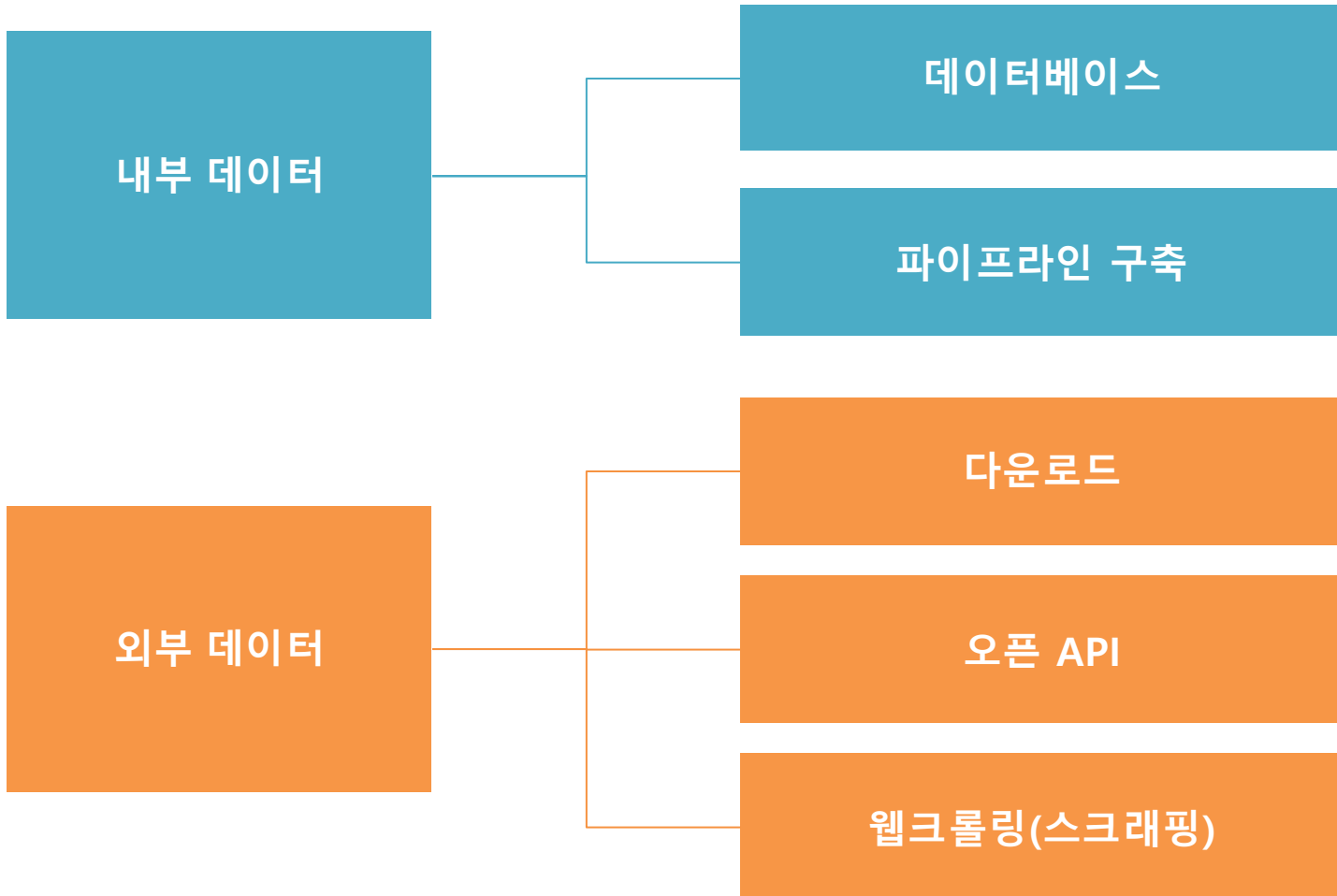


데이터 분석 기술

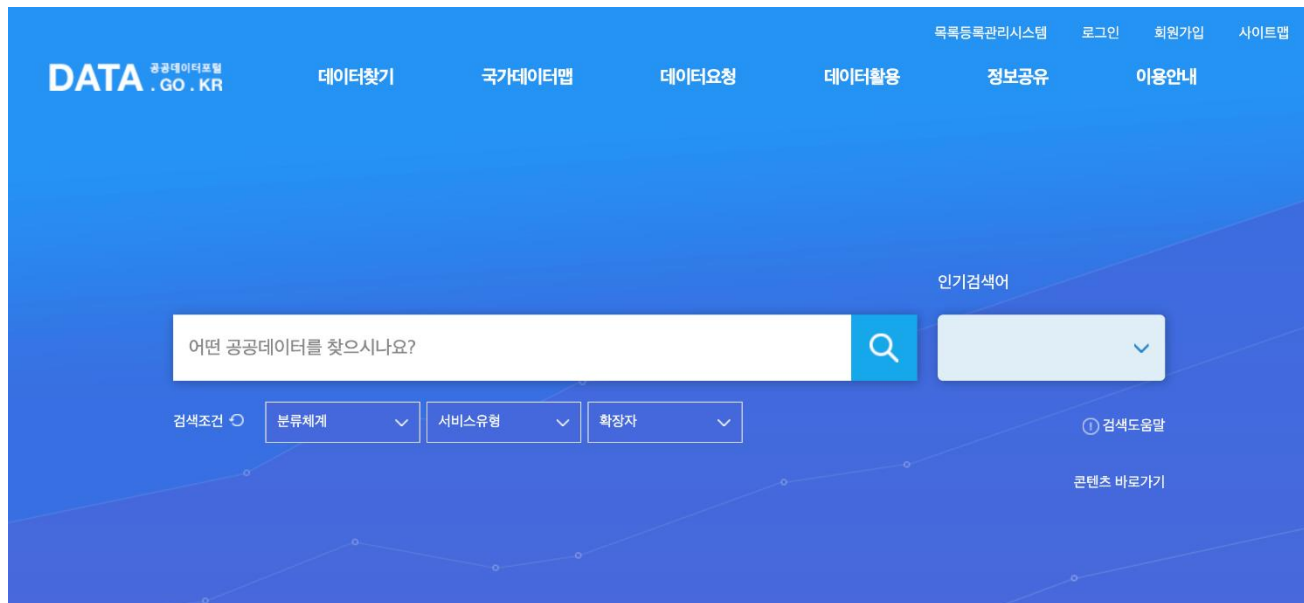
- 데이터 수집 기술



데이터 분석 기술

• 데이터 수집 – 1. 다운로드


- 공공데이터 포털(<https://www.data.go.kr>)
- 공공기관에서 생성 또는 취득하여 관리하는 공공데이터를 제공하는 사이트



데이터 분석 기술

• 데이터 수집 – 1. 다운로드

- 서울열린데이터광장(<http://data.seoul.go.kr/>)
- 서울시의 공공데이터를 제공하는 사이트

 서울 열린데이터 광장

공공데이터

통계

소식&참여

이용안내

[로그인](#) [회원가입](#) [사이트맵](#)

Open API 이용안내

[Home](#) > [이용안내](#) > [Open API 이용안내](#)

열린데이터광장에서 제공하는 오픈API를 사용하기 위해서는 먼저 인증키를 발급받으셔야 합니다.
오픈API는 다양한 서비스와 데이터를 좀 더 쉽게 이용할 수 있도록 공개한 개발자를 위한 인터페이스입니다.

