데이터사이언스를위한 지식관리시스템

3주차: csv, json, xml 데이터 알아보기 + 웹스크래핑

3주차 목표

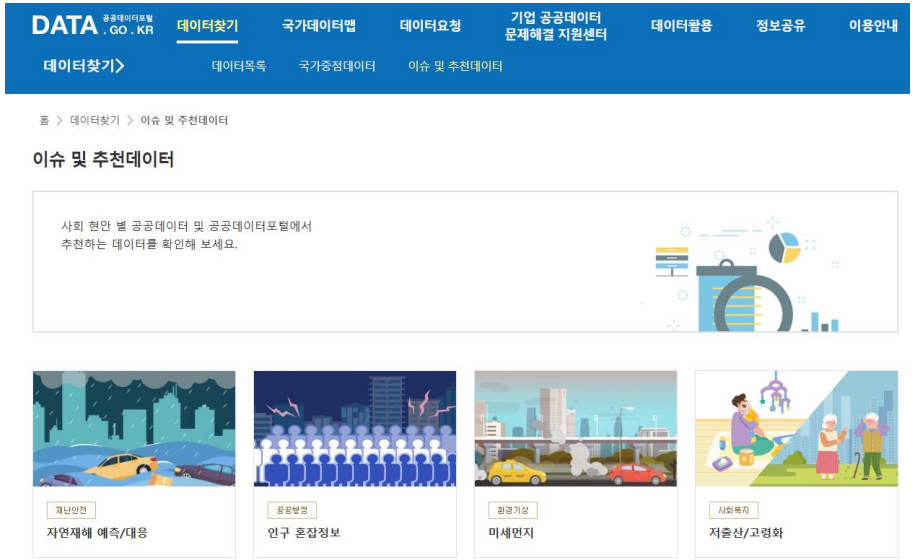
- 파일형태로 배포되는 대표적인 데이터 타입에 대해 알아봅니다
- 아래 데이터 타입의 장단점을 알아봅니다
- csv: comma separated value
- json: javascript object notation
- xml: extensible markup language

CSV

- Comma Separated Value
- 구분자가 다르면 다르게 부르기도 합니다
- 예를 들어 tab으로 구분되면 tsv
- 각 행은 하나의 레코드, 각 열은 데이터의 속성을 의미합니다
- 엑셀과 비슷해서 익숙하게 접근할 수 있습니다

csv파일 다운로드 받기

- 공공데이터 포털 www.data.go.kr 에서 쉽게 확인할 수 있습니다
- 예를 들어 데이터찾기 > 이슈 및 추천데이터 > 자연재해/대응 > 대피소 > 파일데이터 에서 찾을 수 있습니다



csv파일을 python에서 읽기

- 가장 간단한 방법은 pandas를 이용하는 것입니다
- 그러나 몇가지 주의사항이 있습니다
- 인코딩 타입 일치 여부, 헤더 존재 여부, 자료형이 의도된 대로 읽혔는지 확인 필요

```
1  import pandas as pd
```
✓ 0.5s

```
1  pd.read_csv('한국농어촌공사_하류하천 지역 비상대처 대피소 정보_20241231.csv', encoding='cp949')
```
✓ 0.0s

	본부명	지사명	표준시설코드	시설명	대피소명	소재지	전화번호	수용가능인원
0	경북	영덕.울진	4777010127	기사	한국농어촌공사	경상북도 영덕군 영해면 성내리	054-730-5064	NaN
1	경북	영덕.울진	4777010127	기사	황장리경로당	경상북도 영덕군 지품면 원전리	054-732-9834	42.0
2	경북	영덕.울진	4777010127	기사	신안출포경로당	경상북도 영덕군 지품면 신안리	054-733-5967	21.0
3	경북	영덕.울진	4777010127	기사	눌곡리마을회관	경상북도 영덕군 지품면 눌곡리	054-734-2214	26.0
4	경북	영덕.울진	4777010127	기사	지품중학교	경상북도 영덕군 지품면 신안리	054-732-3013	723.0
...	...	...	...	...	...	...	...	...
11695	제주	본부직할	5011010008	송당	송당리사무소	제주특별자치도 제주시 구좌읍 송당리	064-783-4093	NaN
11696	제주	본부직할	5011010008	송당	제주시청	제주특별자치도 제주시 이도이동	064-728-3755	NaN
11697	제주	본부직할	5011010008	송당	제주도청	제주특별자치도 제주시 연동	064-710-3671~8	NaN
11698	제주	본부직할	5011010008	송당	제주	제주특별자치도 제주시 이도이동	064-728-3755	NaN
11699	제주	본부직할	5011010008	송당	세화초등학교	제주특별자치도 제주시 구좌읍 세화리	064-783-2649	180.0

