빅데이터 API 개발

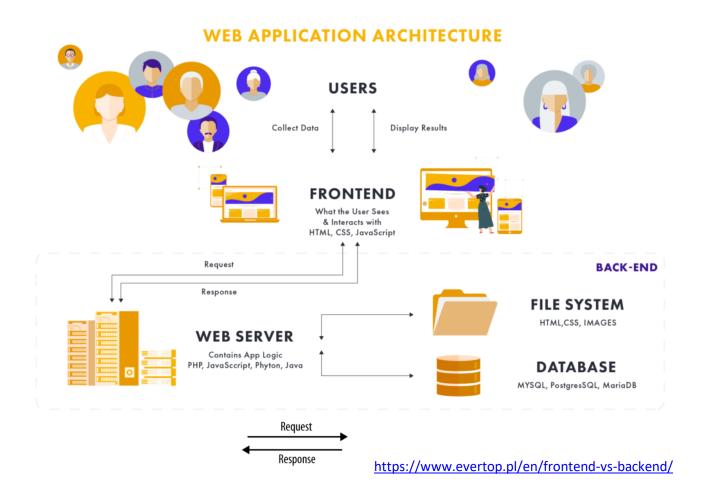
- Database
- Python sqlite3
- Python Requests API
- Python REST API (Fast API)

빅데이터 핵심 하드/소프트 스킬

	대분류	중분류	소분류	
	하드 스킬(Hard Skill)	정보시스템기반기술	■ 운영체제 기초 ■ 네트워크 기초	
		프로그래밍언어활용	■ Python, JAVA, JS 기본 문법 ■ 라이브러리(Library) 활용	
		애플리케이션 설계/개발	■ 객체지향설계/개발	
		테스트 및 배포 (MLOps)	■ 애플리케이션, UI 테스트 ■ 애플리케이션(예측 모델) 배포(Deployment)	
		데이터베이스 활용	■ 데이터베이스 기초 ■ 논리/물리 데이터베이스 설계 ■ SQL 작성 활용	
		데이터 전처리 및 특성 공학	■ 데이터 전처리(Preprocessing) ■ 데이터 변환(Transformation) ■ 특성 공학(Feature Engineering)	
		통계처리 및 분석 지식	■ 데이터 현황분석(EDA, 기초 통계분석) ■ 데이터 통계분석, 데이터 시각화	
		머신러닝 알고리즘 응용	알고리즘 이해 및 적용예측모델 평가 및 선택	
	소프트 스킬(Soft Skill)	Domain 지식 (마케팅, 제조, 특허 등등)		
		사고력, 분석력, 협업 능력, 의사소통 등	5 력	



하드 스킬(Hard Skill)





What is Database

- **데이터베이스(Database, DB)** : 데이터 저장소.
- DBMS(Database Management System, 데이터베이스 관리 시스템) : 데이터베이스를 운영하고 관리하는 소프트웨어.
 - 계층형, 망형, 관계형 DBMS 중 대부분의 DBMS가 테이블로 구성된 관계형 DBMS(RDMBS)형태

DBMS	제작사	작동 운영체제	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	오픈 소스(무료), MySQL 초기 개발자들이 독립 해서 만듦
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	PC용
SQLite	SQLite	Android, iOS	모바일 전용, 오픈 소스(무료)

■ SQL(Structured Query Language): 구조화된 질의 언어라는 뜻으로 관계형 데이터베이스에서 사용되는 언어. 표준 SQL을 배우면 대부분의 DBMS를 사용할 수 있음.



ERD(Entity Relational Diagram)

Professors Table:

- 1. ProfessorID (Primary Key)
- 2. FirstName
- 3. LastName
- 4. Department



to represent the relationship between students, professors & subjects

Course Registration Table:

- 1. RegistrationID (Primary Key)
- 2. StudentID (Foreign Key: Students Table)
- 3. ProfessorID (Foreign Key: Professors Table)
- 4. SubjectID (Foreign Key: Subjects Table)
- 5. Semester
- 6. Grade

Students Table:

- 1. StudentID (Primary Key)
- 2. FirstName
- 3. LastName
- 4. Major

Subjects Table:

- 1. SubjectID (Primary Key)
- 2. SubjectName
- 3. Department



Relational Database (RDB)

