דעה

AI וסטודנטים

אני משתמש ב-AI לרוב עבודת הקוד שלי כי קודם השקעתי עשרות שנים בלימוד היסודות. סטודנטים ומתחילים צריכים גישה אחרת: להשתמש ב-AI כדי לתמוך בעבודה בלי למסור לו את החשיבה.

/Fabio Akita • ailair.akitaonrails.com/he/students

> ***אני, סניור עם שנים של ניסיון, אעשה vibe coding ב-100%. אתם, סטודנטים, ג'וניורים, חייבים לכתוב ביד כמה שיותר, לתת ל-AI לעשות את המשימות השגרתיות יותר במקביל, ואתם חייבים לבדוק הכול ידנית. אתם צריכים עשרות אלפי שעות של תרגול מכוון ולימוד כדי להפוך למישהו שאחר כך יידע מה לבקש מה-AI לעשות בשבילו. לא לפני כן.***
>
> *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.*
>
> ***אני ב-X, 24 בספטמבר 2026***



1. המכפיל

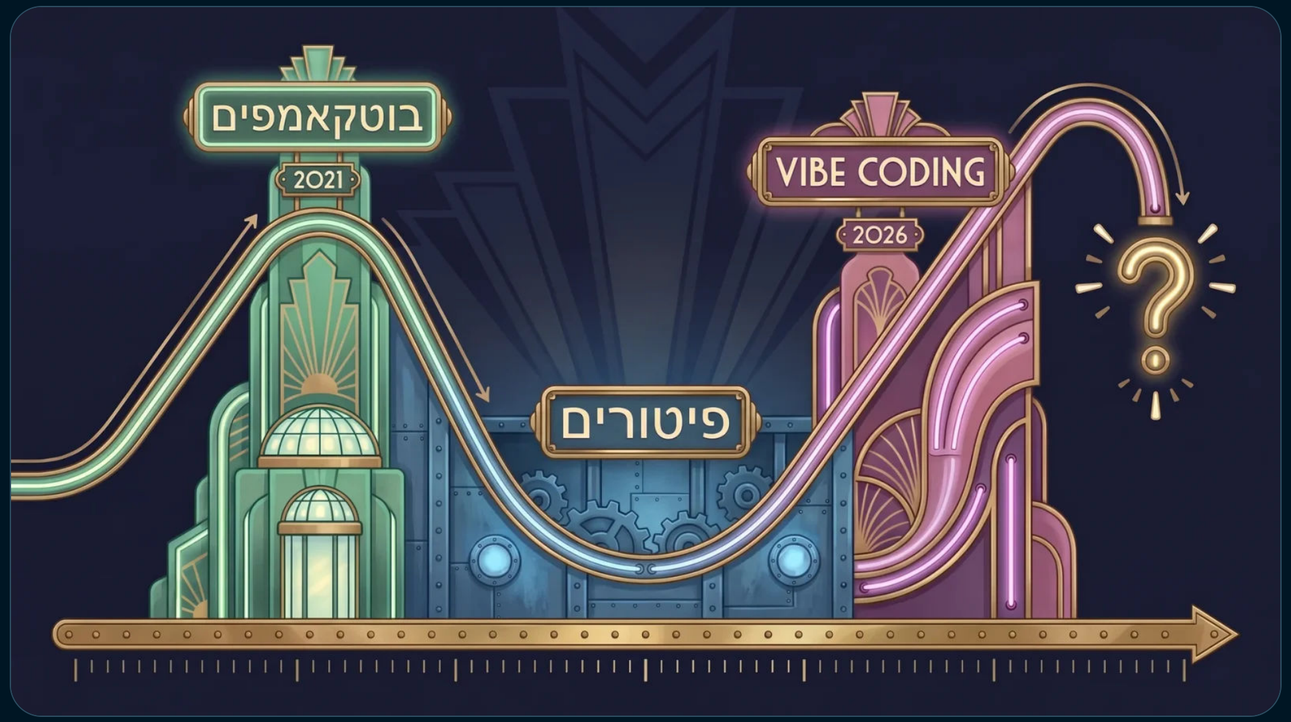
AI מכפיל את מה שאתם מביאים

אותו כלי נותן לסניור יותר עבודה טובה, מהר יותר, ונותן למתחיל יותר ממה שיש למתחיל, כולל הטעויות.

