






개발 업무의 핵심은 '문제 해결'


개발자는 문제 해결사입니다

 "질문해도 될까요?"

 "A 부분에서 B 에러가 발생했는데, C와 D 방법을 시도해봤지만 해결되지 않았어요. 어떤 방향으로 접근하면 좋을까요?"

동등한 입장에서 협업하기

-  소극적 질문이 아닌, 적극적 문제 공유
-  내가 시도한 과정을 명확하게 정리해서 소통
-  문제 상황에 대한 맥락과 배경 정보 함께 제공하기

 개발은 문제 해결의 연속입니다. 문제를 어떻게 접근하고 해결하는지가 개발자의 성장을 결정합니다.

</> 코드 이해도와 책임감

🤖 AI 도구 활용의 올바른 자세

- ✅ AI로 코드를 생성했다면 → 반드시 한 줄씩 이해하고 설명할 수 있어야 함

⚠️ "동작은 하지만 이해 못함" = 시한폭탄

- ✅ 내가 작성하지 않은 코드라도 내 책임으로 받아들이기

📄 인풋/아웃풋 정리의 중요성

- ✅ 중요한 기능의 입력값과 출력값을 문서화
- ✅ 예상되는 상황 고려하기
- ✅ 다른 사람이 봐도 이해할 수 있는 수준으로 정리

💡 이해하지 못하는 코드는 미래의 문제를 야기합니다. 코드의 작동 원리를 이해하는 것이 지속 가능한 개발의 기본입니다.

📈 발표와 보고에서 보여줄 것들

🕒 노력의 과정을 보여주는 방법

- ✔ 시행착오 과정을 정리해서 공유
- ✔ 어려웠던 부분과 해결 과정 설명
- ✔ 배운 점과 개선할 점 솔직하게 언급

≡ 효과적인 발표 구성

- 1 무엇을 해결하려고 했는가?
- 2 어떤 방법들을 시도했는가?
- 3 최종 결과와 한계점은?
- 4 다음에는 어떻게 개선할 것인가

성장하는 개발자의 마인드셋

주도적 학습 태도

- ✔ 모르는 것을 부끄러워하지 말고, 모른 채로 넘어가는 것을 부끄러워하기
- ✔ 이해될 때까지 파고들기
- ✔ 질문하기 전에 탐색과 정리

피드백 받아들이는 자세

- ✗ "죄송합니다, 다음부터 조심할게요"
- ✔ "이 부분이 부족했네요 다음엔 이렇게 개선하겠습니다"

- ✔ 지적받은 부분을 성장 기회로 받아들이기
- ✔ 같은 실수를 반복하지 않는 시스템 만들기

💡 실수는 실패가 아니라 배움의 기회입니다. 성장하는 개발자는 피드백을 두려워하지 않고 오히려 환영합니다.

★ 관심 분야를 찾아 전문성 키우기

🚀 적극성의 원동력은 관심입니다

- ✔ 프론트엔드? 백엔드? AI? 데이터? 인 프라?
- ✔ 내가 흥미를 느끼는 영역을 적극적으로 탐색하기
- ✔ 관심 분야가 생기면 자연스럽게 더 깊이 파고들게 됩니다


📈 전문 분야를 정하면 달라지는 것들

- ✔ 질문의 수준이 높아집니다. "이 기술을 왜 선택하셨나요?"
- ✔ 스스로 학습 방향을 설정할 수 있습니다
- ✔ 업무에서도 해당 분야의 문제에 더 적극적으로 참여하게 됩니다

실무에서 인정받는 인턴이 되려면

신뢰를 쌓는 방법

 진행상황을 정기적으로 공유하기

 문제가 생기면 빨리 알리고 해결책 함께 고민하기

 어려움이 있다면 미리 상의하기

계속 성장하는 개발자가 되기 위해

 관심 분야의 최신 트렌드 꾸준히 팔로우하기

 너만의 개발 노트나 블로그 작성하기

 동료들로부터 적극적으로 피드백 구하기

 실무에서 인정받는 개발자는 기술력뿐만 아니라 소통능력과 성장의지로 주목받습니다.

“ 신뢰는 하루 아침에 쌓이지 않습니다. 매일의 작은 행동과 책임감 있는 태도가 쌓여 만들어집니다.”

마무리 메시지 – 성장의 4가지 핵심

“ 완벽하지 않아도 괜찮습니다. 하지만 성장하려는 노력만큼은 보여주세요 ”

1



문제 해결자 마인드

질문하기 전에 먼저 시도하고 정리하기. 어려움을 마주했을 때 시도한 접근법과 결과를 명확히 공유합니다.

2



주인의식

내가 만든/사용한 모든 코드에 대해 책임지기. 이해하지 못하는 코드는 시한폭탄과 같습니다.

3



관심 분야 개발

나만의 전문성을 키워 적극적인 자세 만들기. 관심 분야가 생기면 자연스럽게 더 깊이 파고들게 됩니다.

4



성장의지

실수를 감추지 말고 개선 방향 제시하기. "죄송합니다"보다 "다음엔 이렇게 하겠습니다"라고 말하세요.



마지막 조언: 관심 있는 분야를 하나 정하고 그 영역에서만만큼은 팀에서 가장 열정적인 사람이 되어보세요.
그 열정이 여러분을 더 나은 개발자로 만들어줄 것입니다.