

SLASH 24

빠르게 성장하고 싶은 주니어 개발자를 위한 소프트 스킬 5가지

임희진
Toss Foundation Leader

본 발표자료의 저작권은 연사에 있으며, 저작권자의 사전 서면 동의 없이 자료의 일부 또는 전부를 이용하거나 배포할 수 없습니다.

또한 해당 자료를 복제하여 SLASH 행사 홈페이지를 제외한 온라인상에 게재하는 행위는 연사가 동의한 저작권 및 배포전송권에 위배됩니다.

토스가 다루는 모든 개인정보는 고객에게 동의를 받은 후에 처리되고 있으며, 접근 권한이 분리되어 있습니다.

개발자는 모든 데이터가 아닌 담당 영역에 한하여 접근·이용할 수 있습니다.



왜 이 발표를 하게 되었나? - 하드 스킬 vs 소프트 스킬

주니어 개발자 멘토링을 해보면 보이는 공통적인 패턴

하드 스킬: 알아서 척척

소프트 스킬: 그런 것도 익혀야 하나요?

하드 스킬
프로그래밍
아키텍처 설계
트러블 슈팅

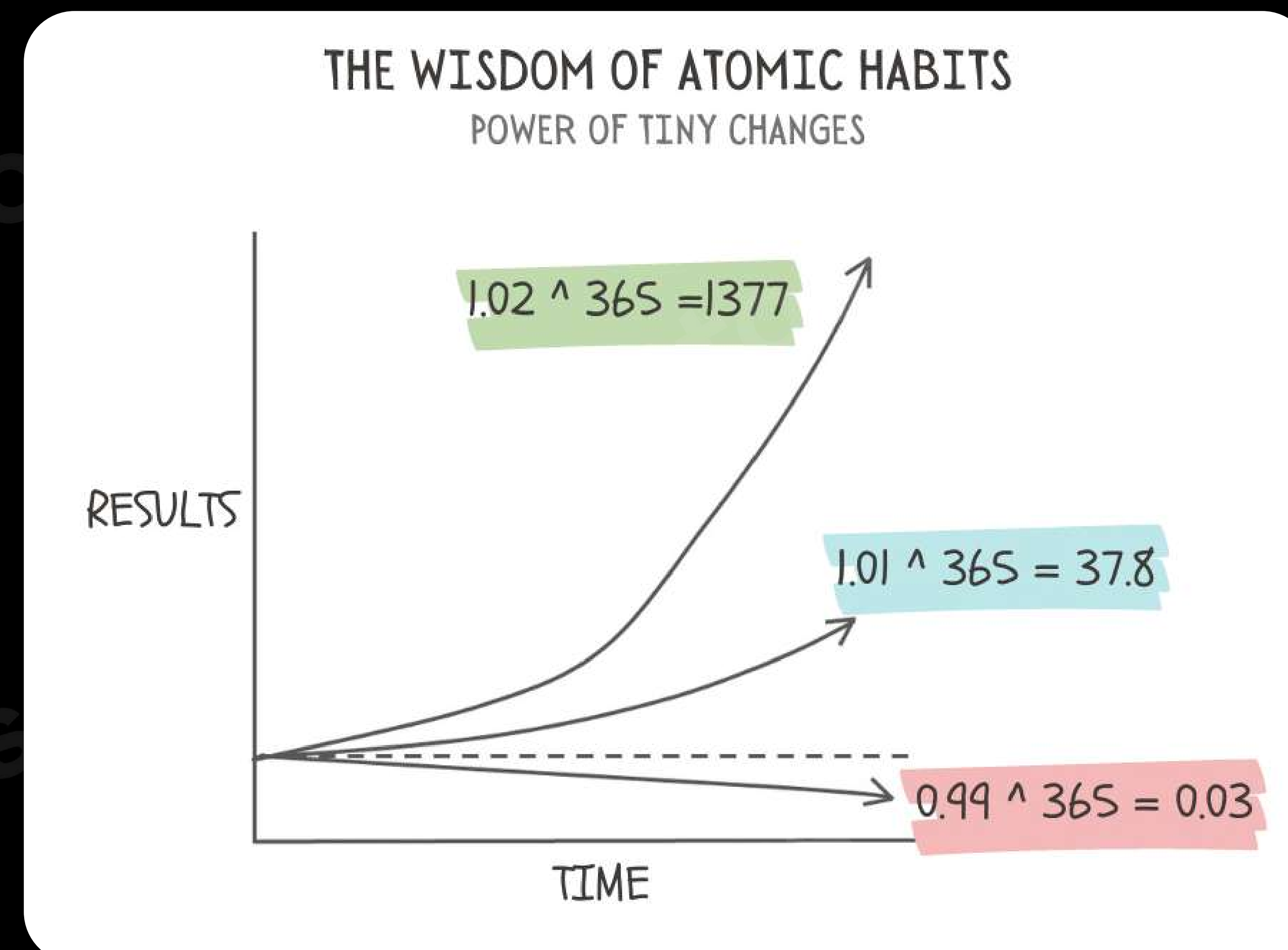
소프트 스킬
커뮤니케이션
시간 관리
문제해결 능력
메타인지 능력

왜 이 발표를 하게 되었나? - 성장의 복리 효과

주니어 때는 성장하는 법을 배우고 성장 곡선의 기울기를 가파르게 만들어야 할 때

하드 스킬이 성장의 더하기라면 소프트 스킬은 곱하기

한 번 익혀두면 평생 써먹을 수 있는 스킬



출처 : <https://wisdomproject.substack.com/p/atomic-habits-key-ideas-notes-and>

발표자는 누구길래?