AI מכפיל את מה שאתם מביאים

- AI יכול להכפיל טעויות כשהיסודות חלשים. עם יסודות חזקים, הוא יכול להכפיל עבודה מועילה.
- דוח DORA 2025 של Google מצא את אותו הדבר בצוותים: "AI לא מתקן צוות; הוא מגביר את מה שכבר קיים."
- אם אתם מנוסים ועדיין מתנגדים, תפסיקו. אין שום הישג בלכתוב Dockerfile או Makefile ביד בפעם המאה. העבודות השחורות הולכות ל-AI; היסודות לא.
- מי שמבין מערכות הפעלה, מסדי נתונים, רשתות, מבני נתונים וקומפיילרים משתמש בסוכנים כשלד חיצוני. מי שלא, משתמש בהם כקב. שלד חיצוני מגביר כוח; קב רק מסתיר חולשה.
- ומישהו צריך לבדוק את התוצר. בבדיקה של Veracode מ-2025 על יותר מ-100 מודלים, 45% מדוגמאות הקוד נכשלו בבדיקות אבטחה והכניסו פגיעויות מה-OWASP Top 10.

> ***מתכנת שמוכן היטב לא יוחלף. אמרתי את זה בבירור: אני לעולם לא אוחלף.***
>
> *.Programador bem preparado não vai ser substituído. Disse claramente: "EU" nunca vou ser*
>
> ***אני ב-X, 28 בספטמבר 2026***



2. הבועה הקודמת

כבר ראינו את הקיצור הזה

לפני ה-AI, ההבטחה הייתה בוטקאמפ: חודשיים ואתם מרוויחים משכורת של Google. זה נגמר בסגירות ובפיטורים.

כבר ראינו את הקיצור הזה

- מספר בוגרי הבוטקאמפים גדל בערך פי שמונה בין 2013 ל-2018. Lambda School פרסמה שיעורי השמה "של עד 86 אחוז"; הלשכה להגנה פיננסית על הצרכן (CFPB) מצאה שהמספרים הפנימיים שלה היו קרובים יותר ל-50%, ולפעמים נמוכים עד 30%.
- פתחתי את Akitando ב-2018 בדיוק בשביל זה: להכין מתכנתים אמיתיים לשרוד את פיצוץ הבועה. ב-2022 אמרתי לאנשים לבטל את הקורסים ולהפסיק להאמין למשפיענים.
- עכשיו אותה הבטחה חוזרת בפנים חדשות: לא צריך ללמוד לתכנת, ה-AI עושה את זה בשבילכם. אף אחד לא הופך למהנדס בבוטקאמפ של חודש, ואף אחד לא הופך למהנדס מכתיבת פרומפטים.
- ואז הכסף הזול נגמר. בוטקאמפים נסגרו ב-2023 וב-2024, 2U הודיעה בסוף 2024 שהיא יוצאת מהתחום, ותעשיית הטק פיטרה יותר מ-150,000 עובדים בשנה ההיא לבדה.
- מה שהקורסים האלה לימדו, טפסים, CRUD, דוחות והעתקים של אותו ממשק, זה בדיוק מה ש-LLMs משכפלים הכי טוב. רוב המשרות האלה כבר נחתכו אחרי 2022.

3. הראיות

למידה על טייס אוטומטי לא נקלטת

המחקרים עד עכשיו מצביעים לאותו כיוון: כשה-AI חושב במקומכם, אתם לא לומדים.

