

13장. 웹 크롤링 및 서버(Server)

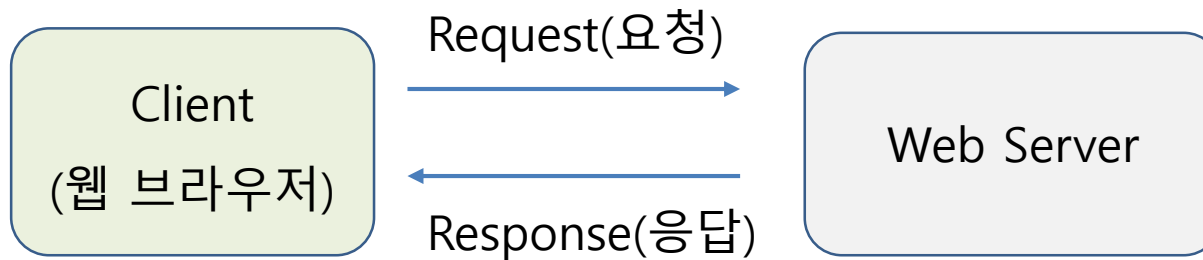


웹 스크래핑 = 웹 크롤링

➤ 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.

로봇 배제 표준

▷ 로봇 배제 표준

```
← → ↻ 🔒 google.com/robots.txt

User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
```

```
← → ↻ 🔒 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/
```

로봇 배제 표준

▷ 로봇 배제 표준

```
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)

- HTML의 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>, <p>~</p>
- 태그는 속성과 함께 사용할 수 있다. <태그 속성="속성값" 속성="속성값" ...>

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
<h1>	Headline을 뜻하고.. <h1>제목 표시</h1> 1~6까지 있고, 1이 가장 크고 6이 가장 작은 크기 제목
<p>	paragraph의 줄임말로 '단락'이란 앞 뒤에 줄바꿈이 있는 텍스트 덩어리
	 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 인터랙티브를 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 인터랙티브를 담당</p>

  <!-- 이미지 삽입 -->
  <br><br>

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

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 찾기
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)
```

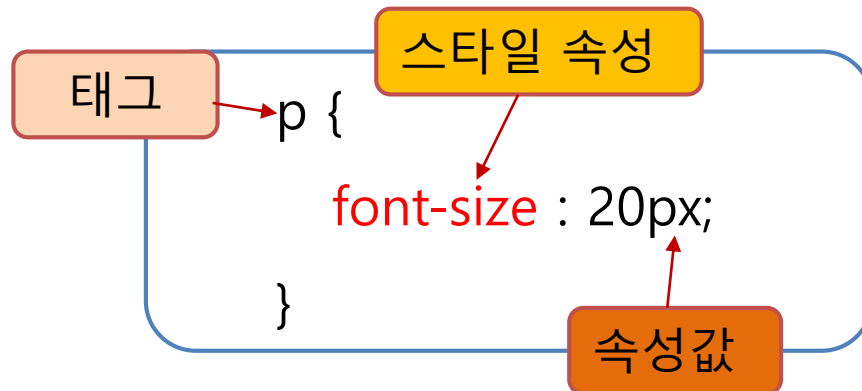
Dictionary 자료구조
{키 : 값}

```
<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
```

CSS

CSS(Cascading Style Sheets)란?

- CSS는 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.

전체/태그 선택자

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

- #(샷)** 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.

```
/* 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

```
<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>
```

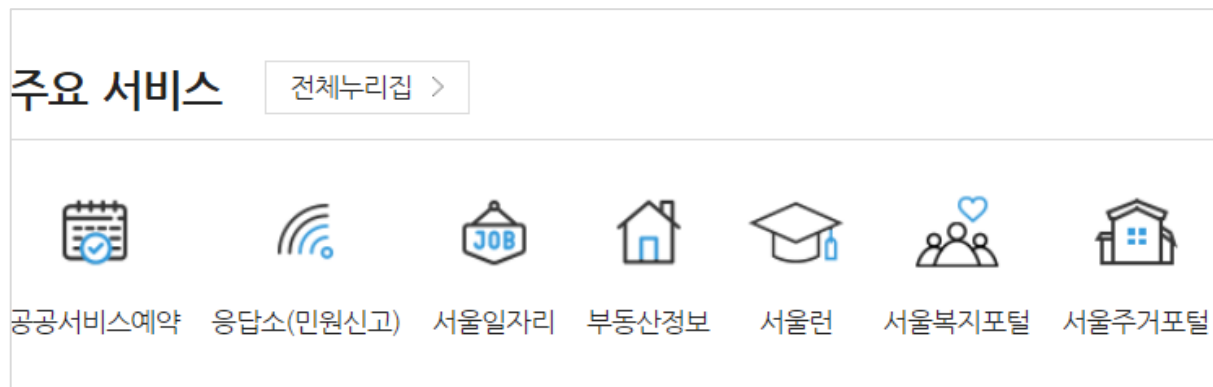
웹 스크레이핑 = 웹 크롤링

❖ 주요 검색 함수

- ✓ `select_one(태그이름.선택자이름)`
첫번째 요소로 찾기
- ✓ `select(태그이름.선택자이름)`
태그에 해당하는 모든 요소 찾기

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기



웹 브라우저 > 우클릭 > 검사(단축키 F12)



서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

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

# 첫번째 메뉴 찾기
first_li = html.select_one('li.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

