



PRODUCT INTRODUCTION

Aira Code چیست؟

عامل کدنویسی داخل ترمینال که پروژه را می‌فهمد، تغییر می‌دهد، تست می‌کند و تصمیم‌ها را به یاد می‌سپارد.

پرداخت ریالی

حافظه پروژه

ترمینال و مرورگر

عامل کدنویسی

کدنویسی واقعی فقط «جواب دادن» نیست؛ باید روی **خود پروژه** کار کرد

Aira Code تکمیل خودکار چند خط کد یا یک پنجره چت کنار ادیتور نیست. یک عامل است که ابزار دارد و نتیجه را در مخزن واقعی تحویل می‌دهد.

چت کدنویسی معمولی

شما: «این باگ ورود را درست کن.»

پاسخ: «این نمونه کد را امتحان کن...»؛ بدون دیدن پروژه و بدون اجرای تست.

Aira Code به عنوان عامل

شما: «این باگ ورود را درست کن.»

آیرا: مسیر احراز هویت را پیدا می‌کند، تغییر محدود می‌دهد، تست مربوط را اجرا می‌کند و نتیجه را تحویل می‌دهد.

مخزن را می‌خواند

ساختار پروژه، فایل‌های مرتبط، قراردادها و قوانین موجود را پیدا می‌کند.

تغییر واقعی می‌دهد

فایل‌ها را در محدوده کار ویرایش می‌کند؛ فقط یک قطعه کد برای کپی‌کردن نمی‌فرستد.

صحت را بررسی می‌کند

دستور، تست، build و بررسی خروجی را اجرا می‌کند و نتیجه را گزارش می‌دهد.

Aira Code یک همکار اجرایی برای توسعه نرم افزار است

تجربه از ترمینال شروع می شود، اما پشت آن عامل، حافظه، مدل ها و حساب آیراچت به هم وصل اند.

۱ رابط توسعه دهنده

CLI تعاملی، دستور یک مرحله ای و نشست های مرورگر برای کارهای طولانی تر.

۱

۲ عامل و ابزارها

خواندن، جست و جو، ویرایش، اجرای shell، تست، برنامه ریزی و کار با MCP.

۲

۳ حافظه و مسیریابی

حافظه CogX تصمیم های پروژه را نگه می دارد و Aira Levels مسیر مدل و عمق فکر را تنظیم می کنند.

۳

تعریف ساده

کاری را توضیح می دهی؛ آیرا کد داخل پروژه می گردد، تغییر می دهد، بررسی می کند و گزارش می دهد.

هدف این است که فاصله میان «پیشنهاد کد» و «تغییر قابل استفاده در پروژه» کوتاه شود.

از نصب تا اولین کار، همه چیز در چند قدم روشن انجام می شود

Aira Code روی Linux، macOS و Windows در دسترس است و حساب Aira Chat، کیف پول و حافظه را یکپارچه می کند.

aira - zsh

```
$ curl -fsSL https://airacode.ir/install.sh | bash
```

```
✓ AiraCode installed
```

```
$ aira login
```

```
Opening AiraChat in your browser...
```

```
$ cd my-project && aira
```

```
AiraCode · project memory connected
```

```
• این خطای ورود را پیدا کن، اصلاح کن و تستها را اجرا کن >
```

```
Reading repository structure...
```

```
✓ Relevant files found · plan ready
```

۱ نصب در یک خط

نصب کننده سیستم عامل و shell را تشخیص می دهد.

۲ ورود با Aira Chat

دستور login مرورگر را برای تأیید حساب باز می کند.

۳ باز کردن پروژه

به پوشه مخزن برو و نشست تعاملی را شروع کن.

۴ توضیح کار با زبان طبیعی

مثلاً «این خطا را پیدا کن، اصلاح کن و تست مربوط را اجرا کن.»

عامل تا وقتی نتیجه قابل بررسی نباشد، کار را تمام شده فرض نمی‌کند

Aira Code یک حلقه تکرارشونده دارد: بفهم، برنامه‌ریزی کن، تغییر بده، تست کن و نتیجه را با شواهد جمع‌بندی کن.





