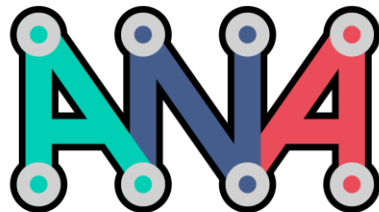


ANA Python 교육

Python으로 할 수 있는 재미있는 것들

충남대학교 알고리즘 동아리 ANA



김시온

- 충남대학교 컴퓨터공학과 18학번
- 2023 ANA 동아리 회장
- SW Maestro 13th 수료
- SW Academy 2nd ~ 3rd 백엔드 TA



홍성빈



- 충남대학교 컴퓨터공학과 18학번
- BOB 7th 수료
- 교육용 서버 구축

전진호



- 충남대학교 수학과 17학번
- Samsung SW 역량 테스트 B형 통과
- 2022-1 컴퓨터과학적사고 실습 조교

Python으로 아래 문제를 해결해봅시다

1. 로또 당첨 번호 분석(?)

가장 많이 등장한 당첨번호는 무엇일까요?

2. 포켓몬 도감 만들기

웹 사이트에 있는 포켓몬 도감 정보를 저장하고 싶어요

Python

Python은 배우기 쉽고, 무궁무진한 활용법을 가진 언어입니다.



브라우저 매크로, Discord 봇, 인공지능, 이미지 처리, 데이터 분석 ...

KAIST Python 공익 사건

6개월간 마우스와 키보드로 해야 할 일을 하루만에 끝냈다.

필자가 고용노동부 자동지청 사회복무요원으로 근무하던 시기의 이야기입니다.

병현 씨, 최근 1년간 발송된 모든 등기우편의
발송내역을 조회한 후 종이에 인쇄해주세요.

방금 확인해보니 3,900건 이상인데다가
매일 발송내역이 쌓여요.

우체국 사이트에서 등기번호마다 조회해서
인쇄해야 하니 넉넉히 6개월 정도 걸린다고
생각하시고 천천히 진행해주세요.

네, 확인해보겠습니다.

- 30분 후 -

주무관님, 오늘 말씀하신 일은 모두 처리하였습니다.

제가 자동화 프로그램을 만들어봤으니 앞으로
이 프로그램으로 사용하시면 될 것 같습니다.
보내드릴테니 확인 부탁드립니다.



웹 프로그래밍에 대한 약간의 지식 + Python Selenium

```
crawler = pc.crawler()

done_list = []
temp = os.listdir(out_dir)

for el in temp:
    if ".png" in el:
        done_list.append(el)

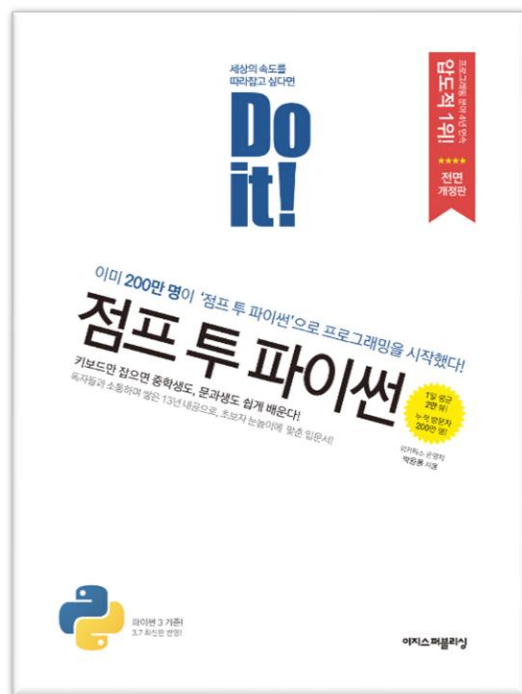
for line in csv:
    line_split = line.strip().split(",")
    if len(line_split) != 2:
        continue
```

[post_crawler Github](#)



점프 투 파이썬으로 Python을 공부하기

책을 구입하지 않아도 [점프 투 파이썬 위키독스](#)로 공부할 수 있습니다.



점프 투 파이썬 - Wikidocs

검색어를 입력하세요.

점프 투 파이썬

00장 들어가기 전에

- 00-1 머리말
- 00-2 저자소개
- 00-3 동영상 강의
- 00-4 소스 코드
- 00-5 책 구입 안내
- 00-6 주요변경이력

01장 파이썬이란 무엇인가?

- 01-1 파이썬이란?
- 01-2 파이썬의 특징
- 01-3 파이썬으로 무엇을 할 수 있을까?
- 01-4 파이썬 설치하기
- 01-5 파이썬 둘러보기
- 01-6 파이썬과 에디터

02장 파이썬 프로그래밍이 갖는 특이점

점프 투 파이썬

지은이 : 박응용
최종 편집일시 : 2023년 9월 11일 5:31 오후
저작권 : CC BY-NC-ND
e-book 판매가 : 8,000원 (구매하기)
4,577 명이 추천

이 책은 파이썬이란 언어를 처음 접해보는 독자와 프로그래밍을 한 번도 해 본적이 없는 사람들을 대상으로 한다. 프로그래밍을 할 때 사용되는 전문적인 용어들을 알기 쉽게 풀어서 쓰려고 노력하였으며, 파이썬이란 언어의 개별적인 특성만을 강조하지 않고 프로그래밍 전반에 관한 사항을 파이썬이란 언어를 통해 알 수 있도록 알기 쉽게 설명하였다.

파이썬에 대한 기본적인 지식을 알고 있는 사람이라도 이 책은 파이썬 프로그래밍에 대한 흥미를 가질 수 있는 좋은 안내서가 될 것이다. 이 책의 목표는 독자가 파이썬을 통해 프로그래밍에 대한 전반적인 이해를 갖게하는 것이며, 또 파이썬이라는 도구를 이용하여 원하는 프로그램을 쉽고 재미있게 만들 수 있게 하는 것이다.