```
# 모든 항목 찾기
all_li = html.select('div.m_service > ul > li')
print(all_li)

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

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

공공서비스예약

응답소(민원신고)

서울일자리

부동산정보

서울런

서울복지포털

서울주거포털

청년몽땅정보통

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

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

# 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:
    print(li.text)
print(all_li[1].text)
```

실습1. 국립중앙박물관 관람 정보

- 1) 국립중앙박물관 사이트에 접속한다.
- 2) robots.txt를 확인한다.
- 3) 관람시간과 관람료를 크롤링한다.

관람시간

월, 화, 목, 금, 일요일: 10:00 ~ 18:00 (입장 마감: 17:30)

수, 토요일: 10:00 ~ 21:00 (입장 마감: 20:30)

· 옥외 전시장(정원)은 오전 7시부터 오후 10시까지 관람하실 수 있습니다.

관람료

무료

상설전시관, 어린이박물관, 무료 특별전시 해당

유료

유료 특별전시 해당

관람권 구입하는 곳: 특별전시실 1 앞 매표소

관람권 판매시간: 관람 종료 30분 전까지

KBS 뉴스 기사

➤ 뉴스 기사 크롤링하기



```
import requests
from bs4 import BeautifulSoup

# 메인 기사 스크랩
url = "https://news.kbs.co.kr/news/pc/view/view.do?ncd=8287486"
response = requests.get(url)
html = BeautifulSoup(response.text, 'html.parser')
```

KBS 뉴스 기사

➤ 뉴스 기사 크롤링하기

```
# 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())
```

[앵커] 내란특검이 어제 저녁 윤석열 전 대통령에 대해 특수공무집행방해죄 등으로 체포영장을 청구했습니다. 자세한 내용 알아보겠습니다. 신현욱 기자, 체포영장을 법원이 계속 검토하고 있는 거죠? [기자] 네, 발부 여부 나올 수 있습니다. 체포영장 심사는 통상 수사기관이 밀행성을 유지하기 때문에 구속영장과는 다르게 피의자가 죄를 범했다고 의심할 만한 상당한 이유가 있는지, 정당한 이유 없이 출석요구에 응하지 않았는지를 기준으로 전 대통령에 대한 체포영장을 어제 오후 5시 50분쯤 청구했다고 밝혔습니다. 체포영장에는 윤 전 대통령이 지하고, 계엄에 가담한 군사령관들의 비화폰 기록을 삭제하라고 지시한 혐의가 담겼습니다. 내란특검은 윤 전 대통령 석 요구에 모두 응하지 않았으며, 이후 소환에도 응하지 않을 의사를 밝혔다고 설명했는데요. 윤 전 대통령 측은

KBS 뉴스 기사

➤ 뉴스 기사 크롤링하기

```
# 데이터 프레임 만들기
import pandas as pd

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

df = pd.DataFrame(data)
df

# csv 파일로 만들기
df.to_csv('kbs_news1.csv', index=False)
```

✓ 0.0s

	뉴스url	제목	
0	https://news.kbs.co.kr/news/pc/view/view.do?nc...	전국에 가끔 구름...오늘도 폭염 속 소나기	\n\n\n 오늘 전국에 가 는

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

삼성 패키징 조직 통합...AVP 개발, TSP 총괄로 이관

발행일 : 2024-08-19 14:21

삼성전자가 반도체 패키징 경쟁력 강화를 위해 흩어진 인력을 통합하는 조직개편을 단행했다. 구현할 첨단 패키징과 기존 패키징 기술을 하나로 묶어 시너지를 극대화하려는 의도로 풀이된다.

19일 업계에 따르면 삼성전자 반도체 사업 부문(DS)은 부문장 직속에 있던 '첨단 패키징(AVP)' '테스트앤시스템패키지(TSP)' 총괄 내로 이관했다. AVP 개발실장이었던 최경세 부사장과 관련 TSP 총괄 부사장 산하로 이동했다.

AVP 개발실은 2.5차원(D)와 3D 패키징 등 첨단 패키징 기술 개발을 전담하는 조직이다. 반도체 한계를 극복하기 위해 첨단 패키징 기술이 급부상하자 경계현 사장이 DS 부문장으로 있던 2021년 만들었다. 경 사장은 여기서 그치지 않고 첨단 패키징을 D램이나 시스템 반도체처럼 사업화를 사업 관련 인력까지 배치해 'AVP사업팀'을 만들었다.

명언 크롤링

● Quotes to Scrape

quotes.toscrape.com

Quotes to Scrape

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."

by [Albert Einstein](#) (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

"It is our choices, Harry, that show what we truly are, far more than our abilities."

by [J.K. Rowling](#) (about)

Tags: [abilities](#) [choices](#)