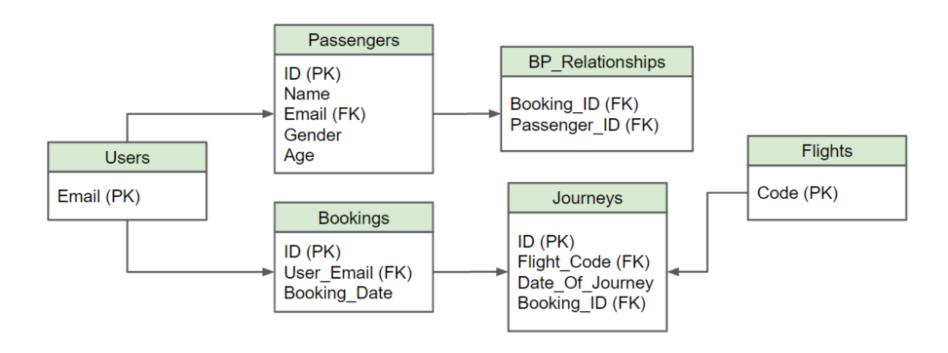
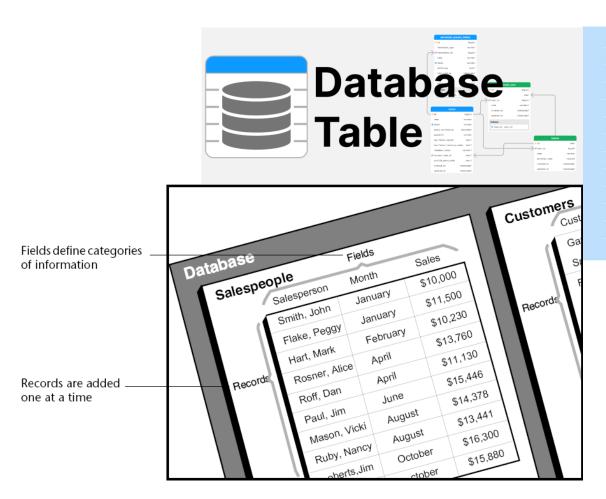
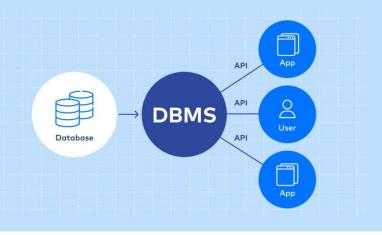




