◈ 意見

AIと学生

私は何十年もかけて基礎を学んだので、コーディングの大半にAIを使っています。学生と初心者には別の使い方が必要です。考える作業を渡さず、仕事を支えるためにAIを使ってください。

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

> 私はシニアで、何年も経験があるので、100%バイブコーディングします。学生やジュニアのあなたは、できるだけ多くを手で書かなければいけません。ありふれた作業はAIに並行してやらせて、すべてを手作業でレビューしなければいけません。あとでAIに何をやらせるべきかわかる人間になるには、何万時間もの意図的な練習と勉強が必要です。その前ではありません。
>
> *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.*
>
> ***私のX投稿、2026年9月24日***

◆ 1. 掛け算

AIはあなたが持ち込んだものを掛け算する

同じツールが、シニアにはより多くのよい仕事をより速く与え、初心者には初心者が持っているものを、間違いも含めてもっと多く与えます。



◈ 1. 掛け算

AIはあなたが持ち込んだものを掛け算する

- 基礎が弱いと、AIは間違いを増幅します。基礎が強ければ、役に立つ仕事を増やせます。
- GoogleのDORA 2025レポートも、チームについて同じことを見つけています。「AIはチームを直さない。すでにあるものを増幅する」。
- 経験があるのにまだ抵抗しているなら、やめてください。DockerfileやMakefileを100回目も手で書くことに価値はありません。雑用はAIに任せ、基礎は任せません。
- OS、データベース、ネットワーク、データ構造、コンパイラを理解している人は、エージェントを外骨格として使います。理解していない人は松葉杖として使います。外骨格は力を増幅し、松葉杖は弱さを隠すだけです。
- そして誰かが出力をチェックしなければいけません。Veracodeが2025年に100以上のモデルを試したテストでは、コードサンプルの45%がセキュリティテストに落ち、OWASP Top 10の脆弱性を持ち込んでいました。

◈ 1. 掛け算

> **よく準備されたプログラマーは置き換えられません。はっきり言いました。「私」は決して置き換えられません。**
>
> *Programador bem preparado não vai ser substituído. Disse claramente: "EU" nunca vou ser.*
>
> ***私のX投稿、2026年9月28日***

◈ 2. 前回のバブル

この近道は前にも見た

AIの前は、ブートキャンプが約束でした。2か月でGoogle並みの給料がもらえるというものです。結末は閉鎖とレイオフでした。



◈ 2. 前回のバブル

この近道は前にも見た

- ブートキャンプの修了者は2013年から2018年のあいだに約8倍に増えました。Lambda Schoolは就職率を「最大86%」と宣伝していましたが、米消費者金融保護局（CFPB）は、社内の数字が50%に近く、ときには30%まで下がっていたことを突き止めました。
- 私が2018年にAkitandoを始めたのは、まさにこのためです。本物のプログラマーがバブル崩壊を生き延びられるように準備することです。2022年には、講座を解約してインフルエンサーを信じるのをやめるよう言っていました。
- 今、同じ約束が新しい顔で戻ってきています。コードを学ぶ必要はない、AIがやってくれる、というものです。1か月のブートキャンプでエンジニアになる人はいませんし、プロンプトを書くだけでなる人もいません。
- やがて安いお金が尽きました。ブートキャンプは2023年と2024年に閉鎖し、2Uは2024年末にこの事業から撤退すると発表し、テック業界はその年だけで15万人以上の職を削りました。
- そうした講座が教えていたもの、つまりフォーム、CRUD、レポート、同じUIのコピーは、まさにLLMが最も得意とする再現の対象です。そうした仕事の大半は、2022年以降すでに削られています。

