

startcamp 7일차

프레임워크

요청을 받고 응답을 주는 것이 반복되는데 그 사이의 핵심 로직만 수정하여 사용할 수 있도록 생산성을 높여준 프로그램

서버 ← → 클라이언트

서버 자체에 대한 장고의 부담을 줄이기 위해 정적인 파일 문서(HTML, CSS, JS, 이미지...)를 분리할 수 있게 front와 back 서버를 분리하여 만들 것 ⇒ **Client Side Rendering**

Server Side Rendering VS Client Side Rendering

	Server Side Rendering(서버 사이드 렌더링)	Client Side Rendering(클라이언트 사이드 렌더링)
특징	- 서버에서 페이지를 그려 클라이언트(브라우저)로 보낸 후 화면에 표시하는 기법 - 하나의 서버로 back과 front(Django의 경우 template(HTML,CSS))를 모두 구현 하며 Django의 코드를 HTML로 묶어 보여줌	- web server(vue3) + web application server(Django) - js가 Django의 서버를 json파일로 받아 활용
장점	검색 엔진 최적화와 빠른 페이지 렌더링 서버에서 미리 그려서 브라우저로 보내주기 때문에 페이지를 그리는 시간을 단축할 수 있다.	서버 부하가 적다. 초기 전송되는 페이지의 속도는 빠르다.
단점	서버 부하, 초기 로딩 이후 페이지 이동시 속도가 다소 느리다.	빈 HTML 페이지를 받아 브라우저에서 그리는 서비스에서 필요한 데이터를 클라이언트(브라우저)에서 추가로 요청하여 재구성해야 하기 때문에 전체적인 페이지 완료 시점은 SSR보다 느려진다.

라이브러리 vs 프레임워크

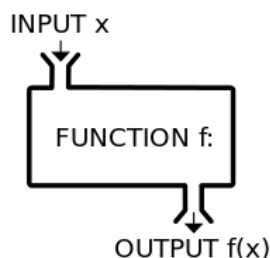
프레임워크 : 뼈대가 정해져있고 이를 반드시 지켜줘야한다. → 형식적인 틀이 존재

ex) vue3

라이브러리 : 형식적인 틀이 없어서 자유롭게 만들 수 있다.

ex) React

함수



x를 넣었을 때 f(x)를 출력하는 코드

- python 내장함수
 - abs(), sum(), str(), list(), int(), type()...

```
# 새로운 함수를 정의
def 함수이름(매개변수1, 매개변수2):
    return(매개변수1 + 매개변수2)
```

```
# 리스트가 주어지면 그 리스트의 평균을 반환하는 함수
grade = [1,2,3,4,5,6]

def average(grade_list):
    return sum(grade_list) / len(grade_list)

print(average(grade))
```