17 נק׳

פחות נקודות בבוחן למהנדסים, רובם ג׳וניורים, שלמדו ספרייה חדשה עם AI

17%

ציונים נמוכים יותר במבחן לתלמידים שתרגלו עם GPT-4 רגיל, אחרי שנלקח מהם

19%

איטיים יותר: מפתחים מנוסים עם AI, שעדיין האמינו שהם מהירים יותר

למידה על טייס אוטומטי לא נקלטת

- המחקר של Anthropic עצמה, ינואר 2026: מהנדסים, רובם ג'וניורים, שלמדו ספרייה חדשה (Trio) עם AI קיבלו 50% בבוחן, לעומת 67% אצל מי שכתב קוד ביד, בלי חיסכון משמעותי בזמן. מי שהשתמש ב-AI כדי להסביר מושגים שמר על רוב הלמידה; מי שהעביר לו את הקוד קיבל מהציונים הנמוכים ביותר.
- הניסוי של METR מ-2025: מפתחי קוד פתוח מנוסים עבדו 19% יותר זמן עם AI, ואחר כך עדיין האמינו שהוא הפך אותם למהירים ב-20%.
- במחקר ב-PNAS עם כ-1,000 תלמידי תיכון, GPT-4 רגיל העלה את ציוני התרגול ב-48% ואז הוריד את ציוני המבחן ב-17% אחרי שנלקח מהם. מורה פרטי שנבנה כדי לגרום להם לחשוב מנע את רוב ההפסד.
- פרה-פרינט של MIT, שעדיין לא עבר ביקורת עמיתים, מצא שאנשים שכתבו חיבורים עם ChatGPT התקשו לצטט את העבודה של עצמם.

מסלול הכניסה נעשה צר יותר

חברות עדיין מגייסות אנשים שיודעים לבדוק ולהחליט. הן מגייסות הרבה פחות מתחילים כדי להקליד.

65%

פחות גיוס של בוגרים טריים בחברות הטק הגדולות מאשר ב-2019

4.5%

ממודעות הדרושים בפיתוח תוכנה הן למשרות כניסה (רבעון 1 של 2026)

19%

פחות בני 22 עד 25 מהצפוי במשרות החשופות ביותר ל-AI

מסלול הכניסה נעשה צר יותר

- SignalFire, 2026: גיוס בוגרים טריים ירד בכ-65% בחברות הטק הגדולות וב-76% בסטארטאפים בשלב מוקדם בהשוואה ל-2019, בזמן שגיוס מהנדסים בחברות הטק הגדולות ירד רק ב-11%.
- מחקר ה-“Canaries” של Stanford: התעסוקה של בני 22 עד 25 במשרות החשופות ביותר ל-AI, כולל מפתחי תוכנה, נמוכה ב-19% ממה שהייתה אילו שמרה על הקצב של משרות פחות חשופות, בעיקר בגלל פחות גיוסים. החוקרים מקפידים לקרוא לזה תיאורי ולא סיבתי.
- Indeed: משרות כניסה היו 4.5% ממודעות הדרושים בפיתוח תוכנה בתחילת 2026. מודעות הדרושים צמחו בכמעט 15% אחרי ש-Claude Code יצא בפברואר 2025, ו-71% מהצמיחה הזו היו משרות סניור.
- לבוגרים טריים במדעי המחשב יש אחד משיעורי האבטלה הגבוהים מכל תחום לימוד (7.0%) אבל אחד משיעורי התת-תעסוקה הנמוכים (19%, לעומת 39% בסך הכול): כשמגייסים אותם, זה בדרך כלל למשרה שדורשת את התואר. מסלול הכניסה הרגיל הוא עדיין תואר ראשון.



5. הפער

AI לא יכול לנחש מה אתם צריכים

הוא עושה מה שביקשתם, והוא לא עושה מה שלא ביקשתם. לדעת מה לבקש, זו העבודה.