점프 투 파이썬 개정 2판이 종이책으로 출간되었습니다. (2023년 6월 15일)

점프 투 파이썬

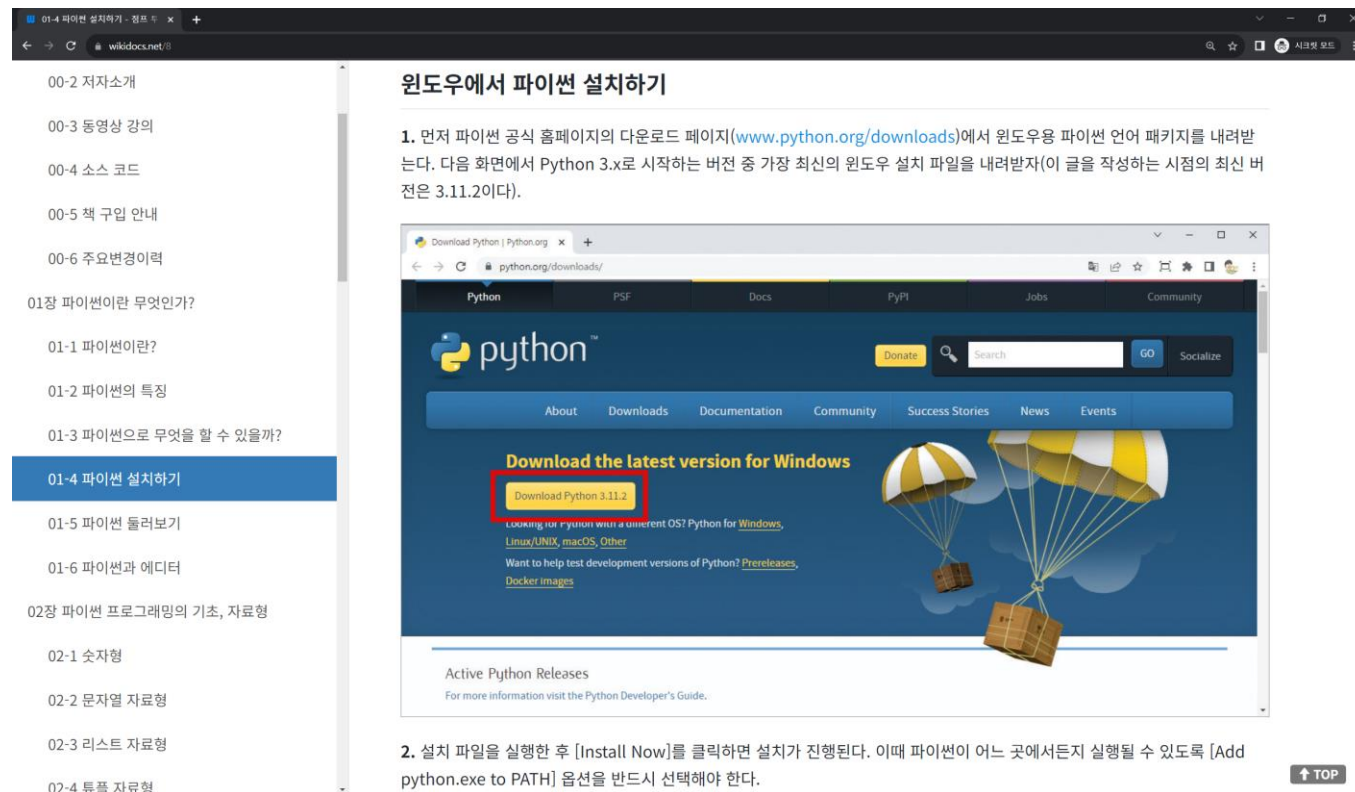
무료 e-book 안내

위키독스의 "점프 투 파이썬"은 "되새김 문제"와 "파이썬 코딩 면허 시험 20제"를 제외한 모든 챕터를 무료로 공개하고 있습니다. "되새김 문제"와 "파이썬 코딩 면허 시험 20제" 그리고 해당 문제의 정답 및 풀이는 유료 E-book 또는 종이책에서만 확인할 수 있습니다.

↑ TOP

Python 설치하기

Python 공식 홈페이지에서 Python을 설치할 수 있습니다.



00-2 저자소개

00-3 동영상 강의

00-4 소스 코드

00-5 책 구입 안내

00-6 주요변경이력

01장 파이썬이란 무엇인가?

01-1 파이썬이란?

01-2 파이썬의 특징

01-3 파이썬으로 무엇을 할 수 있을까?