서버 개발자 10년, Technical PO 3년차

서버 개발자: 카카오 광고 플랫폼, 토스페이, 토스증권, 토스 슈퍼앱, 토스인증서

Technical PO: 토스인증서, 송금, 토스 비즈니스



SLASH 23 - 토스 시니어 개발자가 말하는 커리어 패스



THE TEaM: 토스팀 워크스토리

오늘 얘기할 소프트 스킬 5가지

1. 그림으로 소통하기
2. 나만의 TODO List 만들기
3. 코드를 보는 안목 기르기
4. 왜? 라고 묻기
5. 나만의 개발 공부법 찾기

1. 그림으로 소통하기

개발자는 커뮤니케이션을 못하는 것이 아닙니다

커뮤니케이션 ≠ 말하기

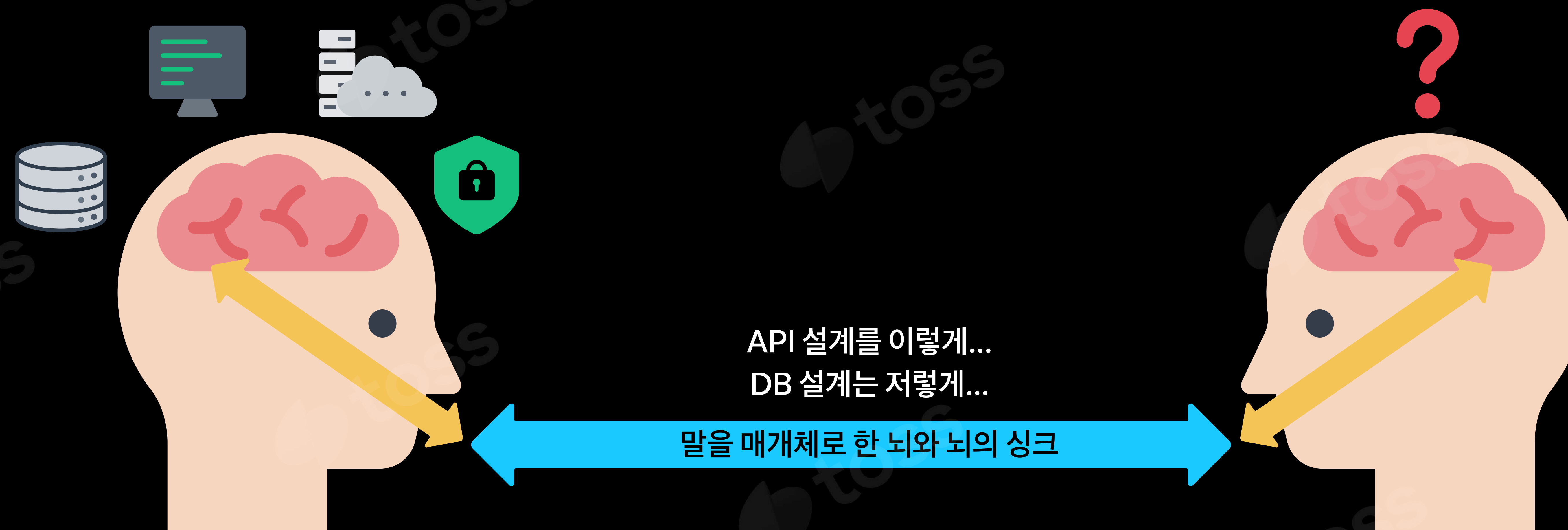


1. 그림으로 소통하기

개발자는 커뮤니케이션을 못하는 것이 아닙니다

커뮤니케이션 ≠ 말하기

말을 매개체로 전달하고자 하는 정보가 말로 표현하기에 너무 복잡한 것

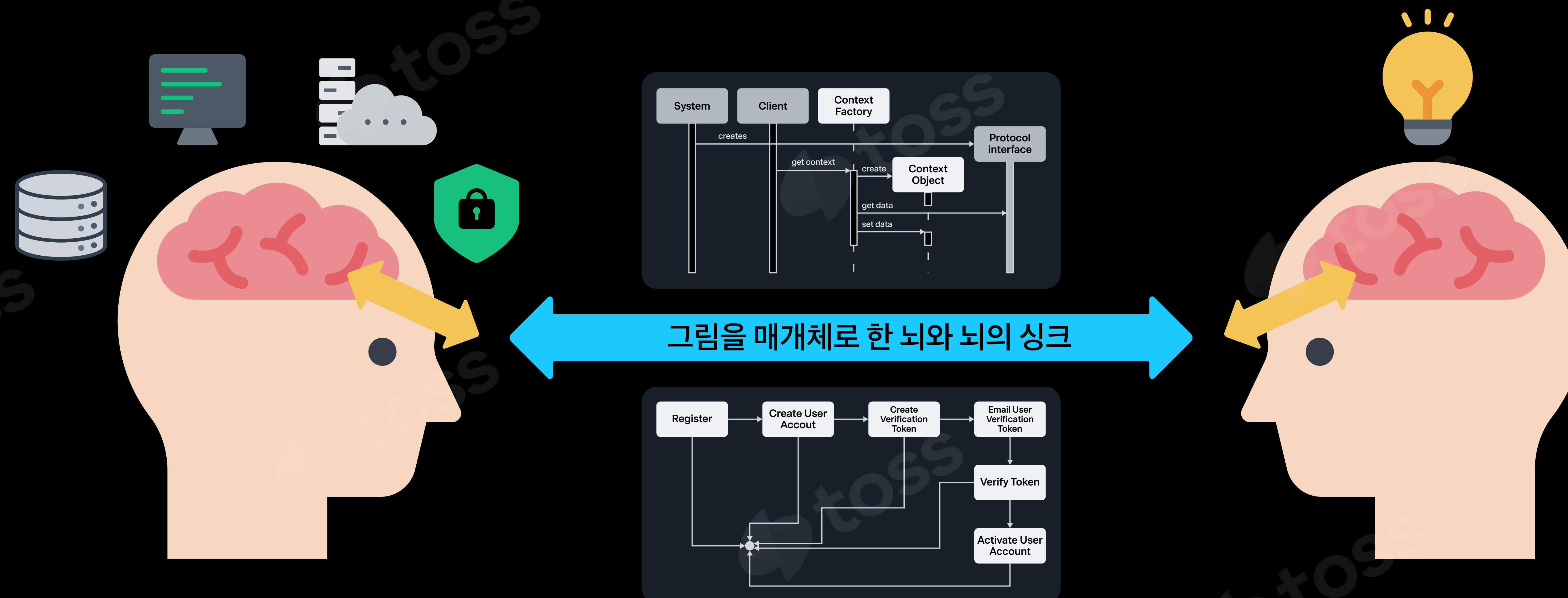


1. 그림으로 소통하기

생각/이미지 → 구어 → 생각/이미지 (정보유실 매우 큼)

생각/이미지 → 문장 → 생각/이미지 (정보유실 큼)

생각/이미지 → 도표 → 생각/이미지 (정보유실 상대적으로 적음)



