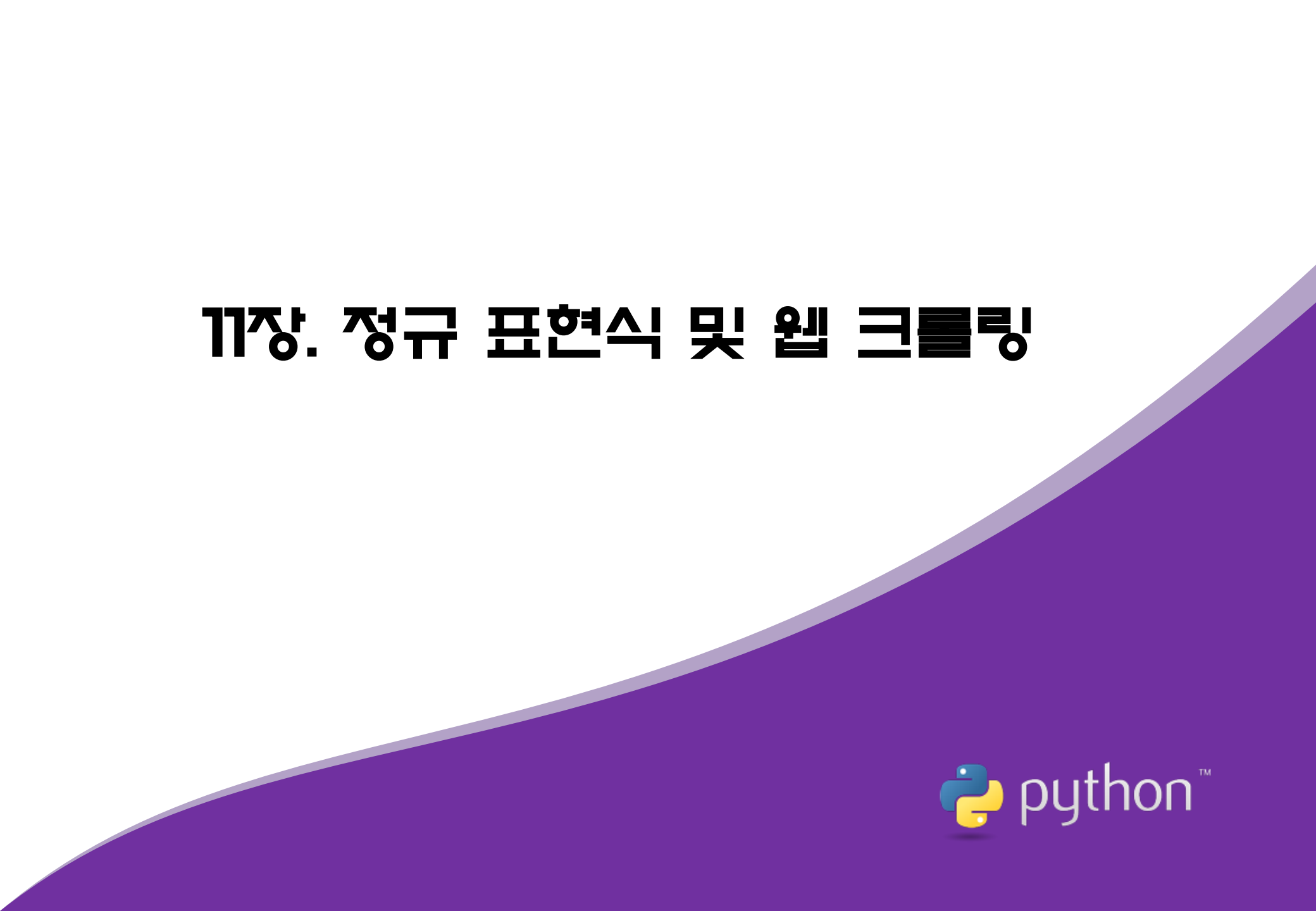
11장. 정규 표현식 및 웹 크롤링



정규식 – Regular Expression

❖ ***정규표현식이란?***

- 특정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는 형식 언어이다. 문자열의 검색과 치환을 지원한다.

정규식 – Regular Expression

- 자주 사용하는 정규 표현식

표현식	설 명
^	정규식 시작
$	정규식 끝
^[0-9]*$	숫자
^[a-zA-Z]*$	영문 대, 소문자
^[가-힣]*$	한글
^010[-](d{3}|\d{4})[-]\d{4}$	휴대폰
^\d{6}[-][1-4]{6}$	주민등록번호

정규식 – Regular Expression

- 정규표현식에 사용되는 메타문자

메타문자	설 명(사용 예)
[ ]	대괄호는 [ ]사이의 문자들과 일치함, [x]
-	문자의 범위를 지정하는 하이픈(-), [1-4]
^	부정을 나타내는 캐럿, [^0-9]
*****	0번 이상 반복, 1번 이상 반복(+)
{m}	m은 반복횟수, {3,4} – 3개 또는 4개
( )	소괄호는 서브 클래스. 그룹을 만들 때 사용
\d	숫자 – [0-9]
\w	알파벳 + 숫자
\s	공백

정규표현식 지원 – re 모듈

- 정규 표현식 활용

1. re.compile(**'[a-z]+'**) : 정규 표현식을 컴파일 한다.
2. match("**korea**") : 문자열의 시작 부분에서 정규 표현식과 일치하는 부분을 찾음

<match 객체의 주요 메서드>

메서드	기 능
group()	매치된 문자열을 돌려준다.
start()	매치된 문자열의 시작위치를 돌려준다.
end()	매치된 문자열의 끝위치를 돌려준다
span()	매치된 문자열의 (시작, 끝)에 해당하는 튜플 반환.