01-4 파이썬 설치하기

01-5 파이썬 둘러보기

01-6 파이썬과 에디터

02장 파이썬 프로그래밍의 기초, 자료형

02-1 숫자형

02-2 문자열 자료형

02-3 리스트 자료형

02-4 튜플 자료형

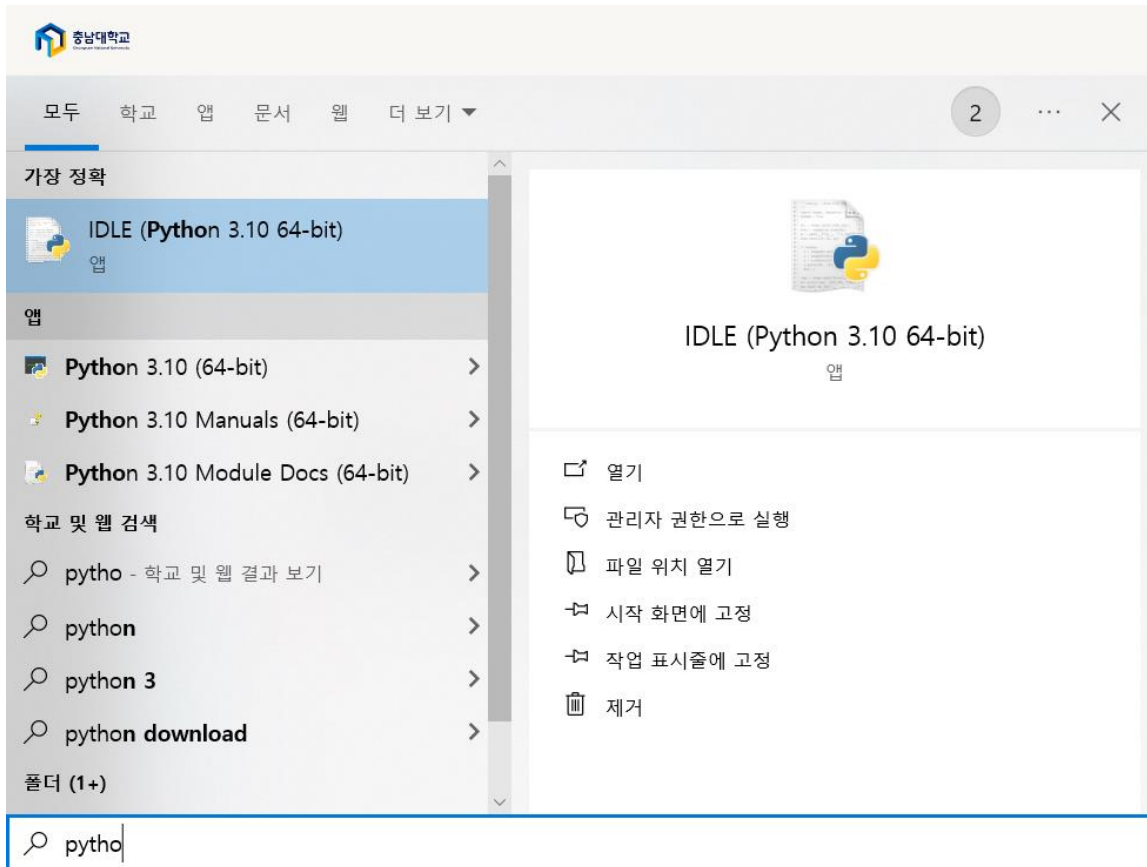
윈도우에서 파이썬 설치하기

1. 먼저 파이썬 공식 홈페이지의 다운로드 페이지(www.python.org/downloads/)에서 윈도우용 파이썬 언어 패키지를 내려받는다. 다음 화면에서 Python 3.x로 시작하는 버전 중 가장 최신의 윈도우 설치 파일을 내려받자(이 글을 작성하는 시점의 최신 버전은 3.11.2이다).

2. 설치 파일을 실행한 후 [Install Now]를 클릭하면 설치가 진행된다. 이때 파이썬이 어느 곳에서든지 실행될 수 있도록 [Add python.exe to PATH] 옵션을 반드시 선택해야 한다.

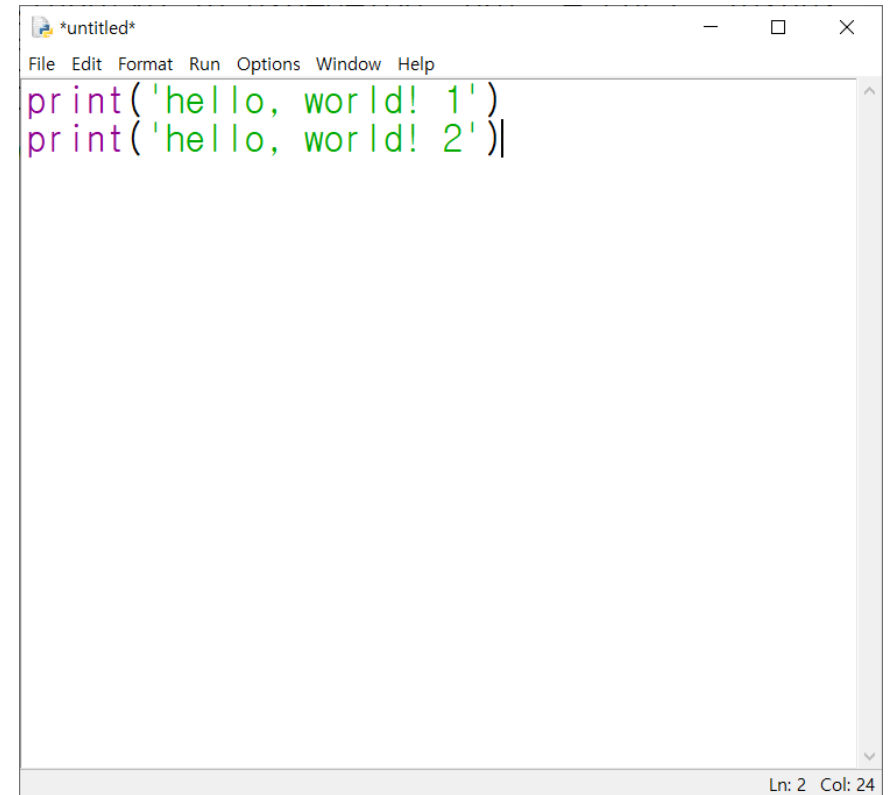
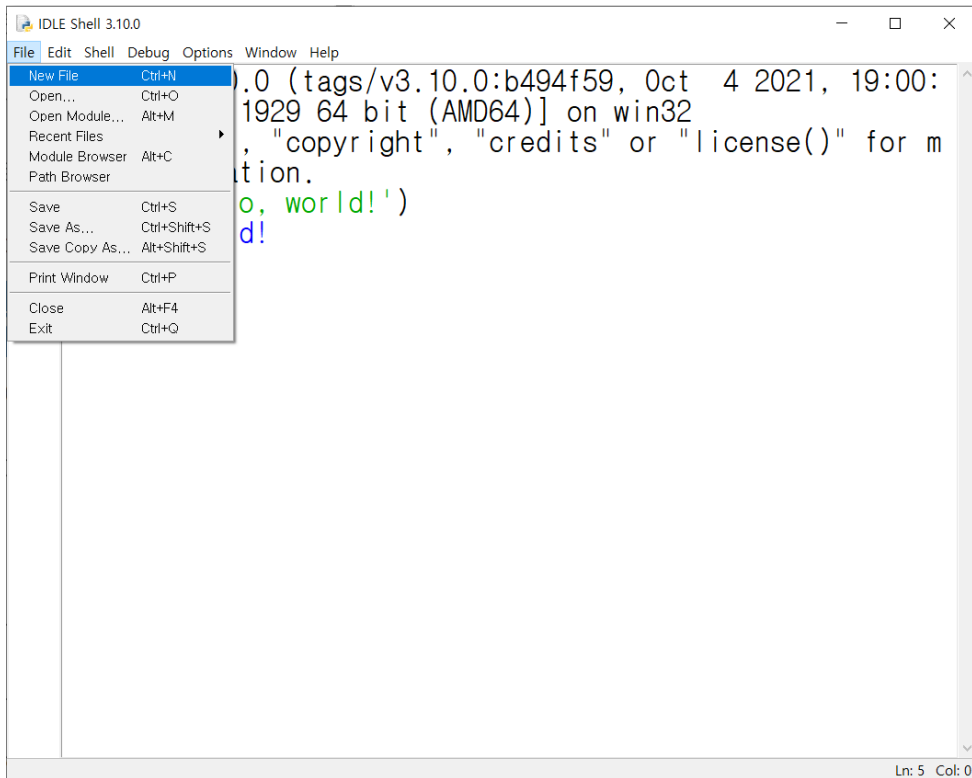
Python 코드 실행하기