◆ 5. הפער

AI לא יכול לנחש מה אתם צריכים

- סוכן קוד צריך שתסבירו לו את המטרה, ההקשר, האילוצים, סדרי העדיפויות, מה אסור שיישבר, אילו דרכים מקובלות ואיך מודדים הצלחה. הרבה מתכנתים אף פעם לא למדו להגיד את זה.
- אפילו באפליקציה קטנטנה, כל דבר שנבנה ב-vibe coding ועולה לפרודקשן חייב לעבור בדיקה של מתכנת מנוסה. זה לא אופציונלי.
- הדרך שלי לדבר עם סוכן בנויה מארבעה חלקים: מה אני רוצה, איך, מה אני לא רוצה, ואיך אנחנו בודקים. אי אפשר למלא אותם בלי להכיר את התחום.

6. מיתוס השפה

בדיקות חשובות יותר מהשפה

באוקטובר 2026 נתתי ל-4 מודלים את אותה משימה קשה ב-7 שפות. אף שפה לא יצאה בראש.

112

ניסיונות: 7 שפות, 4 מודלים, 2 תנאים, 2 הרצות לכל אחד

24

תוצאות נכונות משני המודלים החלשים כשנתתי להם את הבדיקות

16

תוצאות נכונות מאותם מודלים בלי הבדיקות

בדיקות חשובות יותר מהשפה

- שני מודלי החזית פתרו נכון כל ניסיון שנספר, בכל שפה.
- למדו לכתוב בדיקות, והפכו כל באג לבדיקה. המיומנות הזו עובדת בכל שפה, עם כל מודל.
- כל כישלון שבר כלל של הבעיה עצמה. אף אחד מהם לא היה שגיאת טיפוסים, כך שאף קומפיילר לא היה תופס אותם.



7. השאלה

אם AI יכול לעשות את כל התכנות, למה ללמוד?

כי הוא לא יכול לעשות את כולו. הוא משחזר את מה שראה, ומה שהוא לא ראה זה בדיוק המקום שבו נמצאות המשרות הבאות.

אם AI יכול לעשות את כל התכנות, למה ללמוד?

- המודלים למדו מקוד ציבורי. רוב הקוד בעולם פרטי, 81.5% מהתרומות ב-GitHub לבד, ובעיה שאף אחד לא פתר קודם לא נמצאת באף סט אימון.
- עם קוד legacy זה אותו סיפור. אין קהילת קוד פתוח גדולה של COBOL: הקוד שמריץ בנקים, חברות ביטוח וממשלות הוא פרטי, והמודלים כמעט לא ראו אותו.
- לדעתי, זו ההזדמנות. ככל שפחות אנשים מוכנים ללמוד את היסודות בדרך הקשה, כך שווים יותר אלה שכן. הם יהיו הדור שסוגר את הפערים שהמודלים לא מצליחים לסגור לבד.
- איפה שמשהו דומה פורסם, AI משכפל אותו מהר. איפה שלא, הוא מסתובב בלולאות ארוכות ויקרות, ומישהו צריך לקחת את ההגה: להגדיר מה זה "נכון", לבנות את האורקל, לכתוב בדיקות עם יעדים אובייקטיביים, למצוא את המקור שאף אחד לא פרסם, ולשים לב כשהסוכן מתחיל להסתובב במעגלים.
- כל אחת מהעבודות האלה צריכה מישהו שמבין את הבעיה מלמטה: איך מחשבים, מערכות הפעלה, קומפיילרים, רשתות ומסדי נתונים באמת עובדים. זה הידע הקשה שרוב האנשים מדלגים עליו.

> ***המיומנות שהמכונה הכי צריכה מכם היא זו שהיא הכי פחות מסוגלת ללמוד מהאימון שלה.***
>
> *A habilidade que a máquina mais precisa de você é a que ela menos consegue aprender com o treino dela.*



8. דוגמה אמיתית

מה באמת עמד מאחורי “ה-AI עשה את זה”

ai-memory, מערכת הזיכרון בקוד פתוח שלי לסוכני קוד, נכתבה ברובה על ידי AI. כל מה ששומר עליה נכונה הגיע מהבדיקה וההכוונה שלי, שבוע אחרי שבוע.

