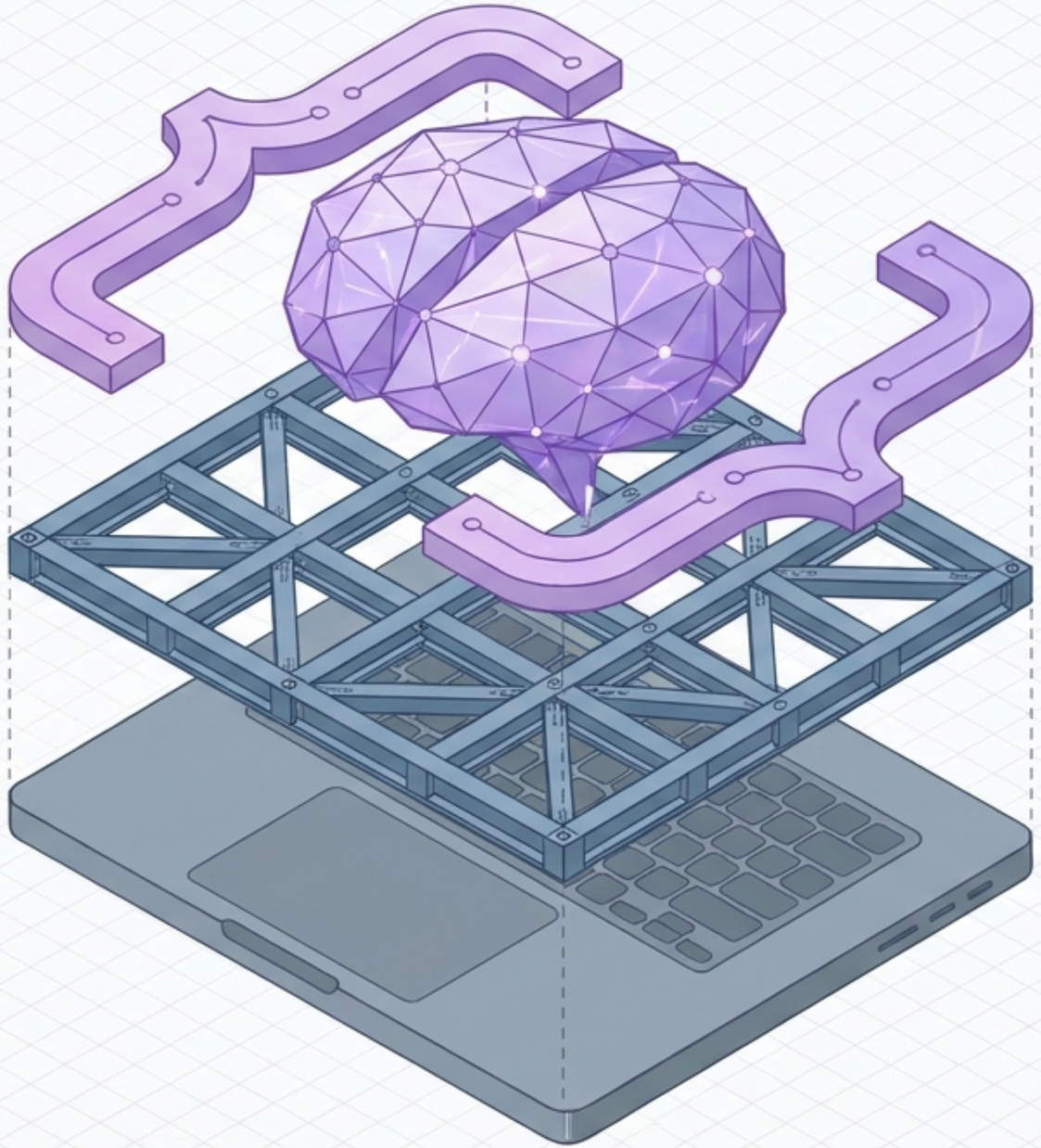
Építsd fel a saját 20 dolláros AI kódoló operációs rendszerdet

Teljes útmutató a Claude Code, a Zed és a T3 Code tökéletes integrálásához Mac-en.



