학습 정리

팀	이진조	구성원	이규진

일정	발제자	주제
5/30 (목)	이규진	requests 모듈 기초, 웹 브라우저 없는 스크랩핑 및 파싱 실습

주요 내용 요약

섹션 4. 파이썬 다양한 데이터 형식 가공하기(1)

다양한 데이터 전송 형식 개요

- 데이터 전송 표준 형식 종류
- 바이너리(Binary) 데이터 vs 텍스트(Text) 데이터

• 종류	• 장점	• 단점
• 텍스트 데이터	• 텍스트 에디터 편집 가능, 가독성 좋음, 즉시 수정	• 크기가 큼
• 바이너리 데이터	• 크기 작음, 동영상, 이미지 등	• 에디터 편집 불가, 데이터 저장 영역의 위치 정의

- 바이너리(Binary), 텍스트(Text) 데이터 생성 실습

```
import pickle # (객채,텍스트) 직렬화, 역직렬화

# 파일 이름과 데이터
bfilename = 'C:/Users/student/Desktop/section4/test.bin'
tfilename = 'C:/Users/student/Desktop/section4/test.txt'
```

```
data1 = 77
data2 = "Hello, world!"
data3 = ["car", "apple", "house"]

# 바이너리 쓰기
with open(bfilename, 'wb') as f:
    pickle.dump(data1,f) # dumps(문자열 직렬화)
    pickle.dump(data2,f)
    pickle.dump(data3,f)

# 텍스트 쓰기
with open(tfilename, 'wt') as f:
    f.write(str(data1))
    f.write('\n')
    f.write(data2)
    f.write('\n')
    f.writelines('\n'.join(data3))

# 바이너리 읽기
with open(bfilename, 'rb') as f:
    b = pickle.load(f) # Loads(문자열 역직렬화)
    print(type(b), 'Binary Read1 | ',b)
    b = pickle.load(f)
    print(type(b), 'Binary Read2 | ',b)
    b = pickle.load(f)
    print(type(b), 'Binary Read3 | ',b)
# <class 'int'> Binary Read1 | 77
# <class 'str'> Binary Read2 | Hello, world!
# <class 'list'> Binary Read3 | ['car', 'apple', 'house']


# 텍스트 읽기
with open(tfilename, 'rt') as f:
    for i, line in enumerate(f,1):
        print(type(line), 'Text Read' + str(i) + ' | ',line, end='')
# <class 'str'> Text Read1 | 77
# <class 'str'> Text Read2 | Hello, world!
# <class 'str'> Text Read3 | car
# <class 'str'> Text Read4 | apple
# <class 'str'> Text Read5 | house
```

파이썬으로 XML 데이터 다루기

- XML 데이터 간단 개요
- XML 기상청 날씨 데이터 조회

```
import sys
import io
import urllib.request as req
from bs4 import BeautifulSoup
import os.path
```

```
sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding = 'utf-8')
sys.stderr = io.TextIOWrapper(sys.stderr.detach(), encoding = 'utf-8')

# 다운로드 url
url = "http://www.weather.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108"
savename = "C:/Users/student/Desktop/section4/forecast.xml"

if not os.path.exists(savename):
    req.urlretrieve(url, savename)
```

- XML 기상청 날씨 데이터 지역별 파싱 및 출력

```
# BeautifulSoup 파싱
xml = open(savename, 'r', encoding="utf-8").read()
soup = BeautifulSoup(xml, 'html.parser')

# 지역 확인
info = {}
for location in soup.find_all("location"): # find가 효율적이다.
    loc = location.find("city").string
    # print(loc)
    weather = location.find_all("tmn")
    # print(weather)
    if not (loc in info):
        info[loc] = []
    for tmn in weather:
        info[loc].append(tmn.string)
# print(info)

# 각 지역별 날씨 텍스트 쓰기
with open('C:/Users/student/Desktop/section4/forecast.txt', 'wt') as f:
    for loc in sorted(info.keys()):
        print("+",loc)
        f.write(str(loc)+'\n')
        for n in info[loc]:
            print("-",n)
            f.write('\t'+str(n)+'\n')
```