정규식을 사용한 문자열 검색

- 정규 표현식 활용

```
import re

pat = re.compile("[a-z]") #정규 표현식
mat = pat.match("korea")   #조사할 문자열
print(mat)
print(mat.group())
print(mat.start())
print(mat.end())
print(mat.span())

if mat:
    print('문자열 있음: ', mat.group())
else:
    print('문자열 없음')
```

정규식을 사용한 문자열 검색

- 정규 표현식 활용
  - ✓ 메타 문자 * 과 +의 차이

```
# *은 0개이상, +는 1개 이상
pat = re.compile("a*b")
mat = pat.match("b")  #aaab
# print(mat)
if mat:
    print('문자열 있음: ', mat.group())
else:
    print('문자열 없음')
```

정규식을 사용한 문자열 검색

- 유효성 검사
  - ✓ fullmatch() 함수 – 문자열 전체가 정규 표현식과 일치하는지를 찾음

```
# 전화번호 검증
# phone_pat = re.compile('010-\d{3,4}-\d{4}')
phone_pat = re.compile("010-[0-9]{3,4}-[0-9]{4}")
mat = phone_pat.fullmatch("010-12-5678")
print(bool(mat)) #False

# 한글과 전화번호 패턴 검사
name_pat = "제갈수연";
pat = re.compile("[가-힣]{2,5}")
mat = pat.fullmatch(name_pat)
print(bool(mat)) #True
```

정규식을 사용한 문자열 검색

- 유효성 검사 예제

```
# 전화번호 패턴 유효성 검사
def validate_phone_number(phone):
    """전화번호 유효성 검사 (010-XXXX-XXXX 형식)"""
    phone_pat = re.compile("010-\d{3,4}-\d{4}")
    return bool(phone_pat.fullmatch(phone))

phone_list = [
    "010-1234-5678",  # 유효
    "010-123-4567",   # 유효
    "010-12-5678",    # 무효
    "012-1234-5678",  # 무효
    "01012345678",    # 무효
    "010-1234-567"    # 무효
]

print("=== 전화번호 검증 결과 ===")
for phone in phone_list:
    print(f"{phone}: {validate_phone_number(phone)}")
```

정규식을 사용한 문자열 검색

- 유효성 검사 예제

```
# 한글이름 패턴 유효성 검사
def validate_name(user_name):
    pattern = re.compile("^[가-힣]{2,5}$")
    return bool(pattern.fullmatch(user_name))

while True:
    user_name = input("한글 이름 입력(2~5자): ")

    if validate_name(user_name):
        print(f"이름: {user_name}")
        break
    else:
        print("올바른 한글 이름이 아닙니다. 다시 입력하세요")
```

그루핑(Grouping)

- 그루핑(Grouping)

  문자열 중에서 특정 부분의 문자열만 추출하고 싶을 때 사용한다.

  소괄호()를 사용해서 그룹을 구분한다.

group(인덱스)	설 명
group(0)	매치된 전체 문자열
group(1)	첫 번째 그룹에 해당하는 문자열
group(2)	두 번째 그룹에 해당하는 문자열
group(n)	n 번째 그룹에 해당하는 문자열

그루핑(Grouping)

- 이름과 전화번호를 구분하여 문자열 추출

```
# 그룹 - 소괄호()
phone = "jang 010-1234-5678"
pat = re.compile("(\w+)\s{1,2}(010-\d{3,4}-\d{4})")
mat = pat.match(phone)
print(mat.group())
print(mat.group(1)) #jang
print(mat.group(2)) #010-1234-5678
```

그루핑(Grouping)

- **sub()를 사용한 문자 마스킹 처리**

sub(\g <그룹 인덱스>)

```
# 폰번호 뒷 4자리 마스킹 처리
pattern = re.compile("(\w+)\s{1,2}(010-\d{3,4})-\d{4}")

print(pattern.sub("\g<1>", phone)) #jang
print(pattern.sub("\g<2>-****", phone)) #010-1234-****
```

그루핑(Grouping)

- **sub()를 사용한 문자 마스킹 처리**

sub(\g <그룹 인덱스>)

```
# 주민등록번호 마스킹 처리
data = """
kim 920815-1234567
lee 031011-4123456
"""

pat = re.compile("(\d{6})[-]\d{7}")
print(pat.sub("\g<1>-*******", data))

pat2 = re.compile("(\d{6}[-]\d{1})\d{6}")
print(pat2.sub("\g<1>******", data))
```

```
kim 920815-*******
lee 031011-*******


kim 920815-1******
lee 031011-4******
```

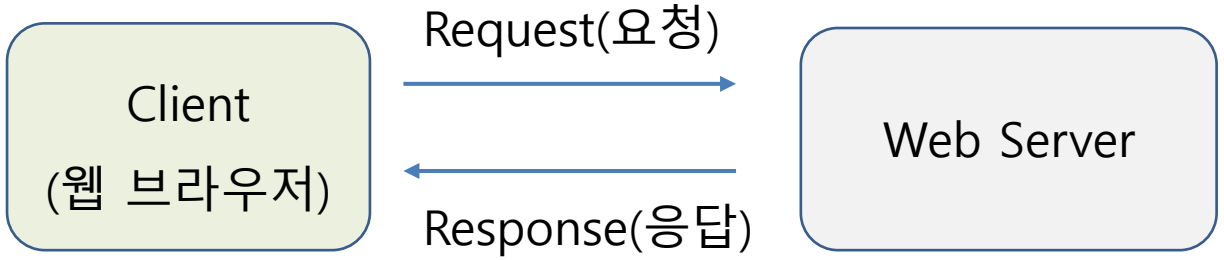
웹 스크래핑 = 웹 크롤링

➢ ***웹 스크래핑(Scraping)이란?***

인터넷에 있는 웹 페이지를 방문해서 자료를 수집하는 일.

웹 크롤링이라고도 한다.

▶ 웹 서버에 요청하고 응답받기



웹 스크래핑 = 웹 크롤링

▷ requests 모듈(라이브러리)

Python 프로그래밍 언어용 HTTP 라이브러리이다.

HTTP 프로토콜을 이용하여 웹 사이트로부터 데이터를 송수신할 수 있다.

url 요청 - requests.get(url)

```
import requests

url = "https://www.python.org"
response = requests.get(url)  # url 객체 저장
print(response)
print(response.status_code)
html = response.text  # html 코드 저장
# print(html)

url2 = "https://www.python.org/3"
response = requests.get(url2)
print(response)
```

```
<Response [200]>
<Response [404]>
```

정상

페이지 없음

로봇 배제 표준

▷ 로봇 배제 표준

로봇 배제 표준이란?

웹사이트에 로봇이 접근하는 것을 방지하기 위한 규약 robots.txt에 기술하고 있다.