מה באמת עמד מאחורי "ה-AI עשה את זה"

4,743
מילים של כללים לסוכנים ב-AGENTS.md

16
אינווריאנטים שאסור להם להישבר אף פעם

22
גבולות אבטחה נאכפים, כל אחד ממופה לשומר ולבדיקות עוינות

+3,000
בדיקות יחידה

+600
בדיקות אינטגרציה

- בערך 65 הנחיות מפורשות, 28 הערות זיכרון אצורות שקיימות בגלל טעויות ספציפיות, ויותר מ-60 מסמכים עם מחקר, תחקירים והחלטות ארכיטקטורה.
- כל שכבה מסמנת מקום שבו ה-AI לבד לא הצליח, והייתי צריך לדעת איך לתקן.
- שום דבר מזה לא היה שם ביום הראשון. אלה שבועות של מעקב אחרי ה-AI שעושה את רוב העבודה כל יום וליטוש שלה, והליטוש תלוי בניסיון.

8. דוגמה אמיתית

שום דבר מזה לא אוטומטי, לא נעשה על ידי ה-AI. כל זה הוא המקום שבו ה-AI לבד לא מצליח ואני צריך לדעת איך לכוונן. אף חובבן לא יכול לעשות את זה.

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

אני ב-25 ,X בספטמבר 2026



◆ 9. מה לעשות

אם אתם סטודנטים או ג׳וניורים

השתמשו ב-AI כמו שסניור משתמש במתמחה: לעבודות השחורות. הלמידה חייבת לעבור דרך הידיים שלכם.

אם אתם סטודנטים או ג׳וניורים

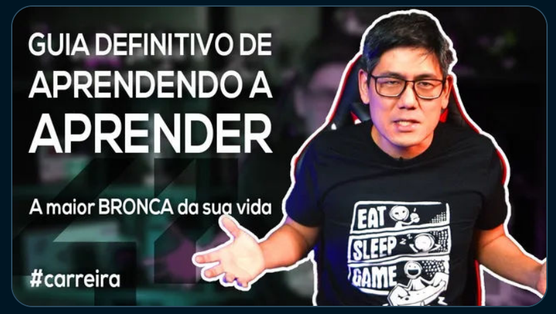
- כתבו ביד כמה שאתם יכולים. תשברו, תדבגו, תתקנו. להיתקע זה חלק מהעניין: החוקרים של Anthropic כותבים שמאמץ קוגניטיבי, "אפילו להיתקע בצורה כואבת", כנראה חשוב לשליטה.
- בדקו כל מה שהוא מייצר, שורה אחרי שורה. אם אתם לא יכולים להסביר שינוי, אתם לא מוכנים לקבל אותו.
- סניור הוא מי שהיה ג׳וניור בכל נושא שהוא שולט בו.
- תנו ל-AI לטפל במשימות השגרתיות במקביל, ובקשו ממנו להסביר מושגים, לא לכתוב את הקוד שאתם אמורים ללמוד ממנו.
- תספרו את השעות. מאפס ועד משרת הג׳וניור הראשונה יש לפחות בערך 1,400 שעות של לימוד ותרגול; לדעתי, שליטה דורשת עשרות אלפים. לכנרים הטובים ביותר של Ericsson היו יותר מ-10,000 עד גיל 20, ומחקרים מאוחרים יותר מראים שתרגול הוא לא כל הסיפור, אבל שום דבר לא מחליף אותו.

