◈ 의견

AI와 학생

저는 먼저 수십 년 동안 기초를 익혔기 때문에 지금은 코딩 대부분에 AI를 씁니다. 학생과 초보에게는 다른 접근이 필요합니다. 생각까지 넘기지 말고 AI로 작업을 보조하세요.

Fabio Akita · ailair.akitaonrails.com/ko/students/

> ***경력이 몇 년 되는 시니어인 저는 100% 바이브 코딩을 합니다. 학생이나 주니어인 여러분은 최대한 많이 손으로 써야 하고, 더 단순한 일은 AI에게 병렬로 맡기고, 모든 걸 직접 리뷰해야 합니다. 나중에 AI에게 무엇을 시킬지 아는 사람이 되려면 수만 시간의 의식적인 연습과 공부가 필요합니다. 그 전에는 안 됩니다.***
>
> *Eu, sênior, anos de experiência, vou fazer 100% vibe codar. você, estudante, júnior, DEVE escrever o máximo possível na mão, deixar a IA fazer tarefas mais mundanas, em paralelo, e você DEVE revisar tudo manualmente. Você precisa de dezenas de milhares de horas de prática deliberada, estudo, pra se tornar alguém que, depois, vai saber o que mandar a IA fazer por você, não antes.*
>
> ***X에 쓴 글, 2026년 9월 24일***

◈ 1. 승수

AI는 여러분이 가진 것을 곱합니다

같은 도구가 시니어에게는 좋은 결과물을 더 많이,
더 빨리 주고, 초보에게는 초보가 가진 것을
실수까지 포함해 더 많이 줍니다.



1. 승수

AI는 여러분이 가진 것을 곱합니다

- 기초가 약하면 AI가 실수를 키울 수 있습니다. 기초가 탄탄하면 유용한 작업을 늘릴 수 있습니다.
- Google의 DORA 2025 보고서도 팀 단위에서 같은 결과를 냈습니다. "AI는 팀을 고쳐 주지 않습니다. 이미 있는 것을 증폭합니다."
- 경험이 있는데도 아직 버티고 있다면 그만두세요. Dockerfile이나 Makefile을 백 번째로 손으로 쓰는 데에는 아무 가치도 없습니다. 잡일은 AI에게 맡기고, 기초는 직접 쌓으세요.
- 운영체제, 데이터베이스, 네트워크, 자료구조, 컴파일러를 이해하는 사람은 에이전트를 외골격으로 씁니다. 모르는 사람은 목발로 씁니다. 외골격은 힘을 키워 주고, 목발은 약점을 가려 줄 뿐입니다.
- 그리고 누군가는 결과물을 확인해야 합니다. Veracode가 2025년에 100개가 넘는 모델을 테스트했을 때 코드 샘플의 45%가 보안 테스트를 통과하지 못했고 OWASP Top 10 취약점을 만들어 냈습니다.

◈ 1. 승수

> ***잘 준비된 프로그래머는 대체되지 않습니다. 분명히 말했습니다. "저는" 절대 대체되지 않을 겁니다.***
>
> *Programador bem preparado não vai ser substituído. Disse claramente: "EU" nunca vou ser.*
>
> ***X에 쓴 글, 2026년 9월 28일***

◈ 2. 지난번 거품

이 지름길은 전에도 봤습니다

AI 전에는 부트캠프가 약속이었습니다. 두 달이면 Google 수준 연봉을 받는다고요. 결말은 폐업과 해고였습니다.