1. 그림으로 소통하기

제안

그림으로 소통하는 습관을 들여야 합니다

다이어그램 툴 하나를 정해서 익숙해져야 합니다

2. 나만의 TODO List 만들기

우리는 우리가 평생 해내온 TODO의 결과물

모두에게 공평하게 주어진 시간. 그 시간을 더 잘 쓰기 위한 TODO List

그런데 의외로 많은 개발자분들이 TODO List가 없습니다

PM이 관리하는 프로젝트 JIRA를 TODO List로 쓰는 경우도 많습니다

메모장이나 슬랙으로 TODO를 관리하는 경우도 많습니다

2. 나만의 TODO List 만들기

내가 중심인 TODO List를 만들어야 합니다

그 안엔 일도 있고

공부할 것도 있고

매일 되새겨야 할 명언도 있고

사적으로 챙겨야 할 일도 있어야 합니다

즉, 내 뇌의 백업처럼 동작해야 합니다

TODAY ▾

+ 할 일 추가 ⋮

- ☐ [어려운 것을 먼저하라.
쉬운 것은 알아서 해결될 것이다.]
- ☐ 정산 배치 속도 올리기
 - ☐ 멀티 스레드 적용해볼까
 - ☐ 타 팀 사례 수집해보기
- ☐ CS 밀린 것 해결하기
- ☐ 회식 장소 예약하기
- ☐ SLASH24 발표자료 만들기
- ☐ 리팩토링 책 읽기
- ☐ 10월 여행계획 짜고 예약하기

