2강

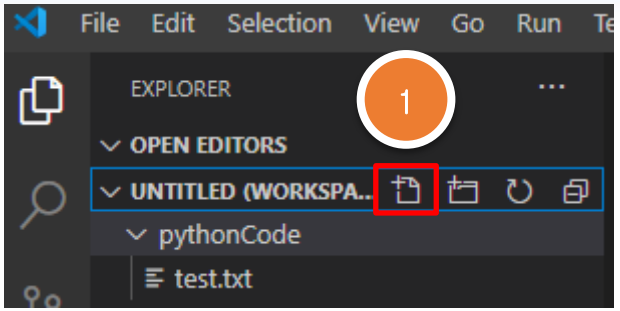
스크래이핑 기초

Urllib 활용

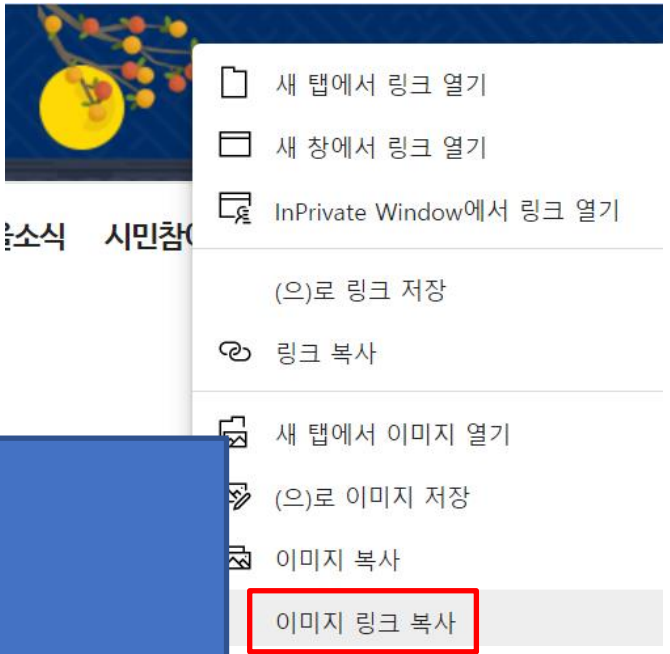
이해하기

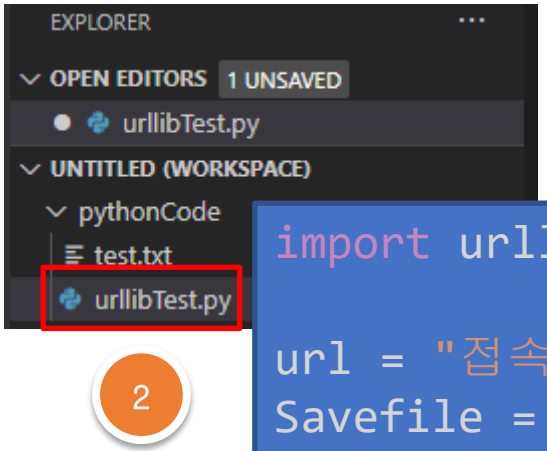
- ❖ 특징
  - 다운로드 – urllib.request.urlretrieve
  - 읽기 – urllib.request.urlopen.read
- ❖ API 이해
  - xxx의 인코딩 설정 – xxx.decode

파일 만들기



https://www.seoul.go.kr/





4

```
import urllib.request

url = "접속할 URL"
Savefile =  "저장할파일명"

urllib.request.urlretrieve(url, savefile)
```

메모리 데이터 저장

```
import urllib.request

url = "https://www.seoul.go.kr"

req = urllib.request

mem = req.urlopen(url).read()

print(mem)
```

현재 코드는 URL에 접속하여 웹을 다운받고 콘솔창에 출력하는 내용이다.
1에서 b는 binary를 의미하는 것으로 코드에서 ₩x9c₩xec와 같은 코드를 의미하며 기계어코드라고도 한다.