Gemini Notebook

A rendszer anatómiája: Egymásra épülő rétegek

Interfész

T3 Code: Feladatok és ágak vizuális menedzsere

Szerkesztő

Zed: Gyors, natív, tiszta felület

AI Motor

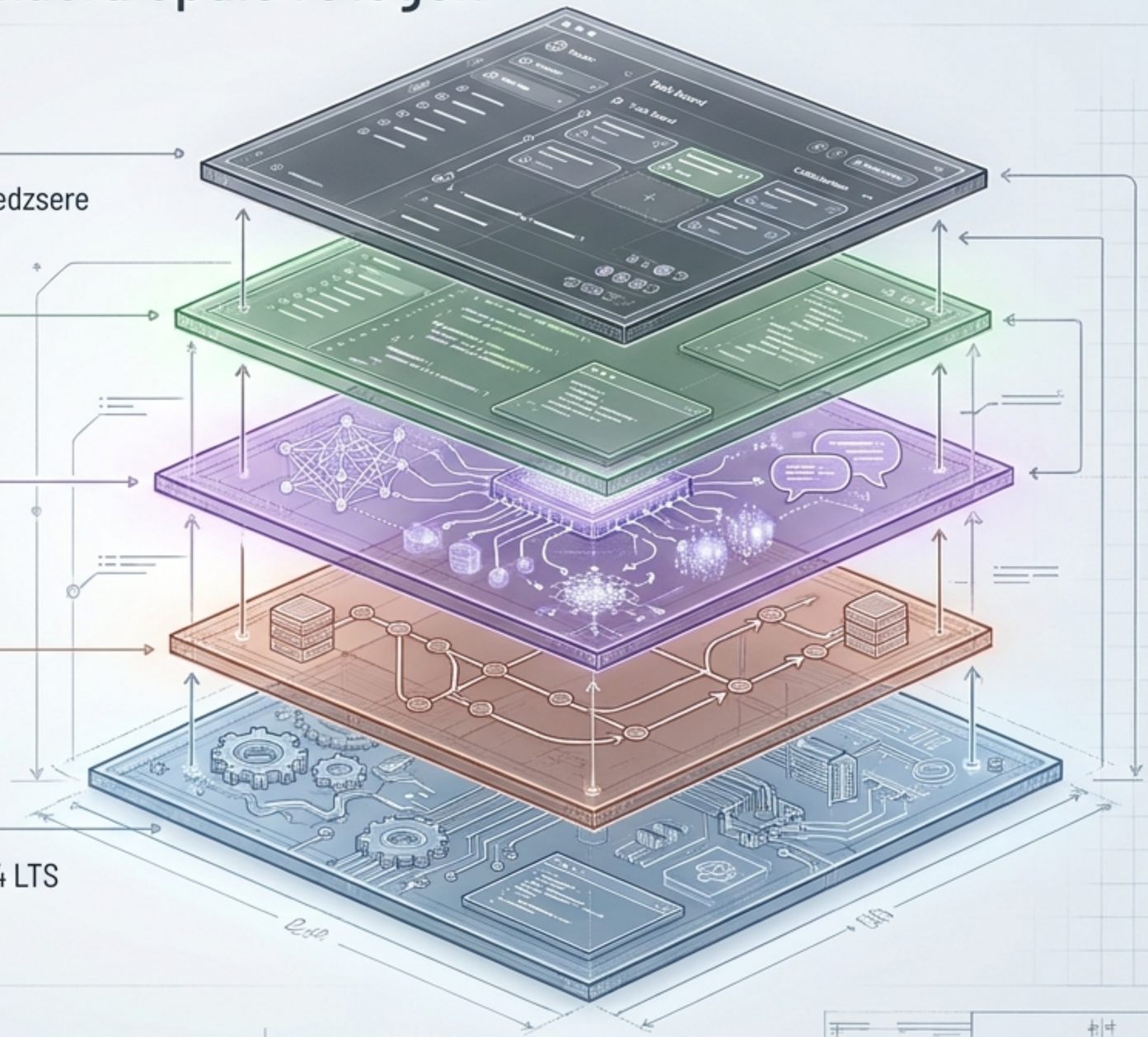
Claude Code CLI

Verziókövetés

Git + GitHub CLI (gh)