- 로봇에 의한 접근이 허용되는 경우라도 웹 서버에 무리가 갈 만큼 반복적으로 웹 페이지를 요청하는 것과 같이 서비스 안정성을 해칠 수 있는 행위를 하지 않아야 함
- 크롤링(또는 스크래핑)으로 취득한 자료를 임의로 배포하거나 변경하는 등의 행위는 저작권을 침해할 수 있으므로 저작권 규정을 준수해야 함

로봇 배제 표준

▷ 로봇 배제 표준

템플릿 태그	설 명
User-agent: * Disallow: /	모든(*) 로봇(검색엔진 봇)에게 루트 디렉터리(/) 이하 모든 문서에 대한 접근을 차단한다.
User-agent: * Allow: /	모든(*) 로봇에게 루트 디렉터리(/) 이하 모든 문서에 대한 접근을 허락한다.
User-agent: * Disallow: /temp/	모든(*) 로봇에게 특정 디렉터리(/temp/)에 대한 접근을 차단한다.

✓ **User-agent: 지침을 적용할 크롤러 이름**

로봇 배제 표준

▷ 로봇 배제 표준 – python.org

https://www.python.org/robots.txt

```
# Directions for robots.  See this URL:
# http://www.robotstxt.org/robotstxt.html
# for a description of the file format.

User-agent: HTTrack
User-agent: puf
User-agent: MSIECrawler
Disallow: /

# The Krugle web crawler (though based on Nutch) is OK.
User-agent: Krugle
Allow: /
Disallow: /~guido/orlijn/
Disallow: /webstats/

# No one should be crawling us with Nutch.
User-agent: Nutch
Disallow: /

# Hide old versions of the documentation and various large
User-agent: *
Disallow: /~guido/orlijn/
Disallow: /webstats/
```

HTTrack, puf, MSIECrawler 같은 **사이트 복제용 툴**이 웹사이트 전체를 긁어가지 못하게 차단하는 역할

로봇 배제 표준

▷ 로봇 배제 표준

```python
import requests

urls = ["https://www.naver.com/", "https://www.python.org/"]
filename = "robots.txt"

# print(urls[0] + filename)

for url in urls:
    url_path = url + filename
    print(url_path)
    response = requests.get(url_path)
    print(response)
```

```
https://www.naver.com/robots.txt
<Response [200]>
https://www.python.org/robots.txt
<Response [200]>
```

HTML이란?

- **HTML(HyperText Markup Language)**
  - 하이퍼텍스트를 마크업 하는 언어, HTML5(현재 버전)
  - **하이퍼텍스트** : 웹 사이트에서 링크를 클릭해 다른 문서나 사이트로 이동하는 기능
  - **마크업** : **tag**(태그)를 사용해 문서에서 어느 부분이 제목이고 본문인지, 어느 부분이 사진이고 링크인지 표시하는 명령어(코드)

- **HTML**의 역사

  인물 : **팀버너스리** – 웹의 아버지, 영국의 컴퓨터 과학자

  WorldWideWeb(월드와이드웹) 하이퍼텍스트 시스템 고안 – cern(유럽 입자 물리연구소)에서 개발됨.

  URL, HTML, HTTP 최초 설계, W3C 창립

웹브라우저와 웹편집기

- **웹 브라우저(Web Browser)**
  - 웹 사이트를 둘러 볼때 사용하는 프로그램
  - 크롬, 익스플로러, 파이어 폭스, 사파리 등

- **웹 편집기**
  - **비주얼 스튜디오 코드(VS code)**,

- **HTML 태그(Tag)**
  - 웹 문서의 내용중 '<' 와 '>'로 묶인 부분을 태그라 한다.
  - 대부분의 태그는 여는 태그와 닫는 태그로 구성된다. <h2>~</h2>, ~
  - 태그는 속성과 함께 사용할 수 있다. <태그 속성="속성값" 속성="속성값" ...>

    <img src="images/first.jpg" width="300" height="350" alt="시계이미지">

HTML 태그(tag)

◆ HTML 태그(Tag)

태그	설명
<!DOCTYPE html>	현재 문서가 HTML5 언어로 작성된 웹 문서라는 뜻 문서 유형을 지정.
<html>~</html>	웹 문서의 시작과 끝을 나타내는 태그
<head>~</head>	웹 브라우저가 웹 문서를 해석하기 위해 필요한 정보들을 입력하는 부분. <meta>태그 – 문자 세트를 비롯한 문서 정보 <title>태그 – 브라우저의 제목 표시줄에 표시
<body>~</body>	실제로 웹 브라우저 화면에 나타날 내용

HTML 태그(tag)

◆ HTML 태그(Tag)

태그	설명
<hn>	Headline을 뜻하고.. 제목 표시 1~6까지 있고, 1이 가장 크고 6이 가장 작은 크기 제목
<p>	paragraph의 줄임말로 '단락'이란 앞 뒤에 줄바꿈이 있는 텍스트 덩어리
<img>	**<img src="경로" 속성="값">** **alt 속성** - 이미지를 설명하는 대체 텍스트이다. **width, height 속성** - 이미지의 크기 (너비, 높이)를 조정할 수
<a>	<a>태그는 클릭했을 때 다른 페이지로 넘어가는 기능을 가지고 있다. 텍스트
	목록을 나타내는 태그이다.
	목록안의 항목(item)을 표시하는 태그이다.