◈ 3. 証拠

自動操縦の学習は身につかない

これまでの研究は同じ方向を指しています。AIが考えてしまうと、あなたは学びません。

17ポイント

クイズの点数が低下：新しいライブラリをAIで学んだ、主にジュニアのエンジニア

17%

試験の点数が低下：素のGPT-4で練習した生徒から、それを取り上げたあと

19%

遅くなった：AIを使った経験豊富な開発者。それでも速くなったと信じていた

◈ 3. 証拠

自動操縦の学習は身につかない

- Anthropic自身の研究、2026年1月：新しいライブラリ（Trio）をAIで学んだ主にジュニアのエンジニアはクイズで50%で、手でコードを書いた人たちの67%を下回り、時間の節約も有意ではありませんでした。AIに概念を説明させた人は学びの大半を保ち、コードを任せきりにした人は最も低いグループに入りました。
- METRの2025年の試験：経験豊富なオープンソース開発者はAIを使うと19%長く時間がかかり、終わったあとも20%速くなったと信じていました。
- 高校生約1,000人を対象にしたPNASの研究では、素のGPT-4で練習の点数は48%上がり、取り上げられると試験の点数が17%下がりました。考えさせるように作られたチューターは、この低下をおおむね避けられました。
- まだ査読を受けていないMITのプレプリントでは、ChatGPTでエッセイを書いた人は自分の文章を引用するのに苦労しました。

◈ 4. 市場

未経験者の入り口は狭くなった

企業はレビューして決められる人をまだ雇っています。タイプするだけの初心者は、ずっと少ししか雇いません。

65%

大手テック企業の新卒採用の減少（2019年比）

4.5%

ソフトウェア開発の求人のうち未経験者向けの割合（2026年第1四半期）

19%

予想より少ない22〜25歳の雇用（AIの影響を最も受ける職）

◈ 4. 市場

未経験者の入り口は狭くなった

- SignalFire、2026年：2019年と比べて、新卒採用は大手テック企業で約65%、アーリーステージのスタートアップで76%減りました。一方、大手テック企業のエンジニア採用は11%しか減っていません。
- スタンフォード大学の「Canaries」研究：ソフトウェア開発者を含む、AIの影響を最も受ける職での22〜25歳の雇用は、影響の少ない職と同じペースで伸びていた場合より19%低く、その大半は採用が減ったことによるものです。著者たちは慎重に、これを因果的ではなく記述的なものと呼んでいます。
- Indeed：2026年初め、ソフトウェア開発の求人のうち未経験者向けは4.5%でした。2025年2月にClaude Codeが出てから求人は約15%増え、その増加の71%はシニア向けでした。
- 最近のコンピューターサイエンスの卒業生は、全専攻の中でも失業率が最も高い部類（7.0%）ですが、不完全雇用率は最も低い部類です（19%、全体では39%）。雇われるときは、たいてい学位が必要な仕事に就きます。典型的な入り口は今も学士号です。

◈ 5. すき間

AIはあなたに必要なものを推測できない

AIは頼んだことをやり、頼まなかったことはやりません。何を頼むべきかわかっていることが仕事です。



◈ 5. すき間

AIはあなたに必要なものを推測できない

- AIコーディングエージェントには、目的、コンテキスト、制約、優先順位、壊してはいけないもの、許される道筋、成功をどう測るかを説明する必要があります。多くのプログラマーはそれを言葉にすることを学んでいません。
- 小さなアプリでも、バイブコーディングで作って本番に出るものは、経験のあるプログラマーがレビューしなければいけません。これは任意ではありません。
- 私のエージェントへの話し方は4つの部分からなります。何がほしいか、どうやるか、何がほしくないか、どう確認するか。ドメインを知らなければ、これは埋められません。

◈ 6. 言語の神話

言語よりテストが大事

2026年10月、4つのモデルに同じ難しいタスクを7つの言語で与えました。どの言語も抜きん出ませんでした。

112

回の試行：7言語、4モデル、2条件、各2回

24

テストを渡したときの、弱いほうの2モデルの正解数

16

テストなしでの、同じモデルの正解数

◈ 6. 言語の神話

言語よりテストが大事

- 2つのフロンティアモデルは、対象になったすべての試行を、どの言語でも正解しました。
- テストの書き方を学び、見つけたバグはすべてテストにしてください。このスキルはどの言語でも、どのモデルでも通用します。
- 失敗はどれも、問題そのもののルールを破ったものでした。型エラーは1つもなかったので、どのコンパイラーでも捕まえられなかったはずです。