Table in Database

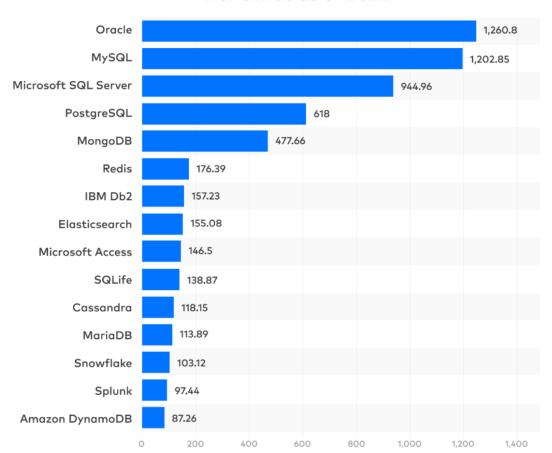






The Popular Database

Most popular database management systems worldwide as of 2022

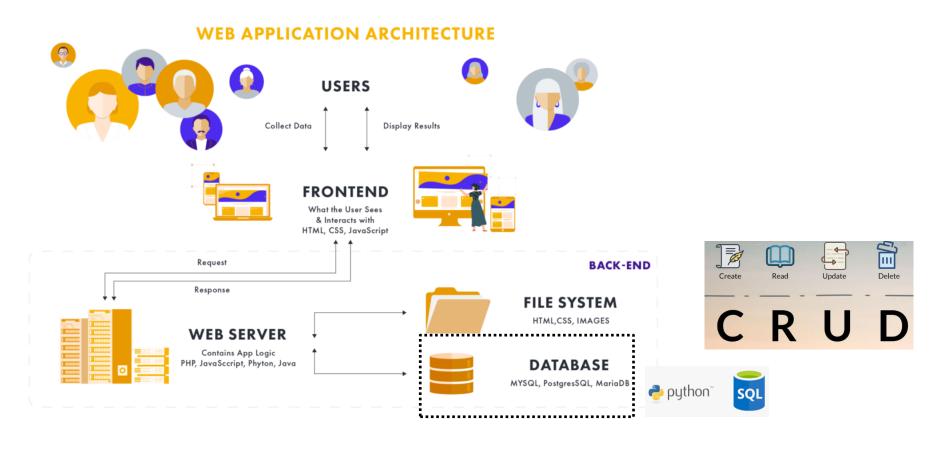






파이썬프로그래밍(2)

- Database
- Python sqlite3
- Python Requests API
- Python REST API (Fast API)



https://www.evertop.pl/en/frontend-vs-backend/



- SQLite는 MySQL나 PostgreSQL와 같은 데이터베이스 관리 시스템
- 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스
- '에스큐엘라이트'또는 '시퀄라이트'라고 읽는다. [위키백과]
- https://www.sglite.org/index.html



Small. Fast. Reliable. Choose any three.

License Support Home About Documentation Download Purchase

What Is SQLite?

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. More Information...

11

The SQLite file format is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way through the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems [1] [2] [3] and as a long-term archival format for data [4]. There are over 1 trillion (1e12) SQLite databases in active use [5].

SQLite source code is in the public-domain and is free to everyone to use for any purpose.

Latest Release

Version 3.43.0 (2023-08-24). Download Prior Releases

Search

Common Links

- Features
- · When to use SQLite
- Getting Started
- Try it live!
- Prior Releases
- SQL Syntax
 - Pragmas
 - SQL functions
 - Date & time functions
 - Aggregate functions
 - Window functions
 - Math functions
 - JSON functions
- C/C++ Interface Spec
 - Introduction
 - List of C-language APIs
- The TCL Interface Spec
- Ouirks and Gotchas
- · Frequently Asked Questions Commit History
- Bugs
- News



I 빅데이터과 I

w02_1_python connect to sql.ipynb

import sqlite3

Connect to a database (creates a new one if it doesn't exist) conn = sqlite3.connect('example.db')



Create a cursor object to execute SQL commands cursor = conn.cursor()





```
# Create the table
cursor.execute(""
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL,
        age INTEGER
    )
"")
```

Insert data into the table

```
[5]

cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", ('John Doe', 30))

cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", ('Jane Smith', 25))

conn.commit()
```



▼ Retrieve data from the table

```
cursor.execute("SELECT * FROM users")
rows = cursor.fetchall()

# Display the retrieved data
for row in rows:
    print(f"ID: {row[0]}, Name: {row[1]}, Age: {row[2]}")

# Close the cursor and the connection
cursor.close()
conn.close()

ID: 1, Name: John Doe, Age: 30
ID: 2, Name: Jane Smith, Age: 25
```

- 데이터베이스로부터 선택된 데이터를 모두 가져와서 rows라는 변수에 저장, 선택된 결과 집합을 행의 목록으로 반환
- ✓ 데이터베이스 연결 및 커 서를 올바르게 열고 닫는 것은 데이터베이스 작업을 수행할 때 중요

```
query = "SELECT * FROM users"
  import pandas as pd
  df = pd.read_sql_query(query, conn)
  df
ProgrammingError
                                          Traceback (most recent call last)
<ipython-input-8-4aa4d0ffd3f4> in <cell line: 4>()
      3 import pandas as pd
----> 4 df = pd.read_sql_query(query, conn)
      5 df
                                     2 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/sql.py in execute(self, *args, **kwargs)
  2014
            def execute(self, *args, **kwargs):
  2015
-> 2016
               cur = self.con.cursor()
   2017
               trv:
  2018
                    cur.execute(*args, **kwargs)
ProgrammingError: Cannot operate on a closed database.
 STACK OVERFLOW 검색
```



i d

w02_1_python connect to sql.ipynb

```
[12]     conn = sqlite3.connect('example.db')
     cursor = conn.cursor()

[13]     query = "SELECT * FROM users"

     import pandas as pd
     df = pd.read_sql_query(query, conn)
     df
```

name age

30

John Doe

2 Jane Smith

데이터베이스 연결을 다시 열고 커서를 만들면 OK



실전문제

w02_1_python connect to sql.ipynb

fake 모듈을 사용하여 가상의 데이터(성명, 주소, 이메일) 1만개를 만들고 db에 저장하라

C Edit on GitHub





_ _ _ _ _ _	_ _ _ _ _ _ _ _	_i _i _i_i _i _i	_ _ _ _ _ _ _ _ _	_ _ _ _ _ _	
version	v19.3.1 () Python Test	s passing	coverage	96% lic
Compatibility					

Docs » Welcome to Faker's documentation!