```
▼<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">
  ▼<span class="text" itemprop="text">
    ▼<font style="vertical-align: inherit;">
      <font style="vertical-align: inherit;">"우리가 만든 세상은 우리의 사고
        다. 우리의 사고를 바꾸지 않고는 세상을 바꿀 수 없습니다."</font>
    </font>
  </span>
  ▼<span>
    ▶<font style="vertical-align: inherit;">☹</font>
    ▼<small class="author" itemprop="author">
      ▼<font style="vertical-align: inherit;">
        <font style="vertical-align: inherit;">아인슈타인 </font>
      </font>
    </small>
    ▼<a href="/author/Albert-Einstein">
      ▼<font style="vertical-align: inherit;">
        <font style="vertical-align: inherit;">(정보)</font>
      </font>
    </a>
  </span>
  ▶<div class="tags">☹</div>
```

명언 크롤링

- Quotes to Scrape

```
import requests as req
from bs4 import BeautifulSoup

url = "https://quotes.toscrape.com/"
res = req.get(url)

html = BeautifulSoup(res.text, 'html.parser')
# print(html)

# 명언(find_all)
quote_div = html.find_all('div', class_='quote')
# print(quote_div)
quote_div_span = html.find_all('span', class_='text')
# print(quote_div_span)
# print(len(quote_div_span))
# 리스트 내포
# [i.find_all('span', class_='text')[0].text for i in quote_div]
```

명언 크롤링

```
# 명언 추출(select)
quote_text = html.select('div.quote > span.text')
# print(quote_text)
# print(len(quote_text))
# print(quote_text[0].text)

# for quote in quote_text:
#     print(quote.text)

# 리스트 내포
[i.text for i in quote_text]

# 명언 말한 사람
quote_author = html.select('div.quote > span > small.author')
# print(quote_author)
[i.text for i in quote_author]

# 말한 사람 정보 링크
quote_link = html.select('div.quote > span > a')
# print(quote_link[0].attrs)
# print(quote_link[0]['href'])
[i['href'] for i in quote_link]
```

```
['/author/Albert-Einstein',
 '/author/J-K-Rowling',
 '/author/Albert-Einstein',
 '/author/Jane-Austen',
 '/author/Marilyn-Monroe',
 '/author/Albert-Einstein',
 '/author/Andre-Gide',
 '/author/Thomas-A-Edison',
 '/author/Eleanor-Roosevelt',
 '/author/Steve-Martin']
```

실습3. 환율정보 크롤링하기

- 네이버 > 증권 > 시장지표 > 환전 고시 환율에 접속하여, 아래 출력결과 이미지대로 크롤링 하세요

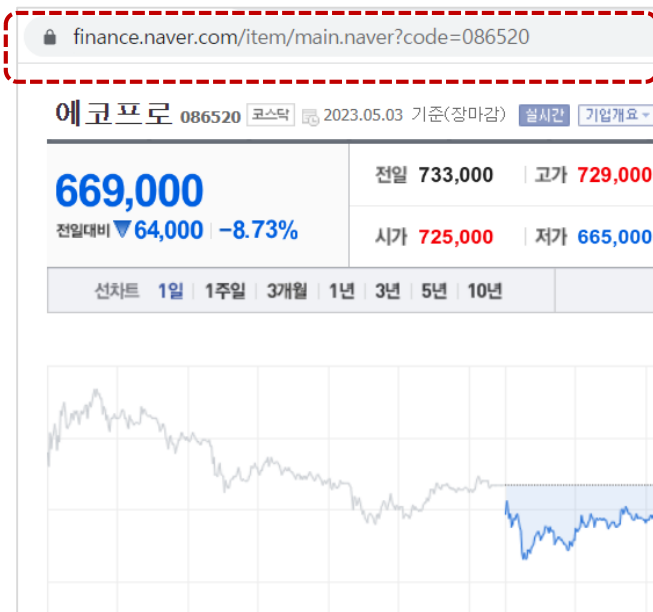


USD 1,335.00
JPY(100엔) 916.11
EUR 1,484.32
CNY 187.25

주식 정보

◆ 주식 정보 가져오기

네이버 > 증권(금융 홈) > 주식 종목(우측 하단)