◆ 7. 問い

AIがプログラミングを全部できるなら、なぜ学ぶのか？

全部はできないからです。AIは見たことのあるものを再現します。見たことのないものこそ、次の仕事がある場所です。



7. 問い

AIがプログラミングを全部できるなら、なぜ学ぶのか？

- モデルは公開コードから学びました。世界のコードの大半は非公開で、GitHubのコントリビューションだけでも81.5%がそうです。そして誰もまだ解いていない問題は、どの学習データにも入っていません。
- 似たものが公開されていれば、AIはすばやく再現します。公開されていなければ、長くて高くつくループの中で空回りし、誰かが引き継がなければいけません。「正しい」とは何かを定義し、オラクルを作り、客観的な目標を持つテストを書き、誰も公開しなかった参考資料を見つけ、エージェントが堂々巡りを始めたら気づく人です。
- レガシーも同じです。COBOLの大きなオープンソースコミュニティはありません。銀行、保険会社、政府を動かすコードは非公開で、モデルはほとんど見たことがありません。
- こうした仕事はどれも、問題を根っこから理解している人を必要とします。コンピューター、OS、コンパイラ、ネットワーク、データベースが実際にどう動くか。ほとんどの人が飛ばす、難しい知識です。

◈ 7. 問い

AIがプログラミングを全部できるなら、なぜ学ぶのか？

- 私の考えでは、そこにチャンスがあります。基礎を苦労して学ぼうとする人が少ないほど、学んだ人の価値は上がります。モデルが自力では埋められないすき間を埋めるのは、その世代になるでしょう。

> ***機械があなたに最も必要としているスキルは、機械が学習から最も学びにくいスキルです。***
>
> *A habilidade que a máquina mais precisa de você é a que ela menos consegue aprender com o treino dela.*

◈ 8. 実例

「AIがやった」に実際に必要だったもの

コーディングエージェント用の私のオープンソースのメモリシステム、ai-memoryは、大部分をAIが書いています。それを正しく保っているものはすべて、毎週続けた私のレビューと指示から来ています。



◈ 8. 実例

「AIがやった」に実際に必要だったもの

4,743

AGENTS.mdに書いたエージェント向けルールの語数

16

絶対に壊してはいけない不変条件

22

強制されたセキュリティ境界。それぞれがガードと敵対的テストに対応

3,000+

ユニットテスト

600+

インテグレーションテスト

- 明示的な指示が約65個、具体的なミスがあったからこそ存在する厳選されたメモリノートが28個、調査、ポストモーテム、アーキテクチャの決定をまとめたドキュメントが60以上あります。
- どの層も、AIだけではうまくいかず、私が直し方を知っていなければならなかった場所を示しています。
- 初日からあったものは1つもありません。AIが毎日作業の大部分をこなすのを何週間も見て、磨いてきた結果です。そして磨くには経験が要ります。

8. 実例

> **これはどれも、AIが自動で作ったものではありません。すべて、AIだけではできず、私が調整の仕方を知っていなければならない部分です。素人にはできません。**
>
> *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.*
>
> — 私のX投稿、2026年9月25日

