1. settings.py -> 배포 시 DEBUG = False로 설정해야 하는 이유

텍스트이(가) 표시된 사진

자동 생성된 설명

**오류 날 때 소스의 일부를 보여주기 때문에 보안 상의 문제**

2. 협업할 때 패키지를 팀원과 공유할 수 있는 것

**- pip freeze > requirements.txt**

설치한 패키지 목록이 txt 파일로 저장됨

텍스트이(가) 표시된 사진

자동 생성된 설명

**- pip install -r requirements.txt**

Requirements에 있는 패키지를 설치

3. url 패스 3번째 인자 (name = ‘ ’ ) => **별칭**

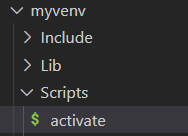
**url 주소 대신 별칭으로 접근 가능**

* 나중에 url 주소가 바뀌더라도 일일이 수정할 필요 X
* 장고는 urls.py 변경을 통해 ‘각 뷰에 대한’ url이 변경되는 유연한 시스템을 갖고 있다.
* url이 변경 되더라도 url reverse가 변경된 url을 추적한다. (누락의 위험이 적다)

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4. 가상환경 실행 시 source 명령어 역할



**activate -> Bash 파일 (환경설정 파일)**

※ Bash: 텍스트 형식으로 저장되어서 한 줄씩 순차적으로 읽어 실행되도록 작성된 프로그램

텍스트이(가) 표시된 사진

자동 생성된 설명

**Source 명령어는 bash 파일을 리부팅 없이 즉시 적용하는 역할!**

리눅스 환경설정 파일들을 수정하기만 한다고 바로 내용이 적용되는 것은 아니다. 리부팅이나 쉘에 재로그인 하지 않고 수정된 새로운 환경 설정 내용을 즉시 적용하기 위해서 source 명령어가 사용된다.

5. http 각각의 매서드 이름 + 역할 알아오기

**GET :** 요청받은 URI의 정보를 검색하여 응답한다. GET 요청은 오로지 데이터를 읽을 때만 사용되고 수정할 때는 사용하지 않는다.

따라서 이런 이유로 사용하면 안전하다고 간주된다. 즉, 데이터의 변형의 위험없이 사용할 수 있다는 뜻이다.

게다가 GET 요청은 여러 번 하더라도 변함없이 항상 같은 응답을 받을 수 있다. 그러므로 GET을 데이터를 변경하는 등의 안전하지 않은 연산에 사용하면 안된다.

|  |
| --- |
| \*URI : 통합 자원 식별자(Uniform Resource Identifier), 웹 기술에 사용하는 논리적, 물리적 리소스를 식별하는 역할. 위 두 주소는 모두 index.html을 식별하여 가리키고 있으므로 URI이다. 그러나 첫 번째는 주소는 웹 서버의 실제 파일 주소를 나타내므로 URL이 될 수 있고, 두 번째 주소는 index라는 파일은 실제 존재하지 않기 때문에 서버 내부에서 처리하는 과정을 거치므로 URL이 될 수 없다. |

**POST :** 요청된 자원을 생성(CREATE)한다. 클라이언트(보통 브라우저)는 서버로 HTTP Body에 Data를 전송한다. 새로 작성된 리소스인 경우 HTTP헤더 항목 Location : URI주소를 포함하여 응답한다

**HEAD :** GET방식과 동일하지만, 응답에 BODY가 없고 응답코드와 HEAD만 응답한다. 웹서버 정보확인, 헬스체크, 버젼확인, 최종 수정일자 확인등의 용도로 사용된다.

**PUT :** 요청된 자원을 수정(UPDATE)한다. 내용 갱신을 위주로 Location : URI를 보내지 않아도 된다. 클라이언트측은 요청된 URI를 그대로 사용하는 것으로 간주함.

**PATCH :**   PUT과 유사하게 요청된 자원을 수정(UPDATE)할 때 사용한다. PUT의 경우 자원 전체를 갱신하는 의미지만, PATCH는 해당자원의 일부를 교체하는 의미로 사용.

**DELETE :** 요청된 자원을 삭제할 것을 요청함.  (안전성 문제로 대부분의 서버에서 비활성)

**CONNECT :**   요청한 리소스에 대해 양방향 연결을 시작하는 메소드이다. 예를 들어, 클라이언트는 원하는 목적지와의 TCP 연결을 HTTP 프록시 서버에 요청한다. 그러면 서버는 클라이언트를 대신하여 연결의 생성을 진행한다. 한번 서버에 의해 연결이 수립되면, 프록시 서버는 클라이언트에 오고가는 TCP 스트림을 계속해서 프록시한다.

|  |
| --- |
| \* TCP (전송 제어 프로토콜)은 두 개의 호스트를 연결하고 데이터 스트림을 교환하게 해주는 중요한 네트워크 프로토콜이다. TCP는 데이터와 패킷이 보내진 순서대로 전달하는 것을 보장해준다. |

**TRACE :** 원격지 서버에 클라이언트가 요청한 자원에 도달하기 까지의 경로를 기록하는 루프백 메시지를 호출하기 위해 테스트용으로 사용.

**OPTIONS :** 웹서버에서 지원되는 메소드의 종류를 확인할 경우 사용.