Open API 이용방법



열린데이터광장 접속

01



Open API
인증키 신청

02



Open API
검색 / 확인

03



Open API 활용 /
애플리케이션 제작

04



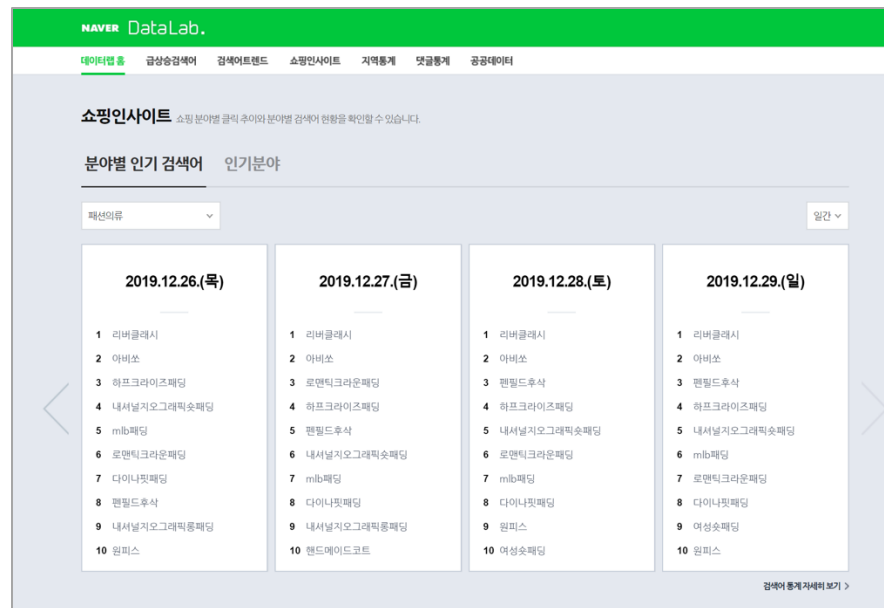
애플리케이션 등록

05

데이터 분석 기술

• 데이터 수집 – 1. 다운로드

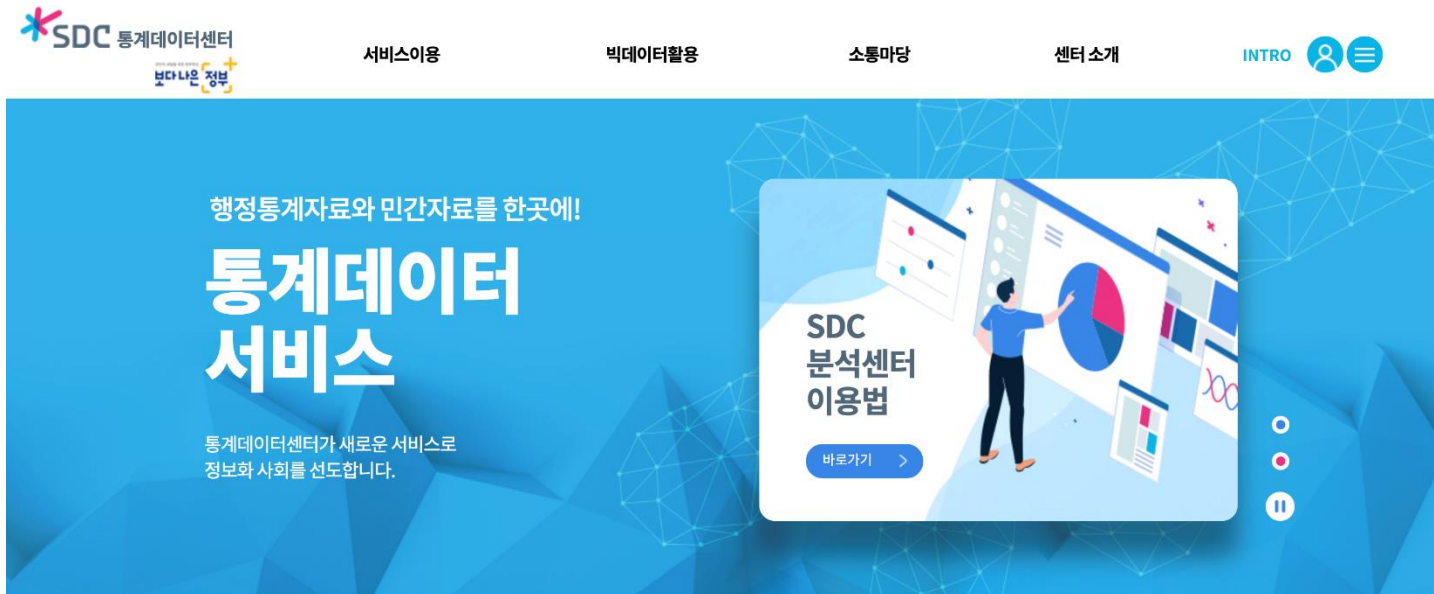
- 네이버데이터랩(<https://datalab.naver.com/>)
- 네이버 검색어, 쇼핑 등을 기반으로 한 통계 데이터를 제공하는 사이트



데이터 분석 기술

• 데이터 수집 – 1. 다운로드

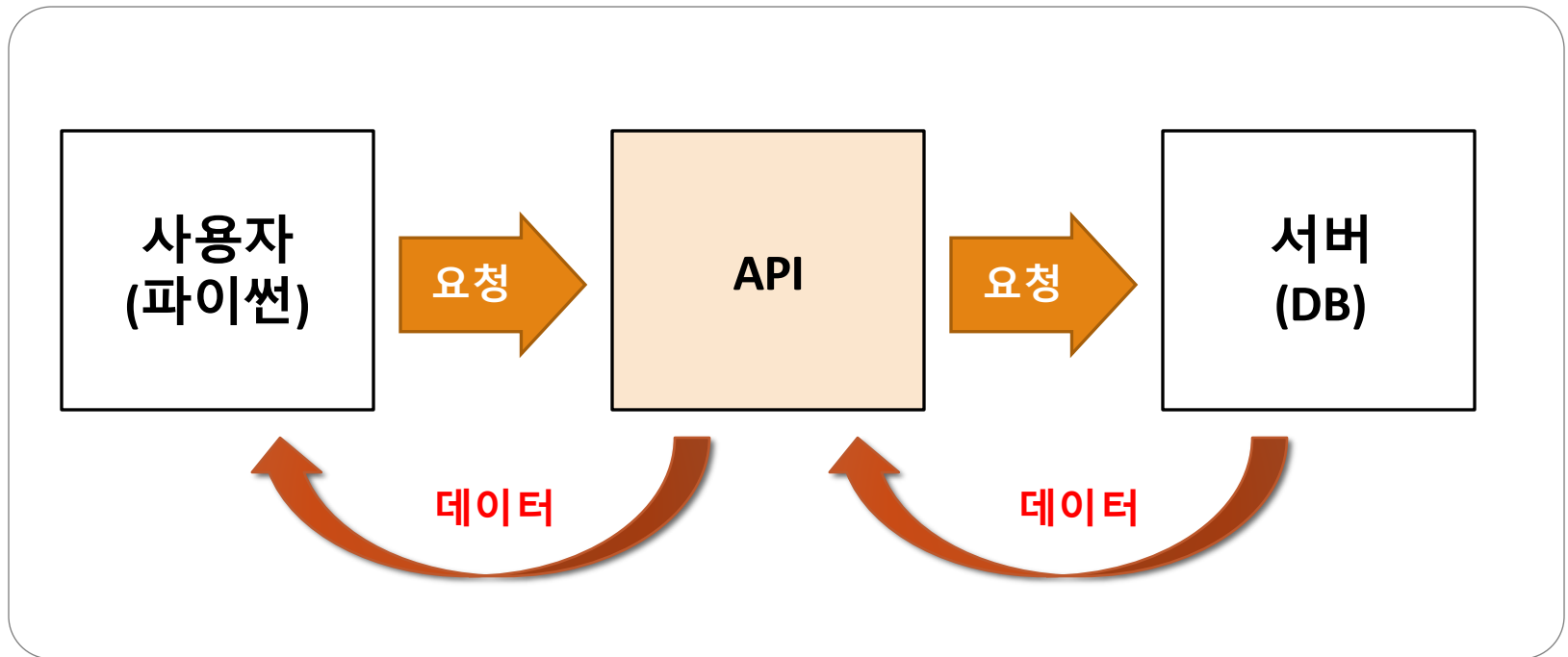
- 통계빅데이터센터(<http://data.kostat.go.kr/sbchome/index.do>)
- 통계자료 및 민간자료를 편리하게 이용할 수 있는 플랫폼



데이터 분석 기술

• 데이터 수집 – 2. API(Application Programming Interface)

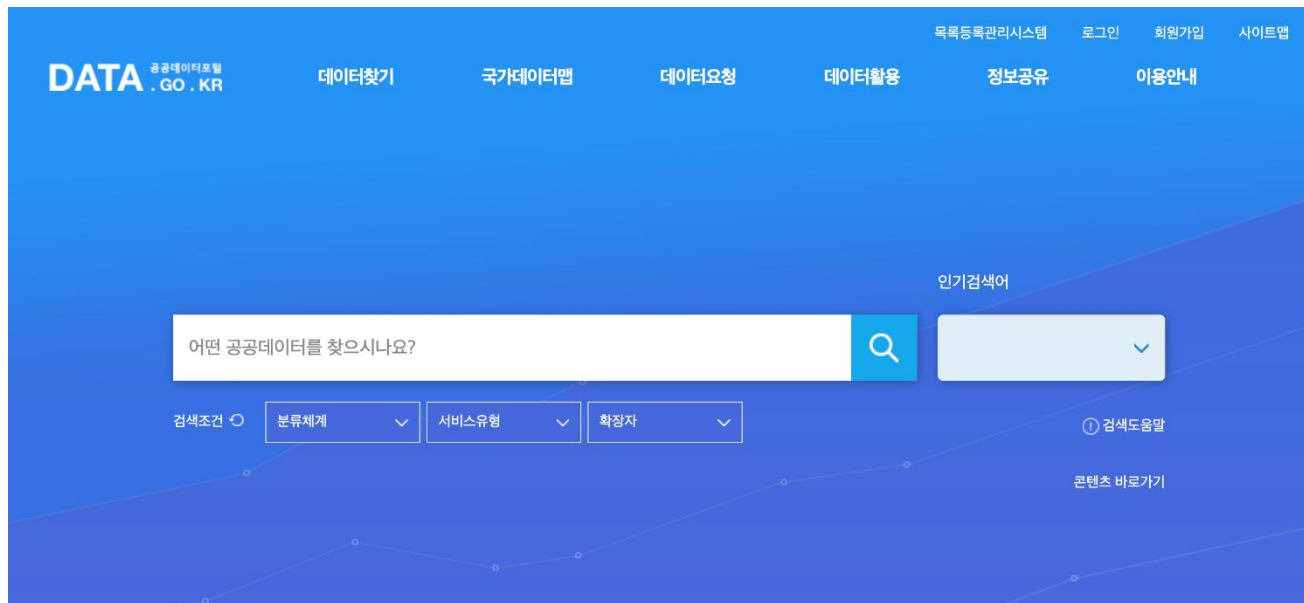
- 서버와 데이터베이스에 접근하는 창구 역할 (허용된 사람만 접근 가능)
- 접속 방법을 표준화 (동일한 원칙 적용)