2. 나만의 TODO List 만들기

내 뇌의 백업인 TODO List는 어디서든 손쉽게 접근할 수 있어야 하고,
항상 동기화 되어 있어야 합니다

회사든 집이든 출근길이든 휴가지든

휴대폰이든 태블릿이든

앱이든 웹이든

Mac이든 Windows든

iOS든 Android든

2. 나만의 TODO List 만들기

계속 손이 가는 TODO List를 만들어야 합니다

복잡한 기능이 많은 TODO List는 오히려 손이 안 감

일, 프로젝트, 공부, 취미, 여행 등 목적별로 만들면 관심 있는 TODO List만 열심히 봄

TODO가 계속 쌓이면 잘 안 보게 됨

2. 나만의 TODO List 만들기

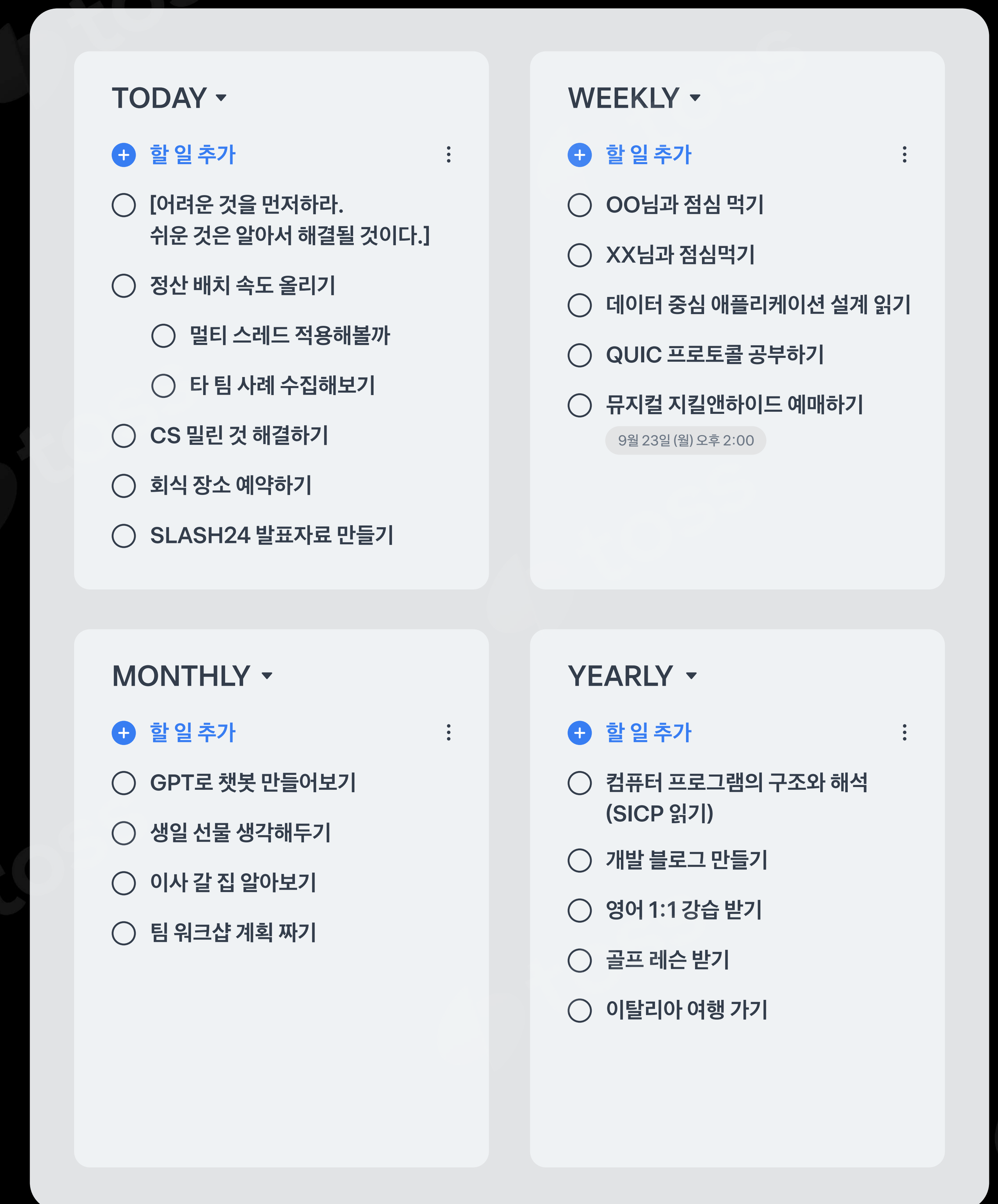
제가 10년간 써 온 방식

Google Task

Today, Weekly, Monthly, Yearly로 주기별로 TODO List를 나눈다

평소엔 Today 위주로 본다

각 주기에서 오래 머문 TODO는 다음 주기로 이동한다



2. 나만의 TODO List 만들기

제안

내가 중심인 TODO List를 만들어야 합니다

