

프레임 워크 고르는 방법

프레임워크를 고민해서 고르는 이유

1. 한번 프레임워크를 결정하면 바꾸는데에 시간과 비용이 많이 든다.
2. 프레임워크에 따라 개발의 속도나, 성능, 즉 비용의 차이가 날 수 있다.
3. 필요이상의 고성능을 추구하느라, 개발 속도의 저하나 팀원 뽑기가 어려워질 수도 있다.
4. 빠른 개발 속도를 추구하느라, 대용량 트래픽을 못 견디게 되는 경우도 발생할 수 있다.

좋은 프레임워크는 상황마다 다르다.

1. 혼자서 다 개발해야하고, 보편적인 서비스 어플리케이션을 만드는 경우는 성능이 좀 떨어지더라도 기본적인 기능이 갖춰진 프레임워크가 좋다.
2. 복잡한 비즈니스 로직이 필요하고, 방대한 코드를 다루게 되어야하는 경우 코드의 재사용을 잘 지원해주는 데코레이터나 코드 제네레이션을 해주는 `cli`가 잘 지원되는 프레임워크가 좋을 수 있다.
3. 극한의 성능이 필요한 경우 인기가 없고, 구현 예시는 별로 없는 힙한 언어, 프레임워크로 갈 수도 있다.

프론트앤드에서의 예시

1. 머신러닝이 필요한 어플리케이션을 개발해야한다. 혹은 동영상을 직접 편집하고 수정하는 어플리케이션을 만든다. -> 고성능 필요 -> 네이티브 어플리케이션으로 개발해야한다.(ex. 안드로이드 앱, 스위프트 IOS 개발)
2. 보편적인 어플리케이션들을 만들려고 한다. 그리고 내가 모바일 프론트앤드 개발이 처음이다 -> 크로스 플랫폼 프레임워크. ex. 플러터
3. 나는 웹 모바일 프론트앤드 개발을 많이 해봤다. -> 웹 앱. ex. Next.js

백엔드에서의 예시

1. 계산 성능은 많이 필요없는데, 커스터마이징과 트래픽만 잘 받아주면 되는 경우 -> **Nest.js**
2. 계산 성능도 많이 필요하고, 커스터마이징이랑 트래픽도 잘 받아줘야하는 경우 -> **Spring + Webflux**
3. 시간이 없어서 남이 잘 만든 기능을 가져다 써야한다 -> **Ruby on rails, python Django**
4. API 서버만 띄우면 되고 성능이 제일 중요 -> **Rust Actix, go gin**

부록) 데이터베이스 고르는 방법

빠른 **API** 반응속도가 필요하고, 데이터 인테그레이션이 깨진다고 해서 세상이 멸망하지는 않는 경우, 혹은 **Schemaless** 하게 진행해야하는 경우 -> **mongodb**

데이터 인테그레이션이 중요하고 이게 깨지면 회사로 안끝나는 경우, 마치 은행, 군대 기록 -> **mysql, postgres**