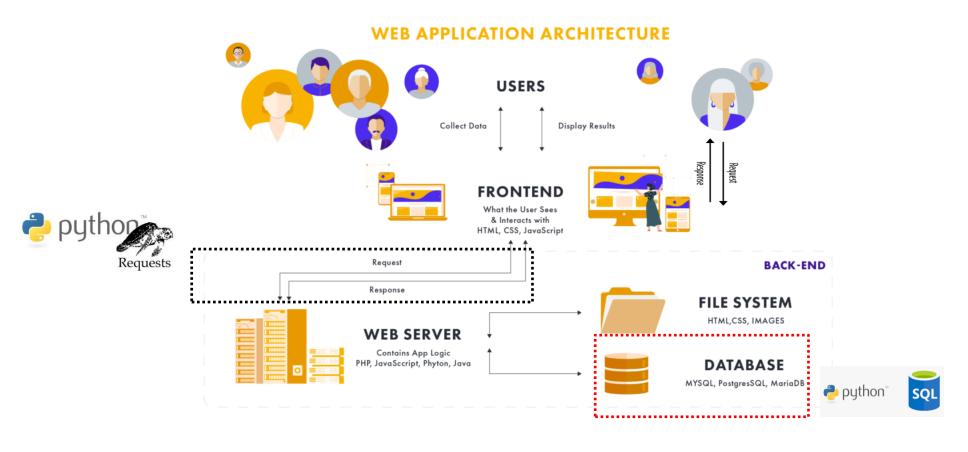
	i d	name	emai I	address
0	1	Cindy Kennedy	tylerrogers@example.com	Unit 4873 Box 8070₩nDPO AE 33956
1	2	lan Gomez	valerie 77@ example.com	2704 Francis Via₩nNicoleside, OH 31894
2	3	Michael Curry	janet 17@ example.com	3804 Matthew Trail Apt. 286₩nLuisville, SD 03707
3	4	Andrew Conway	tiffanyhuerta@example.com	933 Katherine Key Apt. 368₩nMooretown, NY 06222
4	5	John Melendez	crystal 18@ example.net	36907 Robinson Tunnel Suite 878₩nSeanport, NY
9995	9996	Becky Blanchard	hurleytracy@example.net	80128 Fisher Parkways₩nPooleburgh, NE 78127
9996	9997	Jennifer Robinson	bmitchell@example.org	68156 Edwards Locks₩nFletchertown, MH 68230
9997	9998	Daniel Stewart	erica fernandez@example.com	1926 Welch Lakes₩nBellshire, PR 11767
9998	9999	Joseph Vance	kramerrodney@example.org	59988 Rebecca Rapids₩nEast Lisaberg, NE 29881
9999	10000	Kelsey Daniels	patrick12@example.org	93212 Kimberly Vista Apt. 115₩nEast Jacobfurt,

10000 rows × 4 columns



파이썬프로그래밍(2)

- Database
- Python sqlite3
- Python Requests API
- Python REST API (Fast API)



https://www.evertop.pl/en/frontend-vs-backend/



Requests is an elegant and simple HTTP library for Python, built for human beings.

Star 50,156

Requests is an elegant and

currently looking at the

documentation of the

development release.

Useful Links

Quickstart

Advanced Usage

API Reference

Extensions

Issue Tracker

Release History

Contributors Guide

Requests @ GitHub Requests @ PyPI

Recommended Packages and

simple HTTP library for Python,

built for human beings. You are



Requests: HTTP for HumansTM

Release v2.31.0. (Installation)

downloads/month 322M license Apache 2.0 wheel yes python 3.7 | 3.8 | 3.9 | 3.10 | 3.11

Requests is an elegant and simple HTTP library for Python, built for human beings.

