

프레임워크 학습을 앞두고

1. 알고리즘 vs 프레임워크: 사고의 전환
2. 프레임워크란 무엇인가?
 - 2.1 규칙: "자유를 위한 약속"
 - 2.2 왜 프레임워크를 사용할까?
3. 어떻게 학습해야 할까?

프레임워크 학습을 앞두고

지금까지 여러분은 Python과 알고리즘을 통해 '하나의 문제를 푸는 능력'을 길렀습니다. 이제 Django 프레임워크를 통해, 여러 기능이 유기적으로 연결된 '하나의 완성된 서비스'를 만드는 과정을 시작합니다.

알고리즘이 '정답'을 찾아가는 과정이었다면, 프레임워크는 정해진 규칙 안에서 '최적의 경험'을 만들어가는 과정입니다.

1. 알고리즘 vs 프레임워크: 사고의 전환

구분	알고리즘 학습	프레임워크 학습
목표	입력 → 처리 → 출력 구현	웹 애플리케이션 전체의 큰 틀과 흐름 이해
핵심	효율과 정확성	개발 속도와 유지보수를 위한 구조
초점	로직 구현(Implementation)	기능 활용(Utilization)

2. 프레임워크란 무엇인가?

프레임워크는 웹 개발을 위한 "밀키트"와 같습니다. 훌륭한 웹 서비스를 만들기 위해 필요한 모든 재료(데이터베이스 연동, URL 관리 등)가 손질되어 있고, 따라야 할 레시피(정해진 구조)가 함께 제공됩니다.

우리는 모든 재료를 직접 기르고 손질할 필요 없이, 밀키트의 레시피에 따라 핵심적인 요리(핵심 비즈니스 로직)에만 집중하여 빠르고 맛있게 요리를 완성할 수 있습니다.

2.1 규칙: "자유를 위한 약속"

프레임워크가 제시하는 규칙들은 개발자의 자유를 뺏는 제약이 아니라, 팀 전체의 생산성을 높이기 위한 약속입니다. 이 약속을 따름으로써 모든 팀원이 일관성 있고 효율적인 구조의 코드를 작성하게 되어, 협업이 쉬워지고 유지보수 비용이 크게 줄어듭니다.

2.2 왜 프레임워크를 사용할까?

- **개발 속도 향상:** 반복적인 작업을 대신 처리해줘 핵심 기능에 집중할 수 있습니다.
- **유지보수 편리:** 정해진 구조를 따르므로 코드가 체계적이고, 다른 사람과 협업하기 좋습니다.
- **실무 표준:** 대부분의 실제 웹 서비스는 프레임워크를 기반으로 개발됩니다.

3. 어떻게 학습해야 할까?

1. 도구의 '내부'보다 '쓰임새'에 집중하세요.

- 프레임워크가 제공하는 강력한 기능(도구)들의 내부 코드를 처음부터 파고들 필요는 없습니다. 지금은 **'이 도구로 무엇을, 어떻게 만들 수 있는가'**라는 목적에 집중하고, 필요할 때 공식 문서를 찾아 활용하는 능력이 더 중요합니다.

2. "어떻게 만들지?"보다 "어떻게 조합할지?"를 고민하세요.

- 알고리즘처럼 모든 것을 직접 구현하는 대신, Django가 제공하는 로그인, DB 관리 등의 기능들을 **어떻게 연결하고 조합해야** 내가 원하는 서비스가 완성될지 고민해야 합니다.

3. "일단 만들어보세요": 작은 성공의 경험이 중요합니다.

- 가장 좋은 학습법은 간단한 게시판이나 할 일 목록 같은 **작은 프로젝트를 직접 만들어보는 것**입니다. 이론만 보는 것보다, 하나의 완전한 흐름(Create, Read, Update, Delete)을 직접 경험하는 것이 훨씬 더 많은 것을 남깁니다.

프레임워크 학습은 “조립” 과 “활용” 에 초점을 맞추는 것이 핵심입니다.

알고리즘처럼 한정된 문제를 효율적으로 해결하는 것이 아니며, 주어진 기능을 조합해 실제 서비스를 만드는 과정입니다.

충분한 실습과 함께 점차 Django가 제공하는 기능들을 익혀나가면, 웹 개발 전반을 이해하게 될 것입니다.