◈ 9. 何をすべきか

学生やジュニアなら

シニアがインターンを使うようにAIを使ってください。雑用のためです。学びは自分の手を通さなければいけません。



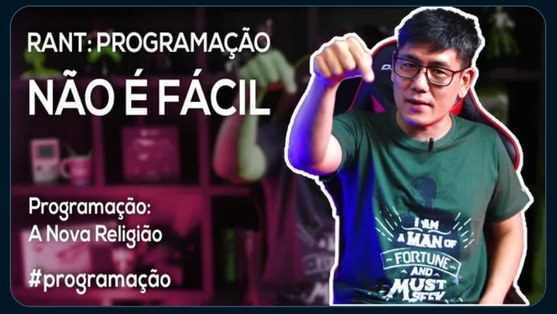
◈ 9. 何をすべきか

学生やジュニアなら

- できるだけ多くを手で書いてください。壊して、デバッグして、直す。行き詰まるのも学びのうちです。Anthropicの研究者は、「つらいほど行き詰まること」も含めた認知的な努力が、習熟にはおそらく重要だと書いています。
- AIが出すものはすべて、1行ずつレビューしてください。変更を説明できないなら、まだ受け入れる準備ができていません。
- シニアとは、自分が習熟したどのテーマでも、かつてジュニアだった人のことです。
- ありふれた作業はAIに並行してやらせ、概念を説明してもらってください。あなたが学ぶべきコードを書かせるのではありません。
- 時間を数えてください。ゼロから最初のジュニアの仕事までには、少なくとも約1,400時間の勉強と練習が要ります。私の考えでは、習熟には何万時間もかかります。エリクソンの研究で最も優れたバイオリニストは20歳までに1万時間以上を積んでいました。その後の研究では練習がすべてではないとわかっていますが、練習に代わるものはありません。

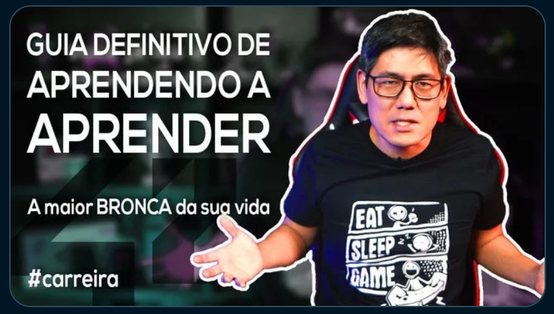
◈ どこから始めるか

Akitandoのアーカイブ



RANT: Programação NÃO É Fácil

2020年2月 · 32:22



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

2020年4月 · 1:04:37



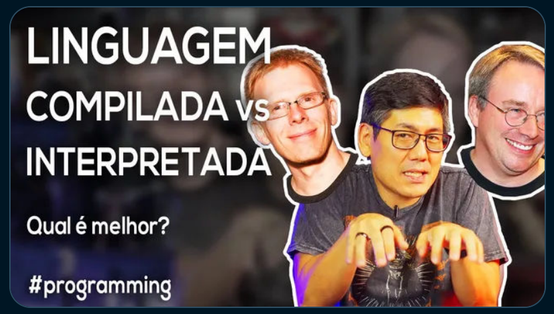
O Guia +Hardcore de Introdução à COMPUTAÇÃO

2020年6月 · 1:18:28



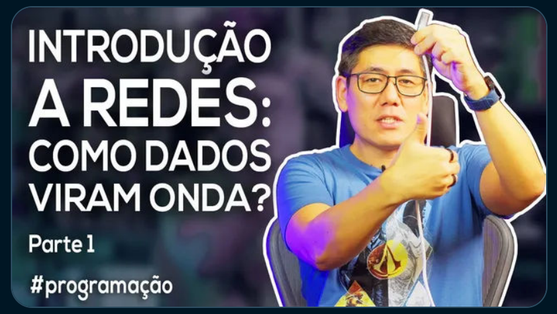
Hello World Como Você Nunca Viu! | Entendendo C

2021年3月 · 1:09:22



Linguagem Compilada vs Interpretada | Qual é melhor?

2022年4月 · 1:11:44



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

2022年7月 · 37:59

まとめ

1

AIはあなたがすでに知っていることを掛け算します。掛けるものが少なければ、得られるものもわずかです。

2

キャリアとして売られた近道は、すでに一度崩壊しました。新しいものも同じ約束です。

3

自動操縦の学習は身につきません。これまでの研究は一致しています。

4

今も価値のある仕事には、出力を判断できる人が必要です。一方、未経験者の採用は縮んでいます。

5

AIはあなたの目的も、限界も、トレードオフも推測できません。それを知っていることが仕事です。

6

私のベンチマークでは、モデルにテストを渡すと結果が変わりました。言語は問題に合ったものを選んでください。

7

AIは公開されているものを再現します。AIが残すすき間、つまり非公開の仕事と新しい仕事こそが仕事で、それを埋める方法が難しい基礎です。

8

手で書き、雑用はAIにやらせ、すべてをレビューし、時間をかけてください。

大事なのは客観的な現実だけです。

A única coisa que importa é realidade objetiva.