Behold, the power of Requests:

>>> r = requests.get('https://api.github.com/use
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"...'
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77

See similar code, sans Requests.

Requests allows you to send HTTP/1.1 requests extrem add query strings to your URLs, or to form-encode your nection pooling are 100% automatic, thanks to urllib3.

Beloved Features

Requests is ready for today's web.

• Keep-Alive & Connection Pooling

Quickstart

Eager to get started? This page gives a good introduction in how to get started with Requests.

First, make sure that:

- · Requests is installed
- Requests is up-to-date

Let's get started with some simple examples.

Make a Request

Making a request with Requests is very simple.

Begin by importing the Requests module:

>>> import requests

Now, let's try to get a webpage. For this example, let's get GitHub's public timeline:

https://docs.python-requests.org/en/latest/user/quickstart/



REQUEST_APIs_quickstart.ipynb

▼ Make a Request

```
import requests
# HTTP GET 요청은 웹 서버에서 정보를 검색하는 데 사용됩니다. 이 요청은 데이터를 서버로 보내지 않고,
# 서버에서 제공하는 정보나 리소스를 가져옵니다.
r = requests.get('https://api.github.com/events')
r
# HTTP POST 요청을 'https://httpbin.org/post URL로 보냅니다. '.
# 이 요청으로 보내는 데이터는 {'key': 'value'} 사전 형식입니다. 이 데이터는 요청 본문의 일부로 서버에 전송됩니다.
r = requests.post('https://httpbin.org/post', data={'key': 'value'})
r
r = requests.delete('https://httpbin.org/delete')
r
```



```
# HTTP HEAD 방법은 리소스의 본문 내용을 실제로 요청하지 않고 웹 서버에서
# 리소스의 헤더를 요청하는 데 사용됩니다.
# HTTP HEAD 요청을 보내면 서버는 콘텐츠 유형, 콘텐츠 길이,
#다양한 메타데이터 등의 정보를 포함하여 리소스와 관련된 헤더로 응답하지만
# 리소스의 실제 콘텐츠를 보내지는 않습니다. 이는 리소스에 대한 메타데이터만 필요하고
#전체 리소스를 다운로드하고 싶지 않을 때 유용합니다.
r = requests.head('https://httpbin.org/get')
r.headers
# HTTP OPTIONS 메서드는 서버에서 지원하는 HTTP 메서드 및 헤더 옵션을 조회하기 위해 사용됩니다.
# OPTIONS 요청을 보내면 서버는 해당 엔드포인트 또는 리소스에 대한 지원되는 HTTP 메서드 목록과 다른 헤더 옵
션을 반화합니다.
#이 메서드는 주로 CORS (Cross-Origin Resource Sharing)와 같은 웹 보안 및 권한 관련 설정을 확인하기 위해 사용됩
니다.
r = requests.options('https://httpbin.org/get')
```



REQUEST_APIs_quickstart.ipynb

▼ Passing Parameters In URLs

```
payload = {'key1': 'value1', 'key2': 'value2'}
r = requests.get('https://httpbin.org/get', params=payload)
r
print(r.url)
#?는 query, & 조건문
https://httpbin.org/get?key1=value1&key2=value2
print(r.url)
print(r.text)
r = requests.get('https://httpbin.org/get')
print(r.url)
print(r.text)
payload = {'key1': 'value1', 'key2': ['value2', 'value3']}
r = requests.get('https://httpbin.org/get', params=payload)
print(r.url)
```



REQUEST_APIs_quickstart.ipynb

▼ Response Content

r = requests.get('https://api.github.com/events')
r.text

print(r.text)

"pretty-print"를 의미하는 pprint 모듈을 가져옵니다. 'pprint' 모듈은 데이터 구조(예: 사전 또는 JSON)를 보다 읽기 쉽고 형식화된 방식으로 인쇄하는 모듈

import pprint
pprint.pprint(r.text)

r.encoding

▼ Binary Response Content

r.content



REQUEST_APIs_quickstart.ipynb

▼ JSON Response Content

```
r = requests.get('https://api.github.com/events')
r.json()

d_json = r.json()

type(d_json)

len(d_json)

for text in d_json:
    print(text)
```