Python shell을 통해서 내가 작성한 코드를 엔터를 칠 때마다 즉시 실행해볼 수 있습니다.



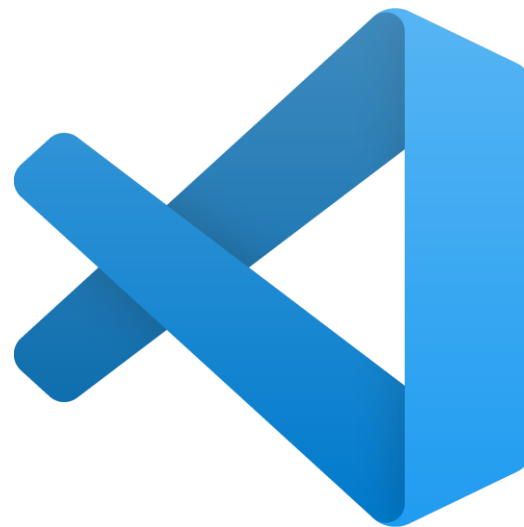
Python 코드 한 번에 실행하기

한 줄 한 줄 실행하지 않고, 코드 작성을 마치고 한 번에 실행하려면 New File을 클릭합니다.



IDE에서 Python 사용하기

PyCharm, VSCode 등등 다양한 IDE를 활용해도 좋습니다.



입출력

```
# input()은 한 줄을 입력받습니다.  
line = input()  
  
# 공백을 기준으로 잘라 리스트를 만듭니다.  
numbers = line.split()  
print(numbers) # ['1', '2', '3', '4', '5']  
  
# map은 각각의 요소를 지정된 함수로 처리해서  
바꿉니다.  
# numbers의 각각의 요소(str)을 int(함수)로  
처리해서 바꿉니다.  
# map은 map object를 반환하므로 list로  
형변환해줍니다.  
numbers = list(map(int, numbers))  
print(numbers) # [1, 2, 3, 4, 5]
```

```
numbers = [1, 2, 3]  
  
# print()로 list를 출력할 수 있습니다  
print(numbers) # [1, 2, 3]  
  
# *(unpacking) 연산자 list 앞에 사용할 수  
있습니다.  
# print(1, 2, 3)과 같은 결과를 만들 수  
있습니다.  
print(*numbers) # 1 2 3  
  
# str.join()을 이용해 list의 요소를 하나의  
문자열로 만들 수 있습니다.  
print(''.join(map(str, numbers))) # 123
```

Lotto API를 이용해 번호 별 등장 횟수 구하기

로또 1회차(2002.12.7.)부터 700회차(2016.04.30.)까지 추첨된 당첨 번호 중에서 $n(1 \leq n \leq 45)$ 이 등장한 횟수를 구하는 프로그램을 작성해보자. (보너스 번호 제외)



1회차 ~ 6회차 당첨 번호

10 23 29 33 37 40

9 13 21 25 32 42

11 16 19 21 27 31

14 27 30 31 40 42

16 24 29 40 41 42

14 15 26 27 40 42

$$n = 40$$

$$\rightarrow 4$$

$$n = 42$$

$$\rightarrow 4$$

$$n = 1, 2, \dots, 8$$

$$\rightarrow 0$$

분명 무작위로 추출한 공일 텐데 특정 번호가 많이 나오는 것처럼 보입니다.
정말로 그런지 확인해볼까요?

로또 추첨 결과를 입력으로 받는다고 가정합니다.

첫째 줄에 정수 m, n 이 주어지고, 둘째 줄부터 m 개의 줄에 각 회차의 추첨 결과가 주어진다.