데이터 분석 기술

• 데이터 수집 – 2. API

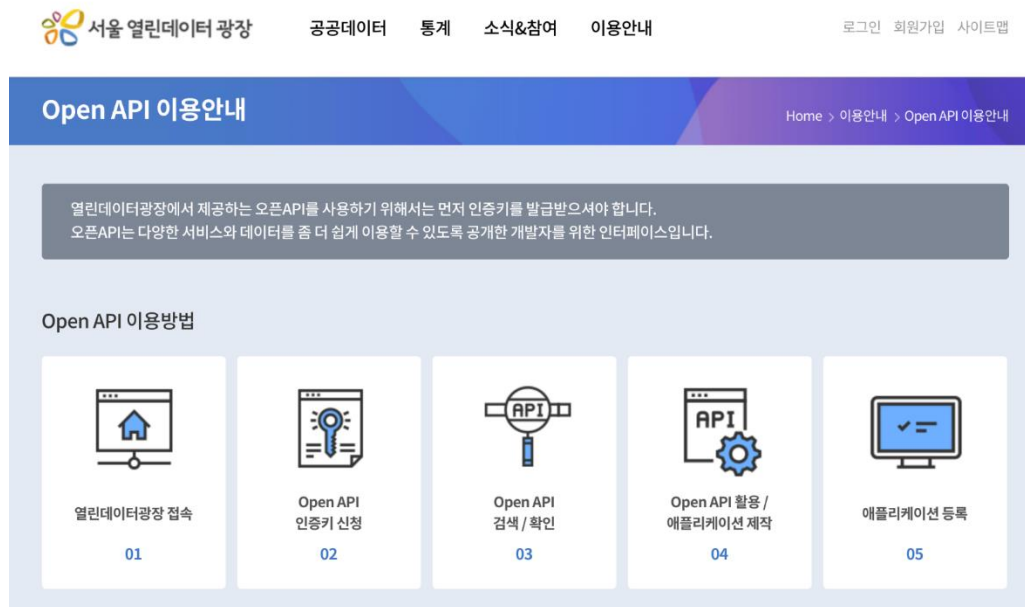
- 공공데이터 포털(<https://www.data.go.kr>)
- 공공기관에서 생성 또는 취득하여 관리하는 공공데이터를 제공하는 사이트



데이터 분석 기술

• 데이터 수집 – 2. API

- 서울열린데이터광장(<http://data.seoul.go.kr/>)
- 서울시의 공공데이터를 제공해주는 사이트



데이터 분석 기술

• 데이터 수집 – 2. API

- 공공데이터포털(<https://www.data.go.kr>) 접속
- 회원가입 & 로그인 진행
- [데이터셋] 메뉴에서 [파일데이터] 혹은 [오픈API] 선택
- 파일데이터와 오픈API 2가지 방법으로 데이터 제공

전체(33,608) 파일데이터(30,190) 오픈API(3,299) 표준데이터(119)

파일데이터 30,190건을 찾았습니다. Q 기관별 검색

파일데이터 [30,190건] 정확도 날짜 제목 조회수 다운로드

도로교통공단_교통사고 통계 조회수 : 22,698 다운로드수 : 26,582
수정일 : 2019.09.16 기관 : 도로교통공단 서비스유형 : 다운로드 LINK
- 경찰에서 조사, 처리한 교통사고에 대한 통계 정보로 인적 피해가 있는 사고만 집계 됨- 사고유형별, 법규위반별, 시간대별 ...

기상청 날씨예보 정보 조회수 : 18,006 다운로드수 : 257
수정일 : 2019.11.08 기관 : 기상청 서비스유형 : 배송 및 기타
기상청에서 생산하는 초단기예보, 동네예보, 중기예보 등을 제공합니다.

CSV XLS

공공행정 과학기술

전체(33,608) 파일데이터(30,190) 오픈API(3,299) 표준데이터(119)

오픈API 3,299건을 찾았습니다. Q 기관별 검색

오픈API [3,299건] 정확도 날짜 제목 조회수 활용신청

도로명주소조회서비스 조회수 : 29,887 활용신청건수 : 3,954
수정일 : 2018.12.28 기관 : 과학기술정보통신부 우정사업본부 서비스유형 : REST
우정사업본부에서는 도로명주소 체계로 변경되는 새우편번호(2015.8.1 시행) 및 기존 우편번호 정보를 조회하는 기능의 오...

기상청 날씨예보 정보 조회수 : 18,006 활용신청건수 : 16,427
수정일 : 2019.07.25 기관 : 기상청 서비스유형 : REST
기상청에서 생산하는 초단기예보, 동네예보, 중기예보 등을 제공합니다.

XML

공공행정 과학기술

데이터 분석 기술

• 데이터 수집 – 2. API

파일 데이터의 경우 직접 제공을 하거나 외부 링크로 다운 가능

도로교통공단_교통사고 통계

- 경찰에서 조사, 처리한 교통사고에 대한 통계 정보로 인적 피해가 있는 사고만 집계 됨
- 사고유형별, 법규위반별, 시간대별 등 각종 부문별 교통사고 통계자료 제공
- 교통사고분석시스템(http://taas.koroad.or.kr)의 데이터를 바탕으로 함

매체유형 : 텍스트 파일, 링크 건수 : 78 전체 행 수 : N/A 확장자 : CSV / XLS 다운로드 횟수(바로그가) : 249

☐ 전체 **선택 다운로드** ※ 서비스 오류가 있을시 오류신고 버튼을 이용해주세요.

☒ CSV 도로교통공단_월별_주야별 교통사고(...) ☐ CSV 도로

멀티다운로드 **Q 찾기** **오류신고** **★** **멀티다운로드**

☐ CSV ☐ XML

도 0910

업 다운로드

비용부과유무 무료 비용부과기준 및 단위

[파일 다운로드를 제공하는 경우]

기상청 날씨예보 정보 **ENGLISH**

기상청에서 생산하는 초단기예보, 동태예보, 중기예보 등을 제공합니다.

매체유형 : 텍스트 파일, 링크 건수 : 1 전체 행 수 : N/A 확장자 : CSV 다운로드 횟수(바로그가) : 249

☐ 전체 **선택 다운로드** ※ 서비스 오류가 있을시 오류신고 버튼을 이용해주세요.

☐ CSV **과거 동태예보(일간)** **다운로드** **Q 찾기** **오류신고** **★**

과거 동태예보(일간)

업데이트 주기	일간	차기등록예정일	2019-12-28
비용부과유무	무료	비용부과기준 및 단위	없음
다운로드 횟수	249		
등록일	2019-11-08	수정일	2019-11-08
이용허락범위	공공저작물_출처표시		
제공형태	전자기록매체 저장 제공		
URL	https://www.data.go.kr/dataset/FileDownload.do?atchFileId=FILE_000000001588772&fileDetailS=1		
설명	기상청에서 발표한 동태예보의 과거자료입니다. 실험분석자료, 초단기예보, 단기예보의 지역별, 요소별 자료를 제공합니다.		
기타유의사항	본 자료는 매일 생산되는 대용량자료로, 기상자료개방포털(data.kma.go.kr)에서 로그인 후 제공 받으실 수 있습니다.		

[외부 URL을 이용하여 제공하는 경우]

데이터 분석 기술

• 데이터 수집 – 2. API

- 금융위원회 금융통계손해보험정보 가져오기
- 공공 데이터 포털에서 “보험 손해율” 검색한 뒤에, 활용신청을 클릭합니다.

전체(386건)

파일데이터(335건)

오픈 API(51건)

표준데이터셋(0건)

정확도순 ▼ 10개씩 ▼

오픈 API (51건)

재정금융

국가행정기관

미리보기

JSON 금융위원회_금융통계손해보험정보

타이틀 기준년월을 조회하여 손해보험사일반현황, 손해보험사재무현황, 손해보험사주요경영지표, 손해보험사주요영업활동 등의 정보를 제공하는 금융위원회_금융통계손해보험정보

제공기관 금융위원회 수정일 2020-07-22 조회수 1458 활용신청 32 키워드 금융통계,손해보험,현황

활용신청

재정금융

공공기관

미리보기

XML **JSON** 신용보증기금_매출채권보험 정보

신용보증기금의 매출채권보험 인수금액을 받기 및 업종에 따라 조회할 수 있는 서비스

제공기관 신용보증기금 수정일 2019-06-19 조회수 975 활용신청 32 키워드 신용보험,매출채권보험,인수금액

활용신청

데이터 분석 기술

• 데이터 수집 – 2. API

활용 목적 선택 : 기타

활용목적 선택

*표시는 필수 입력항목입니다.

