

개발 첫걸음, 무엇에 집중해야 할까?

처음 프로그래밍을 시작하면 낯선 개념들이 쏟아져 막막하게 느껴질 수 있습니다. 예를 들어, `float` 타입의 유한 정밀도나 메모리, 성능 문제 등은 분명 중요한 개념입니다. 하지만 처음부터 모든 것을 깊이 파고들기보다는, AI라는 강력한 파트너와 함께 **문제 해결의 큰 그림**을 그리고 **핵심 역량**을 기르는 데 집중하는 것이 훨씬 효과적입니다.

1. 예를들어 "유한 정밀도"는 개념만 이해해도 충분합니다.

- **핵심**
컴퓨터는 실수를 100% 정확하게 표현하지 못하고 '근사값'을 사용한다는 점만 기억하세요.
- **학습 목표**
`0.1 + 0.2`의 결과가 우리가 생각하는 `0.3`과 미세하게 다를 수 있다는 사실을 인지하는 것만으로 충분합니다.
- **심화 학습**
금융 계산처럼 높은 정밀도가 필요할 때는, 나중에 `Decimal` 같은 전문 도구를 사용하면 됩니다. 지금은 스트레스받지 마세요.

2. 메모리, 성능 최적화보다는 "개발 경험"이 우선입니다.

- **기초 단계에서는**
코드의 미세한 성능 차이나 메모리 사용량에 너무 얽매일 필요 없습니다.
- **더 중요한 것**
지금은 에러 메시지를 해결하고, 작은 기능을 직접 완성해보는 실질적인 개발 경험이 훨씬 중요합니다. 개발 경험이 쌓이면 성능에 대한 감각은 자연스럽게 따라옵니다.

3. AI와 함께 문제 해결 능력과 좋은 습관 기르기

- **읽기 쉬운 코드 작성**
AI가 생성한 코드라도 그대로 쓰지 마세요. 의도를 파악해 변수명을 더 명확하게 다듬고, 왜 이렇게 작동하는지 주석을 다는 습관은 여전히 중요합니다.
- **디버깅 역량**
코드가 원하는 대로 동작하지 않는 것이 개발의 진짜 시작입니다.
 - **AI에게 에러 메시지와 코드를 함께 보여주고 "왜 이 에러가 발생했고, 어떻게 해결해야 하는지"라고 직접 물어보세요.** 구글링, 공식 문서를 읽는 것은 여전히 중요합니다.
- **검증은 우리 몫**
AI가 제안한 코드가 정말 올바른지, 더 나은 방법은 없는지 **비판적으로 검토하고 테스트하는 능력**을 키워 나가야 합니다.

4. 깊이 파고들기

- "찾으면 되는" 개념에서 "질문하면 되는" 개념으로

프로그래밍은 평생 학습입니다. 이제 모든 것을 외울 필요 없고 외울 수도 없습니다. 대신, **무엇을 모르는지 파악하고, 그 것을 AI에게 명확하게 질문하여 답을 얻어내는 능력**이 중요해졌습니다.

- 효율적인 학습

당장 쓰지 않는 지식에 몰두하기 보다는, 과제를 하거나 프로젝트를 진행하면서 부딪히는 문제를 해결하는 과정에서 자연스럽게 공부하는 편이 효율적입니다.

5. Python을 왜 배우고, 어떤 가치가 있는지 고민하기

- 언어를 배우는 이유

프로그래밍 언어는 컴퓨터에게 명령을 내리는 수단입니다. 결국 장기적으로는 “무엇을 만들 것인가” 이전에 “왜 프로그래밍을 하는가”를 고민해야 합니다. Python을 배우는 과정은 여러 도구 중 하나를 익히는 것이지만, 본질적으로는 문제를 해결 하고 컴퓨터와 대화 하는 능력을 기르는 데 의미가 있습니다.

- 왜 Python이라는 도구인가

Python은 문법이 비교적 간단해 알고리즘에서의 논리적 사고 표현이 유리하고, 데이터 분석 · 웹 개발 · AI 등 다양한 분야에서 활용도가 높아 실무나 개인 프로젝트에서 빨리 성과를 내기 좋습니다. 또한 풍부한 라이브러리와 커뮤니티 덕분에 문제에 부딪힐 때 해결책을 찾기 수월합니다.

- 결국 언어보다 중요한 것

프로그래밍 언어는 말 그대로 도구일 뿐입니다. “어떤 문제를 해결하고, 무엇을 만들어낼 수 있는지”가 더 중요합니다. Python 자체를 완벽히 익히려 애쓰기보다는, 작은 프로젝트를 통해 코드를 실제로 동작시켜보고, 그 과정에서 문제 해결 능력을 키우는 것이 장기적인 성장에 훨씬 도움이 됩니다.

정리하면

처음 프로그래밍을 배울 때, 모든 것을 세세하게 완벽히 이해하려고 하면 오히려 길을 잃기 쉽습니다. `float`의 유한 정밀도처럼 알아두면 유용한 지식이지만, 그 이면의 메모리 구조나 성능 문제까지 초반부터 너무 깊이 파고들면 정작 중요한 **문제 해결 능력**에 투자할 시간이 줄어듭니다.

AI 시대에는 그 어느 때보다도 빠른 속도로 새로운 기술이 등장합니다. 이런 환경에서 살아남기 위해서는 **기초 개념을 정확히 잡으면서도, 계속해서 실습과 프로젝트로 코딩 감각을 익히는 과정을 반복해 나가야 합니다**. 과거처럼 모든 문법을 외우고 코드를 빨리 짜는 능력에만 의존할 수는 없습니다. 그 역할은 이제 AI가 상당 부분 대신하기 때문입니다.

이제 유능한 개발자는 단순 '코더(Coder)'가 아닌, AI라는 강력한 도구를 지휘하는 '**문제 해결 전문가**'입니다. 여러분의 진짜 경쟁력은 "**무엇을 만들고 싶은가?**"라는 본질적인 질문을 던지는 능력, 그리고 AI에게 정확한 요구사항을 전달하고 그 결과물을 비판적으로 검증하는 능력에서 나옵니다. 이러한 능력은 이론 공부만으로는 길러지지 않으며, 오직 **실습과 프로젝트를 통해 아이디어를 실현하는 과정에서 단련됩니다**.

결국 AI 시대에도 변치 않는 핵심은 "**코드를 통해 아이디어를 실현하고, 실용적인 결과물을 만드는 과정**"을 즐기고 습관화하는 것입니다. SSAFY에서 AI와 협업하는 습관을 기르며 이 과정에 최선을 다해보세요. 그렇게 꾸준히 학습하고 경험을 쌓는 경험이 여러분을 대체 불가능한 개발자로 성장시킬 것이라 확신합니다.