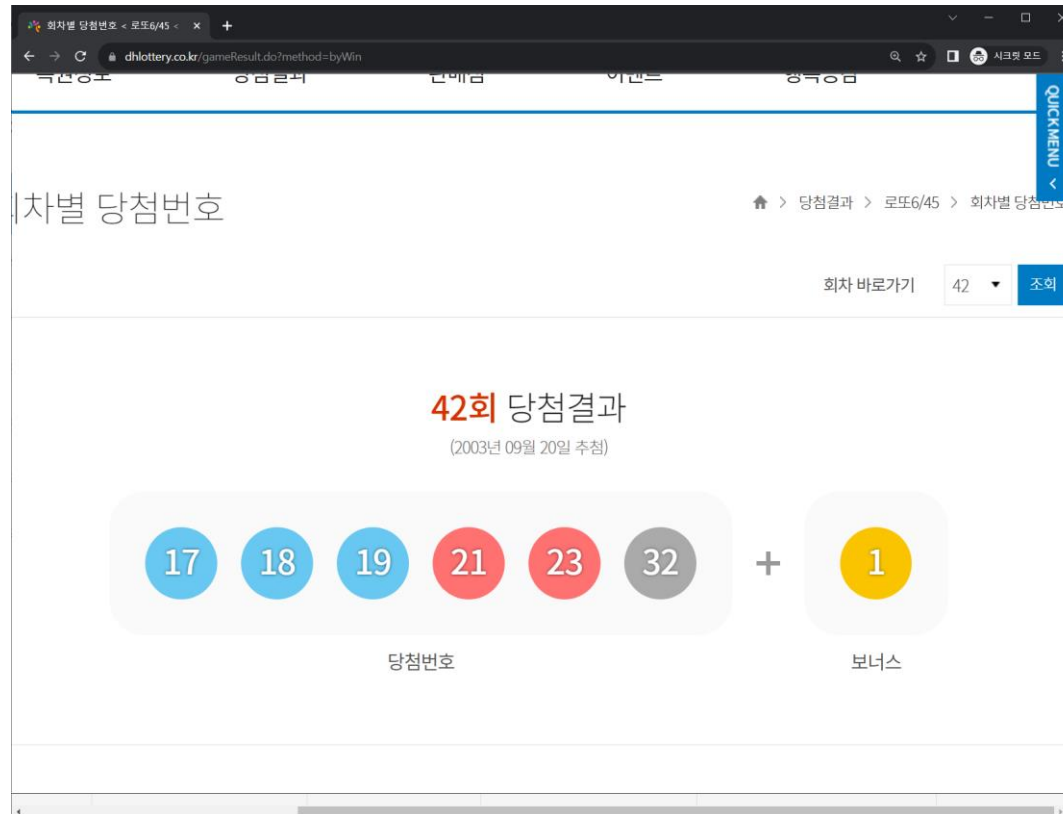
6	42				
10	23	29	33	37	40
9	13	21	25	32	42
11	16	19	21	27	31
14	27	30	31	40	42
16	24	29	40	41	42
14	15	26	27	40	42

List

List를 마치 그릇으로 활용할 수 있습니다. 그릇에 로또 공을 넣는다고 생각합시다.

```
m, n = map(int, input().split())
lotto = [0] * 46
for _ in range(m):
    d1, d2, d3, d4, d5, d6 = map(int, input().split())
    lotto[d1] += 1
    lotto[d2] += 1
    lotto[d3] += 1
    lotto[d4] += 1
    lotto[d5] += 1
    lotto[d6] += 1
print(lotto)
```

로또 추첨 결과를 구하는 방법



동행복권 홈페이지에서 회차별 당첨 번호를 알아낼 수 있습니다.
이걸 700번 반복해야 할까요?

API(Application Programming Interface)

API는 프로그램이 서비스의 자원이나 기능을 쉽게 이용하기 위해 해당 서비스를 만든 사람이 제공하는 규칙입니다.

로또 회사에서 프로그램으로 간단하게 가져올 수 있는 방법을 만들어 제공하고 있습니다. 그 방법은 <https://dhlottery.anacnu.kr/lotteries/42> 에 있는 정보를 가져오는 것입니다.

우리는 Python으로 해당 URL에 있는 정보를 쉽게 가져올 수 있습니다. 바로 로또 회사에서 제공해주는 복권에 대한 API 덕분입니다.

로또 API 사용 방법

<https://dhlottery.anacnu.kr/lotteries/<회차 번호>>

```
{ "returnValue": "success", "drawNumber": 42, "drawDate": "2003-09-20", "totalSellAmount": 119909740000, "firstWinAmount": 6899280100, "numbers": [17, 18, 19, 21, 23, 32], "firstWinnerCount": 6, "firstAccumulatedAmount": 0 }
```

```
{  
    "returnValue": "success",  
    "drawNumber": 42,  
    "drawDate": "2003-09-20",  
    "totalSellAmount": 119909740000,  
    "firstWinAmount": 6899280100,  
    "numbers": [17, 18, 19, 21, 23, 32],  
    "firstWinnerCount": 6,  
    "firstAccumulatedAmount": 0  
}
```

requests 모듈 설치하기

requests 모듈을 이용해서 로또 API를 이용해봅시다.

```
C: \> pip install requests
```

```
Collecting requests
```

```
  Using cached requests-2.31.0-py3-none-any.whl (62 kB)
```

```
Collecting idna<4,>=2.5
```

```
  Using cached idna-3.4-py3-none-any.whl (61 kB)
```

```
Collecting charset-normalizer<4,>=2
```

```
  Using cached charset_normalizer-3.3.0-cp310-cp310-win_amd64.whl (98 kB)
```

```
Collecting urllib3<3,>=1.21.1
```

```
  Using cached urllib3-2.0.6-py3-none-any.whl (123 kB)
```

```
Collecting certifi>=2017.4.17
```

```
  Using cached certifi-2023.7.22-py3-none-any.whl (158 kB)
```

```
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
```

```
Successfully installed certifi-2023.7.22 charset-normalizer-3.3.0 idna-3.4 requests-2.31.0 urllib3-2.0.6
```