◈ 2. 지난번 거품

이 지름길은 전에도 봤습니다

- 부트캠프 수료생은 2013년부터 2018년 사이에 약 8배로 늘었습니다. Lambda School은 취업률이 "최대 86%"라고 광고했지만, 미국 소비자금융보호국(CFPB)은 내부 수치가 50%에 가깝고 때로는 30%까지 낮았다고 밝혔습니다.
- 제가 2018년에 Akitando를 시작한 이유가 바로 이겁니다. 진짜 프로그래머가 거품이 꺼져도 살아남도록 준비시키려고요. 2022년에는 강의를 해지하고 인플루언서를 믿지 말라고 말하고 있었습니다.
- 이제 같은 약속이 새 얼굴로 돌아왔습니다. 코딩을 배울 필요가 없고 AI가 대신해 준다는 약속입니다. 한 달짜리 부트캠프로 엔지니어가 되는 사람은 없고, 프롬프트만 쳐서 엔지니어가 되는 사람도 없습니다.
- 그러다 싼 돈이 바닥났습니다. 2023년과 2024년에 부트캠프들이 문을 닫았고, 2U는 2024년 말 이 사업에서 철수한다고 발표했으며, 테크 업계는 그해에만 15만 개가 넘는 일자리를 줄였습니다.
- 그 강의들이 가르친 것, 즉 폼, CRUD, 리포트, 똑같은 UI의 복사본은 바로 LLM이 가장 잘 복제하는 것입니다. 그런 일자리 대부분은 2022년 이후 이미 사라졌습니다.

◈ 3. 근거

자동 조종으로 한 공부는 남지 않습니다

지금까지의 연구는 같은 방향을 가리킵니다. AI가 생각을 대신하면
배우지 못합니다.

17점

AI로 새 라이브러리를 배운 엔지니어(대부분 주니어)의 퀴즈 점수 하락

17%

일반 GPT-4로 연습한 학생들이 GPT-4가 없어진 뒤 받은 시험 점수 하락

19%

더 느림: AI를 쓴 숙련 개발자들, 그런데도 본인은 더 빨랐다고 믿음

◈ 3. 근거

자동 조종으로 한 공부는 남지 않습니다

- Anthropic 자체 연구(2026년 1월): AI로 새 라이브러리(Trio)를 배운 엔지니어들은 대부분 주니어였고, 퀴즈에서 50%를 받았습니다. 손으로 코딩한 쪽은 67%였고, 시간은 유의미하게 줄지 않았습니다. AI에게 개념 설명을 맡긴 사람들은 배운 것을 대부분 지켰고, 코드를 통째로 맡긴 사람들은 가장 낮은 점수대에 들었습니다.
- METR의 2025년 실험: 숙련된 오픈 소스 개발자들이 AI를 쓰니 19% 더 오래 걸렸고, 끝난 뒤에도 AI 덕분에 20% 빨라졌다고 믿었습니다.
- 고등학생 약 1,000명을 대상으로 한 PNAS 연구에서 일반 GPT-4는 연습 점수를 48% 올렸지만, 없애고 나자 시험 점수를 17% 떨어뜨렸습니다. 학생이 생각하도록 설계한 튜터는 이 손실을 대부분 피했습니다.
- 아직 동료 심사를 거치지 않은 MIT 프리프린트에서는 ChatGPT로 에세이를 쓴 사람들이 자기 글을 인용하는 데 어려움을 겪었습니다.

4. 시장

좁아진 신입 취업문

회사는 여전히 리뷰하고 결정할 줄 아는 사람을 뽑습니다. 타이핑할 초보는 훨씬 적게 뽑습니다.

65%

2019년 대비 빅테크 기업의 신입 졸업생 채용 감소

4.5%

소프트웨어 개발 채용 공고 중 신입 비중 (2026년 1분기)

19%

AI 노출이 가장 큰 직무에서 예상보다 적은 22~25세 인원

좁아진 신입 취업문

- SignalFire(2026): 2019년과 비교해 신입 졸업생 채용이 빅테크 기업에서는 약 65%, 초기 스타트업에서는 76% 줄었습니다. 같은 기간 빅테크 기업의 엔지니어 채용은 11%만 줄었습니다.
- Stanford의 "Canaries" 연구: 소프트웨어 개발자를 포함해 AI 노출이 가장 큰 직무에서 22~25세 고용은, 노출이 적은 직무와 같은 속도로 늘었을 경우보다 19% 낮습니다. 주로 채용이 줄어든 결과입니다. 저자들은 이 결과가 인과 관계를 보여주기보다 현상을 기술한 것이라고 신중하게 밝힙니다.
- Indeed: 2026년 초 소프트웨어 채용 공고 중 신입 직무는 4.5%였습니다. 2025년 2월 Claude Code 출시 이후 공고는 15% 가까이 늘었고, 그 증가분의 71%가 시니어 직무였습니다.
- 최근 컴퓨터 과학 졸업생의 실업률은 전공 중 가장 높은 편(7.0%)이지만, 하향 취업률은 가장 낮은 편(19%, 전체 평균 39%)입니다. 채용되면 대개 학위가 필요한 일자리로 간다는 뜻입니다. 일반적인 진입 경로는 여전히 학사 학위입니다.