1

```
(base) root@a5581f69b902:/pythonCode# python urllibTest.py
b'<!DOCTYPE html>₩r₩n<html lang="ko">₩r₩n<head>₩r₩n<meta ch
4₩x9c₩xec₩x9a₩xb8₩xed₩x8a₩xb9₩xeb₩xb3₩x84₩xec₩x8b₩x9c</titl
 content="₩xec₩x84₩x9c₩xec₩x9a₩xb8₩xed₩x8a₩xb9₩xeb₩xb3₩x84₩
₩xb8, ₩xed₩x95₩xab₩xec₩x9d₩xb4₩xec₩x8a₩x88, ₩xeb₩xb6₩x84₩xe
```

인코딩

```
import urllib.request

url = "https://www.seoul.go.kr"

req = urllib.request

mem = req.urlopen(url).read()
#euc-kr or utf-8 실행
decodeMem = mem.decode("utf-8")
print(decodeMem)
```

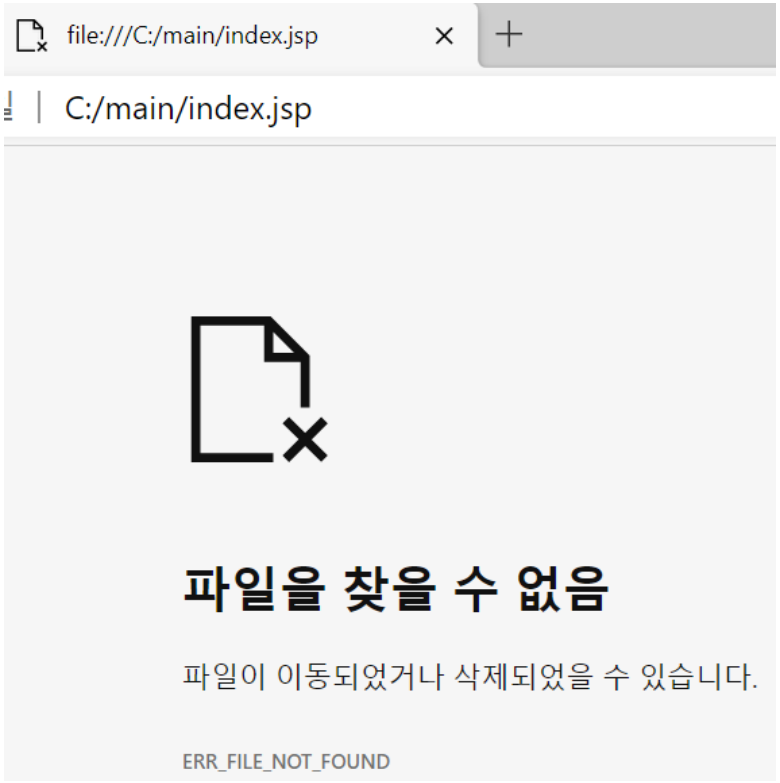
```
(base) root@a5581f69b902:/pythonCode# python urllibTest.py
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>서울특별시</title>
        <meta name="description" content="서울특별시 메인, 핫
```

파일 저장

```
import urllib.request

url = "https://www.seoul.go.kr"

req = urllib.request

mem = req.urlopen(url).read()
#euc-kr or utf-8 실행
decodeMem = mem.decode("utf-8")

#파일 생성
with open("seoul.html", mode="wb") as f:
    f.write(mem)
    print("파일 생성 완료")
```