requests 모듈 사용법

requests 모듈을 이용해서 로또 API를 이용해봅시다.

```
import requests

url = 'https://dhlottery.anacnu.kr/lotteries/42'
# 주어진 URL로 HTTP GET 요청을 보내고 응답을 받아온다.
response = requests.get(url)

# 응답은 하나의 문자열입니다.
result = response.text

print(result)
```

```
result = '{"returnValue":"success","drawNumber":42,"drawDate":"2003-09-20","totalSellAmount":119909740000,"firstWinAmount":6899280100,"numbers":[17,18,19,21,23,32],"firstWinnerCount":6,"firstAccumulatedAmount":0}'

print(type(result)) # str

# response.text는 JSON 형식의 문자열입니다. 이를 dict로 변환할 수 있습니다.
result = response.json()

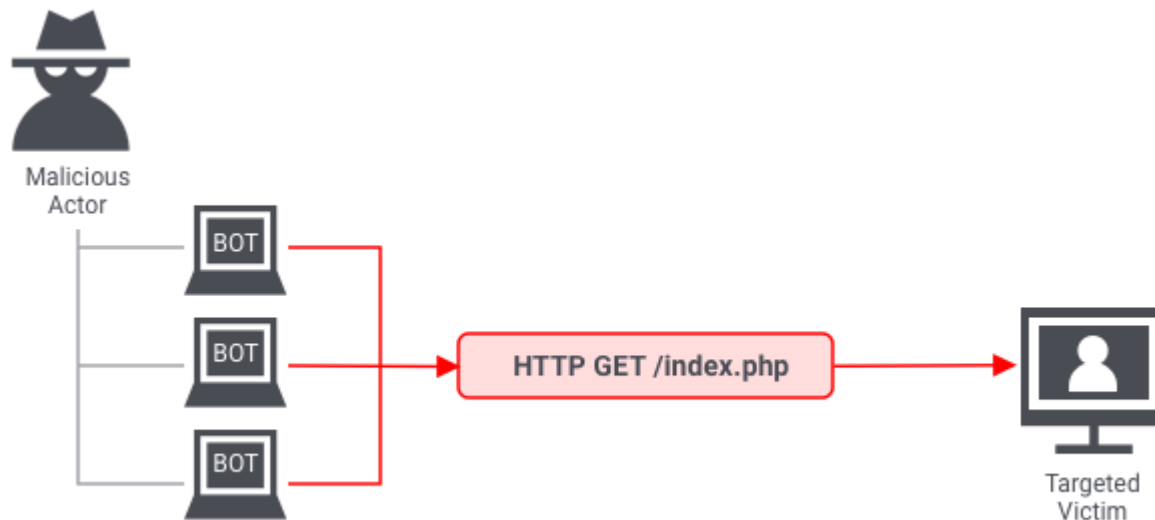
print(type(result)) # dict

result = {'returnValue': 'success', 'drawNumber': 42, 'drawDate': '2003-09-20',
'totalSellAmount': 119909740000, 'firstWinAmount': 6899280100, 'numbers': [17,
18, 19, 21, 23, 32], 'firstWinnerCount': 6, 'firstAccumulatedAmount': 0}
```

API의 응답 결과는 단순한 문자열이지만, 이 문자열이 JSON 형식을 지키고 있기 때문에 쉽게 Python dict 자료형으로 변환할 수 있습니다.

단기간에 너무 많은 요청을 보내면 안됩니다.

사람의 손으로는 불가능한 속도로 많은 요청을 보내면 IP가 차단당할 수 있습니다.
하지만, 우리의 API는 연습용이니 그런 걱정을 할 필요는 없습니다.




```
import requests

def get_result(round):
    url = f'https://dhlottery.anacnu.kr/lotteries/{round}'

    # 주어진 URL로 HTTP GET 요청을 보내고 응답을 받아온다.
    response = requests.get(url)

    # response.text는 JSON 형식의 문자열입니다. 이를 dict로 변환할 수 있습니다.
    result = response.json()

    return result['numbers']
```

로또 API를 사용해 round 회차의 추첨 결과를 가져오는 함수입니다.

```
f = [0] * 46

for round in range(1, 701):
    print(round)
    result = get_result(round)
    for number in result:
        f[number] += 1

print(f[1:]) # [0, 103, 88, 91, 99, ..., ]
```

```
f = [0, 103, 88, 91, 99, ..., 93]

n = int(input())

print(f[n])
```

requests를 라이브러리를 포함하고 있는 왼쪽 소스코드를 제출하면 안됩니다.

1. 백준 온라인 저지 Python 환경에는 Python 표준 라이브러리만 사용할 수 있습니다.
2. 만약 사용할 수 있다고 하더라도, 700번 인터넷에 접속하는 시간이 오래 걸리기 때문에 시간 초과가 발생합니다.