▼ 데이터 구조 이해 해보기

```
import pandas as pd

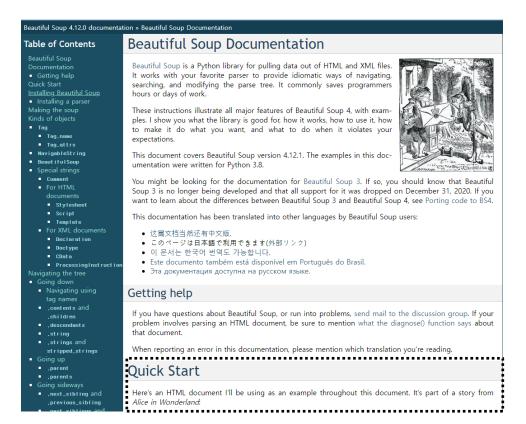
for text in d_json:
    display(pd.DataFrame(text))
    print('*'*100)
```



Beautiful Soup Documentation

REQUEST_APIs_quickstart.ipynb

<u>Beautiful Soup</u> is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.



html_doc = """<html><head><title>The Dormouse's
story</title></head> <body> The Dormouse's
story Once upon a time there were three
little sisters; and their names were Elsie,
<a href="http://example.com/lacie" class="sister"
id="link2">Lacie and <a href="http://example.com/tillie"
class="sister" id="link3">Tillie; and they lived at the bottom of a
well. ... """

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
(soup.prettify())
soup.title
soup.p
```

https://www.crummy.com/software/BeautifulSoup/bs4/doc/



26 | 빅데이터과 |

Beautiful Soup Documentation

REQUEST_APIs_quickstart.ipynb

Quick Start

```
from bs4 import BeautifulSoup

# HTML
soup = BeautifulSoup("Some<b>bad<i>HTML")
soup
print(soup.prettify())
soup.find(string="bad")
soup.i

# XML
soup = BeautifulSoup("<tag1>Some<tag2/>bad<tag3>XML", "xml")
print(soup.prettify())
```

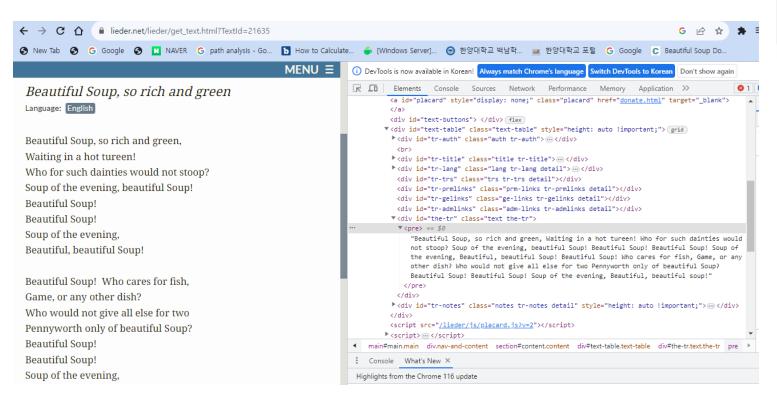


실전문제:

REQUEST_APIs_quickstart.ipynb

요 사이트에 가서 가사(Turtle soup)를 parsing 하고 대문자 소문자 구분없이 'Beautiful' 이 몇 번 나오는 지 count 하라

https://www.lieder.net/lieder/get_text.html?TextId=21635







실전문제:

REQUEST_APIs_quickstart.ipynb

▼ BeautifulSoup

```
[12]
       from bs4 import BeautifulSoup
       url = 'https://www.lieder.net/lieder/get_text.html?TextId=21635'
[14]
       response = requests.get(url)
        response
     <Response [200]>
[15]
       soup = BeautifulSoup(response.text)
       print(soup.prettify())
[16]
[29]
       div_element = soup.find('div', {'id': 'the-tr', 'class': 'text the-tr'})
[30]
       pre_element = div_element.find('pre')
       text = pre_element.get_text()
       pprint.pprint(text)
```