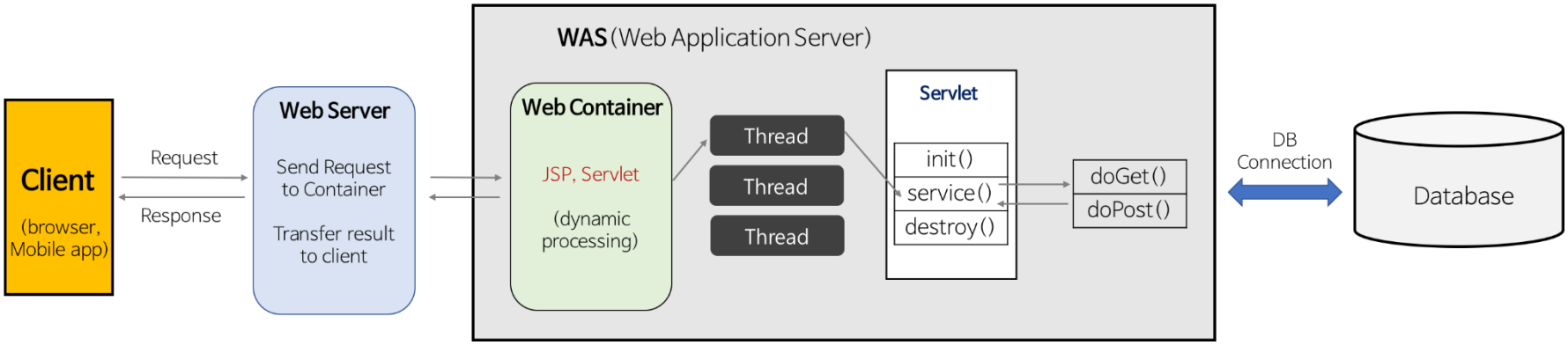
Quiz

❖ 이전 화면에서 동작시킨 seoul.html이 정상적으로 동작되도록 코드를 수정하시오



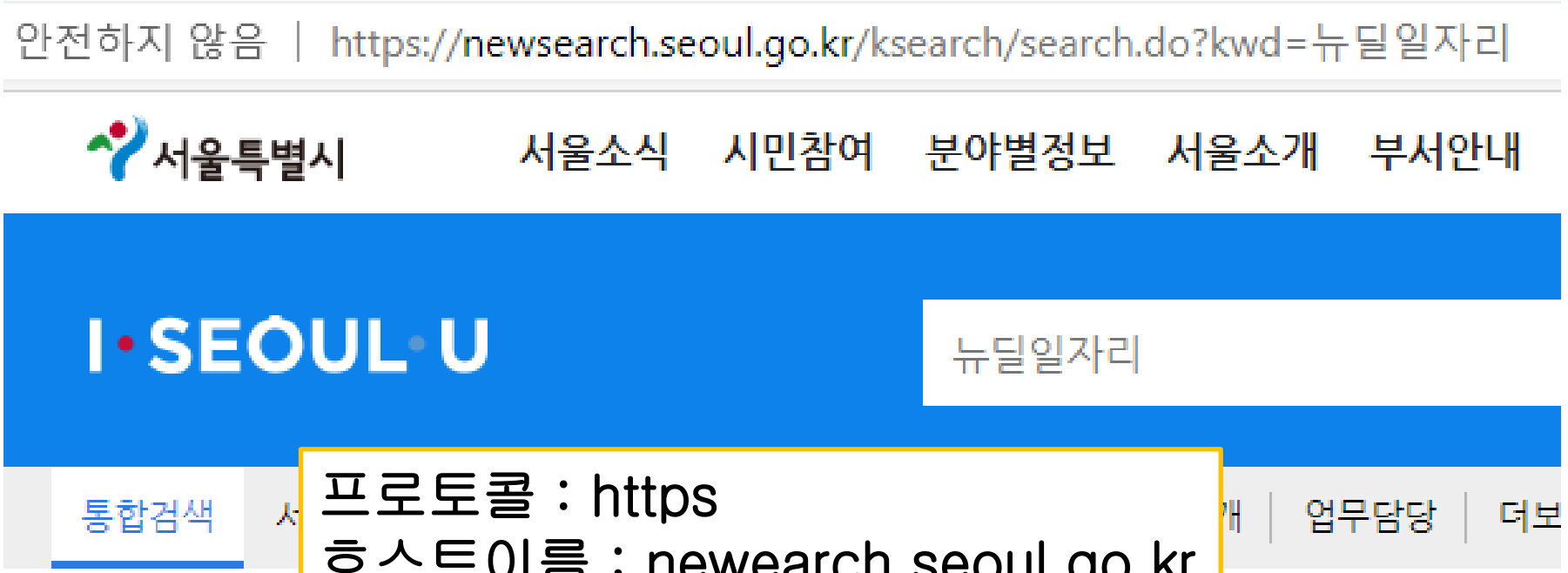
웹 구조 이해

Web Service Architecture



참조 : https://gmlwjd9405.github.io/2018/10/27/webserver-vs-was.html

URL 분석(get 방식)