◆ 5. 빈틈

AI는 여러분에게 필요한 걸 짐작하지 못합니다

AI는 시킨 일을 하고, 시키지 않은 일은 하지 않습니다. 무엇을 시킬지 아는 것이 일입니다.



◈ 5. 빈틈

AI는 여러분에게 필요한 걸 짐작하지 못합니다

- 코딩 에이전트에게는 목표, 맥락, 제약, 우선순위, 깨지면 안 되는 것, 받아들일 수 있는 경로, 성공을 측정하는 방법을 설명해 줘야 합니다. 이걸 말로 표현하는 법을 배운 적 없는 프로그래머가 많습니다.
- 아주 작은 앱이라도 바이브 코딩으로 만들어 프로덕션에 올라가는 것은 경험 있는 프로그래머가 리뷰해야 합니다. 선택 사항이 아닙니다.
- 제가 에이전트와 대화하는 방식은 네 부분입니다. 무엇을 원하는지, 어떻게 할지, 무엇을 원하지 않는지, 어떻게 확인할지. 도메인을 모르면 이걸 채울 수 없습니다.

◈ 6. 언어 신화

언어보다 중요한 테스트

2026년 10월, 모델 4개에 같은 어려운 과제를 7개 언어로 줬습니다. 앞서는 언어는 없었습니다.

112

시도: 언어 7개, 모델 4개, 조건 2개, 각각 2회 실행

24

약한 모델 두 개에 테스트를 줬을 때의 정답 결과

16

같은 모델에 테스트 없이 돌렸을 때의 정답 결과

◈ 6. 언어 신화

언어보다 중요한 테스트

- 프런티어 모델 두 개는 모든 언어에서 대상이 되는 시도를 전부 맞혔습니다.
- 테스트 작성법을 배우고, 모든 버그를 테스트로 만드세요. 이 기술은 어떤 언어, 어떤 모델에서도 통합니다.
- 실패는 모두 문제 자체의 규칙을 어긴 것이었습니다. 타입 오류는 하나도 없었으니, 어떤 컴파일러도 잡아내지 못했을 겁니다.

◆ 7. 질문

AI가 프로그래밍을 다 한다면 왜 배워야 할까

AI가 전부 할 수는 없기 때문입니다. AI는 본 것을 재현하고, 보지 못한 곳에 다음 일자리가 있습니다.



◈ 7. 질문

AI가 프로그래밍을 다 한다면 왜 배워야 할까

- 모델은 공개 코드로 학습했습니다. 세상 코드의 대부분은 비공개이고, GitHub 기여만 봐도 81.5%가 비공개입니다. 아무도 풀어 본 적 없는 문제는 어떤 학습 데이터에도 없습니다.
- 레거시도 마찬가지입니다. 큰 오픈 소스 COBOL 커뮤니티는 없습니다. 은행, 보험사, 정부를 돌리는 코드는 비공개이고, 모델은 그 코드를 거의 본 적이 없습니다.
- 제 생각에는 여기에 기회가 있습니다. 기초를 어렵게 배우려는 사람이 적을수록 그렇게 배운 사람의 가치는 커집니다. 모델이 혼자 메우지 못하는 빈틈을 메우는 세대가 바로 그들일 겁니다.
- 비슷한 것이 공개된 적 있으면 AI는 빠르게 재현합니다. 그런 적이 없으면 길고 비싼 루프를 돌고, 누군가 이어받아야 합니다. "맞다"가 무슨 뜻인지 정하고, 오라클을 만들고, 객관적인 목표가 있는 테스트를 쓰고, 아무도 공개하지 않은 참고 자료를 찾고, 에이전트가 제자리를 맴돌기 시작하면 알아채야 합니다.
- 이 일들은 모두 문제를 밑바닥부터 이해하는 사람이 필요합니다. 컴퓨터, 운영체제, 컴파일러, 네트워크, 데이터베이스가 실제로 어떻게 동작하는지 아는 사람입니다. 대부분이 건너뛰는 어려운 지식입니다.