웹페이지 만들기

- **웹 페이지 만들기**

웹 기초 기술

HTML - 웹 페이지의 구조를 담당

CSS - 웹페이지의 디자인을 담당

Javascript(자바스크립트) - 웹 페이지의 인터렉티브를 담당

청와대 홈페이지로 이동

웹페이지 만들기

- **웹 페이지 만들기**

```
<!DOCTYPE html>
<html>
<head>
    <!-- 한글 인코딩 -->
    <meta charset="utf-8">
    <title>웹페이지 만들기</title>
</head>
<body>
    <h1>웹 기초 기술</h1>
    <hr>
    <p>HTML - 웹 페이지의 구조를 담당</p>
    <p>CSS - 웹페이지의 디자인을 담당</p>
    <p>Javascript(자바스크립트) - 웹 페이지의 인터렉티브를 담당</p>

    <!-- 이미지 삽입 -->
    <img src="white-owl.jpg" alt="흰부엉이" width="640" height="480"><br><br>

    <!-- 하이퍼 링크 -->
    <a href="https://www.opencheongwadae.kr/mps" target="_blank">
        청와대 홈페이지로 이동</a>
</body>
</html>
```

웹 스크레이핑 = 웹 크롤링

- **html 목록 태그**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>목록 만들기</title>
</head>
<body>
    <!-- ul(unordered list) : 순서없는 목록 -->
    <ul class="item">
        <li>인공지능</li>
        <li>빅데이터</li>
        <li>로봇공학</li>
    </ul>
    <ul class="comlang">
        <li>Python</li>
        <li>C/C++</li>
        <li>Java</li>
    </ul>
</body>
</html>
```

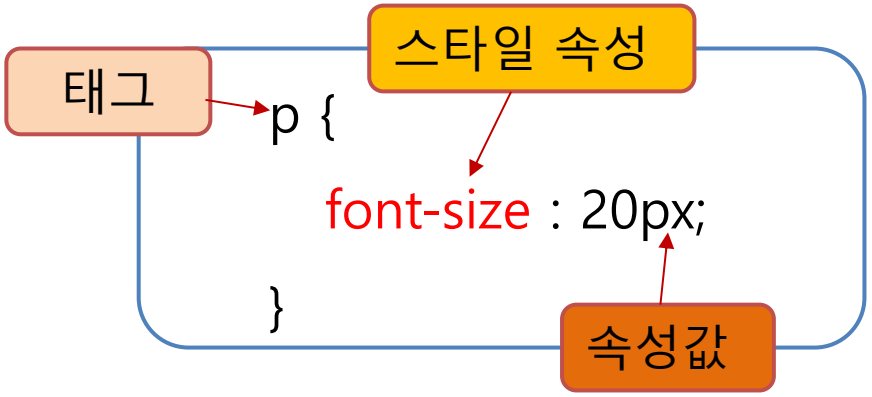
- 인공지능
- 빅데이터
- 로봇공학

- Python
- C/C++
- Java

CSS(Cascading Sytle sheets)

CSS(Cascading Style Sheets)란?

- **CSS**는 HTML과 함께 웹 표준 기술이다.
- **HTML**이 텍스트나 이미지, 표 같은 각 요소를 문서에 넣어 뼈대를 만드는 것이라면 **CSS**는 텍스트 색상이나 크기, 이미지 크기나 위치, 표 색상, 배치 방법 등 웹 문서의 디자인 요소를 담당한다.
- **스타일 형식**



세미콜론(;)으로 구분하여 중괄호{ }안에 나열한다.

선택자(Selector)

선택자(Selector)란?

스타일의 속성을 적용하는 요소를 '선택자(selector)'라고 부른다. 이 선택자는 태그 하나가 될 수도 있지만 여러 개의 요소를 묶어 별도의 선택자로 지정할 수도 있다.

- **태그 선택자**

  문서에서 특정 태그를 사용한 모든 요소에 스타일 적용

- **id 선택자, class 선택자**

  ✓ **id 선택자** : 문서 안에서 **한 번만 사용**한다면 id 선택자로 정의

  #(샵) 다음에 id 이름 지정

  ✓ **class 선택자** : 문서 안에서 **여러 번 반복할 스타일**이라면 클래스 선택자로 정의.

  마침표(.) 다음에 클래스 이름 지정

id & class 선택자

레드향

껍질에 붉은 빛이 돌아 레드향이라 불린다.
레드향은 한라봉과 귤을 교배한 것으로 일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.
비타민 C와 비타민 P가 풍부해 혈액순환, 감기예방 등에 좋은 것으로 알려져 있다.

```css
/* id 선택자 - 유일하게 스타일 적용 */
#container{
  width: 500px;
  border: 1px solid #333;
  padding: 10px;
}
/* class 선택자 - 여러 곳에 같은 스타일 적용*/
.redtext{
  color: red;
}
```

selector.html

```html
<div id="container">
  <h1>레드향</h1>
  <p>껍질에 붉은 빛이 돌아 <span class="redtext">레드향</span>이라 불린다.<br>
    <span class="redtext">레드향</span>은 한라봉과 귤을 교배한 것으로
    일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.<br>
    비타민 C와 비타민 P가 풍부해 혈액순환, 감기예방 등에 좋은 것으로 알려져 있다.</p>
</div>
```

웹 스크레이핑 = 웹 크롤링

❖ BeautifulSoup 라이브러리(모듈)

HTML과 XML 문서를 파싱하기 위한 파이썬 라이브러리이다.

웹 서버로 부터 HTML 소스코드를 가져온 다음에는 HTML 태그 구조를 해석하기 위한 과정이 필요하다.

HTML 소스 코드를 해석하는 것을 파싱(parsing)이라고 부른다.

▶ BeautifulSoup 설치

pip install BeautifulSoup4

▶ BeautifulSoup 사용

from bs4 import BeautifulSoup

웹 스크레이핑 = 웹 크롤링

❖ BeautifulSoup 라이브러리(모듈)

- ✓ soup.find(태그)

  처음 나오는 태그로 찾기

- ✓ soup.find_all(태그)

  태그에 해당하는 모든 요소 찾아서 리스트로 반환함

- ✓ soup.find(태그, attrs={'class': css_selector})

  태그에 해당하는 선택자로 찾기

웹 스크레이핑 = 웹 크롤링

- **html 태그 크롤링**

```
from bs4 import BeautifulSoup

html_str = """
<!DOCTYPE html>
<html>
<body>
    <ul class="item">
        <li>인공지능</li>
        <li>빅데이터</li>
        <li>로봇공학</li>
    </ul>
    <ul class="lang">
        <li>Python</li>
        <li>C/C++</li>
        <li>Java</li>
    </ul>
</body>
</html>
"""
```

웹 스크레이핑 = 웹 크롤링

- **find(), find_all() 사용**

```
soup = BeautifulSoup(html_str, "html.parser")
# print(soup)
#find('ul') - 처음 나오는 ul 태그 찾음
first_ul = soup.find('ul')
print(first_ul)
print(first_ul.text)