11700 rows × 8 columns

json

- Javascript Object Notation
- 오늘날 가장 많이 쓰이는 데이터 포맷을 꼽으라면 이 형식이 나올 것입니다
- 키:값 형태로 데이터를 저장하는 방법 중 하나 입니다
- csv에 비해 좋은 점은 자료형을 좀더 명확하게 서술할 수 있다는 것입니다

json파일을 python에서 읽기

- 2주차 강의자료에서 수집했던 json파일을 이용합니다
- 이렇게 읽은 데이터의 자료형은 dict가 됩니다
- Python의 dict 역시 키:값 형태의 자료형입니다

```python
import json

with open('test.json', 'r') as fp:
    data = json.load(fp)

data
```

✓ 0.0s


```
{'latitude': 52.52,
 'longitude': 13.419998,
 'generationtime_ms': 0.2065896987915039,
 'utc_offset_seconds': 0,
 'timezone': 'GMT',
 'timezone_abbreviation': 'GMT',
 'elevation': 38.0,
 'current_units': {'time': 'iso8601',
  'interval': 'seconds',
  'temperature_2m': '°C',
  'wind_speed_10m': 'km/h'},
 'current': {'time': '2025-09-19T21:45',
  'interval': 900,
  'temperature_2m': 19.0,
  'wind_speed_10m': 7.2},
 'hourly_units': {'time': 'iso8601',
  'temperature_2m': '°C',
  'relative_humidity_2m': '%',
  'wind_speed_10m': 'km/h'},
 'hourly': {'time': ['2025-09-19T00:00',
```

xml

- csv, json, xml 중에서 가장 대중적이지 않은 자료형을 꼽으라면 이 자료형입니다
- 공공데이터포털에서 조회되는 내용을 보면 이해할 수 있습니다
- 그럼에도 불구하고 이것을 다루는 이유는 여전히 이 데이터를 사용하는 경우가 있기 때문입니다

xml파일 다운로드 받기

- 공공데이터 포털 www.data.go.kr 에서 확인할 수 있으나 비중이 적어서 쉽게 찾아지지는 않습니다
- 데이터찾기 > 데이터목록 > xml로 검색하여 아래와 같이 찾을 수는 있습니다
- 왜 xml은 상대적으로 대중적이지 못한 것일까요?
- 데이터를 텍스트 에디터로 열어보면 가독성, 용량측면에서 csv나 json에 비해 불리함을 짐작할 수 있습니다

xml파일을 python에서 읽기

- 2주차 강의자료에서 수집했던 json파일을 이용합니다
- 이렇게 읽은 데이터의 자료형은 dict가 됩니다
- Python의 dict 역시 키:값 형태의 자료형입니다

```python
import xml.etree.ElementTree as ET

tree = ET.parse("GSTN_SCENE_PT_UTM52N.shp.xml")
root = tree.getroot()

print("최상위 태그:", root.tag)

for esri in root.findall("Esri"):
    create_date = esri.find("CreaDate").text
    create_time = esri.find("CreaTime").text
    sync_once = esri.find("SyncOnce").text

    print(f"CreateDate: {create_date}, CreateTime: {create_time}, SyncOnce: {sync_once}")

```

✓ 0.0s

```
최상위 태그: metadata
CreateDate: 20140612, CreateTime: 20091300, SyncOnce: FALSE
```

정리

- 우리가 데이터를 수집하면서 자주 다루게 될 데이터 타입에 대해 알아보았습니다

	CSV	JSON	XML
특징	comma로 필드 구분	키:값을 { }로 표현	키:값을 태그로 표현
장점	가독성이 좋음	자료형이 비교적 명확 키:값 형태이기 때문에 헤더 정보가 누락될 염려가 없음	자료형이 비교적 명확 키:값 형태이기 때문에 헤더 정보가 누락될 염려가 없음 주석을 지원함
단점	자료형이 모호할 수 있음 Header가 누락될 수 있음	주석이 지원되지 않음 csv에 비해서 상대적으로 가독성이 떨어짐	동일한 데이터를 표현하기 위해 큰 용량 필요 가독성이 떨어짐