어디서든 손쉽게 접근할 수 있어야 하고, 항상 동기화 되어있어야 합니다

계속 손이 가는 TODO List를 만들어야 합니다

3. 코드를 보는 안목 기르기

기존에 있던 잘 돌아가는 코드, 선배들의 코드가 답이 아닙니다

기존 코드를 마냥 우수하다고 생각하면 안 됨

다양한 요구사항, 수많은 히스토리, 빡빡한 일정

안목이 없으면 현실적인 코드에 익숙해지게 됨

불편함(Code Smell)을 느낄 수 있는 안목을 길러야 함



3. 코드를 보는 안목 기르기

제안

평소에 쓰는 오픈소스 코드 열어 보기

유명한 오픈소스 프로젝트 코드 통째로 받아서 공부하기

소스코드 가독성, 리팩토링, 디자인 패턴을 다루는 책 다양하게 읽어보기

평소에 코딩 70%, 리팩토링 30% 시간 쓰기

(취미로) 시간 날 때마다 예전에 본인이 짰, 다른 사람이 짰 소스코드 리팩토링 하기

4. 왜? 라고 묻기

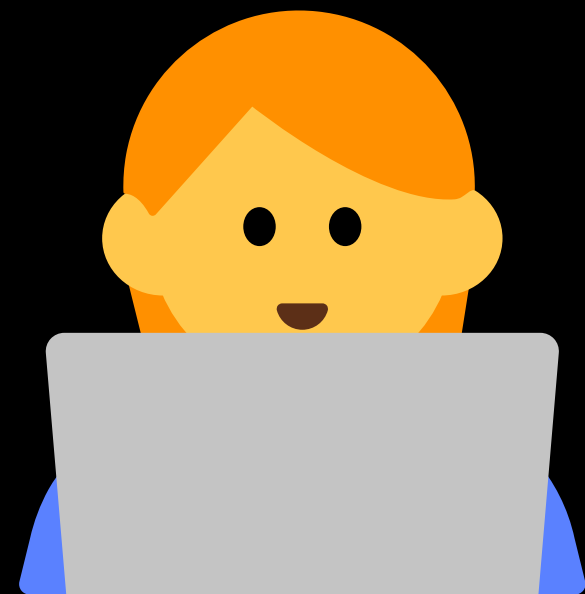
개발자는 개발을 넘어 문제를 해결하는 사람이어야 합니다

최악의 개발: 개발은 실컷하고 문제는 해결 못 함

최고의 개발: 개발도 안 했는데 문제 해결

문제의 본질을 파악하기 위한 습관: "왜?"