برای کار روی پروژه، فقط مدل کافی نیست؛ عامل به ابزار واقعی نیاز دارد

ابزارها کمک می‌کنند پاسخ به یک تغییر قابل اجرا، محدود و قابل اثبات تبدیل شود.



Shell، تست و build

اجرای دستورها و بررسی اینکه تغییر واقعاً در محیط پروژه کار می‌کند.



ویرایش فایل‌ها

اعمال تغییر مشخص در فایل‌های لازم و حفظ محدوده کار.



خواندن و جست‌وجو

پیدا کردن فایل، نماد، الگو و مسیر اجرای مرتبط با درخواست.



زیرعامل‌ها و کار پس‌زمینه

تقسیم کارهای مستقل و ادامه بعضی اجراهای طولانی بدون قفل کردن مسیر اصلی.



MCP و LSP

اتصال به ابزارها و سرویس‌های بیرونی و استفاده از اطلاعات ساختاری کد.



Plan و Worktree

برنامه‌ریزی پیش از اجرا و جداسازی کارهای هم‌زمان در صورت نیاز.



حافظه پروژه یعنی لازم نیست هر نشست، دوباره از صفر ساخته شود

CogX نکات مرتبط را برای کار فعلی بازیابی می‌کند؛ هدف، نگه داشتن تصمیم‌های مفید است، نه ریختن تمام تاریخچه در هر درخواست.

حافظه بازخورد

اصلاح‌های قبلی، قواعد دلخواه و سبکی که در بررسی و تحویل انتظار داری.

حافظه کاربر

ترجیحات پایدار، شیوه کار و اطلاعاتی که میان پروژه‌ها به درد می‌خورند.

حافظه مرجع

مسیر فایل‌های مهم، API‌ها، داشبوردها و runbook‌هایی که باید دوباره پیدا شوند.

حافظه پروژه

تصمیم‌های معماری، رخدادها، استقرارها و نکات خاص همین مخزن.



به جای نام شرکت مدل، نوع کار را انتخاب می‌کنی

هر سطح ترکیبی از مسیر مدل، عمق حافظه، بودجه فکرکردن و اندازه زمینه است. گیت‌وی انتخاب ارائه‌دهنده را پشت صحنه مدیریت می‌کند.

aira-deep

عمیق

برای معماری، تصمیم پیچیده و تحلیل‌هایی که فکر و حافظه بیشتری می‌خواهند.

aira-code

کد

برای دیباگ، refactor و کار چندمرحله‌ای با ابزارهای واقعی مخزن.

aira-balanced

متعادل

پیش‌فرض مناسب برای کدنویسی روزمره؛ تعادل میان کیفیت، سرعت و هزینه.

aira-max

بیشینه

برای کار سنگین چندمرحله‌ای، وقتی کیفیت نهایی از سرعت و هزینه مهم‌تر است.

aira-fast

سریع

برای پاسخ کوتاه، خلاصه، عنوان و کارهای سبک با تأخیر کمتر.

aira-docs

اسناد

برای PDF، تصویر، فایل‌های طولانی و درک زمینه گسترده.

در عمل: برنامه‌نویس نوع مسئله را انتخاب می‌کند؛ لازم نیست دائماً کاتالوگ مدل‌ها و تفاوت ارائه‌دهنده‌ها را دنبال کند.

گیت وی ایراکد، مسیر مدل ها و حساب را یک جا مدیریت می کند

برنامه نویس به جای مدیریت چند حساب، چند کلید و چند روش پرداخت، از هویت Aira Chat و کیف پول ریالی استفاده می کند.

۱. ترمینال به API ایرا وصل می شود

ورود، نشست و وضعیت حساب از یک مسیر داخلی مدیریت می شوند.

۲. حافظه و سطح انتخابی اعمال می شوند

زمینه مرتبط و بودجه کیفیت پیش از درخواست مدل مشخص می شوند.

