

Day 1 - FastAPI

<https://ryuzypjproject.tistory.com/59>

1. Fast API (Application Programming Interface)

- [FastAPI](#)는 Python 기반의 웹 프레임워크로, 주로 API를 빠르게 개발하기 위해 설계되었습니다.
- FastAPI는 자동으로 생성되는 OpenAPI 문서(Swagger UI)를 통해 API의 사용성과 테스트를 쉽게 할 수 있으며, 데이터 유효성 검사와 타입 힌트를 활용한 코드 자동완성 및 오류 방지를 지원합니다.
- 특히, 비동기 처리를 활용하여 속도가 중요한 대규모 애플리케이션 개발에 적합하며, RESTful API와 GraphQL 같은 현대적 웹 API 개발에 유용합니다.

타입 힌팅(Type Hints)

- 타입 힌팅(Type Hints)은 프로그래밍 언어에서 변수, 함수 매개변수, 함수 반환값 등에 대한 데이터 타입 정보를 코드에 명시적으로 제공하는 기술입니다.

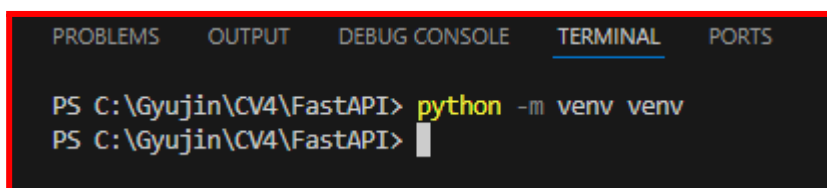
VSCode 가상환경 세팅법:

Terminal → New Terminal

명령어

python -m venv venv

- m : make
- venv : 가상환경 명령어
- venv : 가상환경 이름



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Gyujin\CV4\FastAPI> python -m venv venv
PS C:\Gyujin\CV4\FastAPI>
```

가상환경 실행:

Powershell → Terminal

cd venv\Scripts

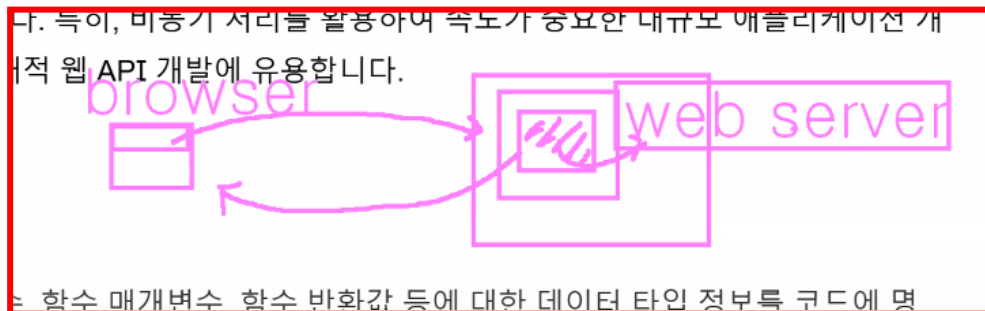
activate.bat

```
(venv) C:\Gyujin\CV4\FastAPI\venv\Scripts>
```

** 윈도우 사용자 이름이 한글이면 안된다. (파이썬이 제대로 작동을 안한다)

Uvicorn 설치:

- FastAPI 애플리케이션을 실행하기 위해 uvicorn이라는 ASGI 서버가 필요합니다. Uvicorn은 Python으로 작성된 가볍고 빠른 웹 서버로, 비동기 웹 애플리케이션과 API를 실행하는 데 사용됩니다. 특히 FastAPI와 잘 어울리며, WebSocket과 HTTP/2 같은 최신 웹 기술을 지원합니다. 개발할 때는 자동 리로드 기능으로 편리하게 작업할 수 있고, 운영 환경에서는 다중 프로세스로 높은 성능을 제공합니다.



- 사이트를 만들거나 응답을 주기위해서 사용 : FastAPI
- FastAPI를 구동시키며, 웹 서버를 만들기위해서 사용 : Uvicorn

```
pip install "uvicorn[standard]"
```

uvicorn[standard]

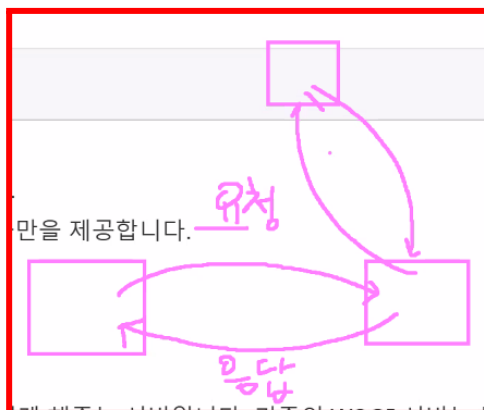
1. 이 패키지는 uvicorn에 기본적으로 포함된 기능 외에도 추가적인 표준 미들웨어를 포함합니다.
2. 표준 미들웨어는 보안, 로깅 및 기타 서버 관련 기능을 추가하는 데 도움이 됩니다.
3. 예를 들어, Gzip 압축, CORS(Cross-Origin Resource Sharing) 지원 등이 기본적으로 포함되어 있습니다.

uvicorn

1. 기본적으로 필요한 최소한의 기능만을 제공하는 패키지입니다.
2. uvicorn[standard]에 비해 미들웨어가 적고, 기본적인 서버 기능만을 제공합니다.

ASGI 서버

- ASGI 서버는 Python에서 비동기 웹 애플리케이션을 실행할 수 있게 해주는 서버입니다.
 - 연산이 많이 필요할때 쉬운 연산순부터 처리
- 기존의 WSGI 서버는 동작이 느리고 실시간 기능(WebSocket, HTTP/2 등)을 지원하지 않았지만, ASGI 서버는 비동기로 동작해 더 빠르고 실시간 데이터 처리도 가능합니다.
- FastAPI 같은 최신 웹 프레임워크가 ASGI 서버와 잘 어울리며, 대표적인 서버로 Uvicorn과 Daphne가 있습니다.
- 이를 통해 많은 요청을 동시에 처리하고, 실시간 기능이 필요한 앱을 쉽게 만들 수 있습니다.



- 웹 사이트는 클라이언트(요청) → 서버(응답) 을 한 뒤 끊어버린다.
- 따라서 게임이나 채팅처럼 실시간으로 연동되는것은 아니다.

블로그에 나와있는 데로 예제 복붙

```
myapp/
├─ main.py
├─ static/
│   └─ script.js
├─ templates/
│   └─ index.html
```

```
main.py > ...
1  from fastapi import FastAPI, Request
2  from fastapi.responses import HTMLResponse
3  from fastapi.staticfiles import StaticFiles
4  from fastapi.templating import Jinja2Templates
5
```

- Jinja2Templates
 - Frontend 를 만들어주기 위해서 사용한다.
 - 페이지를 보여주는 역할을 한다.
- 설치를 먼저 해줘야 한다.
 - **pip install jinja2**

FastAPI 애플리케이션 실행

```
uvicorn main:app --reload
```

- FastAPI는 자동으로 생성된 Swagger UI를 통해 API 문서를 제공합니다.
- 기본적으로 <http://127.0.0.1:8000/docs> 또는 <http://127.0.0.1:8000/redoc>에서 확인할 수 있습니다.
- Swagger UI를 통해 API의 엔드포인트, 매개변수, 응답 형식 등을 살펴볼 수 있습니다.