הארכיון של Akitando



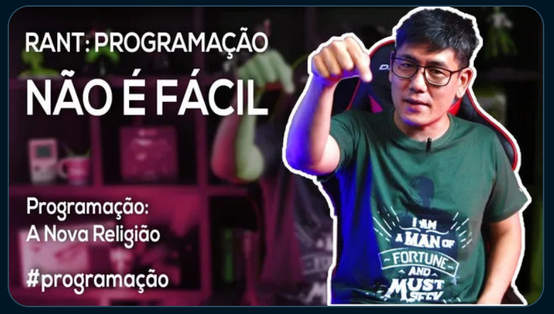
O Guia +Hardcore de Introdução à COMPUTAÇÃO

יוני 2020 • 1:18:28



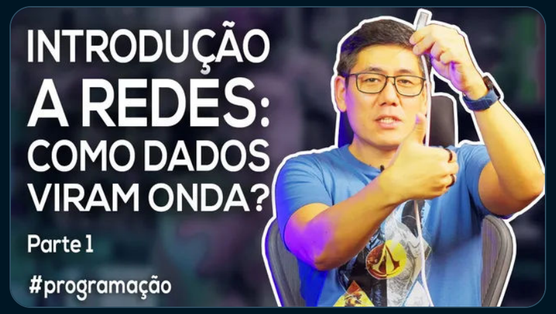
Guia DEFINITIVO de Aprendendo ... Aprender | A maior BRONCA da

אפר׳ 2020 • 1:04:37



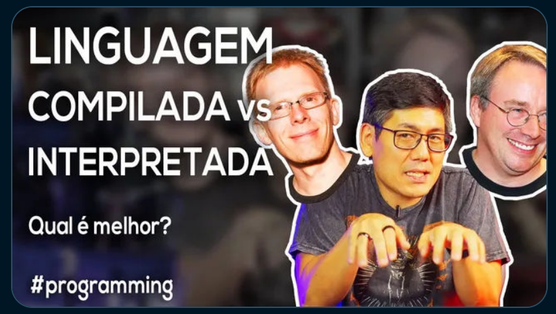
RANT: Programação NÃO É Fácil

פבר׳ 2020 • 32:22



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

יולי 2022 • 37:59



Linguagem Compilada vs ?Interpretada | Qual é melhor

אפר׳ 2022 • 1:11:44



Hello World Como Você Nunca Viu! | Entendendo C

מרץ 2021 • 1:09:22

המסקנות

1

AI מכפיל את מה שאתם כבר יודעים. אם אין הרבה מה להכפיל, מקבלים מעט מאוד.

2

קיצורי דרך שנמכרים כקריירה כבר קרסו פעם אחת. החדש הוא אותה הבטחה.

3

למידה על טייס אוטומטי לא נקלטת. המחקרים עד עכשיו מסכימים.

4

העבודה שנשארת בעלת ערך זקוקה לאנשים שיודעים לשפוט את התוצר, בזמן שהגיוס למשרות כניסה מצטמצם.

5

AI לא יכול לנחש את המטרה, המגבלות או הפשרות שלכם. לדעת אותן, זו העבודה.

6

בבנצ'מרק שלי, לתת למודל בדיקות שינה את התוצאות. בחרו את השפה שמתאימה לבעיה.

7

AI משכפל את מה שציבורי. הפערים שהוא משאיר, בעבודה פרטית וחדשה, הם העבודה, והיסודות הקשים הם הדרך למלא אותם.

8

כתבו ביד, תנו ל-AI את העבודות השחורות, בדקו הכול, והשקיעו את השעות.

הדבר היחיד שחשוב הוא המציאות האובייקטיבית.

A única coisa que importa é realidade objetiva.

לקריאה נוספת: הפוסטים שלי והמקורות

AI מכפיל את מה שאתם מביאים

- מאפס לפוסט-פרודקשן בשבוע אחד: שימוש ב-AI בפרויקטים אמיתיים ↗
- VS Code הוא כרטיס הניקוב החדש ↗
- תפסיקו לחפש תירוצים ותשחררו יותר ↗
- Google DORA 2025: AI מגביר את מה שקיים ↗
- Veracode: פרצות אבטחה בקוד שנוצר ב-AI ↗