*활용목적

☐ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☒ 기타 ☐ 참고자료 ☐ 연구(논문 등)

자료 조사

5/250

첨부파일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

시스템유형

시스템 유형

☒ 일반

데이터 분석 기술

• 데이터 수집 – 2. API

라이선스 동의

상세기능정보 선택

<input checked="" type="checkbox"/>	상세기능	설명	일일 트래픽
<input checked="" type="checkbox"/>	손해보험일반현황조회	타이틀, 기준년월등을 통하여 금융회사명, 인원수, 임직원수, 임직원 구분코드명등을 조회하는 손해보험일반현황조회 기능	10000
<input checked="" type="checkbox"/>	손해보험재무현황조회	타이틀, 기준년월 등을 통하여 금융회사명, 자산요약재무상태표계정과목금액, 자산요약재무상태표계정과목코드, 자산요약재무상태표계정과목구성비율, 법인등록번호등을 조회하는 손해보험재무현황조회 기능	10000
<input checked="" type="checkbox"/>	손해보험주요영업활동조회	타이틀, 기준년월 등을 통하여 금융회사명, 보험종류경과손해율구분금액, 보험종류경과손해율구분코드, 보험종류경과손해율구분코드명등을 조회하는 손해보험주요영업활동조회 기능	10000
<input checked="" type="checkbox"/>	손해보험주요경영지표조회	타이틀, 기준년월등을 통하여 금융회사명, 자본적정성항목코드, 자본적정성항목코드명, 자본적정성항목값내용등을 조회하는 손해보험주요경영지표조회 기능	10000

라이선스 표시

* 이용허락범위

☒ 동의합니다.

취소

활용신청



데이터 분석 기술

• 데이터 수집 – 2. API

마이페이지 선택 → 오픈API 클릭 → 개발계정 선택 → [승인] 여부 확인



The screenshot displays the '마이페이지' (My Page) interface, specifically the '개발계정' (Developer Account) section. The '오픈API' (Open API) tab is selected, showing the '신청 0건' (0 Applications) status. The '활용 18건' (18 Applications) status is also shown. The '증지0건' (0 Certificates) status is shown. The '신청 0건' status is highlighted with a red box.

마이페이지	개발계정
오픈API 개발계정 운영계정 인증키 발급현황	신청 0건 > 신청중인 단계 · 보류 0건 · 반려 0건
DATA	활용 18건 > 승인되어 활용중인 단계 · 변경신청 0건
나의 문의 >	증지0건 > 증지신청하여 운영이 증지된 단계
나의 관심	상세검색 열기 >
나의 제공신청	
나의 분쟁조정	
회원정보 수정 >	

상세검색 열기 >

재정금융	금융위원회
활용신청 [승인] 금융위원회_금융통계서비스정보	
신청일 2021-10-11	만료예정일 2023-10-11

과학기술 기상청

활용신청 [승인] 기상청_동네예보 조회서비스	
신청일 2021-05-26	만료예정일 2023-05-26

데이터 분석 기술

• 데이터 수집 – 2. API

개발계정 상세보기 → 참고문서 다운로드 → 일반 인증키 복사

마이페이지

오픈API

개발계정

운영계정

인증키 발급현황

DATA

나의 문의

나의 관심

나의 제공신청

나의 분쟁조정

회원정보 수정

개발계정 상세보기

기본정보

데이터명	금융위원회_금융통계시스템정보 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 활용신청	처리상태	승인
활용기간	2021-10-11 ~ 2023-10-11		

서비스정보

참고문서	금융통계 활용자가이드 금융통계시스템정보.docx
데이터포맷	JSON
End Point	http://apis.data.go.kr/1160100/service/GetNonInsuComplInfoService
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)	인증키 복사
일반 인증키 (Decoding)	

데이터 분석 기술

• 데이터 수집 – 2. API

활용신청 상세기능정보 → 미리보기 [확인] → 요청 변수 [미리보기] → 결과 확인

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	손해보험일반현황조회	타이틀, 기준년월 등을 통하여 금융회사명, 인원수, 임직원수, 임직원 구분코드명 등을 조회하는 손해 보험일반현황조회 기능	10000	확인
2	손해보험자우선현황조회	타이틀, 기준년월 등을 통하여 금융회사명, 자산요약자우상태표계정과목금액, 자산요약자우상태표계정과목코드, 자산요약자우상태표계정과목성의를, 법인종류번호 등을 조회하는 손해보험자우선현황조회 기능	10000	확인
3	손해보험주요영입활동조회	타이틀, 기준년월 등을 통하여 금융회사명, 보험종류경과손해율구분금액, 보험종류경과손해율구분코드, 보험종류경과손해율구분코드명 등을 조회하는 손해보험주요영입활동조회 기능	10000	확인
4	손해보험주요경영지표조회	타이틀, 기준년월 등을 통하여 금융회사명, 자본적정성항목코드, 자본적정성항목코드명, 자본적정성항목값내용 등을 조회하는 손해보험주요경영지표 조회 기능	10000	확인

요청변수(Request Parameter)

항목명	샘플데이터	설명
numOfRows	1	한 페이지 결과 수
pageNo	1	페이지 번호
resultType	xml	결과형식(xml/json)
serviceKey	공공데이터포털에서 받은	공공데이터포털에서 받은 인증키
title	손보_일반현황_임직원및	손보_일반현황
basYm	201409	기준년월

[미리보기](#)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
    <pageNo>1</pageNo>
    <numOfRows>1</numOfRows>
  </header>
  <body>
    <table>
      <title>손보_일반현황_임직원및설계사현황</title>
      <totalCount>250</totalCount>
      <items>
        <item>
          <basYm>201409</basYm>
          <crno>1101110013328</crno>
          <fncoCd>0010626</fncoCd>
          <fncoNm>메리츠화재상해보험주식회사</fncoNm>
          <xcsmpInpnCnt>2642</xcsmpInpnCnt>
          <xcsmpInpnDcd>A</xcsmpInpnDcd>
          <xcsmpInpnDcdNm>임직원</xcsmpInpnDcdNm>
        </item>
      </items>
    </table>
  </body>
</response>
```


데이터 분석 기술

- 데이터 수집 – 2. API

- requests 설치: `pip install requests`
- bs4 설치: `pip install bs4`

데이터 분석 기술

- 오픈API를 통한 데이터 수집 (파일명: api.ipynb)

- 필요 라이브러리 불러오기

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

- 서버에 데이터 요청

```
ServiceKey = '발급받은 ServiceKey를 입력하세요'  
url = "http://apis.data.go.kr/1160100/service/(엔드포인트)?"  
  
api_url = url + "serviceKey="+ ServiceKey  
  
req = requests.get(api_url)
```

데이터 분석 기술

- 오픈API를 통한 데이터 수집 (파일명: api.ipynb)

- XML 문서

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"
<response>
<header>
<resultcode>00</resultcode>
<resultmsg>NORMAL SERVICE.</resultmsg>
<pageno>1</pageno>
<numofrows>10</numofrows>
</header>
<body>
<table>
<title>손보_주요영업활동_보험종류별 경과손해율</title>
<totalcount>21732</totalcount>
<items>
<item>
<basym>201912</basym>
<crno>1101110013328</crno>
<fncoCd>0010626</fncoCd>
<fnconm>메리츠화재해상보험주식회사</fnconm>
<isukindelpslosratclsfamt>88.48</isukindelpslosratclsfamt>
<isukindelpslosratdcd>A</isukindelpslosratdcd>
<isukindelpslosratdcdnm>경과손해율_자동차</isukindelpslosratdcdnm>
</item>
<item>
<basym>201709</basym>
<crno>1101110013328</crno>
...
</item>
</items>
</table>
</body>
</response>
```

```
basYm = []
crno = []
fncoCd = []
fncoNm = []
ClsfAmt = []
RatDcd = []
RatDcdNm = []

for item in items:
    basYm.append(item.select('basym')[0].text)
    crno.append(item.select('crno')[0].text)
    fncoCd.append(item.select('fncoCd')[0].text)
    fncoNm.append(item.select('fnconm')[0].text)
    ClsfAmt.append(item.select('isukindelpslosratclsfamt')[0].text)
    RatDcd.append(item.select('isukindelpslosratdcd')[0].text)
    RatDcdNm.append(item.select('isukindelpslosratdcdnm')[0].text)
```