안전하지 않음 | https://newsearch.seoul.go.kr/ksearch/search.do?kwd=뉴딜일자리
서울특별시
서울소식
시민참여
분야별정보
서울소개
부서안내
I·SEOUL·U
뉴딜일자리
통합검색
업무담당
더보
프로토콜 : https
호스트이름 : newearch.seoul.go.kr
경로 : ksearch/search.do
데이터 : wkd=뉴딜일자리

URL 분석

https://search.naver.com/search.naver?sm=top_hty&fbm=1&ie=utf8&query=빅데이터

프로토콜 : https
호스트이름 : search.naver.com
경로 : /search.naver
데이터
  sm=top_hty
  &fbm=1
  &ie=utf8
  &query=빅데이터

데이터 얻어오기 - encoding

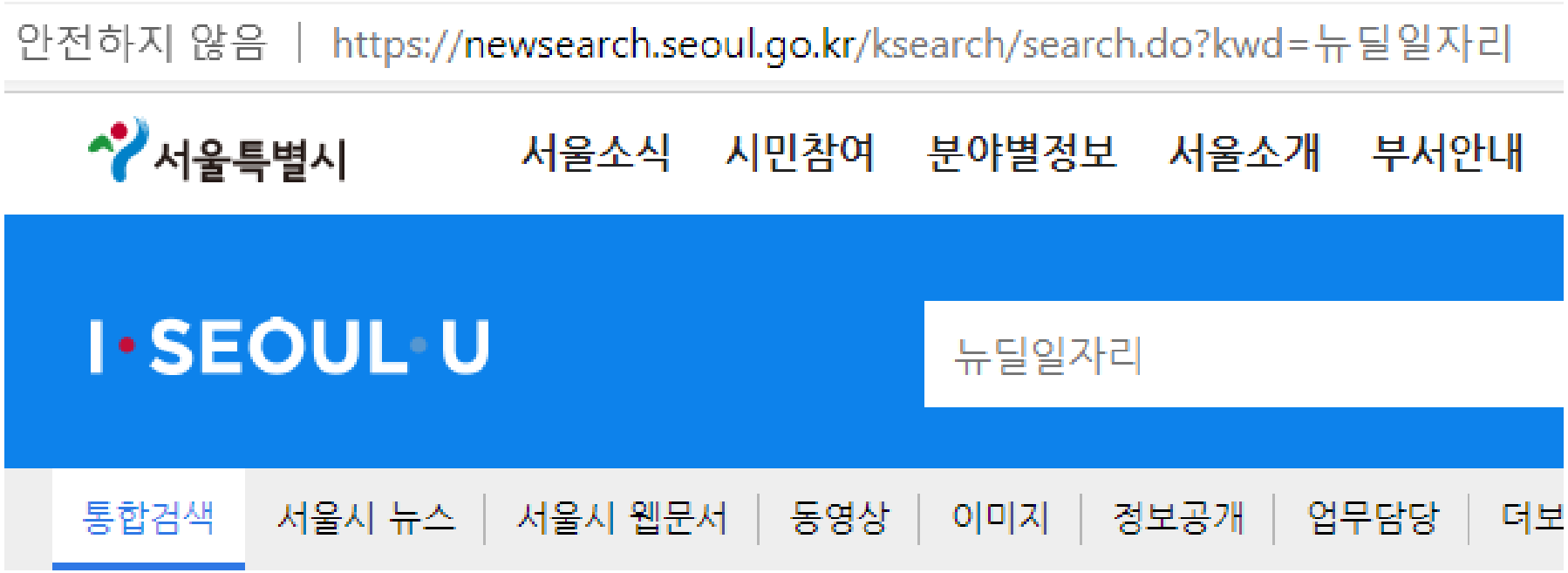
```
import urllib.request

uri = "https://search.naver.com/search.naver"
values={
    "sm":"top_hty",
    "fbm":"1",
    "ie":"utf8",
    "query":"빅데이터"
}
dataEncode = urllib.parse.urlencode(values)

url = uri + "?"+dataEncode

print(url)
```

데이터 얻어 오기

```
import urllib.request

uri = "https://search.naver.com/search.naver"
values={
    "sm":"top_hty",
    "fbm":"1",
    "ie":"utf8",
    "query":"빅데이터"
}
dataEncode = urllib.parse.urlencode(values)

url = uri + "?"+dataEncode

data = urllib.request.urlopen(url).read()

print(data.decode("utf-8"))
```

Quiz

❖ 다음과 같은 내용이 저장될 수 있도록 코딩하시오



scraping

이해하기

❖ 용어 이해

- 스크래이핑 : 스크레이퍼를 사용하여 금속면을 정밀하게 다듬질하는 작업이다.