```
▼<div class="rate_info">
  ▶<div class="today">...</div> == $0
  ▶<table summary="주요 시세(전일종가, 시고저가, 거래량, ...)">
    </div>
  ▶<div class="chart">...</div>
  ▶<a href="javascript:fchartStatus.showChartArea('005930',
    lose" onclick="clickcr(this,'sop.toggle','','',event);">...
  </div>
```

주식 정보

◆ 단일 주식 종목 찾아 오기 - 함수 정의 사용

```
# 주식 1 종목
def getcontent():
    url = 'https://finance.naver.com/item/main.naver?code=086520'
    response = requests.get(url)
    content = BeautifulSoup(response.text, 'html.parser')
    return content

content = getcontent()
today = content.find('div', attrs={'class': 'today'})
# print(today)
price = today.find('span', attrs={'class': 'blind'})
print(price.text)
print(f'에코프로 추가 : {price.text}원')
```

거래중일때 웹에서는 보이지 않음

```
<p class="no_today">
<em class="no_down">
<span class="blind">80,100</span>
<span class="no8">8</span><span cla
</em>
```

주식 정보

◆ 단일 주식 종목 찾아 오기 - 함수 정의 사용

```
def getcontent():  
    url = "https://finance.naver.com/item/main.naver?code=005930"  
    response = requests.get(url)  
    content = BeautifulSoup(response.text, "html.parser")  
    return content  
  
content = getcontent()  
today = content.select_one('div.today')  
print(today)  
price = today.select_one('span.blind')  
print(price)  
print("삼성전자 주가 : {}원".format(price.text))
```

주식 정보

◆ 주식 정보 찾기 – 여러 종목 가격 가져오기

```
def getcontent(item_code): #종목 코드를 매개변수로 입력 받음
    url = "https://finance.naver.com/item/main.naver?code=" + item_code
    response = requests.get(url)
    content = BeautifulSoup(response.text, "html.parser")
    return content

def getprice(item_code):
    content = getcontent(item_code)
    today = content.find('div', attrs={'class': 'today'})
    price = today.find('span', attrs={'class': 'blind'})
    return price
```

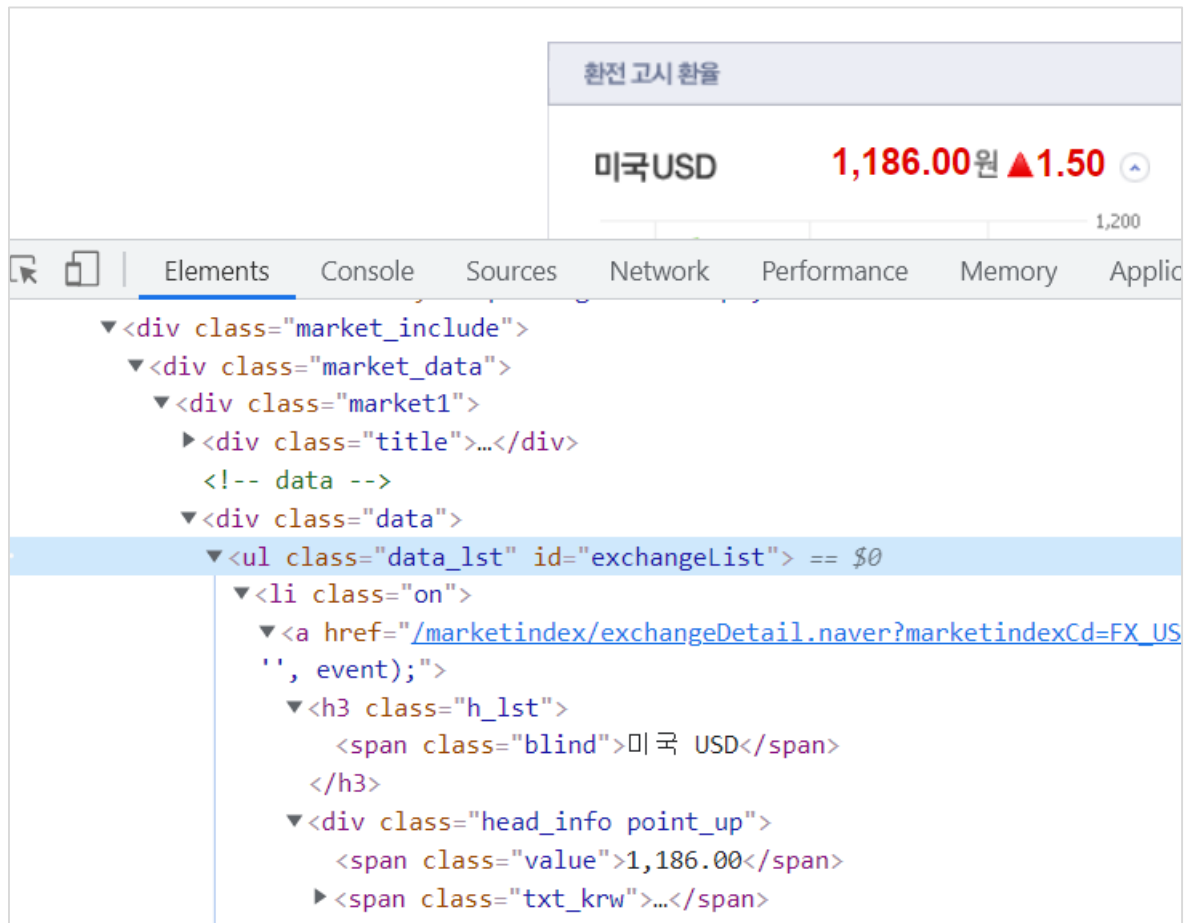
```
에코프로 = getprice('086520')
네이버 = getprice('035420')
NCSOFT = getprice('036570')