데이터 분석 기술

- 오픈API를 통한 데이터 수집 (파일명: api.ipynb)
- 반복문 사용

```
# 손해보험주요영업활동조회
url = "http://apis.data.go.kr/1160100/service/GetNonlInsuCompInfoService/getNonlInsuCompMajoBusiActi?"

api_url = url + "serviceKey="+ ServiceKey + "&numOfRows="+ str(100)

for page in range(1, 218, 1):
    if page % 10 == 0:
        print(page)

    page_url = api_url + "&pageNo=" + str(page)

    req = requests.get(page_url)
    time.sleep(5)

    xml = req.text
    soup = BeautifulSoup(xml, 'html.parser')

    # 한 페이지 아이템 목록
    items = soup.select('item')

    for item in items:
        basYm.append(item.select('basym')[0].text)
        crno.append(item.select('crno')[0].text)
        fncoCd.append(item.select('fnccod')[0].text)
        fncoNm.append(item.select('fnconm')[0].text)
        ClsfAmt.append(item.select('isukindelpslosratclsfamt')[0].text)
        RatDcd.append(item.select('isukindelpslosratdcd')[0].text)
        RatDcdNm.append(item.select('isukindelpslosratdcdnm')[0].text)
```

데이터 분석 기술

- 오픈API를 통한 데이터 수집 (파일명: api.ipynb)
- CSV 파일 저장

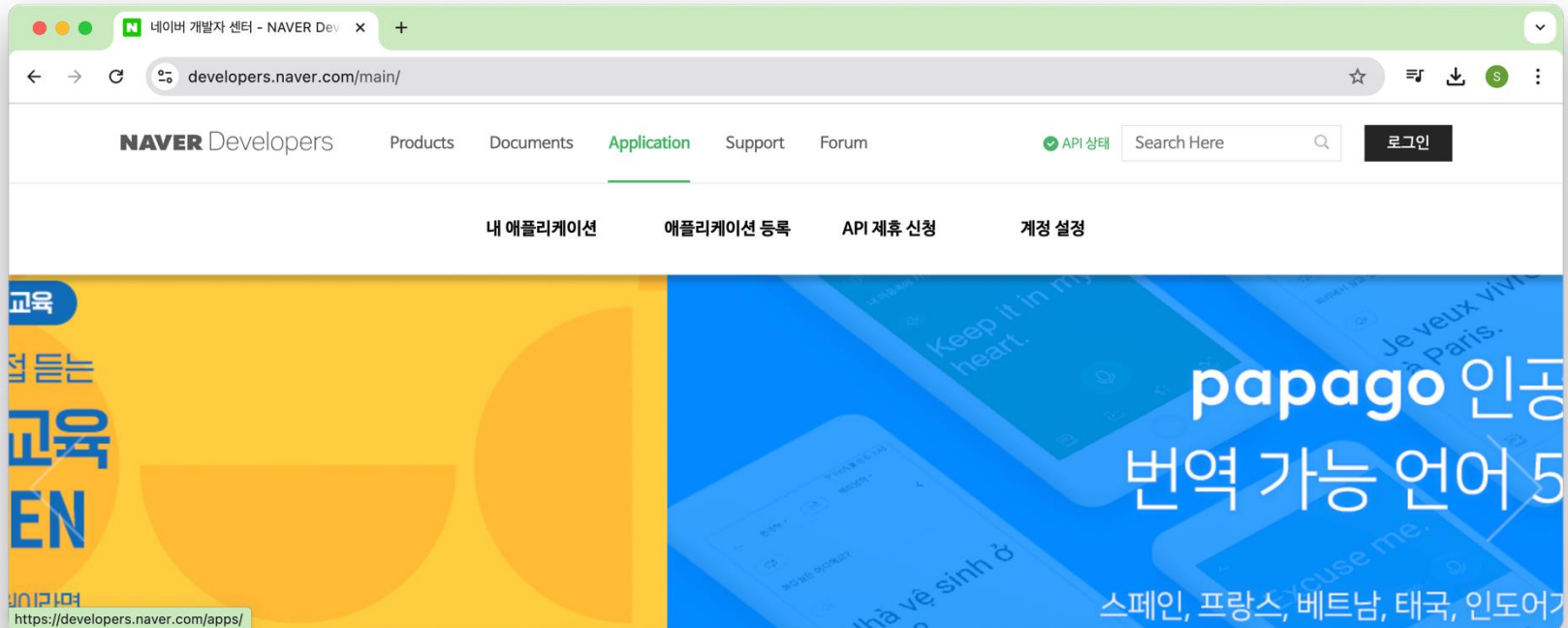
```
data.to_csv('손해보험주요영업활동조회.csv', index=False)
```

basYm	cmo	fncoCd	fncoNm	ClsfAmt	RatDcd	RatDcdNm
201912	1101110013328	10626	화재해상보험주	88.48	A	경과손해율_자동차
201709	1101110013328	10626	화재해상보험주	76.97	A	경과손해율_자동차
202109	1101110013328	10626	화재해상보험주	75.77	A	경과손해율_자동차
201609	1101110013328	10626	화재해상보험주	83.09	A	경과손해율_자동차
201403	1101110013328	10626	화재해상보험주	91.3	A	경과손해율_자동차
202206	1101110013328	10626	화재해상보험주	74.06	A	경과손해율_자동차
202106	1101110013328	10626	화재해상보험주	75.75	A	경과손해율_자동차
202203	1101110013328	10626	화재해상보험주	73.07	A	경과손해율_자동차
201503	1101110013328	10626	화재해상보험주	90.89	A	경과손해율_자동차
201806	1101110013328	10626	화재해상보험주	77.41	A	경과손해율_자동차
201106	1101110013328	10626	화재해상보험주	81.78	A	경과손해율_자동차
201809	1101110013328	10626	화재해상보험주	79.69	A	경과손해율_자동차
201203	1101110013328	10626	화재해상보험주	83.94	A	경과손해율_자동차

데이터 분석 기술

- 네이버 개발자 API 활용
 - 네이버 개발자 센터(<https://developers.naver.com/>)

네이버 뉴스, 블로그, 쇼핑 등 데이터를 검색, 조회할 수 있는 서비스



데이터 분석 기술

• 네이버 개발자 API 활용

• 네이버 API 키 발급

1. 네이버 계정으로 로그인
2. 'Application' 메뉴에서 '애플리케이션 등록' 클릭
3. 애플리케이션 이름과 사용 API 선택 (검색 API)
4. 네이버 API 키 (Client ID, Client Secret) 확인

네이버 개발자 API 키 발급

애플리케이션 이름

- 네이버 로그인할 때 사용자에게 표시되는 이름으로 가급적 10자 이내로 간결하게 설정해주세요
- 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(.), "

선택하세요.

- 검색
- 네이버 로그인
- 네이버 인증서
- 네이버 전자문서
- 네이버페이 배송지 정보
- 단축 URL
- 데이터랩 (검색어트렌드)
- 데이터랩 (쇼핑인사이트)
- 카페

- [애플리케이션 이름] 설정을 확인해 주세요.
- [사용 API] 설정을 확인해 주세요.

데이터 분석 기술

- 네이버 개발자 API 활용

- 네이버 API를 활용하여 뉴스 데이터 수집

뉴스 검색 결과 조회 [↗](#)

설명 [↗](#)

네이버 검색의 뉴스 검색 결과를 XML 형식 또는 JSON 형식으로 반환합니다.

요청 URL [↗](#)

요청 URL	반환 형식
<code>https://openapi.naver.com/v1/search/news.xml</code>	XML
<code>https://openapi.naver.com/v1/search/news.json</code>	JSON

프로토콜 [↗](#)

HTTPS

데이터 분석 기술

- 네이버 개발자 API 활용

- 네이버 API를 활용하여 뉴스 데이터 수집

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: 1000)
sort	String	N	검색 결과 정렬 방법 <ul style="list-style-type: none">- <code>sim</code>: 정확도순으로 내림차순 정렬(기본값)- <code>date</code>: 날짜순으로 내림차순 정렬

데이터 분석 기술

- 네이버 개발자 API 활용
 - 네이버 API를 활용하여 뉴스 데이터 수집

API를 요청할 때 다음 예와 같이 HTTP 요청 헤더에 **클라이언트 아이디**와 **클라이언트 시크릿**을 추가해야 합니다.

```
> GET /v1/search/news.xml?query=%EC%A3%BC%EC%8B%9D&display=10&start=1&sort=sim HTTP/1.1
> Host: openapi.naver.com
> User-Agent: curl/7.49.1
> Accept: */*
> X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 클라이언트 아이디 값}
> X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 클라이언트 시크릿 값}
>
```

요청 예 [↗](#)

```
curl "https://openapi.naver.com/v1/search/news.xml?query=%EC%A3%BC%EC%8B%9D&display=10&start=1&sort=sim" \
-H "X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 클라이언트 아이디 값}" \
-H "X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 클라이언트 시크릿 값}" -v
```