평면용

구멍용

참고 : https://terms.naver.com/entry.nhn?docId=345803&cid=44616&categoryId=44616

❖ Beautifulsoup

- HTML이나 XML의 데이터 추출 용이

BeautifulSoup 기본 이해

```
from bs4 import BeautifulSoup

html = """
<html><body>
    <h1>스크래핑이란?</h1>
    <p>웹 페이지를 분석하는 것</p>
    <p>원하는 부분을 추출하는 것</p>
</body></html>
"""

soup = BeautifulSoup(html, "html.parser")

print(soup.html.body.h1 )
print(soup.html.body.h1.string)
```

```
<h1>스크래핑이란?</h1>
스크래핑이란?
```

Id 정보 추출하기

```
from bs4 import BeautifulSoup

html = """
<html><body>
    <h1 id="title">스크래핑이란?</h1>
    <p id="body">웹 페이지를 분석하는 것</p>
    <p>원하는 부분을 추출하는 것</p>
</body></html>
"""

soup = BeautifulSoup(html, "html.parser")

title = soup.find(id="body")
print(title, " : ", title.string )
```

CSS 선택자 사용하기

- API
  - select_one : 요소 하나 추출
  - select : 여러 개의 리스트 추출
- **추출방법**
  - 태그 접근 : select("[태그명]")
  - ID 접근 : select("[#ID명]") or : select("[태그명#ID명]")
  - Class 접근 : : select("[.class명]") or : select("[태그명.class명]")
  - 하위 접근 : select("[상위태그명] > [하위태그명]")

실습

```
from bs4 import BeautifulSoup

html = """
<html><body>
    <h1 id="title">스크래핑이란?</h1>
    <p id="body">웹 페이지를 분석하는 것</p>
    <p>원하는 부분을 추출하는 것</p>
</body></html>
"""
soup = BeautifulSoup(html, "html.parser")
#title 추출
title = soup.select_one("body > h1")
print(title, " : ", title.string )
#p 추출
pList = soup.select("p")
for p in pList:
    print(p.string)
```