print(f'에코프로 주가 : {에코프로.text}원')
print(f'네이버 주가 : {네이버.text}원')
print(f'에코프로 주가 : {NCSOFT.text}원')
```


네이버 금융 크롤링하기

● 환율정보 수집하기

네이버 > 증권 > 시장지표 > 환전 고시 환율



The screenshot shows the Naver Finance '환전 고시 환율' (Exchange Rate Announcement) page for the US Dollar (미국USD). The current rate is 1,186.00 KRW, with a change of +1.50 KRW. A DOM tree overlay is visible at the bottom, showing the HTML structure of the page. The relevant HTML elements for the exchange rate are highlighted in blue.

```
<div class="market_include">
  <div class="market_data">
    <div class="market1">
      <div class="title">...</div>
      <!-- data -->
      <div class="data">
        <ul class="data_lst" id="exchangeList"> == $0
          <li class="on">
            <a href="/marketindex/exchangeDetail.naver?marketindexCd=FX_US"
              '', event);">
              <h3 class="h_lst">
                <span class="blind">미국 USD</span>
              </h3>
              <div class="head_info point_up">
                <span class="value">1,186.00</span>
                <span class="txt_krw">...</span>
```

네이버 금융 크롤링하기

- 환율정보 수집하기 – find() 사용하여 첫번째 환율 찾기

```
import requests
from bs4 import BeautifulSoup

# 네이버 증권 > 시장지표 > 환전 고시 환율
resp = requests.get("https://finance.naver.com/marketindex")
soup = BeautifulSoup(resp.text, "html.parser")
ul = soup.find('ul', attrs={'class': 'data_lst'}) #첫번째 ul
#print(ul)
li = ul.find('li') #첫번째 li
#print(li)
exchange = li.find('span', attrs={'class': 'blind'}) # 환율 종류
#print(exchange)
value = li.find('span', attrs={'class': 'value'}) # 환율 지수
print(exchange.text, value.text)
```

exchange_find.py

👉 실행 결과

미국 USD 1,186.00

네이버 금융 크롤링하기

- 환율정보 수집하기 – findAll() 사용하기

```
# 전체 환율 추출하기
all_li = ul.findAll('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, ': ', value.text)
    print(exchange.text.split(' ')[-1], ': ', value.text)
```

```
USD : 1,262.00
JPY(100엔) : 955.66
EUR : 1,348.64
CNY : 185.63
```

네이버 금융 크롤링하기

- 환율정보 수집하기

select(태그요소.선택자이름) – 전체 검색(리스트로 반환)

select_one(태그요소.선택자이름) – 1개 검색

exchange_select.py

```
response = requests.get("https://finance.naver.com/marketindex")
soup = BeautifulSoup(response.text, "html.parser")
lis = soup.select("div.market1 ul li") #market1 전체 검색
#lis = soup.select("ul.data_lst li") #market2, 3까지 모두 찾을
#print(lis)

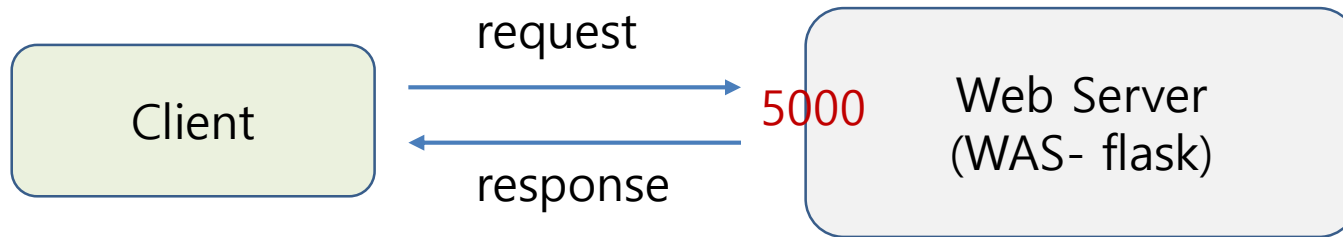
for li in lis:
    exchange = li.select_one("span.blind") #환율 종류
    #print(exchange)
    value = li.select_one("span.value") #환율 지수
    #print(exchange.string, value.string)
    print(exchange.string.split(' ')[-1], ': ', value.string)
```

👉 실행 결과

