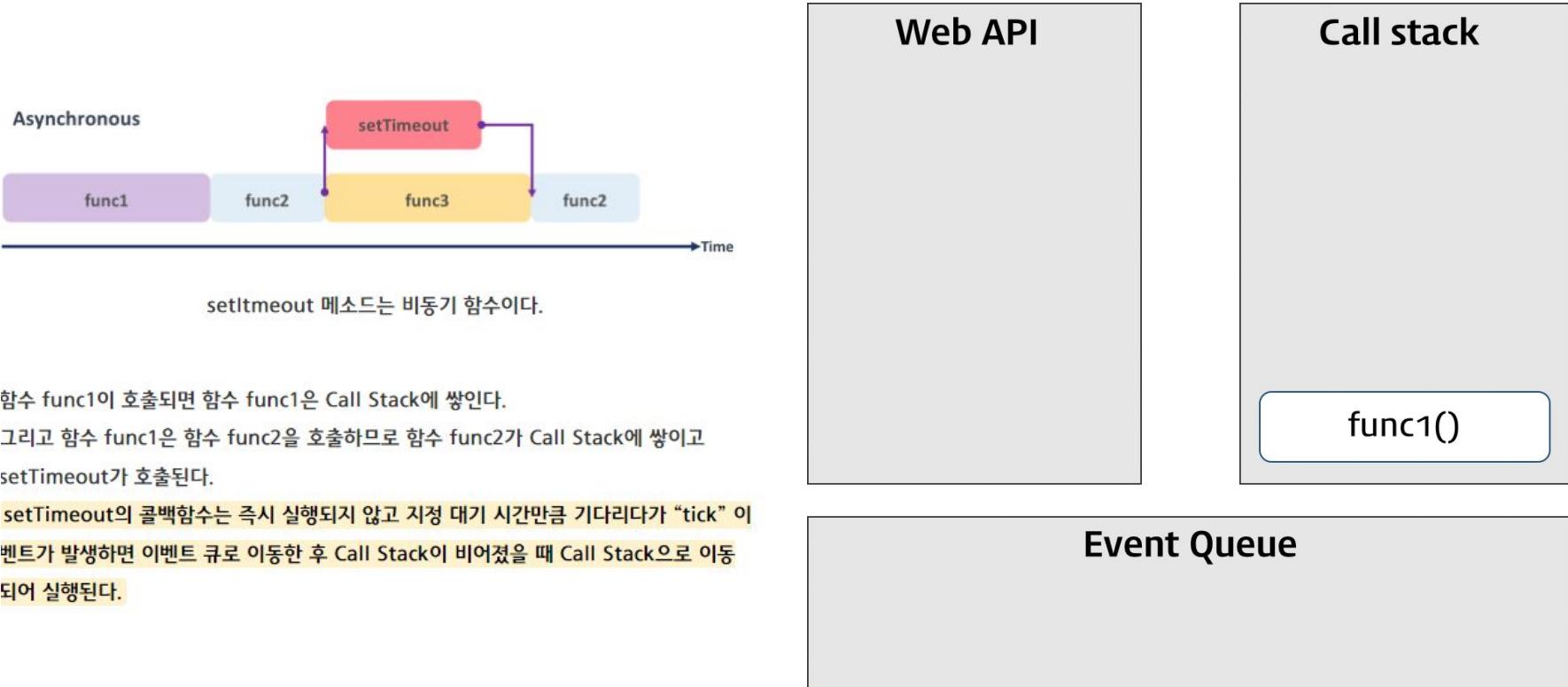


setTimeout 메소드는 비동기 함수이다.

함수 func1이 호출되면 함수 func1은 Call Stack에 쌓인다.

그리고 함수 func1은 함수 func2를 호출하므로 함수 func2가 Call Stack에 쌓이고 setTimeout가 호출된다.

setTimeout의 콜백함수는 즉시 실행되지 않고 지정 대기 시간만큼 기다리다가 “tick” 이벤트가 발생하면 이벤트 큐로 이동한 후 Call Stack이 비어졌을 때 Call Stack으로 이동되어 실행된다.



AJAX

Ajax(Asynchronous JavaScript and XML, 에이잭스)는 비동기적인 웹 애플리케이션의 제작을 위해 아래와 같은 조합을 이용하는 웹 개발 기법이다.

- 표현 정보를 위한 HTML (또는 XHTML) 과 CSS
- 동적인 화면 출력 및 표시 정보와의 상호작용을 위한 DOM, 자바스크립트
- 웹 서버와 비동기적으로 데이터를 교환하고 조작하기 위한 XML, XSLT, XMLHttpRequest (Ajax 애플리케이션은 XML/XSLT 대신 미리 정의된 HTML이나 일반 텍스트, JSON, JSON-RPC를 이용할 수 있다).

기존 기술과의 차이점 [편집]

기존의 웹 애플리케이션은 브라우저에서 폼을 채우고 이를 웹 서버로 제출(submit)을 하면 하나의 요청으로 웹 서버는 요청된 내용에 따라서 데이터를 가공하여 새로운 웹 페이지를 작성하고 응답으로 되돌려준다. 이때 최초에 폼을 가지고 있던 페이지와 사용자가 이 폼을 채워 결과물로써 되돌려 받은 페이지는 일반적으로 유사한 내용을 가지고 있는 경우가 많다. 결과적으로 중복되는 HTML 코드를 다시 한번 전송을 받음으로써 많은 대역폭을 낭비하게 된다. 대역폭의 낭비는 금전적 손실을 야기할 수 있으며 사용자와 대화(상호 반응)하는 서비스를 만들기 어렵게도 한다.

반면에 Ajax 애플리케이션은 필요한 데이터만을 웹서버에 요청해서 받은 후 클라이언트에서 데이터에 대한 처리를 할 수 있다. 보통 SOAP이나 XML 기반의 웹 서비스 프로토콜이 사용되며, 웹 서버의 응답을 처리하기 위해 클라이언트 쪽에서는 자바스크립트를 쓴다. 웹 서버에서 전적으로 처리되던 데이터 처리의 일부분이 클라이언트 쪽에서 처리되므로 웹 브라우저와 웹 서버 사이에 교환되는 데이터량과 웹서버의 데이터 처리량도 줄어들기 때문에 애플리케이션의 응답성이 좋아진다. 또한 웹서버의 데이터 처리에 대한 부하를 줄여주는 일이 요청을 주는 수많은 컴퓨터에 대해서 일어나기 때문에 전체적인 웹 서버 처리량도 줄어들게 된다.