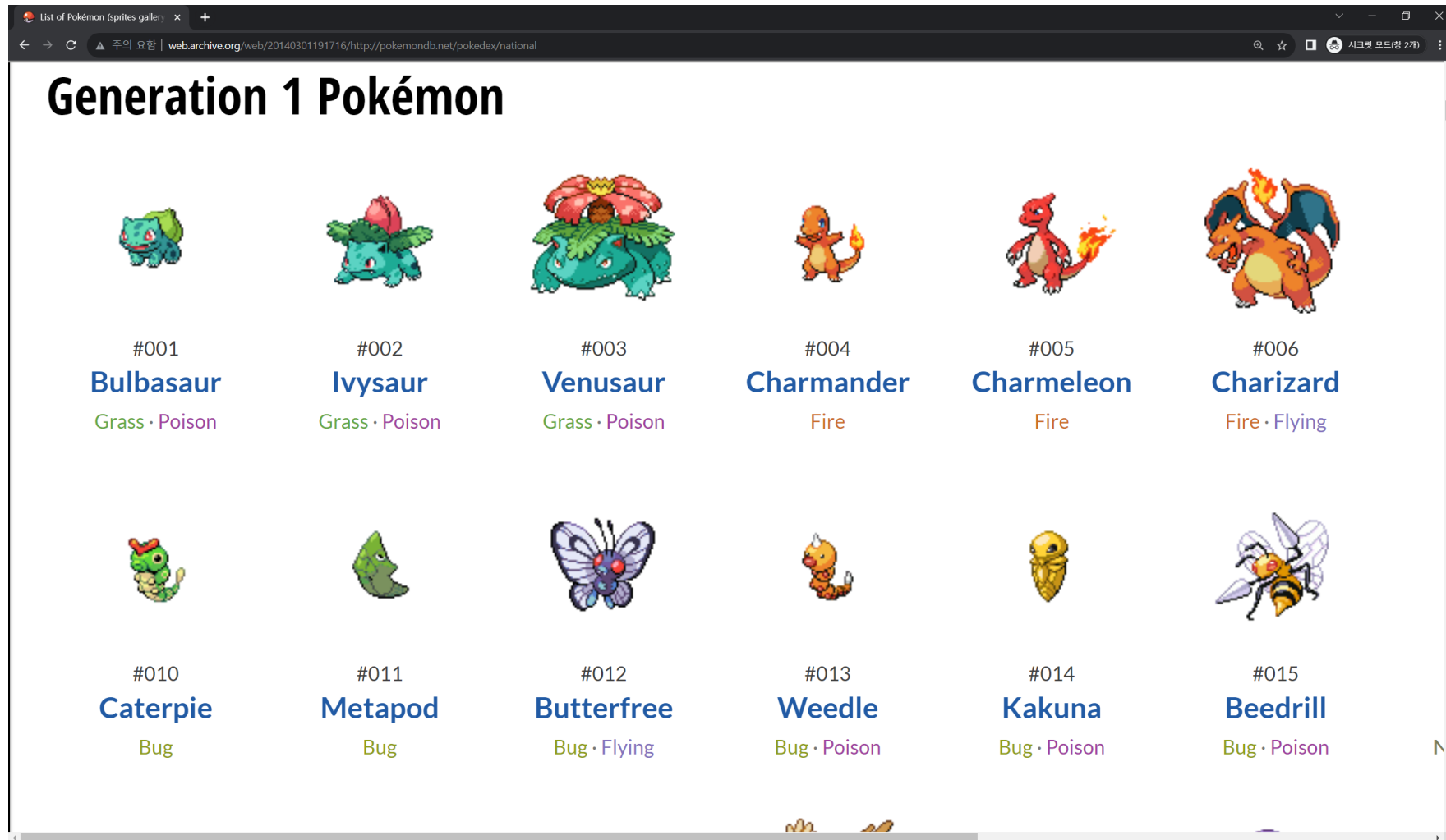
포켓몬 도감 프로그램 만들기

2014.03.01.을 기준으로 도감 번호 $n(1 \leq n \leq 718)$ 번 포켓몬의
영문 이름과 타입(1개 혹은 2개)를 출력하는 프로그램을 작성해보자.



$$n = 25$$

Pikachu
Electric



<https://pokemondb.anacnu.kr/> 에서 각 포켓몬의 영문 이름과 타입을 알아낼 수 있습니다. 그러나, 이번엔 편리한 API가 없다고 가정합시다.

requests 모듈 똑같이 사용해봅시다

이번에도 응답은 하나의 문자열이지만, 특이하게 생겼습니다.

```
import requests

url = 'https://pokemondb.anacnu.kr/'
# 주어진 URL로 HTTP GET 요청을 보내고 응답을 받아온다.
response = requests.get(url)

# 마찬가지로 응답은 하나의 문자열입니다.
result = response.text

print(result)
```

File 입출력으로 문자열 확인하기

문자열이 너무 길어서 출력으로는 확인하기 힘듭니다. 다음과 같은 방법으로 파일에 저장해봅시다.

```
import requests

url = 'https://pokemondb.anacnu.kr/'
response = requests.get(url)
response.encoding = 'utf-8'

html = response.text

file = open('pokemon.html', 'w', encoding='utf-8', newline='')
file.write(html)
```

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <article class="main-content grid-wrapper">
      <h1>List of Pokémon</h1>
      <div class="infocard-tall-list">
        <hr />
        <h2 id="gen-1">Generation 1 Pokémon</h2>
        <span class="infocard-tall">
          <a class="pkg pkgRBY n1 "data-sprite=" pkgRBY n1 "
            href="http://pokemondb.net/pokedex/bulbasaur"
          ></a><br />
          <small>#001</small><br />
          <a class="ent-name" href="http://pokemondb.net/pokedex/bulbasaur"
            ><span id="name-#001">Bulbasaur</span></a>
          자세한 내용은 생략...
        </span>
```

HTML이란?

우리가 보는 웹 페이지는 사실 문자열입니다.

```
<html>
  <head>
    <title>Todolist</title>
  </head>

  <body>
    <h1>오늘 할 일</h1>
    <ol>
      <li>선형대수 과제</li>
      <li>파이썬 공부하기</li>
      <li>
        <strong>컴프 과제</strong>
      </li>
    </ol>
  </body>
</html>
```



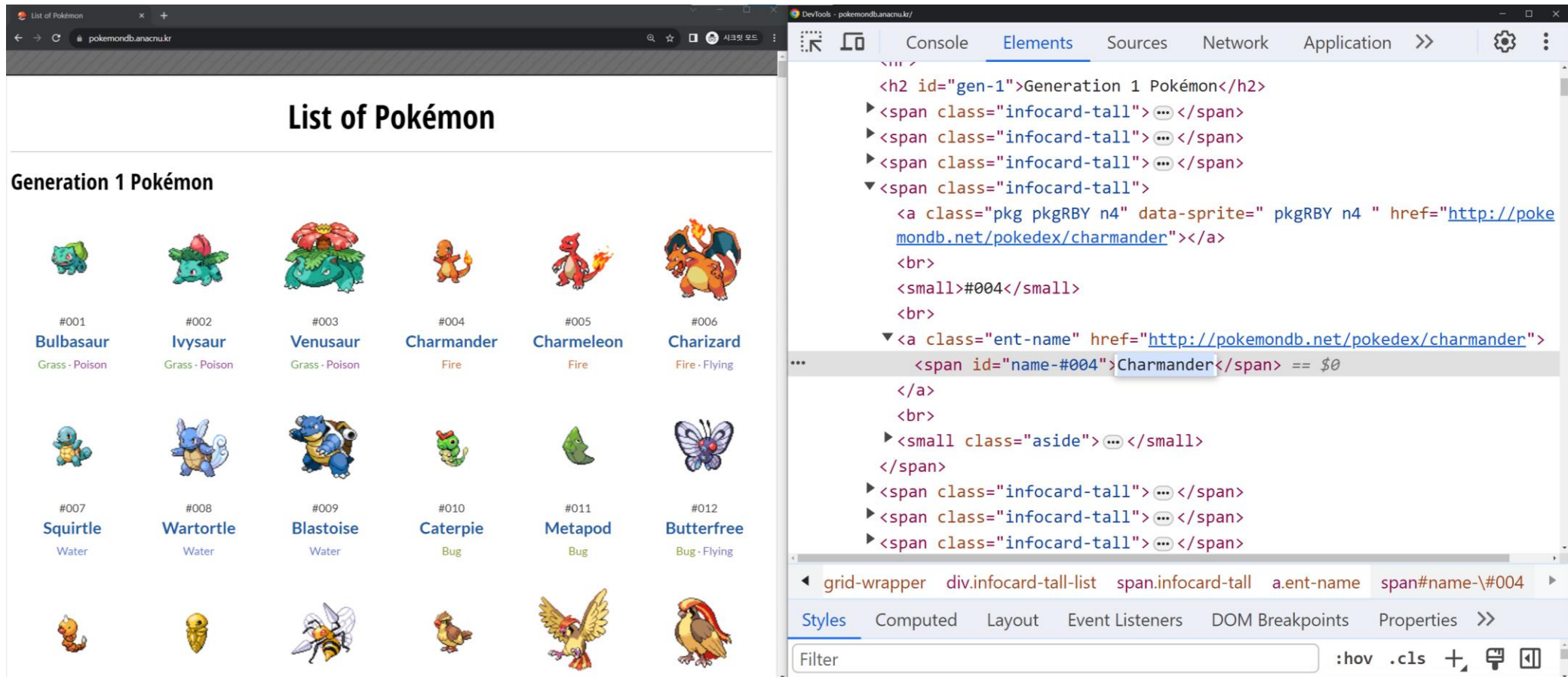
todolist.html

오늘 해야할 일

1. 선형대수 과제
2. 파이썬 공부하기
3. **컴프2 과제**

브라우저에서 F12로 개발자 도구를 사용해봅시다

개발자도구를 통해 내가 보고 있는 페이지가 원래는 어떤 글이었는지 알 수 있습니다.



The screenshot shows a web browser displaying the 'List of Pokémon' page. The page is titled 'List of Pokémon' and shows a grid of Generation 1 Pokémon. The DevTools developer tool is open, showing the 'Elements' panel. The HTML structure of the Charmander entry is visible, including a link to the Charmander page, a small ID number, and a name span.