もっと読む：私の記事と出典

AIはあなたが持ち込んだものを掛け算する

↗ゼロからポストプロダクションまで1週間：実際のプロジェクトでAIを使う
↗VS Codeは新しいパンチカード
↗言い訳はやめて、もっとリリースしよう
↗Google DORA 2025：AIはすでにあるものを増幅する
↗Veracode：AI生成コードのセキュリティ欠陥

この近道は前にも見た

↗愚痴：AIはプログラマーを殺したのか
↗CFPB：Lambda Schoolの就職率の主張
↗Course Report：ブートキャンプ市場、2013年から2018年
↗Inside Higher Ed：ブートキャンプの閉鎖
↗Higher Ed Dive：2Uがブートキャンプから撤退
↗TechCrunch：2024年のテック業界のレイオフ
↗Akitando #132：スタートアップバブルの崩壊（2022年、ポルトガル語）

自動操縦の学習は身につかない

↗新LLMベンチマークv4、パート1：進め方
↗VS Codeは新しいパンチカード
↗DriveClubとshadPS4がAIと私を打ち負かしかけた話：どう学ぶか
↗Anthropic：AIの支援がコーディングスキルに与える影響
↗バスタニほか、PNAS：生成AIは学習を損なうことがある
↗METR：経験豊富な開発者が19%遅くなった
↗MIT Media Lab：Your Brain on ChatGPT（プレプリント）

未経験者の入り口は狭くなった

↗愚痴：AIはプログラマーを殺したのか
↗SignalFire：State of Tech Talent 2026
↗Indeed：労働市場はシニア寄りに傾いている
↗Indeed：Claude Code以降のソフトウェア求人（2026年7月）
↗Stanford Digital Economy Lab：若い労働者とAIの格差（2026年8月）
↗ニューヨーク連銀：最近の卒業生の労働市場
↗BLS：ソフトウェア開発者

もっと読む：私の記事と出典

AIはあなたに必要なものを推測できない

↗ LLMがあなたの会社で失敗する理由
↗ LLMが期待どおりの結果を出さない理由
↗ バイブコーディング：GLM 4.7だけで小さなアプリを作った（TV Clipboard）

言語よりテストが大事

↗ LLMでのプログラミングに向いた言語はあるのか？
↗ ベンチマーク：コード、データ、手法

AIがプログラミングを全部できるなら、なぜ学ぶのか？

↗ LLMの限界：あるものの再現は得意、ないものには高くつく
↗ AIリバースエンジニアリング：Adobeは終わったのか？
↗ VS Codeは新しいパンチカード
↗ GitHub Octoverse 2025：81.5%が非公開
↗ arXiv（2026年）：LLMとCOBOL、めったに公開されないコード
↗ ARC Prize：新しい課題、純粋なLLMは0%

「AIがやった」に実際に必要だったもの

↗ ai-memory 2.0
↗ ソフトウェアに完成はない：4つのプロジェクトとデプロイ後の日々

学生やジュニアなら

↗ DriveClubとshadPS4がAIと私を打ち負かしかけた話：どう学ぶか
↗ 愚痴：AkitaはAIに屈したのか？
↗ AIマラソンの総括：成功か失敗か
↗ Anthropic：「つらいほど行き詰まること」
↗ エリクソンほか（1993年）：意図的な練習
↗ マクナマラとマイトラ（2019年）：追試
↗ Akitando #72：プログラミングは簡単じゃない（ポルトガル語）

すべてのポイントとリンクはウェブで

ailair.akitaonrails.com/ja/students/

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