۳. درخواست به مدل مناسب می رود

گیت وی مسیر ارائه دهنده و پاسخ استریم را پشت صحنه مدیریت می کند.

۴. مصرف با AiraCredit ثبت می شود

شارژاز کارت شتاب انجام می شود و مصرف هر درخواست در کنسول قابل مشاهده است.



از یک خطای کوچک تا یک تغییر چند مرحله‌ای

می‌توان با یک درخواست ساده شروع کرد و سطح اختیار، مدل و عمق بررسی را متناسب با ریسک کار بالا برد.

Refactor محدود

بهبود ساختار بدون تغییر رفتار و با کنترل دقیق فایل‌های دست‌خورده.

عوض نشود API این سرویس را کوچکتر کن؛

ساخت قابلیت

پیدا کردن الگوهای پروژه و پیاده‌سازی feature با کمترین پراکندگی.

فیلتر وضعیت را به همین داشبورد اضافه کن

پیدا کردن باگ

ردگیری مسیر خطا، یافتن علت ریشه‌ای و اصلاح همراه با تست.

این خطای ورود را بازتولید و رفع کن

انتشار و عملیات

اجرای gate‌های انتشار، جمع‌کردن شواهد و گزارش وضعیت واقعی.

ها را اجرا و نتیجه را خلاصه کن release gate

اسناد و توضیح

فهم معماری، ساخت onboarding و هم‌راستا کردن docs با کد.

مسیر پرداخت را برای عضو جدید توضیح بده

تست و کیفیت

نوشتن تست، اجرای verifier و پیدا کردن حالت‌های مرزی مهم.

برای این باگ یک تست رگرسیون اضافه کن

لازم نیست تنظیمات فعلی را از اول بسازی

Aira Code می‌تواند بخش‌های سازگار محیط‌های Claude Code و Codex را پیش‌نمایش و وارد کند؛ کنترل نهایی دست کاربر می‌ماند.

تنظیمات و قواعد

تنظیمات سازگار، فایل‌های حافظه، ruleها و پروژه‌های مورد اعتماد قابل انتقال‌اند.

MCP، دستور و Skill

سرورهای MCP، commandها، MCPها، agentها و skillهای سازگار می‌توانند وارد محیط آیراکد شوند.

پیش‌نمایش پیش از تغییر

حالت dry-run نشان می‌دهد چه چیزی قرار است وارد شود؛ پیش‌فرض‌های خطرناک بدون اجازه کپی نمی‌شوند.

```
import settings
```

```
> /import-settings --dry-run
```

```
Preview compatible settings before import
```



یک موتور، سه روش استفاده برای نوع‌های مختلف کار

از گفت‌وگوی تعاملی داخل ترمینال تا اجرای کوتاه در اسکریپت و نشست طولانی در مرورگر.

`app.airachat.ir/airacode`



نشست در مرورگر

برای اجراهای طولانی که بهتر است ترمینال را آزاد بگذارند و وضعیت از وب دیده شود.

همان هویت، حافظه و کیف پول

`aira -p`



اجرای یک مرحله‌ای

برای پرسش یا کار کوتاه در `command line`، اسکریپت و جریان‌های خودکار بدون ورود به REPL.

مناسب automation و jobهای کوتاه

`aira`



CLI تعاملی

برای کار رفت‌وبرگشتی روی پروژه؛ می‌بینی عامل چه می‌کند، جهت می‌دهی و تا نتیجه جلو می‌روی.

بهترین انتخاب برای توسعه روزمره

در هر سه حالت: حساب Aira Chat، سطح هوش، حافظه پروژه و مصرف AiraCredit یکپارچه می‌مانند.



کد بزن؛ یادش می ماند

Aira Code عامل کدنویسی را به حافظه پروژه، مسیر دسترسی بومی و پرداخت ریالی وصل می کند تا هر نشست ادامه کار قبلی باشد.

airacode.ir

```
curl -fsSL https://airacode.ir/install.sh | bash
```