כבר ראינו את הקיצור הזה

- חפירה: ה-AI הרג את המתכנתים? ↗
- CFPB: הטענות של Lambda School על השמה ↗
- Course Report: שוק הבוטקאמפים, 2013 עד 2018 ↗
- Inside Higher Ed: סגירת הבוטקאמפים ↗
- Higher Ed Dive: 2U יוצאת מהבוטקאמפים ↗
- TechCrunch: הפיטורים בטק ב-2024 ↗
- Akitando #132: בועת הסטארטאפים התפוצצה (2022, בפורטוגזית) ↗

למידה על טייס אוטומטי לא נקלטת

- בנצ'מרק ה-LLM החדש v4, חלק 1: התהליך ↗
- VS Code הוא כרטיס הניקוב החדש ↗
- איך DriveClub ו-shadPS4 כמעט ניצחו את ה-AI ואותי: איך ללמוד ↗
- Anthropic: איך עזרה מ-AI משפיעה על מיומנויות תכנות ↗
- Bastani ואחרים, PNAS: AI גנרטיבי יכול לפגוע בלמידה ↗
- METR: מפתחים מנוסים איטיים יותר ב-19% ↗
- MIT Media Lab: Your Brain on ChatGPT (פרה-פרינט) ↗

מסלול הכניסה נעשה צר יותר

- חפירה: ה-AI הרג את המתכנתים? ↗
- SignalFire: State of Tech Talent 2026 ↗
- Indeed: שוק העבודה נוטה לכיוון הבכירים ↗
- Indeed: מודעות דרושים לתוכנה מאז Claude Code (יולי 2026) ↗
- Stanford Digital Economy Lab: פער ה-AI אצל עובדים צעירים (אוגוסט 2026) ↗
- New York Fed: שוק העבודה של בוגרים טריים ↗
- BLS: מפתחי תוכנה ↗

לקריאה נוספת: הפוסטים שלי והמקורות

AI לא יכול לנחש מה אתם צריכים

- ↗ למה מודלי LLM ייכשלו בחברה שלכם
- ↗ למה מודלי LLM לא נותנים לכם את התוצאה שאתם מצפים לה
- ↗ Vibe code: בניתי אפליקציה קטנה ב-100% עם GLM 4.7 (TV Clipboard)

בדיקות חשובות יותר מהשפה

- ↗ יש שפות טובות יותר לתכנות עם LLMs?
- ↗ הבנצ'מרק: קוד, נתונים ושיטה

אם AI יכול לעשות את כל התכנות, למה ללמוד?

- ↗ המגבלות של מודלי LLM: טובים בשחזור מה שקיים, יקרים במה שלא
- ↗ הנדסה לאחור עם AI: האם Adobe מתה?
- ↗ VS Code הוא כרטיס הניקוב החדש
- ↗ GitHub Octoverse 2025: 81.5% פרטי
- ↗ arXiv (2026): מודלי שפה ו-COBOL, קוד שכמעט אף פעם לא ציבורי
- ↗ ARC Prize: משימות חדשות, 0% למודלי שפה טהורים

מה באמת עמד מאחורי "ה-AI עשה את זה"

- ↗ ai-memory 2.0
- ↗ תוכנה אף פעם לא גמורה: ארבעה פרויקטים והחיים אחרי ה-deploy

אם אתם סטודנטים או ג'וניורים

- ↗ איך DriveClub ו-shadPS4 כמעט ניצחו את ה-AI ואותי: איך ללמוד
- ↗ חפירה: Akita נכנע ל-AI?
- ↗ מסכם את מרתון ה-AI שלי: הצלחה או כישלון?
- ↗ Anthropic: "להיתקע בצורה כואבת"
- ↗ Ericsson ואחרים (1993): תרגול מכוון
- ↗ Macnamara ו-(2019) Maitra: השחזור
- ↗ Akitando #72: תכנות זה לא קל (בפורטוגזית)

כל נקודה, עם כל הקישורים, באתר

/ailair.akitaonrails.com/he/students

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