4. 왜? 라고 묻기



거래 내역 조회할 수 있는 API 알려주시겠어요?

/api/v2/transactions 쓰시면 됩니다



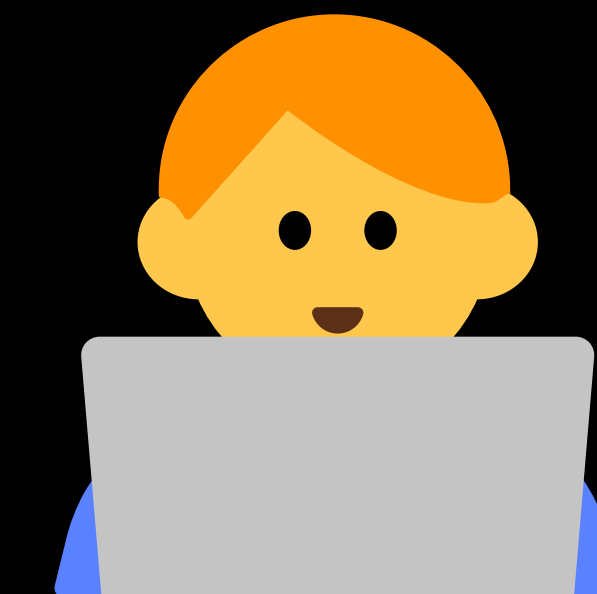
4. 왜? 라고 묻기



사용자 휴대폰 번호 조회할 수 있는 API가 필요해요

이틀 후 개발 끝난 뒤

/api/v2/phone 쓰시면 됩니다



4. 왜? 라고 묻기



거래 내역 조회할 수 있는 API 알려주시겠어요?

혹시 풀고자 하는 문제가 어떻게 되세요?
더 나은 해결 방안이 있을지 같이 고민해보려구요



새벽 배치에서 거래 내역 읽어서
적립금 지급 대상자 추출하려구요

새벽 배치가 임팔라에 쿼리 해서 지급 대상자
추출하면 훨씬 쉽게 해결할 수 있을 것 같아요



4. 왜? 라고 묻기



사용자 휴대폰 번호 조회할 수 있는 API가 필요해요

혹시 풀고자 하는 문제가 어떻게 되세요?
더 나은 해결 방안이 있을지 같이 고민해보려구요



문자 메시지 발송을 위한 휴대폰 번호가 필요해서요

userId로 문자 메시지를 발송할 수 있어요
/api/v2/sms 쓰시면 됩니다



4. 왜? 라고 묻기

내가 요청할 때도 상대방이 "왜?"를 쉽게 알 수 있도록 합니다



새벽 배치에서 거래 내역을 읽어와서
적립금 지급 대상자를 추출하고 싶습니다.

제가 생각한 해결책은 아래와 같습니다.

1. 배치에서 거래 내역 API를 호출한다.
2. 배치에서 거래 내역 DB에 직접 쿼리를 날린다.
3. 혹시 더 좋은 방법이 있을지?

