

roadmap.sh

نقشه‌های راه مرتبط

✓

نقشه راه Claude Code

✓

نقشه راه مهندس هوش مصنوعی

✓

نقشه راه ایجنت‌های هوش مصنوعی

ویب کدینگ

ابزارهای کدنویسی با کمک هوش مصنوعی

Gemini

Claude Code

Cursor

ChatGPT

Replit

Windsurf

متمرکز بر فرانت‌اند

Lovable

v0

ویب کدینگ چیست؟

ذهنیت ویب کدر

قبل از کدنویسی برنامه‌ریزی کنید

- برنامه‌ریزی کنید چه چیزی باید توسعه دهید (MVP، فازهای مختلف)
- به‌جای تلاش برای ساخت همه‌چیز یک‌باره، مرحله به مرحله پیش بروید
- با مثال (ماک‌آپ، نمونه کد، تصاویر) برای هوش مصنوعی روشن کنید چه می‌خواهید

پشته فناوری و کدنویسی

- به‌جای پشته‌های فناوری جدید یا خاص، یک پشته محبوب انتخاب کنید
- اگر ترجیحات سبک یا کدنویسی دارید، آن‌ها را برای هوش مصنوعی مستند کنید
- از هوش مصنوعی بخواهید کد را ماژولار نگه دارد و به سمت ماژول‌ها یا فایل‌های کوچک‌تر برود
- به‌طور مرتب از هوش مصنوعی بخواهید کدبیس را بازبینی و بازآرایی (ریفکتور) کند
- از مهارت‌های (skills) ساخته‌شده توسط دیگران استفاده کنید

بهترین شیوه‌های نوشتن پرامپت

- به‌جای درخواست پنج مورد مختلف، هر بار یک کار بخواهید
- به‌جای دستورالعمل‌های کلی و مبهم، دقیقاً مشخص کنید چه می‌خواهید
- بر اساس جلسات کدنویسی قبلی، به هوش مصنوعی بگویید چه کاری را نباید انجام دهد
- ماک‌آپ، فایل‌های مرجع و موادی که می‌تواند کمک کند را در اختیار هوش مصنوعی بگذارید
- در صورت مفید بودن، از قالب «به‌عنوان ... عمل کن» استفاده کنید (مثلاً به‌عنوان یک پژوهشگر UX عمل کن)
- سند زمینه خود را به‌طور مرتب به‌روزرسانی کنید (مثلاً CLAUDE.md)
- پیش از مسائل پیچیده، صراحتاً از هوش مصنوعی بخواهید «فکر کند» یا «طوفان فکری» انجام دهد

زمینه (Context)

- در صورت وجود و نیاز، از پنجره زمینه بلند استفاده کنید
- اگر هوش مصنوعی پس از سه پرامپت شکست خورد، متوقف شوید و گفتگوی تازه‌ای شروع کنید
- برای کارهای نامرتبط، فعالانه پاک‌سازی کنید و جلسات جدید شروع کنید
- در صورت امکان، از هوش مصنوعی بخواهید از زیرایجنت‌ها (subagents) استفاده کند

دییگ کردن

- پیام خطا را پرامپت کنید و بقیه کار را به هوش مصنوعی بسپارید
- اگر خطاها ادامه داشت، از هوش مصنوعی بخواهید فهرستی از علل احتمالی بسازد
- به هوش مصنوعی بگویید لاگ اضافه کند تا خطا سریع‌تر پیدا شود
- در صورت امکان، MCP (مثلاً Playwright برای مرورگر) را نصب کنید و از هوش مصنوعی بخواهید از آن استفاده کند

بر کنترل نسخه مسلط شوید

- به‌طور مرتب git استفاده کنید (مثلاً پس از هر کار موفق هوش از commit مصنوعی)
- هر ویژگی جدید را با وضعیت پاک Git شروع کنید
- اگر نیاز به بازگشت (revert) دارید، از Git استفاده کنید، نه قابلیت بازگشت داخلی هوش مصنوعی
- از هوش مصنوعی بخواهید کارهای Git و GitHub CLI شما را انجام دهد

تست

- از هوش مصنوعی بخواهید تست بنویسد (تست‌های E2E می‌تواند به ساخت محصولی پایدار کمک کند)
- توسعه مبتنی بر تست (TDD) را در نظر بگیرید
- وقتی باگی پیدا کردید، از هوش مصنوعی بخواهید ابتدا یک تست شککننده بنویسد و سپس آن را رفع کند
- پس از استقرار تست‌ها، به‌طور مرتب بازآرایی کنید