```
USD : 1,168.00
JPY(100엔) : 1,069.06
EUR : 1,381.57
CNY : 181.72
```

클라이언트 & 서버

- 요청(request) 및 응답(response)



WAS(Web Application Server) : 웹 애플리케이션 서버로 함수를 통한 제어, DB 연동 업무를 수행하고 클라이언트에 응답하는 역할을 한다. 파이썬의 플라스크(flask)와 장고(django), 자바의 스프링(spring)이 웹 프레임워크라 할 수 있다.

flask(플라스크)

플라스크(flask)란?

파이썬으로 제작된 마이크로 웹 프레임워크의 하나이며, 웹 sever를 만들고, 웹 애플리케이션을 제작할 수 있다.



Flask 문서에 오신 것을 환영합니다. 시작하기 [설치](#) 한 다음에 대한 개요 얻을 [빠른 시작을](#). Flask를 사용하여 작지만 완전한 애플리케이션을 만드는 방법을 보여주는 더 자세한 [자습서](#) 도 있습니다. 일반적인 패턴은 [Flask 용 패턴](#) 섹션에 설명되어 있습니다. 나머지 문서에서는 [API](#) 섹션 의 전체 참조와 함께 Flask의 각 구성 요소를 자세히 설명 합니다.

flask(플라스크)

- 플라스크 설치

- flask documentation 검색 및 설치

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

pip install Flask

flask 웹서버 만들기

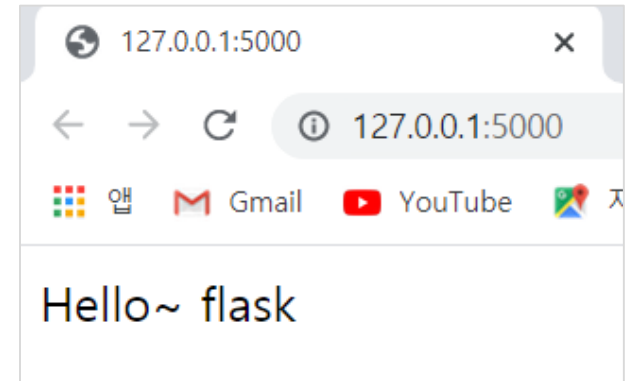
■ 웹 서버 만들기

```
from flask import Flask

app = Flask(__name__) # app - Flask 객체 생성

@app.route('/') # 라우트 ('/'는 루트 경로)
def index():
    return "Hello~ flask"

if __name__ == "__main__":
    app.run()
```



배포용이 아닌 개발용이다.
WSGI(WebServer GateWay Interface)

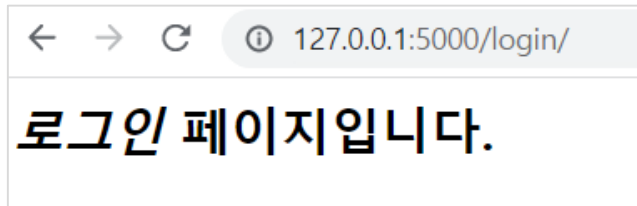
```
* Serving Flask app 'app'
* Debug mode: off
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

```
* Running on http://127.0.0.1:5000
```


flask 웹서버 만들기

■ 페이지 라우팅



```
from flask import Flask

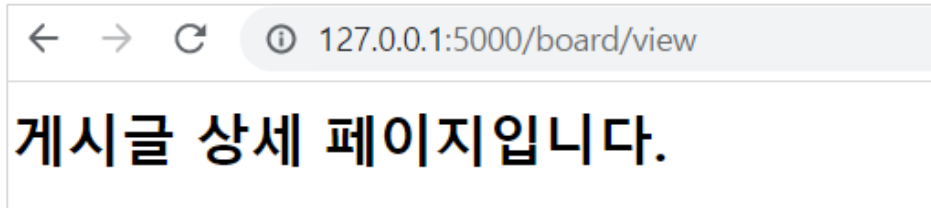
app = Flask(__name__) #app 객체 생성

@app.route('/') #http://127.0.0.1:5000/
def index():
    return "Hello~ Flask"

@app.route('/login') #http://127.0.0.1:5000/login
def login():
    return "<h2><i>로그인</i> 페이지입니다.</h2>"
```

flask 웹서버 만들기

■ 페이지 라우팅



```
@app.route('/register')
def register():
    return "<h2>회원가입 페이지입니다.</h2>"

@app.route('/board/view')
def view():
    return "<h2>게시글 상세 페이지입니다.</h2>"
```

