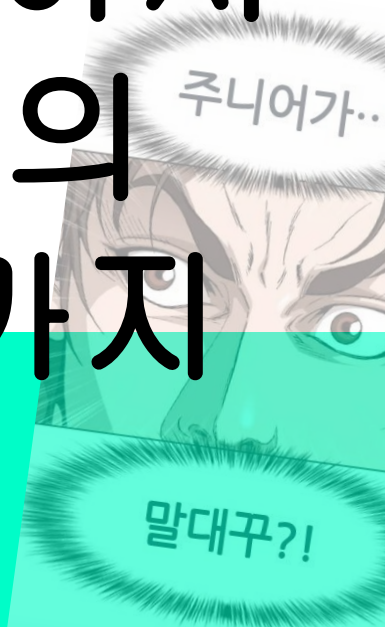


취업은 확정됐는데 아직 입사까지는 시간이 남아서 뭘 해도 애매한 시기의 주니어들을 위한 두 가지 충고



문학과 고령의 만남



2022.02 Roeniss Moon

진리 를 찾고 있습니다

그런데 모두를 위한 정답을 찾기가
어려워서 타겟을 좁혀봤어요



그러므로 이 내용은...

- 매우 주관적임!
- 본 얘기는 빠르게 마무리하고 다같이 이야기해보면 좋을 것 같음

1.

멀쩡한 문서와
친절한 인수인계가
기다릴거라는 생각을
하지마세요

Teams

⚡ [Core Experience Team](#)

👤 [Workspaces Team](#)

🔧 [Infrastructure Team](#)

📡 [API team](#)

🎨 [Design Team](#)

🎲 [Data Team](#)

🦋 [Product Admin](#)

🔧 [Growth Team](#)

🏹 [Notion for Projects](#)

Main

🚜 [Projects](#)

🔍 [QA](#)

🏗️ [Releases](#)

🍄 [Growth](#)

🦋 [Product Tech Talks](#)

Development

🚜 [Dev Getting Started](#)

💻 [Development Setup](#)

🌐 [System Overview](#)

✅ [Running Tests](#)

🎨 [Design System & Deve...](#)

💡 [Prettier](#)

TS [Typescript](#)

🔗 [Embeds](#)

🔥 [Image and File Proxy](#)

🏋️ [Notion Stress Testing ...](#)

🐳 [Docker for Local Devel...](#)

🎮 [Console Helpers](#)

👤 [Development Gripes](#)

⚙️ [Jake's Dev Setup](#)

🐾 [Ad Tracking Iframe](#)

🕒 [Template Picker](#)

Infrastructure

!! [Infra Monitor](#)

🌐 [Infrastructure Architect...](#)

🏠 [Monitoring Links](#)

💻 [Useful Terminal Comm...](#)

💡 [Useful Postgres Com...](#)

🐶 [Datadog Dashboards](#)

🖨️ [Server Console & Deb...](#)

📦 [Redis](#)

📡 [Networking](#)

🔨 [Windows testing work...](#)

🔧 [Terraform](#)

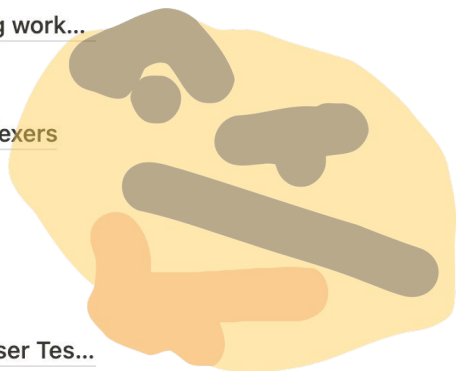
🤖 [Terminal Multiplexers](#)

🦑 [Squid Proxy](#)

🌐 [Elasticsearch](#)

🎯 [CircleCI](#)

100 [QASource Browser Tes...](#)



문서화는 언제나 딜레마

- 문서화 = 유료 = 시간 + 노력 (사바사)
- 문서화 안 할 것들
 - 자주 바뀌는 정보
 - 거의 안 바뀌는 정보
 - 자주 안 보는 정보
- **exception:** 문서화 중독자 join



*팀에 문서화 오타쿠가 들어와서 모든 시니어가 감격하는 장면이다

“그럼 대체 뭘 문서화하지?”