```
for i in text.split():
    if i == 'Beautiful':
        print(i)
    if i == 'Beautiful'.lower():
        print(i)
```

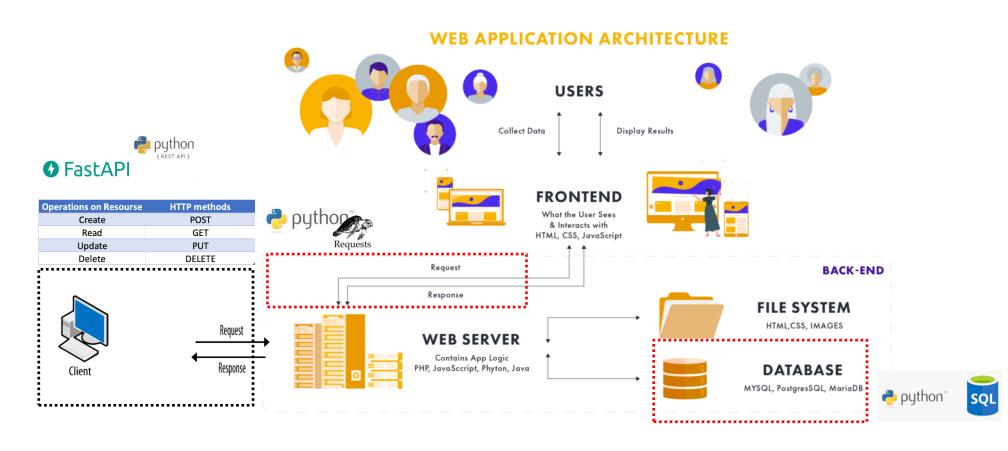
```
Beautiful
```



파이썬프로그래밍(2)

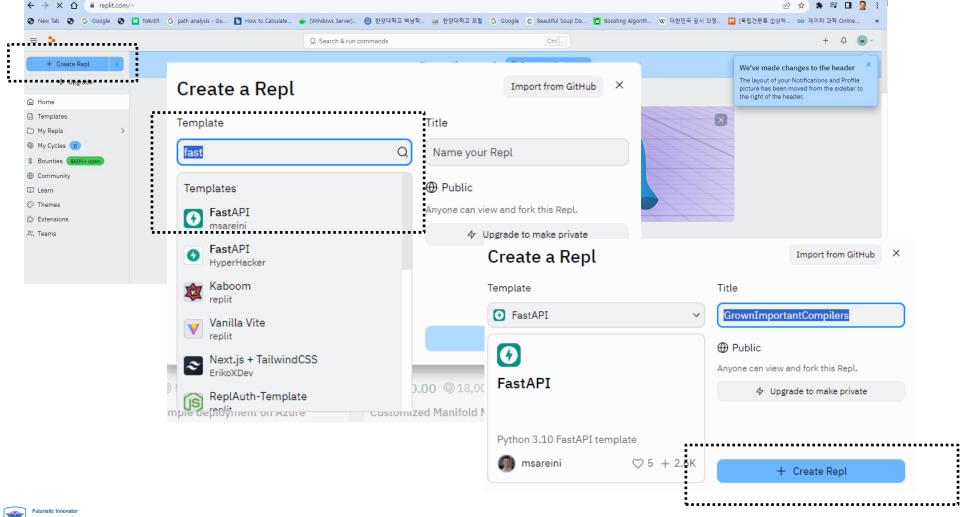
- Database
- Python sqlite3
- Python Requests API
- Python REST API (Fast API)

하드 스킬(Hard Skill)