flask 웹서버 만들기

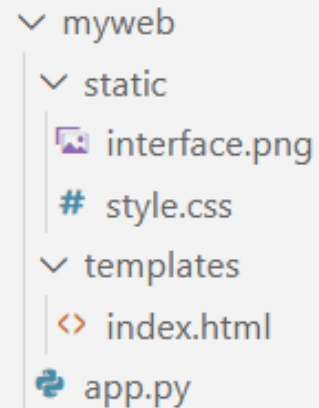
▪ Flask 웹 프레임워크

웹 프레임워크 구조

templates 폴더 -> html 파일

static 폴더 -> css, image 파일

start_app.py -> 실행 파일



```

  myweb
  ├── static
  │   ├── interface.png
  │   └── style.css
  ├── templates
  │   └── index.html
  └── app.py

```

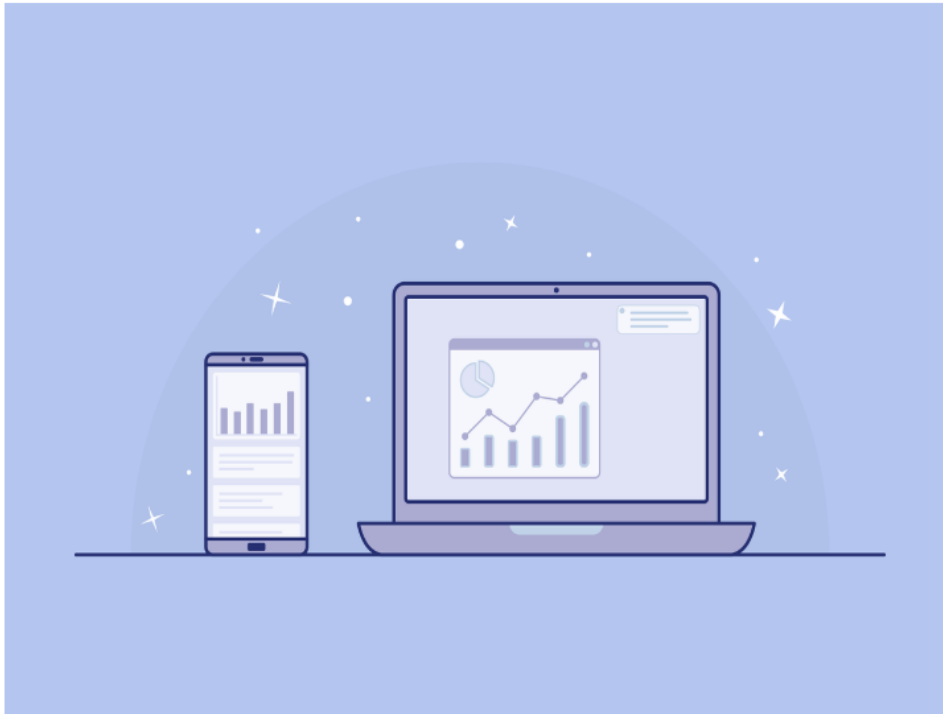
flask 웹서버 만들기

웹 기초 기술

HTML - 웹 페이지의 구조를 담당

CSS - 웹페이지의 디자인을 담당

Javascript(자바스크립트) - 웹 페이지의 인터랙티브를 담당



[픽사베이로 이동](#)

flask 웹서버 만들기

- 인덱스 페이지 서버에 띄우기

app.py

```
from flask import Flask, render_template

app = Flask(__name__) #app 객체 생성

@app.route('/') #http://127.0.0.1:5000/
def index():
    # return "<h1>Welcome~ My Home!!</h1>"
    return render_template('index.html') #인덱스 페이지 렌더링

app.run()
```

flask 웹서버 만들기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>인덱스 페이지</title>
  <link rel="stylesheet" href={{ url_for('static', filename='style.css')}}>
</head>
<body>
  <div id="container">
    <h1>웹 기초 기술</h1>
    <hr>
    <p>HTML - 웹 페이지의 구조를 담당</p>
    <p>CSS - 웹페이지의 디자인을 담당</p>
    <p>Javascript(자바스크립트) - 웹 페이지의 인터랙티브를 담당</p>

    <!-- 이미지 삽입 -->
    <img src={{ url_for('static', filename='interface.png')}}
alt="인터페이스" width="640" height="480"><br><br>

    <!-- 하이퍼 링크 -->
    <a href="https://pixabay.com/ko/" target="_blank">
      Pixabay로 이동</a>
  </div>
</body>
</html>
```

index.html

flask 웹서버 만들기

- 인덱스 페이지 서버에 띄우기