# findAll('li') - 결과를 리스트로 반환함
all_li = first_ul.findAll('li')
print(all_li)
print(all_li[2])
print(all_li[2].text)
```

```
<ul class="item">
<li>인공지능</li>
<li>빅데이터</li>
<li>로봇공학</li>
</ul>

인공지능
빅데이터
로봇공학

[<li>인공지능</li>, <li>빅데이터</li>, <li>로봇공학</li>]
<li>로봇공학</li>
로봇공학
```

웹 스크레이핑 = 웹 크롤링

- **find(), find_all() 사용**

```
# 두번째로 나오는 ul 태그 찾기
# attrs 속성의 클래스(class) 선택자로 찾음
second_ul = soup.find('ul', attrs={'class': 'lang'})
print(second_ul)
all_li = second_ul.find_all('li')
print(all_li)
print(all_li[0])
print(all_li[0].text)
```

dictionary 자료구조
{키 : 값}

```
<ul class="lang">
<li>Python</li>
<ul class="lang">
<li>Python</li>
<li>C/C++</li>
<li>Python</li>
<li>C/C++</li>
<li>C/C++</li>
<li>Java</li>
</ul>
[<li>Python</li>, <li>C/C++</li>, <li>Java</li>]
<li>Python</li>
Python
```

웹 스크레이핑 = 웹 크롤링

❖ **주요 검색 함수**

- ✓ **select_one(태그이름.선택자이름)**

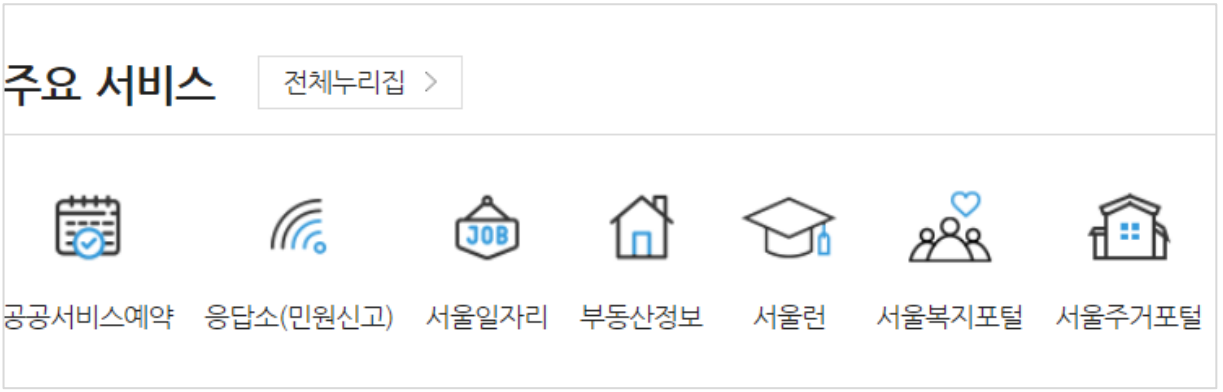
  첫번째 요소로 찾기

- ✓ **select(태그이름.선택자이름 > 하위 태그)**

  태그에 해당하는 모든 요소 찾기

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기



웹 브라우저 >우클릭 > 검사(단축키 F12)

```
▼<div class="m_service"> == $0
  ▼<ul> flex
    ▼<li class="public">
      ▼<a href="//yeyak.seoul.go.kr" onclick="action_logging({tr_code:'serv
        1'});" target="_blank" title="새창">
          <i class="ico_service"></i>
          "공공서비스예약"
        </a>
      </li>
    ▼<li class="answer">
      ▶<a href="//eungdapso.seoul.go.kr" onclick="action_logging({tr_code:'
```

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

```
url = "https://www.seoul.go.kr/main/index.jsp"
response = requests.get(url)
html = BeautifulSoup(response.text, 'html.parser')

# find()로 찾기
first_li = html.find('li', attrs={'class': 'public'})
print(first_li)
print(first_li.text)
```

```
<li class="public">
<a href="//yeyak.seoul.go.kr" onclick="action_logging({tr_code:'service01'});"
예약</a>
</li>

공공서비스예약
```

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

```
# find_all()
div = html.find('div', attrs={'class': 'm_service'})
# print(li)
# print(li.text)
all_li = div.find_all('li')
# print(all_li)

for li in all_li: # 태그 없는 텍스트만 모두 출력
    print(li.text)
print(all_li[1].text)
```

공공서비스예약

응답소(민원신고)

서울일자리

부동산정보

서울런

서울복지포털

서울주거포털

청년몽땅정보통

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

```
# select_one('태그이름.선택자이름')
first_li = html.select_one('li.public')
print(first_li)
print(first_li.text)

# '>' - 자식 선택자, ' '(공백) - 후손 선택자
# all_li = html.select('div.m_service > ul > li')
all_li = html.select('div.m_service ul li')
# print(all_li)

# 메뉴 텍스트 출력
for li in all_li:
    print(li.text)

print(all_li[1].text)
print(all_li[-1].text)
```

실습1. 국립중앙박물관 관람 정보

- 국립 중앙 박물관 관람 정보

  홈페이지 : https://www.museum.go.kr/

  관람 정보 > 관람 안내 > 검사(F12)

실습1. 국립중앙박물관 관람 정보

- 국립 중앙 박물관 관람 정보

```
url = "https://www.museum.go.kr/MUSEUM/contents/M0101000000.do?menuId=tour-guidance"
response = requests.get(url)
html = BeautifulSoup(response.text, 'html.parser')

# 관람안내
# select_one()
first_ul = html.select_one('ul.display-content')
print(first_ul)
print(first_ul.text)
```

실습1. 국립중앙박물관 관람 정보

- 국립 중앙 박물관 관람 정보

```
# select()로 찾기
contents = html.select('ul.display-content-area > li > ul')
print(contents)

for content in contents:
    print(content.text)

# 관람시간
print(contents[0].text)

# 휴관일 및 휴실일
print(contents[1].text)

# 관람료
print(contents[2].text)
```

KBS 뉴스 기사

- 뉴스 기사 크롤링하기

  홈페이지 : https://news.kbs.co.kr/

  KBS > 뉴스 > 메인 기사

KBS 뉴스 기사

- 뉴스 기사 크롤링하기