◈ 7. 질문

> ***기계가 여러분에게 가장 필요로 하는 능력은 기계가 학습으로 가장 얻기 어려운 능력입니다.***
>
> *A habilidade que a máquina mais precisa de você é a que ela menos consegue aprender com o treino dela.*

◈ 8. 실제 사례

"AI가 했다"에 실제로 들어간 것

코딩 에이전트용 오픈 소스 메모리 시스템인 ai-memory는 대부분 AI가 썼습니다. 이걸 올바르게 유지하는 모든 것은 매주 이어진 제 리뷰와 지시에서 나왔습니다.



◈ 8. 실제 사례

"AI가 했다"에 실제로 들어간 것

4,743
AGENTS.md에 적은 에이전트용 규칙의 단어 수

16
절대 깨지면 안 되는 불변 조건

22
강제되는 보안 경계, 각각 가드와 적대적 테스트에 연결됨

3,000+
단위 테스트

600+
통합 테스트

- 명시적 지시 약 65개, 구체적인 실수 때문에 생긴 메모리 노트 28개, 조사와 사후 분석과 아키텍처 결정을 담은 문서 60개 이상이 있습니다.
- 각 층은 AI 혼자서는 제대로 하지 못해서 제가 고치는 법을 알아야 했던 자리를 표시합니다.
- 첫날부터 있던 건 하나도 없습니다. AI가 매일 일의 대부분을 하는 걸 지켜보며 몇 주 동안 다듬은 결과이고, 다듬는 일은 경험에 달려 있습니다.

> ***이 중 어느 것도 AI가 자동으로 만든 게 아닙니다. 전부 AI 혼자서는 못 하고 제가 조정하는 법을 알아야 하는 부분입니다. 아마추어는 이걸 못 합니다.***
>
> *Nada disso é automático, feito pela IA. Tudo isso é onde a IA sozinha não consegue fazer e eu preciso saber com ajustar. Nenhum amador consegue fazer isso.*
>
> ***X에 쓴 글, 2026년 9월 25일***

◆ 9. 무엇을 할까

학생이나 주니어라면

시니어가 인턴을 쓰듯 AI를 쓰세요. 잡일에만요.
배움은 여러분 손을 거쳐야 합니다.



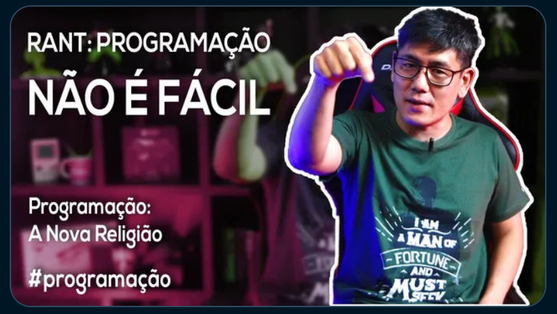
◈ 9. 무엇을 할까

학생이나 주니어라면

- 최대한 많이 손으로 쓰세요. 망가뜨리고, 디버깅하고, 고치세요. 막히는 것도 과정입니다. Anthropic 연구자들은 "고통스럽게 막히는 것까지 포함한" 인지적 노력이 숙달에 중요할 가능성이 높다고 씁니다.
- AI가 만든 건 전부 한 줄씩 리뷰하세요. 설명할 수 없는 변경이라면 아직 받아들일 준비가 안 된 겁니다.
- 시니어란 자기가 숙달한 모든 주제에서 한때 주니어였던 사람입니다.
- 단순한 일은 AI에게 병렬로 맡기고, 개념 설명을 부탁하세요. 여러분이 보고 배워야 할 코드를 써 달라고 하지는 마세요.
- 시간을 세세요. 0에서 첫 주니어 일자리까지는 공부와 연습으로 최소 약 1,400시간이 걸립니다. 제 생각에 숙달에는 수만 시간이 걸립니다. Ericsson의 연구에서 최고 수준의 바이올리니스트들은 20세까지 1만 시간 넘게 연습했습니다. 이후 연구는 연습이 전부는 아니라고 보여 주지만, 연습을 대신하는 것은 없습니다.