4. 왜? 라고 묻기

제안

누군가가 요청할 때 "왜?" 라고 묻는 습관 들이기

내가 요청할 때도 "왜?"를 설명하는 습관 들이기

5. 나만의 개발 공부법 찾기

사람마다 본인에게 맞는 개발 공부법이 있습니다

책과 이론을 바탕으로 공부하는 사람

실전을 바탕으로 공부하는 사람

5. 나만의 개발 공부법 찾기

제가 신입 개발자 시절 1년 반 걸려서 찾은 공부법

1. 무엇을 모르는지 알기 위해 얇은 책이나 오버뷰 읽기

이 기술이 왜 필요한지, 기존 기술보다 왜 좋은지 학습하기

2. 튜토리얼 코드를 짜보면서 경험의 크기를 늘리기

3. 경험이 쌓이면 책보고 이론 쌓으며 경험 다듬기

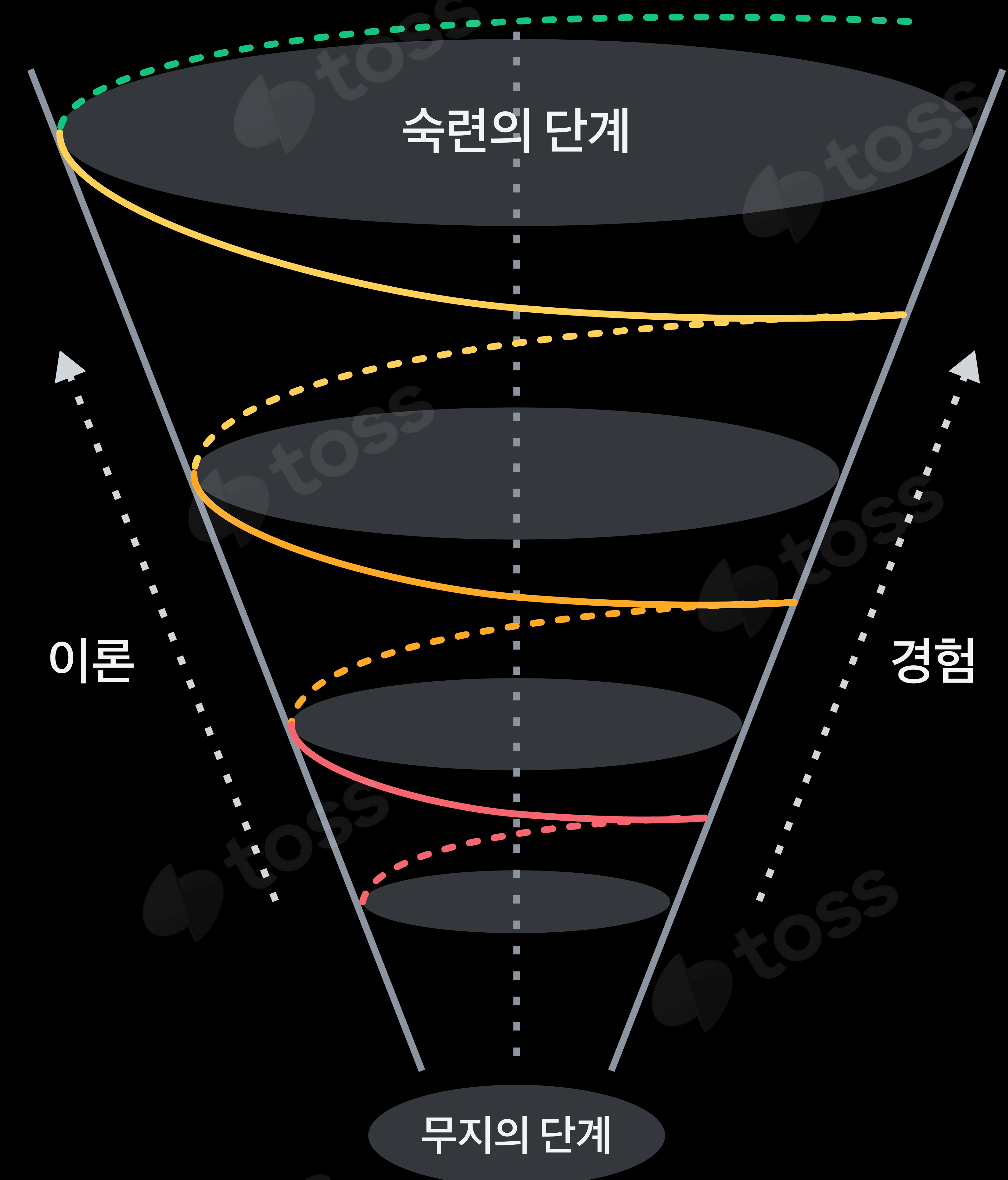
4. 책 바꿔가며 2-3 단계 반복

저자마다 다르게 애길 하기 때문에,

머릿속에 종합해서 나만의 지식으로 만드는 데 도움이 됨