```
# 메인 기사 스크랩
url = "https://news.kbs.co.kr/news/pc/view/view.do?ncd=8295309"
response = requests.get(url)
html = BeautifulSoup(response.text, 'html.parser')

# 제목 스크랩
title = html.select_one("h4.headline-title")
print(title)
print(title.text)

# 내용 스크랩
content = html.select_one('div.detail-body')
print(content.text.strip())
```

KBS 뉴스 기사

- 뉴스 기사 크롤링하기
  - 제목 스크랩

```
<h4 class="headline-title">트럼프 “4일부터 각국에 관세 서한 보낼 것”</h4>
트럼프 “4일부터 각국에 관세 서한 보낼 것”
```

  - 내용 스크랩

```
[앵커] 다음 주 상호 관세 유예 종료를 앞두고 트럼프 대통령이 4일부터 각국에 관세 서한을 보낼 거라고 말했습니다.
복잡하다며 간단한 거래를 하는 편이 낫다는 겁니다. 워싱턴 김지숙 특파원입니다. [리포트] 상호 관세 유예 종료를 앞
장 내일부터 각국에 관세 서한을 보낼 거라고 했습니다. 170여 개국과 상대하고 있는데 얼마나 많은 합의를 할 수 있
겁니다. [도널드 트럼프/미국 대통령/화면출처:폭스TV 유튜브 : "아마 내일부터 여러 나라에 편지를 보내기 시작할 것
데, 그 편지에는 미국에서 사업을 하려면 얼마를 내야 하는지가 적혀 있을 것입니다."] 스콧 베센트 미 재무장관도 합
니다. 최선의 합의를 얻을 수 있다고 생각해 마지막 순간까지 기다리는 국가들은 기존에 책정된 상호 관세율이 적용될
/미 재무장관/CNBC 인터뷰 : "제가 다른 언론 인터뷰에서도 경고했듯, 이 국가들은 조심해야 합니다. 왜냐하면 그들의
있기 때문입니다."] 상호 관세 유예 연장 가능성에 대해선 결승선을 통과해야 할 시점에 공개적으로 연장하겠다고 말하
해 모호성을 견지하면서, 각국에 미국과의 합의를 서두를 것을 압박한 걸로 풀이됩니다. 그러면서 유예 기간이 끝나기
```

KBS 뉴스 기사

- 데이터 프레임 만들기

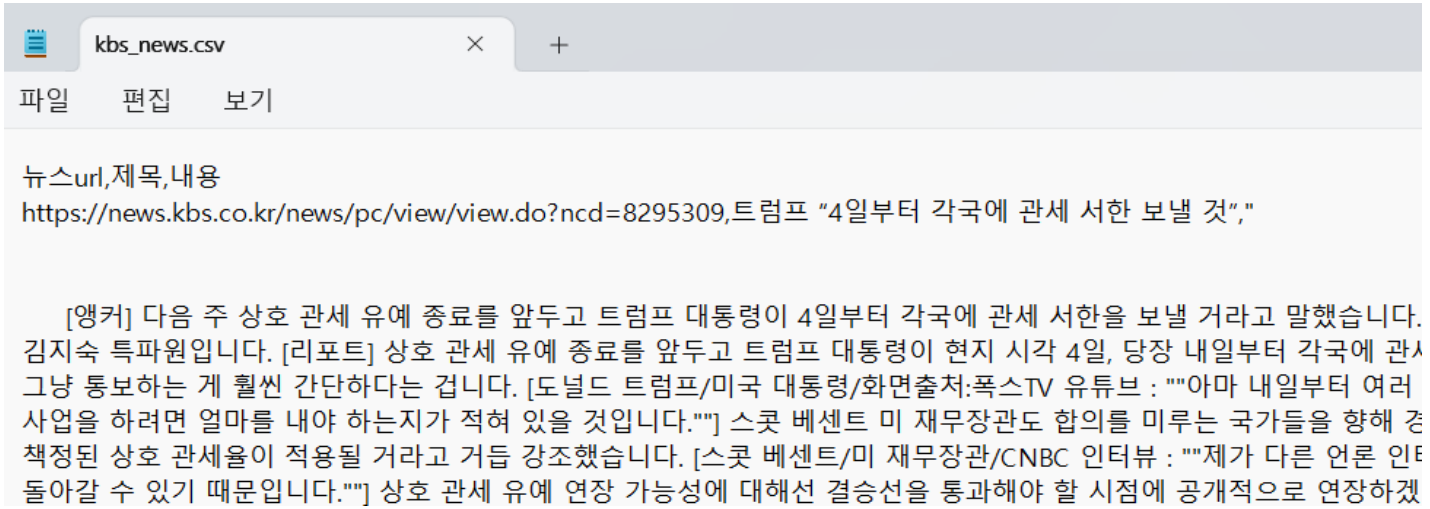
```
data = {
  '뉴스url': [url],
  '제목': [title.text],
  '내용': [content.text]
}

df = pd.DataFrame(data)
# print(df)