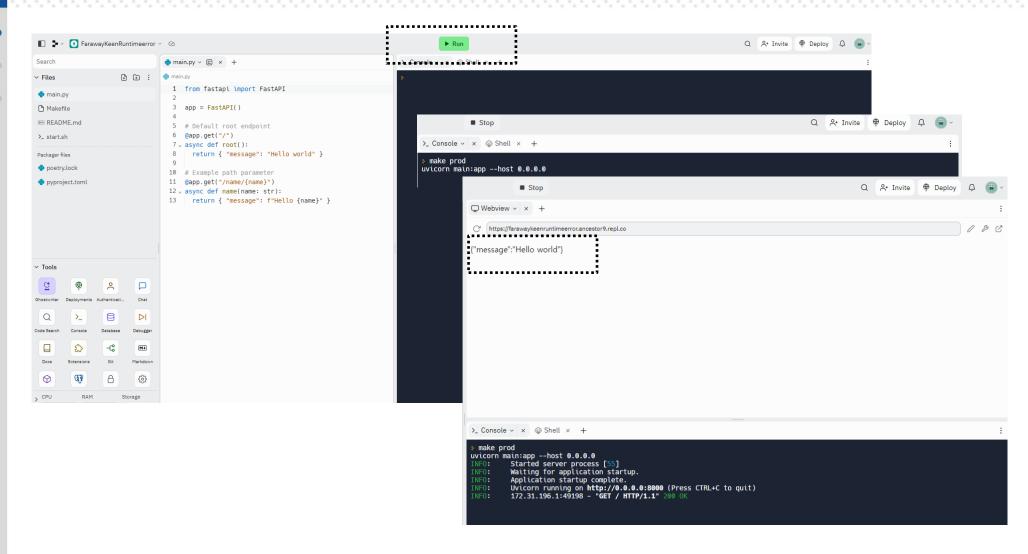


https://www.evertop.pl/en/frontend-vs-backend/

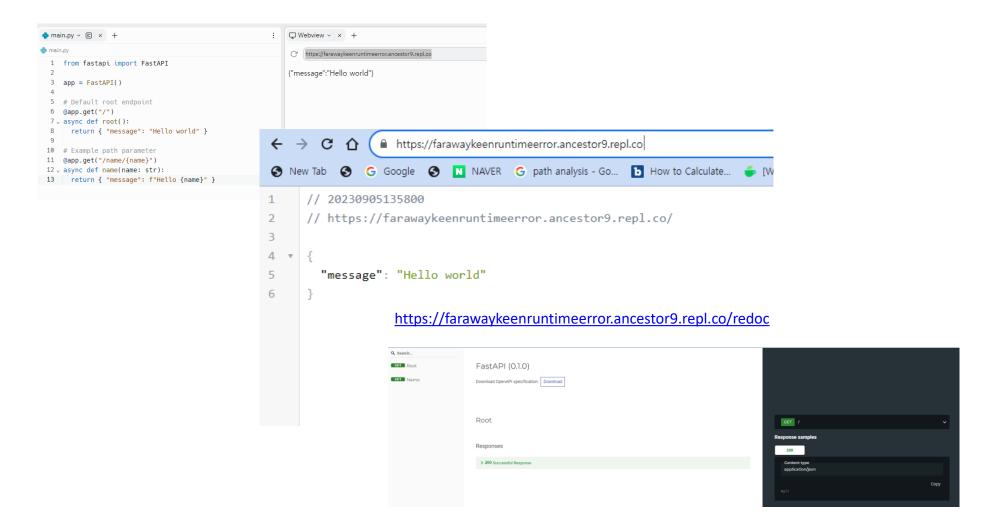






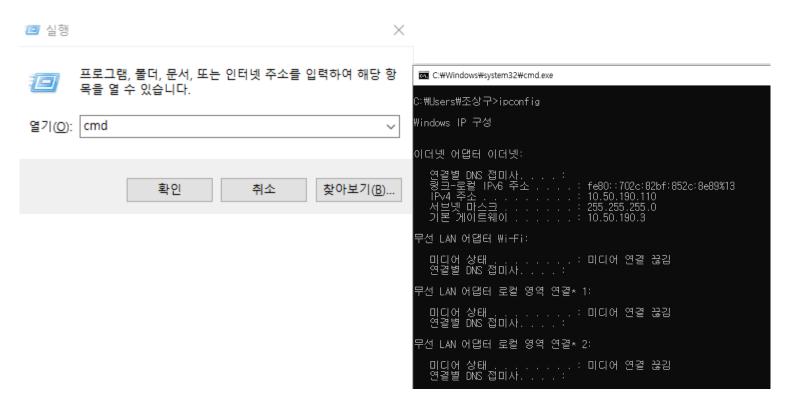








- Windows 실행키 (Windows + R) 을 열면 다음과 같은 화면이 뜬다.
- 'cmd'를 타이핑하고 'Enter'
- prompt 화면에서 'ipconfig'를 치면 현재 로컬컴퓨터의 IP주소를 확인





For ancestor9 eyes only