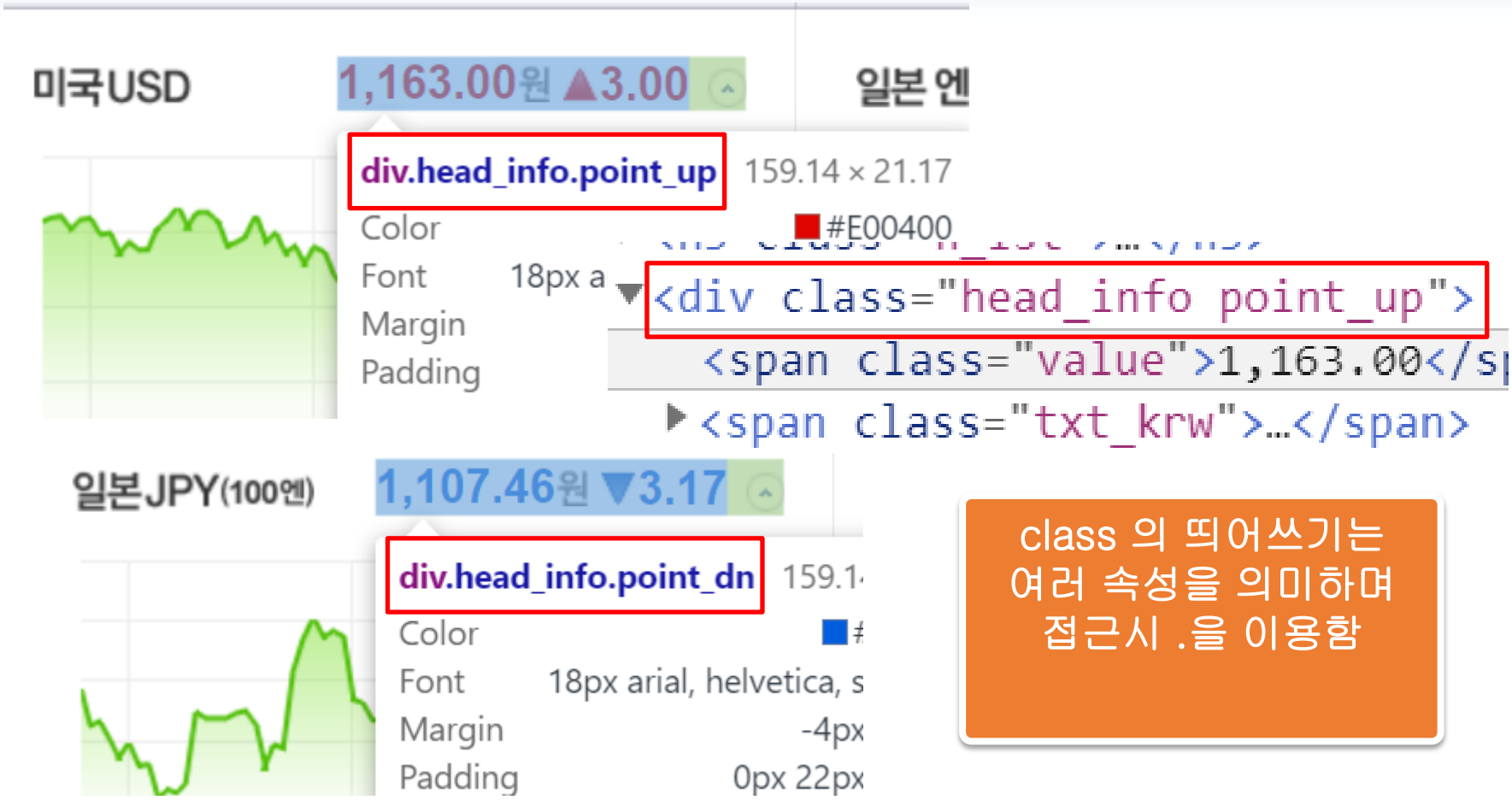
시장지표 – 환전고시환율 얻기

❖ https://finance.naver.com/marketindex/

데이터 추출

```
from bs4 import BeautifulSoup
import urllib.request as req
url="https://finance.naver.com/marketindex/"
html = req.urlopen(url)

soup = BeautifulSoup(html, "html.parser")

#value 추출
value = soup.select_one("span.value")
print(value, " : ", value.string )

#p 추출
values = soup.select("span.value")
for v in values:
    print(v.string)
```

상승한 정보만 추출



class 의 띄어쓰기는
여러 속성을 의미하며
접근시 .을 이용함

Quiz

❖ 국제시장 환율에서 상승 정보만 출력하시오

기사 추출하기

실습

❖ https://news.daum.net/

Anchor 태그의 href 얻기

```
values = soup.select("div.box_peruse div.pop_news.pop_cmt ol li")
for v in values:
    print(v.a.attrs["href"])
```

```
<ol class="list_popcmt">
  <li>
    <a href="https://news.v.daum.net/v/20200924093303542"
    "link_txt" data-tiara-layer="article" data-tiara-id=
    "20200924093303542" data-tiara-type="harmony" data-tiara
    "1" data-tiara-custom="contentUniqueKey=hamny-
    20200924093303542">…</a>
      <span class="info_news">SBS</span>
    </li>
  <li>…</li>
  <li>…</li>
  <li>…</li>
  <li>…</li>
```

기사 정보 관련 DIV 검색

기사정보 추출

```
from bs4 import BeautifulSoup
import urllib.request as req

def getURLInfo(url, tag):
    html = req.urlopen(url)
    soup = BeautifulSoup(html, "html.parser")
    return soup.select(tag)

url="https://news.daum.net/"
tag = "div.box_peruse div.pop_news.pop_cmt ol li"
values = getURLInfo(url, tag)

for v in values:
    articleURL = v.a.attrs["href"]
    articleTag = "#harmonyContainer"
    data = getURLInfo(articleURL, articleTag)

    print(data)
    print("============================")
    print(articleURL)
```