```
/* sytle.css */  
  
/* 페이지 가운데 정렬, 위.아래 여백 30px */  
#container{ width: 80%; margin: 30px auto;}  
  
/* 제목 글자색 - 파랑 */  
#container h1{color: blue;}
```

로또 당첨 번호 확인 앱

➤ 당첨 회차를 입력하면 당첨번호와 보너스 번호 출력

The screenshot shows the DH Lottery website interface. The browser address bar displays 'dhlottery.co.kr/gameResult.do?method=byWin'. The page title is '회차별 당첨번호' (Drawing Results by Drawing). The main content area displays the results for the 1060th drawing, dated March 25, 2023. The winning numbers are 3, 10, 24, 33, 38, and 45, with a bonus number of 36. The numbers are presented in colored circles: 3 (yellow), 10 (yellow), 24 (red), 33 (grey), 38 (grey), 45 (green), and 36 (grey). The text '당첨번호' (Winning Numbers) is below the main numbers, and '보너스' (Bonus) is below the bonus number. A sidebar on the left contains a menu with options like '당첨번호', '당첨확인', '금안내', '다시보기', '참관신청', and '20+'. A top navigation bar includes a home icon, '당첨결과' (Drawing Results), and '로또6/45' (6/45 Lotto). A link '회차 바로가기' (Go to Drawing) is also visible.

회차별 당첨번호

1060회 당첨결과
(2023년 03월 25일 추첨)

3 10 24 33 38 45 + 36

당첨번호 보너스

로또 당첨 번호 확인 앱

➤ 당첨 번호 확인 윈도우

로또 당첨

당첨 회차 입력:

1065

로또 당첨 번호 확인

당첨번호: ['3', '18', '19', '23', '32', '45']

보너스: 24

로또 당첨 번호 확인 앱

```
import requests
from bs4 import BeautifulSoup

num = 1060
url = 'https://dhlottery.co.kr/gameResult.do?method=byWin&drwNo={}'.format(num)
response = requests.get(url)
# print(response.text)

soup = BeautifulSoup(response.text, 'html.parser')
win_result = soup.select_one('div.win_result')
win_result.text
```

'\n1060회 당첨결과\n(2023년 03월 25일 추첨)\n\n\n당첨번호\n\n3\n10\n24\n33\n38\n45\n\n\n'

```
▼<div class="win_result">
  ▶<h4>...</h4>
  <p class="desc">(2023년 03월 25일 추첨)</p>
  ▼<div class="nums">
    ▼<div class="num win">
      <strong>당첨번호</strong>
      ▼<p>
        <span class="ball_645 lrg ball1">3</span>
        <span class="ball_645 lrg ball1">10</span>
        <span class="ball_645 lrg ball3">24</span>
        <span class="ball_645 lrg ball4">33</span>
        <span class="ball_645 lrg ball4">38</span>
        <span class="ball_645 lrg ball5">45</span>
```

로또 당첨 번호 확인 앱

```
▼<form name="frm" method="post" id="frm">
  <span class="unit label">회차 바로가기</span>
  <input type="hidden" name="drwNo" id="drwNo">
  ▼<select id="dwrNoList" name="dwrNoList" title="회차 선택">
    <option value="1065">1065</option> slot
    <option value="1064">1064</option> slot
    <option value="1063">1063</option> slot
```

```
win_list = win_result.text.split('\n')
win_list
```

```
[',',
 '1060회 당첨결과',
 '(2023년 03월 25일 추첨)',
 ',',
 ',',
 '당첨번호',
 ',',
 '3',
 '10',
 '24',
 '33',
 '38',
 '45',
 ',',
 ',',
 ',',
 '보너스',
 '36',
 ',',
 ',',
 ',']
```

로또 당첨 번호 확인 앱

```
win_list = win_result.text.split('₩n')[7:13]
win_list
bonus_num = win_result.text.split('₩n')[-4]
bonus_num
print("당첨번호")
print(win_list)
print("당첨번호")
print(bonus_num)
```

당첨번호

['3', '10', '24', '33', '38', '45']

당첨번호

36

로또 당첨 번호 확인 앱

```
import requests
from bs4 import BeautifulSoup
from tkinter import *

def lotto_win():
    #num = 1060
    num = entry.get() # 입력된 당첨 번호

    url = 'https://dhlottery.co.kr/gameResult.do?method=byWin&drwNo={}'.format(num)
    response = requests.get(url)
    # print(response.text)

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

    win_list = win_result.text.split('\n')[7:13]
    win_list
    bonus_num = win_result.text.split('\n')[-4]

    # 출력 상자에 출력
    output.delete(0.0, END)
    output.insert(END, f'당첨번호: {win_list}\n\n보너스: {bonus_num}')
```

로또 당첨 번호 확인 앱

```
#lotto_win()

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

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

btn = Button(window, text="당첨 번호 확인", command=lotto_win)
btn.grid(row=2, column=0, sticky=W)
# 출력 상자
output = Text(window, bg='skyblue', width=50, height=5)
output.grid(row=3, column=0, sticky=W)

window.mainloop()
```