5. 관련된 오픈소스 파보기

6. 회사에서 일 하면서 써보기



5. 나만의 개발 공부법 찾기

제안

이론/경험 편식하지 않고 골고루 시도해보기

시행착오를 거치면서 나와 가장 잘 맞는 개발 공부법 찾기

오늘 얘기한 소프트 스킬 5가지

1. 그림으로 소통하기
2. 나만의 TODO List 만들기
3. 코드를 보는 안목 기르기
4. 왜? 라고 묻기
5. 나만의 개발 공부법 찾기

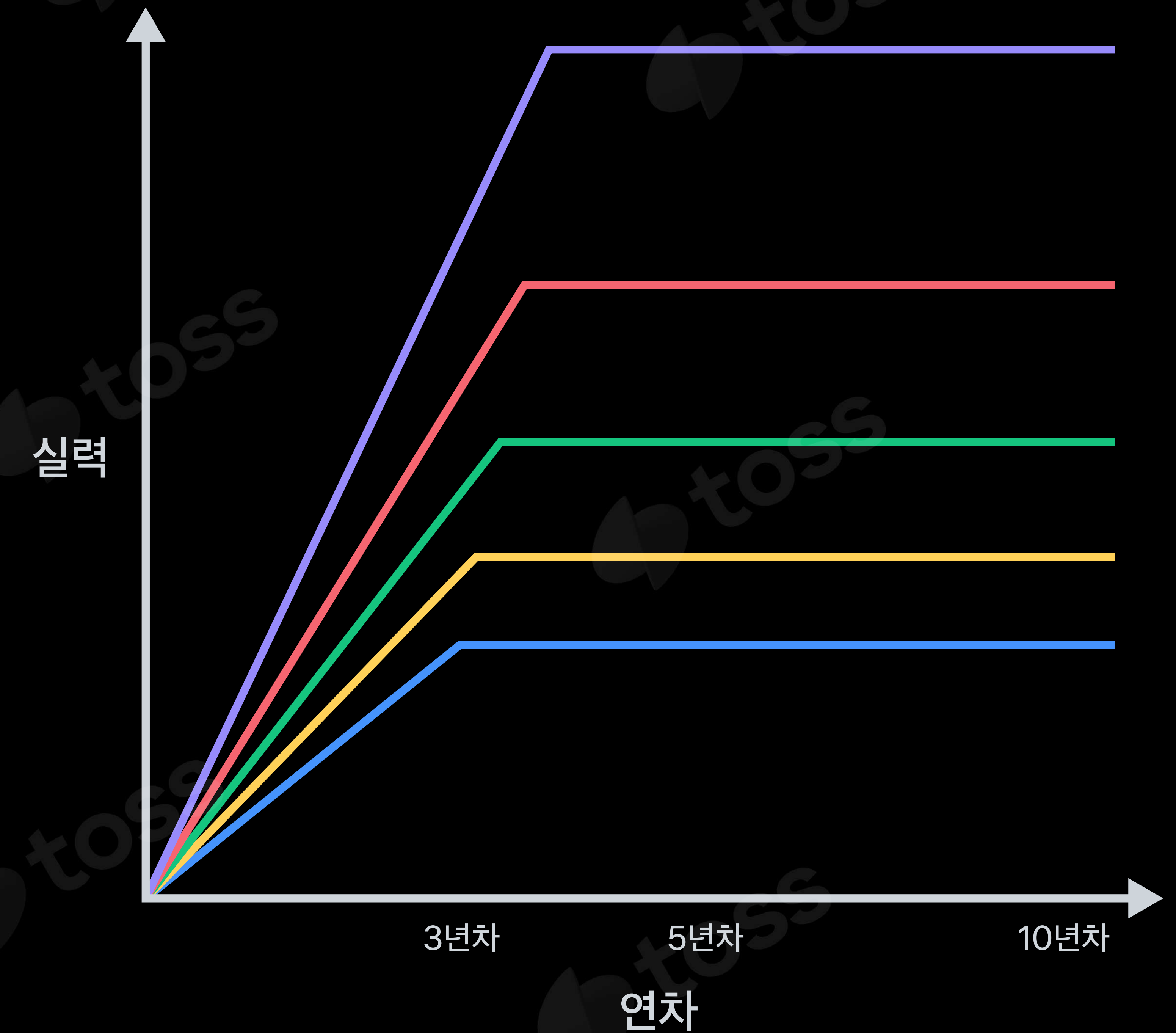
주니어는 성장이 본업이다

주니어 첫 3년에 성장 곡선의 기울기를 바꾸면
10년 후 엄청난 실력 차이가 발생합니다

3년차가 3년차 만큼 한다고 잘 한다고 생각하고 안주하면 안 됩니다

3년차가 5-7년차의 일을, 5년차가 10년차의 일을 하는 개발자가

진정 실력 있는 개발자 입니다



Q&A