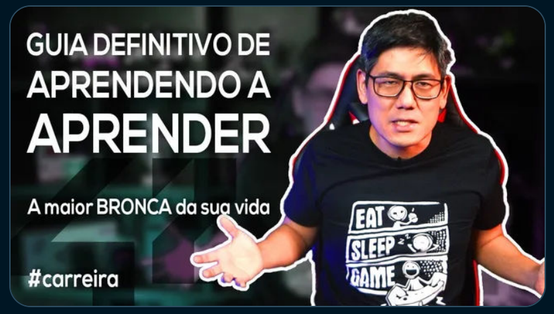
◈ 어디서 시작할까

Akitando 아카이브



RANT: Programação NÃO É Fácil

2020년 2월 · 32:22



Guia DEFINITIVO de Aprendendo a Aprender | A maior BRONCA d...

2020년 4월 · 1:04:37



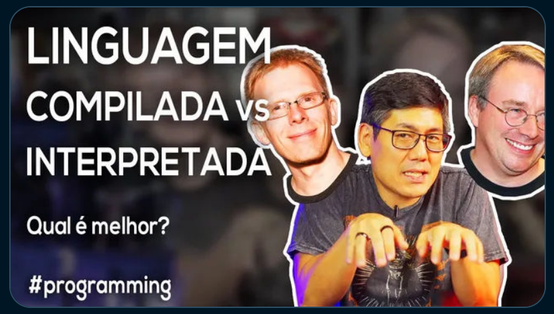
O Guia +Hardcore de Introdução à COMPUTAÇÃO

2020년 6월 · 1:18:28



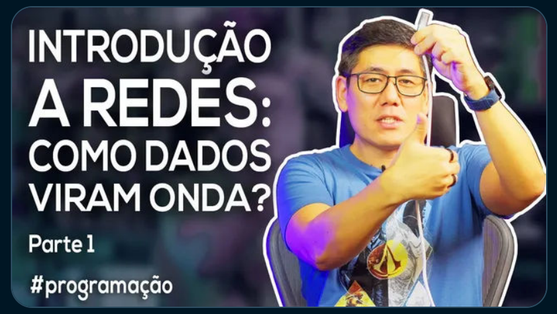
Hello World Como Você Nunca Viu! | Entendendo C

2021년 3월 · 1:09:22



Linguagem Compilada vs Interpretada | Qual é melhor?

2022년 4월 · 1:11:44



Introdução a Redes: Como Dados viram Ondas? | Parte 1

2022년 7월 · 37:59

정리

1

AI는 이미 아는 것을 곱합니다. 곱할 게 적으면 얻는 것도 아주 적습니다.

2

커리어로 팔린 지름길은 이미 한 번 무너졌습니다. 새 지름길도 같은 약속입니다.

3

자동 조종으로 한 공부는 남지 않습니다. 지금까지의 연구가 같은 결론입니다.

4

여전히 가치 있는 일에는 결과물을 판단할 줄 아는 사람이 필요하고, 신입 채용은 줄고 있습니다.

5

AI는 여러분의 목표도, 한계도, 트레이드오프도 짐작하지 못합니다. 그걸 아는 것이 일입니다.

6

제 벤치마크에서는 모델에게 테스트를 주는 것이 결과를 바꿨습니다. 언어는 문제에 맞는 것을 고르세요.

7

AI는 공개된 것을 재현합니다. 비공개 작업과 새로운 작업에서 AI가 남기는 빈틈을 메우는 것이 일이고, 탄탄한 기초가 그 빈틈을 메우는 방법입니다.

8

손으로 쓰고, 잡일은 AI에게 맡기고, 전부 리뷰하고, 시간을 들이세요.

중요한 것은 객관적 현실 하나뿐입니다.