بهترین شیوه‌های امنیتی

- صراحتاً از هوش مصنوعی بخواهید یک ممیزی امنیتی از برنامه انجام دهد
- هرگز اطلاعات ورود را هاردکد نکنید؛ به‌جای آن از متغیرهای محیطی استفاده کنید

ابزارهای هوش مصنوعی می‌توانند در برنامه‌ریزی کمک کنند

برای مثال، اگر از «Claude Code» استفاده می‌کنید، به ابزار بگویید چه چیزی می‌خواهید بسازید، از آن بخواهید در پرداختن ایده کمک کند، فازهای مختلف را مشخص کنید و پس از اتمام، از آن بخواهید همه‌چیز را در یک سند مستند کند تا هنگام ساخت واقعی محصول بتوانید به آن مراجعه کنید.

ایجاد استانداردها از همان ابتدا اهمیت دارد

در ابتدای کار، خروجی‌های اولیه هوش مصنوعی را با دقت بررسی کنید؛ الگوهای کدنویسی، سبک‌ها و تصمیمات معماری. اگر عادت‌های بد را زود تشخیص ندهید، هوش مصنوعی آن‌ها را در هر تکرار تقویت می‌کند و این عادت‌ها به‌سرعت انباشته می‌شوند.

جلسات بازآرایی را به‌طور مرتب اجباری کنید

هوش مصنوعی مسیر کم‌مقاومت‌ترین را انتخاب می‌کند (افزودن کد، بزرگ‌کردن فایل‌ها، رد شدن از پاکسازی). هرازگاهی از آن بخواهید عقب بایستد و بازآرایی کند: تقسیم به ماژول‌های کوچک‌تر، حذف کدهای مرده، بهبود عملکرد.

سند زمینه را به‌روز نگه دارید

اگر متوجه شدید برخی دستورالعمل‌ها را تکرار می‌کنید، آن‌ها را در سند زمینه مستند کنید. نکته کمک‌کننده دیگر این است، پس از یک جلسه گفتگو اگر احساس کردید هوش مصنوعی می‌تواند از آموخته‌های آن جلسه بهره ببرد، از آن بخواهید سند زمینه (مثلاً claude.md) را با آموخته‌های آن جلسه به‌روزرسانی کند.

زمینه را به‌طور مرتب پاک کنید

برای نتایج بهتر و صرفه‌جویی در هزینه توکن، زمینه را پاک کنید. هر زمان که می‌خواهید یک کار جدید یا نامرتب را شروع کنید، بهتر است زمینه را پاک کرده و از نو شروع کنید.

بگذارید هوش مصنوعی دییگ کند، اما بفهمید مشکل چه بود

وقتی چیزی خراب می‌شود، خطا و کد مرتبط را کپی کنید و از هوش مصنوعی بخواهید خطا را توضیح دهد. فقط راه‌حل را نپذیرید، مطمئن شوید آن را می‌فهمید. اگر نمی‌فهمید، از هوش مصنوعی بخواهید آن را به زبان ساده توضیح دهد.

مرتب‌کامیت کنید، با پیام‌های واضح

پس از هر ویژگی یا رفع مشکل کارآمد، یک کامیت برزنید. از هوش مصنوعی بخواهید پیام کامیت واضحی پیشنهاد دهد که توضیح دهد چه چیزی و چرا تغییر کرده است. این کار یک نقطه بازگشت امن به شما می‌دهد تا اگر بعداً مشکلی پیش آمد، به آن بازگردید.

نوشتن تست را به‌طور پیش‌فرض از هوش مصنوعی اجباری کنید

رفتار پیش‌فرض بیشتر ابزارهای هوش مصنوعی، کدی با اولویت پیاده‌سازی و پوشش تست حداقلی تولید می‌کند. هر زمان هوش مصنوعی ویژگی‌ای می‌سازد، آن را مجبور کنید بلافاصله تست‌های پایه بنویسد. این‌طور باگ‌ها زود شناسایی می‌شوند و همیشه می‌دانید چه زمانی چیز جدیدی چیز قدیمی را خراب کرده است.

هرگز اطلاعات محرمانه خود را هاردکد نکنید

هرگز اجازه ندهید هوش مصنوعی رمزهای عبور، کلیدهای API یا توکن‌ها را مستقیم در کد قرار دهد. اگر دیدید این کار را می‌کند، متوقفش کنید و بخواهید به‌جای آن از متغیرهای محیطی استفاده کند. این یکی از رایج‌ترین اشتباهات مبتدیان و یکی از خطرناک‌ترین آن‌هاست.

با مسیرهای مرتبط زیر به یادگیری ادامه دهید

ایجنت‌های هوش مصنوعی

مهندس هوش مصنوعی

Claude Code