데이터 분석 기술

- 네이버 개발자 API 활용
 - 네이버 API를 활용하여 뉴스 데이터 수집 (파이썬)

```
import requests

def naver_news_search(query):

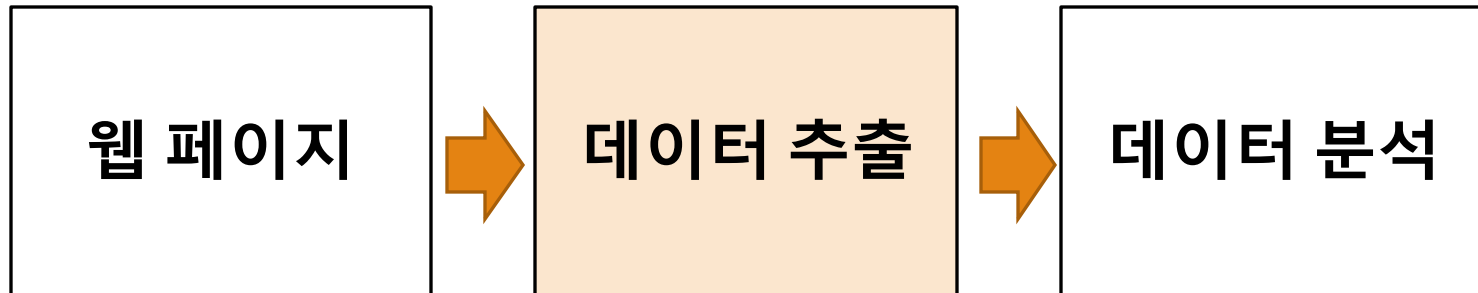
    url = "https://openapi.naver.com/v1/search/news.json"
    headers = {
        "X-Naver-Client-Id": "YOUR_CLIENT_ID",
        "X-Naver-Client-Secret": "YOUR_CLIENT_SECRET"
    }
    params = {"query": query}

    response = requests.get(url, headers=headers, params=params)
    return response.json()
```

데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

- 웹페이지에서 원하는 데이터를 추출하는 행위
- 스크래핑(scraping) 이라고 불리기도 함
- 크롤링하는 소프트웨어를 크롤러(crawler) 라고 부름
- 엑셀, 파이썬 등 다양한 도구를 이용해서 크롤링 가능



데이터 분석 기술

• 웹 브라우저

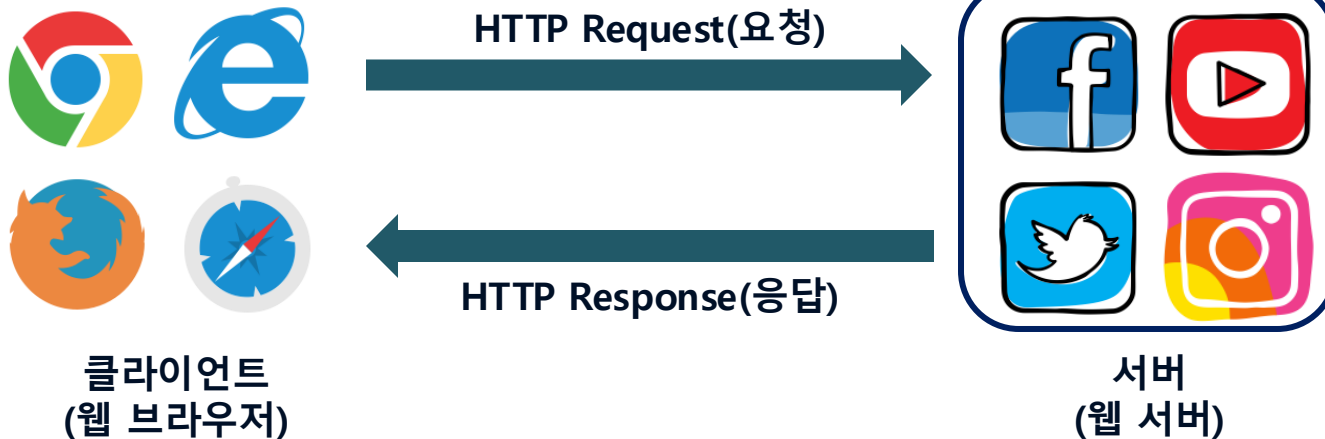
- 웹 브라우저는 페이지를 검색하고 표시하는 소프트웨어 애플리케이션
- 주요 웹 브라우저: 구글 크롬, 모질라 파이어폭스, 사파리, 엣지
- HTML, CSS, JavaScript를 해석하고 실행
- 다양한 웹 표준을 지원



데이터 분석 기술

• 웹 서비스

- 웹서비스는 요청(request)과 응답(response)로 구성
- HTTP(HyperText Transfer Protocol) 또는 HTTPS 사용



데이터 분석 기술

• 요청과 응답 상세

- 요청 (Request):
 - 클라이언트가 서버에 데이터를 요청.
 - HTTP 메서드: GET, POST, PUT, DELETE 등.
- 응답 (Response):
 - 서버가 클라이언트의 요청에 대한 응답을 제공.
 - 상태 코드: 200(성공), 404(찾을 수 없음), 500(서버 오류) 등

데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

크롤링 주의사항

- ✓ **무단으로 크롤링하는 것은 불법이 될 수 있음**
 - ☞ 해당 사이트 약관이나 '사이트 주소/robots.txt' 를 확인 후 크롤링
(ex) <http://www.daum.net/robots.txt>
- ✓ 크롤링을 허용하더라도, 서버에 무리를 주는 행위는 불법이 될 수 있음.
 - ☞ 서버에 무리가 되지 않도록 접속 회수 및 빈도 수를 제한
: 파이썬 `time.sleep()` 명령 활용 등

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링

robots.txt

```
https://www.naver.com/robots.txt  
User-agent: *  
Disallow: /  
Allow : /$
```

`User-agent: *`: 이 규칙은 모든 웹 크롤러에 적용

`Disallow: /`: 웹사이트의 모든 페이지에 대한 크롤링을 금지

`Allow : /\$`: 홈페이지(\$는 URL의 끝을 나타냄)에 대한 접근을 허용

데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

정적(static) 페이지

- 서버에 미리 저장된 파일(HTML, 이미지 등)을 웹브라우저에 표시
- 웹페이지는 내용이 고정되어 변하지 않음

BeautifulSoup 라이브러리 활용
☞ 우선 적용!

동적(dynamic) 페이지

- 서버에서 데이터를 가공하여 실시간으로 웹 페이지를 생성
- 사용자의 요청, 시간, 상황에 따라 웹페이지 내용이 달라지게 됨

Selenium 라이브러리 활용
☞ BeautifulSoup이 안되면 적용!

데이터 분석 기술

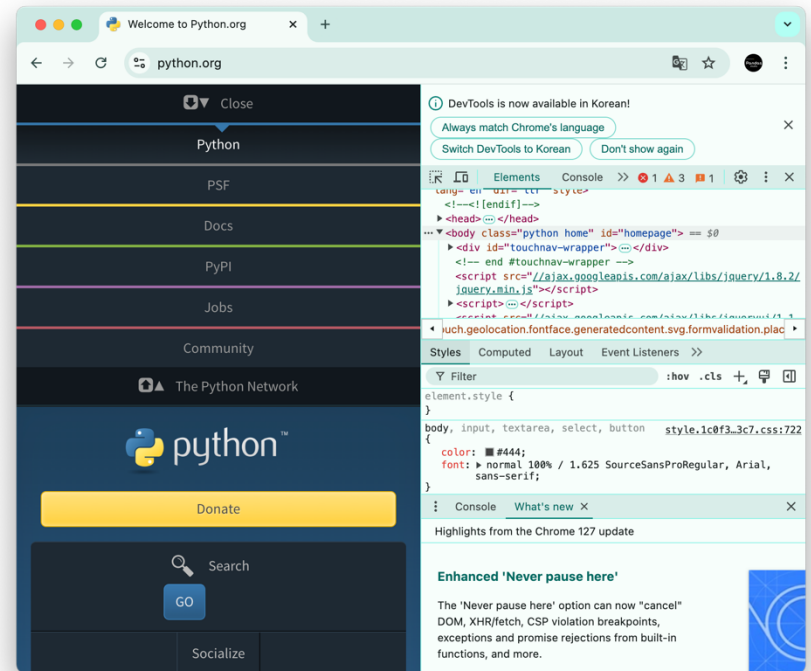
• 데이터 수집 – 3. 웹 크롤링

웹 구성요소	설명
HTML	<ul style="list-style-type: none">• 웹 페이지의 구조를 정의하는 마크업 언어• 브라우저는 HTML을 해석하여 웹 페이지의 콘텐츠와 레이아웃을 구성
CSS	<ul style="list-style-type: none">• HTML로 구성된 웹 페이지의 스타일과 레이아웃을 제어하는 스타일시트 언어• 브라우저는 CSS를 적용하여 웹 페이지의 시각적 표현을 처리
JavaScript	<ul style="list-style-type: none">• 웹 페이지의 동적 기능을 구현하는 스크립트 언어• 브라우저는 JavaScript 코드를 실행하여 웹 페이지의 상호작용성과 동적인 콘텐츠를 제공
미디어 소스	<ul style="list-style-type: none">• 이미지, 비디오, 오디오 등 다양한 멀티미디어 요소