# csv 파일로 만들기
df.to_csv('kbs_news.csv', index=False)

# csv 파일 읽기
news = pd.read_csv('kbs_news.csv')
print(news)
```

KBS 뉴스 기사

- csv 파일 만들기

csv 파일이란? 쉼표로 값을 구분하는 텍스트 파일 형식이다.
각 행은 레코드를 나타내며, 쉼표로 구분된 값들은 해당 레코드의 필드를 의미함.
CSV 파일은 엑셀과 같은 스프레드시트 프로그램에서 데이터를 저장하거나 다른 프로그램 간에 데이터를 교환할 때 주로 사용됨

kbs_news.csv



실습2. 전자 신문 메인 기사 크롤링

1) 전자 신문 사이트에 접속한다.
2) robots.txt를 확인한다.
3) 메인 화면 기사를 크롤링한다.
   (1) 제목 가져오기
   (2) 발행일 가져오기
   (3) 본문 내용 가져오기

실습2. 전자 신문 메인 기사 크롤링

Headline NEWS

이재명 대통령, 국무회의 열고 '31.8조 추경' 의결…“최대한 신속 집행”

이재명 대통령은 5일 국무회의를 주재하고 추가경정예산(추경)안을 심의·의결했다.

ㄴ 2차 추경 1.3조 늘린 31.8조 확정…소비쿠폰 최대 55만…

ㄴ 이재명 대통령 “AI·반도체 투자 아끼지 않겠다”

이재명 대통령, 국무회의 열고 '31.8조 추경' 의결…“최대한 신속 집행”
발행일 : 2025-07-05 11:16

소비쿠폰, 소득 따라 최대 55만원 지원법무부 등 4곳 특수활동비 105억원 반영
(끝)hihong@yna.co.kr홍해인 기자 = 김민석 국무총리가 5일 서울 용산 대통령실 청사에서 열린 국무회의에서 인사말을 하고 있다. 2025.7.5

이재명 대통령은 5일 국무회의를 주재하고 추가경정예산(추경)안을 심의·의결했다.

강유정 대통령실 대변인은 이날 이재명 대통령이 오전 용산 대통령실에서 국무회의를 주재하고 국회를 통과한 추경안을 심의·의결했다고 밝

이 대통령은 국무회의에서 “새 정부의 첫 추경”이라며 “이번 추경은 매우 어려운 국민 경제 상황을 고려해 긴급하게 편성했다”고 말했다.

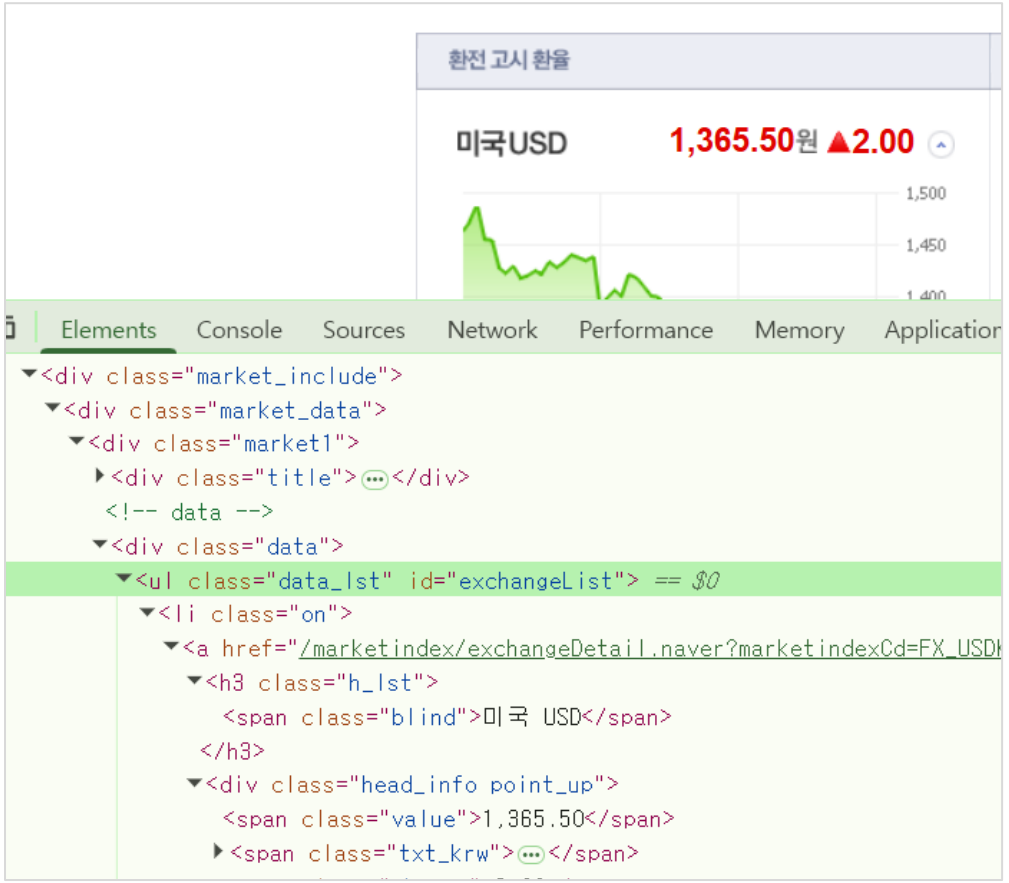
그러면서 “하루라도 빨리 집행돼 국민의 삶에 마중물이 될 수 있도록 해야 해서 주말에 (국무회의를) 갑자기 열었다”며 “최대한 신속하게 집
다해달라”고 당부했다.

이어 행정안전부의 민생회복 소비쿠폰 집행계획을 보고받고 “지급 과정에서 혼선이 발생하지 않게 실무적으로 잘 챙겨달라”고 지시했다.

네이버 금융 크롤링하기

- **환율정보 수집하기**

  네이버 > 증권 > 시장지표 > 환전 고시 환율




```
USD 1,366.30
JPY(100엔) 944.39
EUR 1,607.52
CNY 190.57
```

네이버 금융 크롤링하기

- 환율정보 수집하기 – find() 사용하여 첫번째 환율 찾기

```
resp = requests.get("https://finance.naver.com/marketindex/")
soup = BeautifulSoup(resp.text, "html.parser")

# find()
first_ul = soup.find('ul', attrs={'class': 'data_lst'})
# print(first_ul)
first_li = first_ul.find('li')
print(first_li)

# 환율 종류
exchange = first_ul.find('span', attrs={'class': 'blind'})
print(exchange.text)
print(exchange.text.split(" ")[1]) #미국 USD
```

네이버 금융 크롤링하기

- 환율정보 수집하기 – find_all() 사용

```
# 환율 지수
value = first_ul.find('span', attrs={'class': 'value'})
print(value.text) #1,366.60
print(exchange.text, value.text)

# find_all() - 전체 환율 찾기
all_li = first_ul.find_all('li')
# print(all_li)

for li in all_li:
  exchange = li.find('span', attrs={'class': 'blind'})
  value = li.find('span', attrs={'class': 'value'})
  print(exchange.text.split(" ")[-1], value.text)
```

네이버 금융 크롤링하기

- 환율정보 수집하기 – select() 사용

```
all_li = soup.select("div.market1 ul li")
# all_li = soup.select("ul.data_lst li") #차이 비교
# print(all_li)