A única coisa que importa é realidade objetiva.

더 읽기: 제 글과 출처

AI는 여러분이 가진 것을 곱합니다

↗ 일주일 만에 0에서 포스트 프로덕션까지: 실제 프로젝트에서 AI 쓰기
↗ VS Code는 새로운 천공 카드
↗ 핑계는 그만, 더 많이 배포하기
↗ Google DORA 2025: AI는 있는 것을 증폭한다
↗ Veracode: AI가 생성한 코드의 보안 결함

이 지름길은 전에도 봤습니다

↗ 푸념: AI가 프로그래머를 죽였을까?
↗ CFPB: Lambda School의 취업률 주장
↗ Course Report: 부트캠프 시장, 2013~2018년
↗ Inside Higher Ed: 부트캠프 폐업
↗ Higher Ed Dive: 2U의 부트캠프 철수
↗ TechCrunch: 2024년 테크 업계 해고
↗ Akitando #132: 스타트업 거품 붕괴 (2022, 포르투갈어)

자동 조종으로 한 공부는 남지 않습니다

↗ 새 LLM 벤치마크 v4, 1부: 과정
↗ VS Code는 새로운 천공 카드
↗ DriveClub과 shadPS4가 AI와 저를 거의 이길 뻔한 이야기: 공부하는 법
↗ Anthropic: AI 도움이 코딩 실력에 미치는 영향
↗ Bastani 외, PNAS: 생성형 AI는 학습을 해칠 수 있다
↗ METR: 숙련 개발자가 19% 느려짐
↗ MIT Media Lab: Your Brain on ChatGPT (프리프린트)

좁아진 신입 취업문

↗ 푸념: AI가 프로그래머를 죽였을까?
↗ SignalFire: State of Tech Talent 2026
↗ Indeed: 경력직 쪽으로 기우는 노동 시장
↗ Indeed: Claude Code 출시 이후의 소프트웨어 채용 공고 (2026년 7월)
↗ Stanford Digital Economy Lab: 젊은 노동자와 AI 격차 (2026년 8월)
↗ 뉴욕 연준: 최근 졸업생의 노동 시장
↗ BLS: 소프트웨어 개발자

더 읽기: 제 글과 출처

AI는 여러분에게 필요한 걸 짐작하지 못합니다

↗ LLM이 여러분 회사에서 실패할 이유
↗ LLM이 기대한 결과를 주지 않는 이유
↗ 바이브 코딩: GLM 4.7로 100% 만든 작은 앱 (TV Clipboard)

언어보다 중요한 테스트

↗ LLM으로 프로그래밍하기에 더 좋은 언어가 있을까?
↗ 벤치마크: 코드, 데이터, 방법

AI가 프로그래밍을 다 한다면 왜 배워야 할까

↗ LLM의 한계: 있는 것은 잘 재현하고, 없는 것에는 비싸다
↗ AI 리버스 엔지니어링: Adobe는 끝났나?
↗ VS Code는 새로운 천공 카드
↗ GitHub Octoverse 2025: 81.5%가 비공개
↗ arXiv (2026): LLM과 COBOL, 거의 공개되지 않는 코드
↗ ARC Prize: 새로운 과제, 순수 LLM은 0%

"AI가 했다"에 실제로 들어간 것

↗ ai-memory 2.0
↗ 소프트웨어는 결코 완성되지 않는다: 프로젝트 네 개와 배포 이후의 삶

학생이나 주니어라면

↗ DriveClub과 shadPS4가 AI와 저를 거의 이길 뻔한 이야기: 공부하는 법
↗ RANT: Akita는 AI에 굴복했나?
↗ AI 마라톤 마무리: 성공일까 실패일까?
↗ Anthropic: "고통스럽게 막히는 것"
↗ Ericsson 외 (1993): 의식적인 연습
↗ Macnamara와 Maitra (2019): 재현 연구
↗ Akitando #72: 프로그래밍은 쉽지 않다 (포르투갈어)

모든 요점과 링크는 웹에서

ailair.akitaonrails.com/ko/students/

Fabio Akita · akitaonrails.com · x.com/AkitaOnRails