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링
 - 개발자 도구

- 개발자 도구 열기: F12 또는 우클릭 후 '검사' 선택
- Elements 탭: HTML 요소 구조 확인
- Styles 탭: CSS 스타일 확인 및 수정
- Console 탭: JavaScript 코드 실행 및 디버깅

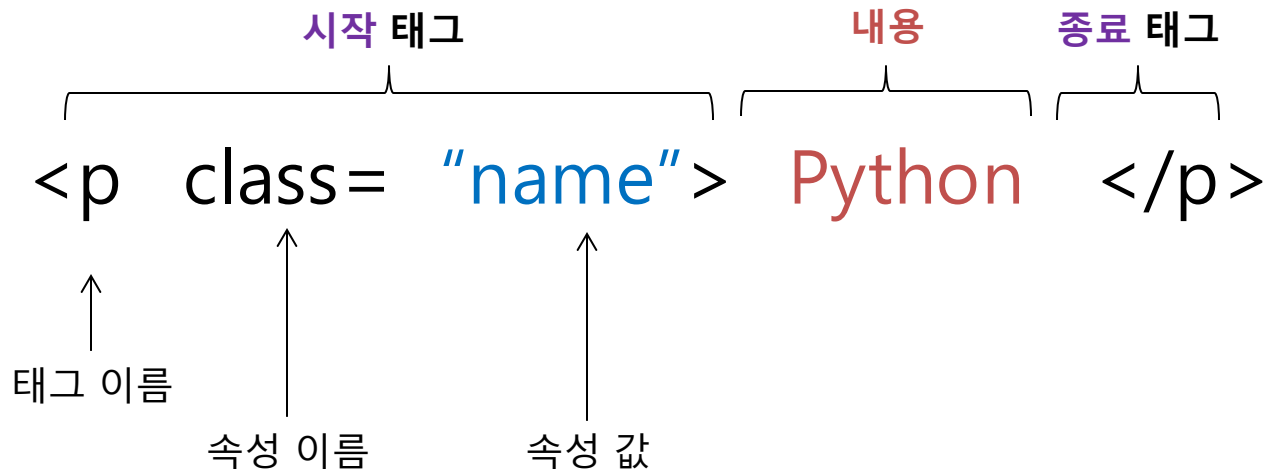


데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

- **HTML 태그(요소), element**

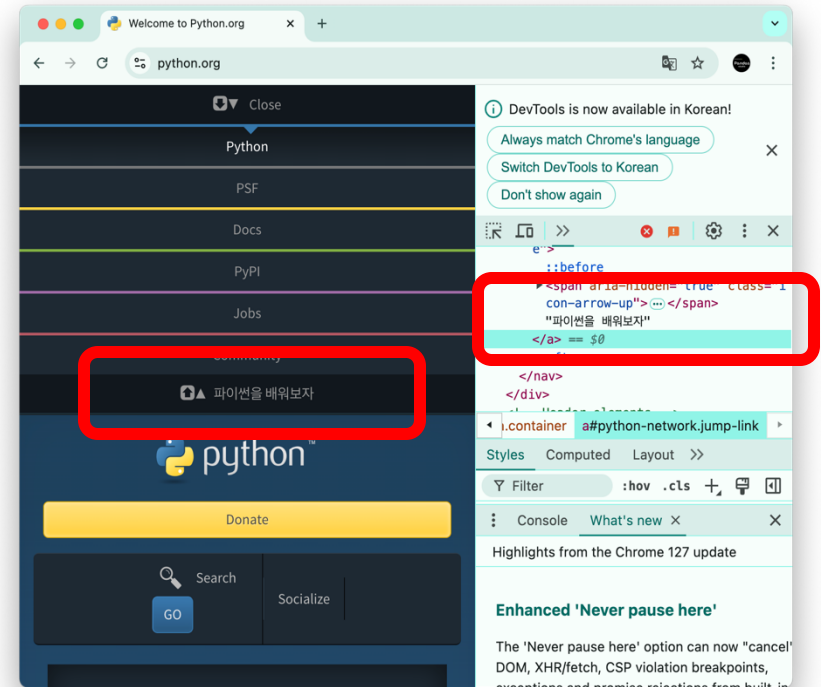
- HTML 문서나 웹 페이지를 이루는 개별적인 요소
- 문서 객체 모델(DOM) 구조로 해석



데이터 분석 기술

- 데이터 수집 - 3. 웹 크롤링
 - 개발자 도구 - HTML 확인 또는 수정

Elements 탭에서 원하는 HTML 요소를
선택하고 수정



데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

• CSS Selector

- CSS 스타일 적용을 하기 위해 HTML의 특정 요소를 선택하는 도구
- 크롬 개발자 도구에서 지원하는 Copy CSS Selector 기능 활용

태그 선택자

HTML 태그 이름을 그대로 사용 (p)

클래스 선택자

주어진 값을 class 속성 값으로 갖는 HTML 요소를 선택 (.myclass)

ID 선택자

주어진 값을 id 속성 값으로 갖는 HTML 요소를 선택 (#myid)

데이터 분석 기술

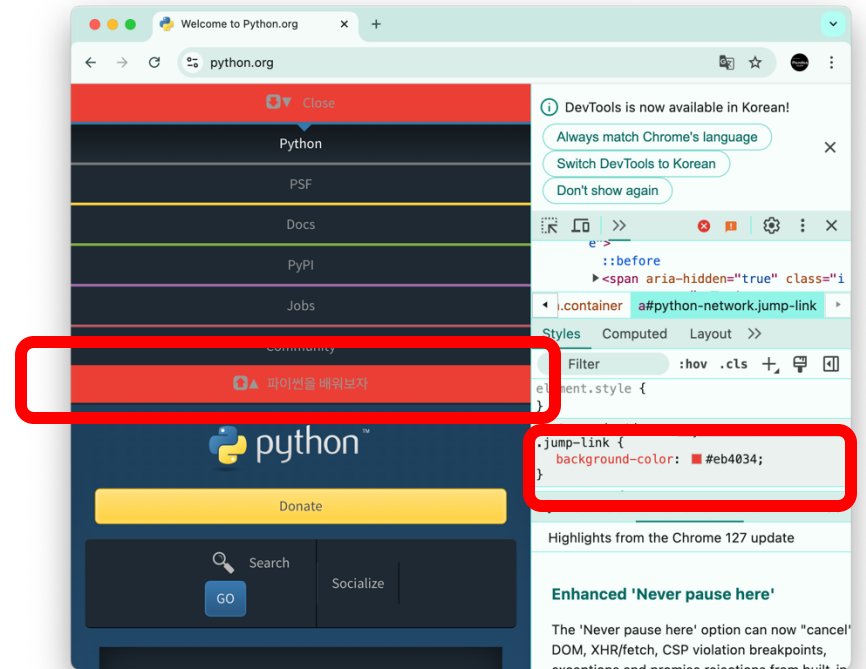
- 데이터 수집 - 3. 웹 크롤링
 - 개발자 도구 - CSS 확인 또는 수정

Styles 탭에서 CSS 속성을 추가, 변경

background-color: #11171d



background-color: #eb4034



데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

- **BeautifulSoup** 라이브러리

- BeautifulSoup는 Python에서 HTML 및 XML 파일을 파싱하는 데 사용
- 웹 스크래핑, 데이터 마이닝, 자동화된 웹 테스트 등 다양하게 활용
- 특히 구조화되지 않은 웹 데이터를 처리하고 분석하는데 매우 유용

find()

조건에 맞는
첫 번째 요소만 반환

find_all()

조건에 맞는
모든 요소를 찾고,

리스트로 반환

select()

CSS 선택자를 사용
하여 요소를 찾고,

리스트를 반환

데이터 분석 기술

• 데이터 수집 – 3. 웹 크롤링

- **Selenium 라이브러리**

- 웹 애플리케이션 테스트 도구로 개발
- 실제 사용자 상호작용을 시뮬레이션하여 포괄적인 테스트 수행 가능
- 클릭, 스크롤, 폼 입력 등 실제 사용자 행동을 정밀하게 재현
- AJAX 요청, JavaScript 렌더링 등으로 실시간 변화하는 웹 요소 포착
- 동적으로 생성되는 데이터를 특정 시점에 캡처하여 분석 가능하다
- 웹 스크래핑, 데이터 추출, 프로세스 자동화 등에 활용
- Chrome, Firefox, Safari, Edge 등 다양한 브라우저 지원

find_element()