구전문학 의문의 1승

결국 “잘 바뀌지 않으면서 & 자주 확인해야만 하는 것들”이 문서화가 됩니다. 그런데 그런게 얼마나 있을까요?

Q. “하지만 전 신규입사자인데...”

A. 물론 신규입사자에게는 당연히 모든게 낯설게 느껴질 수밖에 없습니다.
그런데 한 해에 신규 입사자가 그 팀에 몇명이나 충원될까?
문서화 vs 구전문학 대결에서 후자가 승리할 수밖에 없는 이유입니다.

서론이 길었네요. 남의 코드를 읽으세요

- (1) 내가 (2) 방금 짰 코드는 -> 이해 100% 
- 그런데 (1) 남이 짰 코드나 (2) 반년 전에 짰 나의 코드는 -> 이게 뭐여?? 

“오래전 작성한 자신의 코드를 보거나,
다른 개발자의 코드를 가까이서 지켜보거나
(ex. 적극적으로 서로에게 관여하는 사이드 프로젝트, 페어코딩),
오픈소스를 읽으려고 노력해보세요”

Junior developer

문서랑 코드랑 하나도
안맞고...

패키지 구조도 지멋대로고...

백로그 이슈는 300개인데...

왜 이 시스템은 아직도
이렇게 멀쩡한거지?

대체 내 코드를 어느 위치에
넣어야 되는거야??



Senior developer

센트리에서 또 에러가
잡혔네.

키바나 보니까 **Trace-ID**
문제인데 데브옵스 장애방
보니까 공용 카프카 쪽에서
컨텍스트가 끊기는 이슈가
있다고 공지가 나왔구나.

내일 쯤이면 데브옵스
팀에서 해결할테니 그때
로그 한 번 보고 이슈
완료처리하면 되겠다.



2.

질문은

성격의

문제가 아닙니다

생존입니다

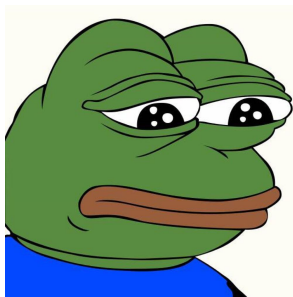
다음 중 가장 좋은 사수는 누구일까요?

- 내 수준에 맞는 태스크를 하나씩 건네주면서 성장을 유도하는 사수
- 하루에 한 번씩 따로 스크럼 시간을 잡아서 업무 진행상황을 공유하고 Q&A 시간을 가지는 사수
- 신규입사자를 위해 필요한 문서와 리소스를 미리 준비해놓고 하나씩 알려주는 사수



당신이 만날 사수는...

- 물어보면 친절히 알려주지만
- 물어보기 전에는 아무 도움도 제공해주지 않는
- 그런... 자판기같은 사수...



왜 그렇게 될 수밖에 없는가

- 개발 업계는 항상 이직모드 -> '언제 나갈 줄 알고?'
- 뭘 모르는지 어떻게 알고? -> '필요하면 지가 오겠지'
- 진짜 바쁨 -> 원래 하던 일은 그대로인데 사수업무가 추가됨

답은 간단하다. 질문하자!

- “저는 i라서 질문을 잘 못해요”
- “저는 A형이라서 나중에 타이밍 봐서 한번에 질문할게요”
- “저는 염소자리라서 인터넷에 검색해보는게 마음이 편해요”

=> 이런 말은 난죽택과 별반 다를 바가 없다



*하스스톤에서 기권하는 장면이다

질문을 방해하는 마음 속 방어기제들

- “질문 받는 것을 안좋아하면 어떡하지?”
 - > 어떻게 알아요? 일단 질문 고고
- “내 질문이 상대방의 몰입을 방해하는 것은 아닐까?”
 - > 낯 유얼 비지니스
- “이거 너무 기초적인 내용 같은데”, “지난번에 물어본건데 또 물어보기는 좀”
 - > 지금이 마지막 찬스 아닐까?



그럼에도 불구하고 질문은 쉽지 않다

“아... 아무때나 무엇이든 질문해도 정말 괜찮은 거구나”

라는 생각이 들 때까지 모두들 힘냅시다!

끝 (여러분의 생각이 무척 궁금합니다)

tl;dr: No good docs/codes, so **read** others' code and do **question**