# 환율 종류 - select_one() : 1개 선택
exchange = soup.select_one("span.blind")
# print(exchange.text) #미국 USD

# 환율 지수
value = soup.select_one("span.value")
# print(value.text) #1,388.80

# 환율 전체 출력
for li in all_li:
  exchange = li.select_one("span.blind")
  value = li.select_one("span.value")
  # 공백문자로 텍스트 분리 - 리스트 반환
  # text 대신 string 가능
  print(exchange.string.split(' ')[-1], value.string)
```

로또 당첨 번호 확인 앱

➢ 로또 당첨번호 가져오기

동행 복권 > 당첨 결과

로또 당첨 번호 확인 앱

➢ 로또 당첨번호 가져오기


```
import requests
from bs4 import BeautifulSoup

num = 1178 # 회차
url = f"https://dhlottery.co.kr/gameResult.do?method=byWin&drwNo={num}"
response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

win_nums = soup.select('div.nums span')
print(win_nums)
```

로또 당첨 번호 확인 앱

➢ 로또 당첨번호 가져오기

```
<div class="win_result">
  <h4>...</h4>
  <p class="desc">(2025년 06월 28일 추첨)</p>
  <div class="nums">
    <div class="num win"> == $0
      <strong>당첨번호</strong>
      <p>
        <span class="ball_645 lrg ball1">5</span>
        <span class="ball_645 lrg ball1">6</span>
        <span class="ball_645 lrg ball2">11</span>
        <span class="ball_645 lrg ball3">27</span>
        <span class="ball_645 lrg ball5">43</span>
        <span class="ball_645 lrg ball5">44</span>
        ::after
      </p>
    </div>
    <div class="num bonus">
      ::before
      <strong>보너스</strong>
      <p>
        <span class="ball_645 lrg ball2">17</span>
        ::after
      </p>
```

로또 당첨 번호 확인 앱

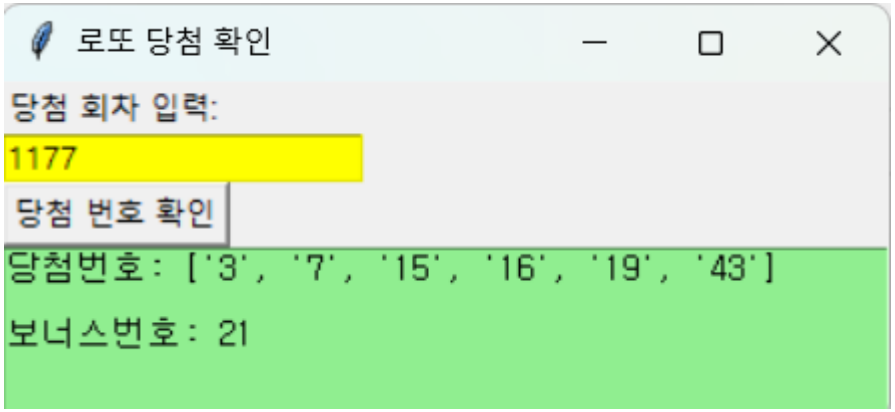
- 로또 당첨번호 가져오기

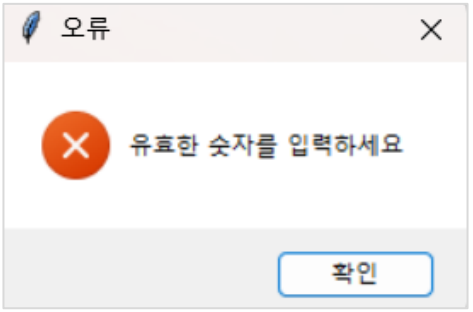
```
win_num_list = []  #당첨 번호 리스트
for num in win_nums:
  print(num.text)
  win_num_list.append(num.text)

print("당첨 번호")
print(win_num_list[:-1])

print("보너스 번호")
print(win_num_list[-1])
```

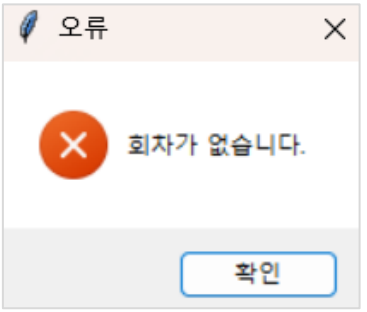
```
당첨 번호
['5', '6', '11', '27', '43', '44']
보너스 번호
17
```

로또 당첨 번호 확인 앱

➢ 로또 당첨 번호 확인 앱







로또 당첨 번호 확인 앱

➢ 로또 당첨 번호 확인 앱

```
window = Tk()
window.title("로또 당첨 확인")

Label(window, text="당첨 회차 입력: ") \
  .grid(row=0, column=0, sticky=W)
entry = Entry(window, bg="yellow")
entry.grid(row=1, column=0, sticky=W)

Button(window, text="당첨 번호 확인", command=lotto_win) \
  .grid(row=2, column=0, sticky=W)
output = Text(window, bg="lightgreen", width=50, height=5)
output.grid(row=3, column=0, sticky=W)

window.mainloop()
```

로또 당첨 번호 확인 앱

➢ 로또 당첨 번호 확인 앱

```
def lotto_win():
    try:
        num = int(entry.get()) #입력된 회차
        if num <= 0 or num > 1179:
            messagebox.showerror("오류", "회차가 없습니다.")
            entry.delete(0, END)  #입력 상자 초기화
            output.delete(0.0, END) #출력 상자 초기화
            return

        # 동행 복권 사이트 - 크롤링
        url = f"https://dhlottery.co.kr/gameResult.do?method=byWin&drwNo={num}"
        response = requests.get(url)
        soup = BeautifulSoup(response.text, 'html.parser')
        win_nums = soup.select('div.nums span')
```

로또 당첨 번호 확인 앱

➢ 로또 당첨 번호 확인 앱

```
    win_num_list = []  #당첨 번호 리스트
    for num in win_nums:
        win_num_list.append(num.text)

    # 출력
    output.delete(0.0, END)
    output.insert(END, f"당첨번호: {win_num_list[:-1]} \
                       \n\n보너스번호: {win_num_list[-1]}")
except ValueError:
    messagebox.showerror("오류", "유효한 숫자를 입력하세요")
    entry.delete(0, END)
    output.insert(END, "오류")
```