```
<h2 id="gen-1">Generation 1 Pokémon</h2>
<span class="infocard-tall">...</span>
<span class="infocard-tall">...</span>
<span class="infocard-tall">...</span>
<span class="infocard-tall">
  <a class="pkg pkgRBY n4" data-sprite=" pkgRBY n4 " href="http://poke
mondb.net/pokedex/charmander"></a>
  <br>
  <small>#004</small>
  <br>
  <a class="ent-name" href="http://pokemondb.net/pokedex/charmander">
    <span id="name-#004">Charmander</span> == $0
  </a>
  <br>
  <small class="aside">...</small>
</span>
<span class="infocard-tall">...</span>
<span class="infocard-tall">...</span>
<span class="infocard-tall">...</span>
```

```
<span class="infocard-tall">
  <a class="pkg pkgRBY n4" data-sprite=" pkgRBY n4 "
    href="http://pokemondb.net/pokedex/charmander"></a>
  <small>#004</small>
  <a class="ent-name" href="http://pokemondb.net/pokedex/charmander">
    <span id="name-#004">Charmander</span>
  </a>
  <small class="aside">
    <a class="itype fire" href="http://pokemondb.net/type/fire">
      <span id="type1-#004">Fire</span>
    </a>
  </small>
</span>
```

문자열의 형태가 HTML이긴 하지만, 그냥 문자열에서 특정 단어를 찾는 일과 같습니다. 이렇게 문자열을 분석해서 원하는 정보를 찾아내는 일을 파싱(Parsing)이라고 합니다.

원하는 내용을 찾아낼 때 유용한 문자열 함수들

```
s = "Hello, world!"

# 문자열의 위치는 0부터 시작합니다.
s.find('H') # 0
s.find('w') # 7
# 만약 없는 문자를 찾으려고하면 -1을 반환합니다.
s.find('?') # -1
# 같은 문자가 여러 개 있을 경우 가장 앞에 있는 문자의 위치를 반환합니다.
s.find('l') # 2

s = 'Hello, world! world!'
# 문자를 찾는것 뿐만 아니라 문자열도 찾을 수 있습니다.
s.find('world') # 7
```

원하는 내용을 찾아낼 때 유용한 문자열 함수들

```
s = 'Hello, world!'

s.find('l') # 2
# 두 번째 인자로 시작 위치를 지정할 수 있습니다.
s.find('l', 3) # 3
s.find('l', 4) # 9

<span id="name-#004">Charmander</span>

a = s.find('name-#004')
s.find('</span>', a)
```

```
def find_name(html: str, n: int):  
    # name-#007을 시작 위치를 찾는다.  
    start = html.find(f'name-#{n:03d}') + 11  
  
    # 포켓몬 이름이 끝나는 위치를 찾는다.  
    end = html.find('</span>', start)  
  
    return html[start:end]
```

f-string으로 찾아야할 문자열을 만들 수 있습니다. `f'name-#{n:03d}'` 에서 `03d`는 정수 `n`을 앞에서부터 3개로 0을 채워서 만들라는 뜻입니다.

```
def find_type(html: str, n: int):  
    result = []  
  
    start = html.find(f'type1-#{n:03d}') + 12  
    end = html.find('</span>', start)  
    result.append(html[start:end])  
  
    if html.find(f'type2-#{n:03d}') != -1:  
        start = html.find(f'type2-#{n:03d}') + 12  
        end = html.find('</span>', start)  
  
        result.append(html[start:end])  
  
    return result
```

어떤 포켓몬은 타입이 한 개고, 어떤 포켓몬은 두 개입니다. 타입 두개를 찾기를 시도해서 없으면 타입이 하나인 것입니다.

bs4와 selenium을 이용한 웹 페이지 파싱

HTML 파일에서 단순히 find()로 문자열을 찾는 것이 아니라, 이를 트리 형태의 객체로 구조화시켜서 트리에서 탐색할 수 있게 해주는 라이브러리가 있습니다. bs4(Beautiful soup)라고 부릅니다.

특정 사이트들은 클릭해야만 데이터를 동적으로 불러오기도 합니다. 마치 사람이 클릭하는 것처럼, 브라우저를 조작해서 웹을 돌아다니면서(크롤링) 페이지의 정보를 가져오게 해주는 라이브러리가 있습니다. Selenium이라고 부릅니다.