기사정보 추출 - delay

```
from bs4 import BeautifulSoup
import urllib.request as req
import time
def getURLInfo(url, tag):
    html = req.urlopen(url)
    soup = BeautifulSoup(html, "html.parser")
    return soup.select(tag)

url="https://news.daum.net/"
tag = "div.box_peruse div.pop_news.pop_cmt ol li"
values = getURLInfo(url, tag)

for v in values:
    articleURL = v.a.attrs["href"]
    articleTag = "#harmonyContainer"
    data = getURLInfo(articleURL, articleTag)
    print(data)
    print("============================")
    print(articleURL)
    time.sleep(1)
```

문자만 출력

```
from bs4 import BeautifulSoup
import urllib.request as req
import time
def getURLInfo(url, tag):
    html = req.urlopen(url)
    soup = BeautifulSoup(html, "html.parser")
    return soup.select(tag)


url="https://news.daum.net/"
tag = "div.box_peruse div.pop_news.pop_cmt ol li"
values = getURLInfo(url, tag)

for v in values:
    articleURL = v.a.attrs["href"]
    articleTag = "#harmonyContainer"
    data = getURLInfo(articleURL, articleTag)
    print(data[0].text)
    time.sleep(1)
```

안티 크롤링

직접적 접속 불가

```
from bs4 import BeautifulSoup
import urllib.request as req

url="https://news.naver.com/"
html = req.urlopen(url)
```

```
File "scrapping.py", line 11, in <module>
  html = req.urlopen(url)
File "/opt/conda/lib/python3.7/urllib/request.py", line 222, in urlopen
  return opener.open(url, data, timeout)
File "/opt/conda/lib/python3.7/urllib/request.py", line 531, in open
  response = meth(req, response)
File "/opt/conda/lib/python3.7/urllib/request.py", line 641, in http_response
  'http', request, response, code, msg, hdrs)
File "/opt/conda/lib/python3.7/urllib/request.py", line 569, in error
  return self._call_chain(*args)
File "/opt/conda/lib/python3.7/urllib/request.py", line 503, in _call_chain
  result = func(*args)
File "/opt/conda/lib/python3.7/urllib/request.py", line 649, in http_error_default
  raise HTTPError(req.full_url, code, msg, hdrs, fp)
rllib.error.HTTPError: HTTP Error 500: Internal Server Error
```

안티 크롤링 회피

참조 : https://book.coalastudy.com/data_crawling/week3/stage3

requests를 통한 우회

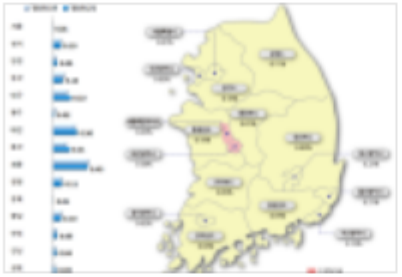
```
from bs4 import BeautifulSoup
import requests

urlHeader=requests.get("https://news.naver.com", headers={'User-Agent':'Mozilla/5.0'})
html = BeautifulSoup(urlHeader.text, "html.parser")
print(html.text)
```

Quiz(헤드라인 뉴스 글들을 추출하시오)

❖ https://news.naver.com/main/main.nhn?mode=LSD&mid=shm&sid1=101

헤드라인 뉴스 Beta

18 서울 집값 16주째 • 전셋값 65주째 상승 이어져 >



서울 집값 16주째, 전셋값 65주째 상승 이어져

서울 집값이 16주째, 서울 전셋값이 65주째 상승을 이어갔다. 24일 한국감정원이 발표한 주간아파트 가격동향에 따르면 이달 셋째 주(21일) 기준 서울 ...

조선비즈 | 30+

벌써 5주째... 오르지도, 내리지도 않는 서울 아파트값 한국일보 | 10+

수도권 아파트 전셋값 59주 연속 상승...일부지역 상승폭 확대 연합뉴스

"집값 언제 떨어지나"...서울은 버티고, 수도권은 더 올라 서울경제