특정 요소 한 개를 찾을 때 사용

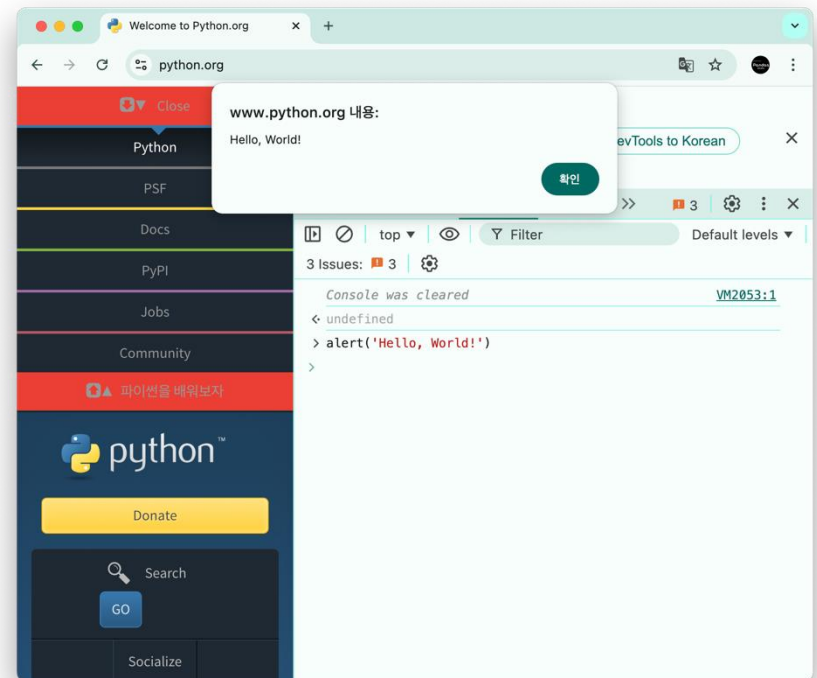
find_elements()

조건을 만족하는 모든 요소를
전부 찾을 때 사용

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링
 - 개발자 도구 - Javascript Alert 실행하기

1. Console 탭 열기: F12 또는 우클릭 후 '검사' 선택, Console 탭 클릭
2. JavaScript 코드 입력: `alert('Hello, World!');` 입력 후 엔터
3. 결과 확인: 브라우저에서 팝업 창으로 'Hello, World!' 메시지 확인



데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링

- selenium 설치: `pip install selenium`
- 웹드라이버 설치: `pip install webdriver_manager`

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - 실시간 랭킹 뉴스 수집

```
# 뉴스 사이트
url = "https://news.daum.net/"

# 에이전트 설정
agent = 'Mozilla/2.0'

# requests.get
resp = requests.get(url)
print(resp)
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - BeautifulSoup – find 메소드

```
# find - 가장 먼저 나타나는 태그를 찾는다  
soup.find(name='ul')
```

```
<ul class="doc-relate" data-tiara-layer="GNB service">  
<li><a class="link_services" data-tiara-layer="enter" href="https://entertain.daum.net">연예</a></li>  
<li><a class="link_services" data-tiara-layer="sports" href="https://sports.daum.net">스포츠</a></li>  
</ul>
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - BeautifulSoup – find_all 메소드

```
# find_all - 모든 태그를 찾는다
ul_data = soup.find_all(name='ul')
len(ul_data)
```

```
# class 속성이 "list_newsissue"인 ul 태그를 모두 찾는다
newsissue = soup.find_all(name='ul', attrs={'class': 'list_newsissue'})
len(newsissue)
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - BeautifulSoup – select 메소드

```
# tag 이름이 'ul'인 것을 모두 찾아서 ul_list에 저장
ul_list = soup.select('ul')
len(ul_list)
```

```
# class 속성값이 list_newsissue인 경우
class_list = soup.select('.list_newsissue')
len(class_list)
```

```
# id="kakaoServiceLogo"

id_list = soup.select('#kakaoServiceLogo')
len(id_list)
```


데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - BeautifulSoup – select 메소드

```
# class 속성값이 list_newsissue인 ul 태그의 자식 li 태그를 모두 찾아서 li_list에 저장
li_list = soup.select('.list_newsissue > li')
len(li_list)
```

- `select()` 메서드는 CSS 선택자를 사용하여 HTML 요소를 찾습니다.
- '.list_newsissue'는 클래스 이름이 'list_newsissue'인 요소를 선택합니다.
- '>' 기호는 직계 자식 요소를 의미합니다.
- 'li'는 태그를 선택합니다.
- 클래스가 'list_newsissue'인 요소의 직계 자식 중 모든 태그"를 선택합니다.

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)

- Selenium – 드라이버 생성

1. Selenium과 필요한 모듈을 임포트하여 Chrome 웹 드라이버를 설정
2. ChromeDriverManager를 사용해 최신 Chrome 드라이버를 자동으로 설치하고 서비스 객체를 생성
3. 설정된 서비스와 옵션을 바탕으로 Chrome 웹 드라이버 인스턴스를 초기화하여 웹 자동화 작업을 위한 준비를 완료

```
# Selenium 드라이버 생성
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager

# Chrome 옵션 설정
options = webdriver.ChromeOptions()

# 드라이버 서비스 생성
service = Service(ChromeDriverManager().install())

# 웹 드라이버 초기화
driver = webdriver.Chrome(service=service, options=options)
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 웹 페이지의 HTML 소스 확인

1. 다음 뉴스 사이트에 접속하여 페이지를 로드합니다.
2. 로드된 페이지의 전체 HTML 소스를 가져옵니다.

```
# 뉴스 사이트
url = "https://news.daum.net/"

driver.get(url)

# HTML 코드
driver.page_source
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 요소 선택 (By.TAG_NAME)

1. By 클래스는 요소를 찾는 다양한 방법(예: ID, 클래스 이름, 태그 이름 등)을 제공
2. driver.find_element() 메서드를 사용하여 웹 페이지에서 요소를 찾음
3. By.TAG_NAME을 사용하여 태그 이름으로 요소를 찾겠다고 지정
4. 'a'는 찾고자 하는 태그 이름이고, 여기서는 첫 번째 <a> 태그(링크)를 검색
5. 찾은 요소는 tag 변수에 저장

```
from selenium.webdriver.common.by import By  
  
tag = driver.find_element(By.TAG_NAME, 'a')
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 요소 선택 (By.CSS_SELECTOR)

1. By.CSS_SELECTOR를 사용하여 CSS 선택자로 요소를 찾겠다고 지정
2. CSS 선택자: '#loginMy > div > div.box_g.box_login > div > a'
3. #loginMy: ID가 'loginMy'인 요소를 선택
4. > div: 'loginMy' 요소의 직계 자식 중 div 태그를 선택
5. > div.box_g.box_login: 다음 직계 자식 중 'box_g'와 'box_login' 클래스를 모두 가진 div를 선택
6. > div: 다시 그 아래의 div를 선택
7. > a: 최종적으로 그 아래의 a 태그(링크)를 선택
8. login =: 찾은 요소(로그인 버튼)를 login 변수에 저장.

```
from selenium.webdriver.common.by import By
```

```
# 로그인 버튼 찾기
```

```
login = driver.find_element(By.CSS_SELECTOR, '#loginMy > div > div.box_g.box_login > div > a')
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 요소 선택 (By.XPATH)

1. By.XPATH를 사용하여 XPath로 요소를 찾겠다고 지정
2. 절대 XPath를 사용하여 페이지 구조 내에서 정확한 위치의 요소를 선택
3. 주의사항: 절대 XPath는 웹사이트의 구조가 조금이라도 변경되면 작동하지 않을 수 있음

```
from selenium.webdriver.common.by import By

# 뉴스 메뉴 찾기
news = driver.find_element(By.XPATH,
                           '/html/body/div[2]/header/div[2]/div[1]/ul/li[4]/a')
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 요소 선택 (By.ID)

1. By.ID를 사용하여 ID 속성으로 요소를 찾겠다고 지정
2. 'q':찾고자 하는 요소의 ID 값
3. ID를 사용하여 요소를 찾는 것은 가장 빠르고 신뢰할 수 있는 방법 중 하나
4. ID를 사용하여 요소를 찾는 방식은 일반적으로 매우 안정적 (ID는 페이지 내에서 유일)
5. 다른 선택자에 비해 성능이 좋고 코드가 간결한 특징

```
from selenium.webdriver.common.by import By  
  
daum_search = driver.find_element(By.ID, 'q')
```

데이터 분석 기술

- 데이터 수집 – 3. 웹 크롤링 (파일명: crawling.ipynb)
 - Selenium – 요소 선택 (By.TAG_NAME)

1. By 클래스는 요소를 찾는 다양한 방법(예: ID, 클래스 이름, 태그 이름 등)을 제공
2. driver.find_element() 메서드를 사용하여 웹 페이지에서 요소를 찾을
3. By.TAG_NAME을 사용하여 태그 이름으로 요소를 찾겠다고 지정
4. 'a'는 찾고자 하는 태그 이름이고, 여기서는 첫 번째 <a> 태그(링크)를 검색
5. 찾은 요소는 tag 변수에 저장

```
from selenium.webdriver.common.by import By  
  
tag = driver.find_element(By.TAG_NAME, 'a')
```


데이터 분석 기술

- 웹 크롤링 (파일명: crawling.ipynb)

다음 포털에서 환율지표 정보를 가져와서 정리
<https://finance.daum.net/exchanges>

(실습 30분)