Alapozás

Mac OS Terminal + Homebrew, Node.js 24 LTS



Gemini Notebook

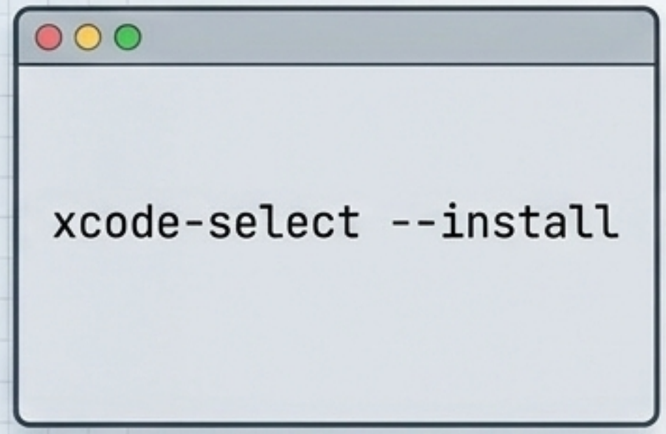
A havi 20 dolláros Pro csomag az ideális belépési pont

Ingyenes ($0)	Pro ($20/hó) - Ajánlott	Max ($100-$200/hó)	Team / Enterprise ($25/fő-től)
Csak webalapú chatbot. Figyelem: Nem tartalmazza a Claude Code CLI hozzáférést!	Teljes Claude Code CLI hozzáférés. 5 órás munkamenet limitek. Ideális tanuláshoz és fejlesztéshez.	Megnövelt limitek és dedikált API kreditek. Csak akkor válts, ha a Pro limitek már hátráltatnak.	Megosztott adminisztráció és számlázás. Minimum 2 fős csapatoknak.

Gemini Notebook

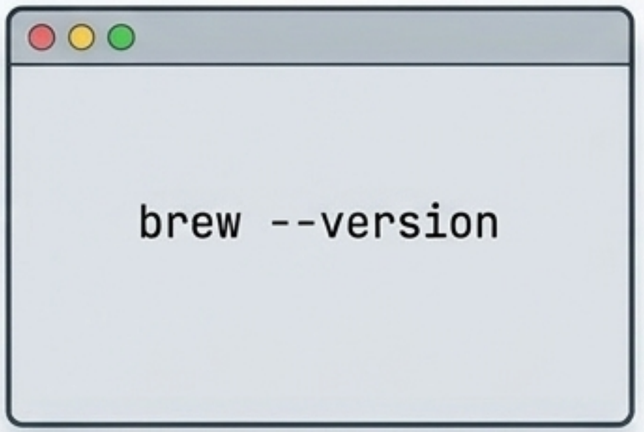
A Mac környezet előkészítése az AI fejlesztéshez

1. Xcode Eszközök



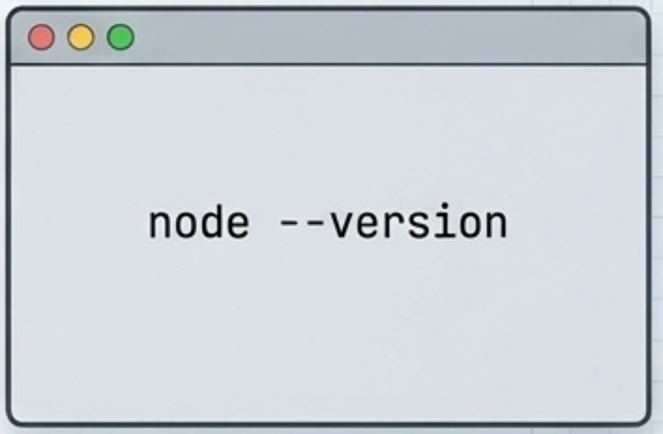
Apple natív fejlesztői parancssori eszközei.

2. Homebrew



Csomagkezelő. Futtasd a telepítő végén kapott 'path' parancsokat!

3. Node.js LTS



Node 24 (LTS) telepítése a weboldalról. Tartalmazza az npm-et.

Gemini Notebook

Verziókövetés inicializálása a GitHub CLI segítségével



Lokális Git

Telepítés:

```
brew install git gh
```

Tipp: A commit előzmények tisztán tartása érdekében használd a GitHub beállításokban található privát e-mail címedet!

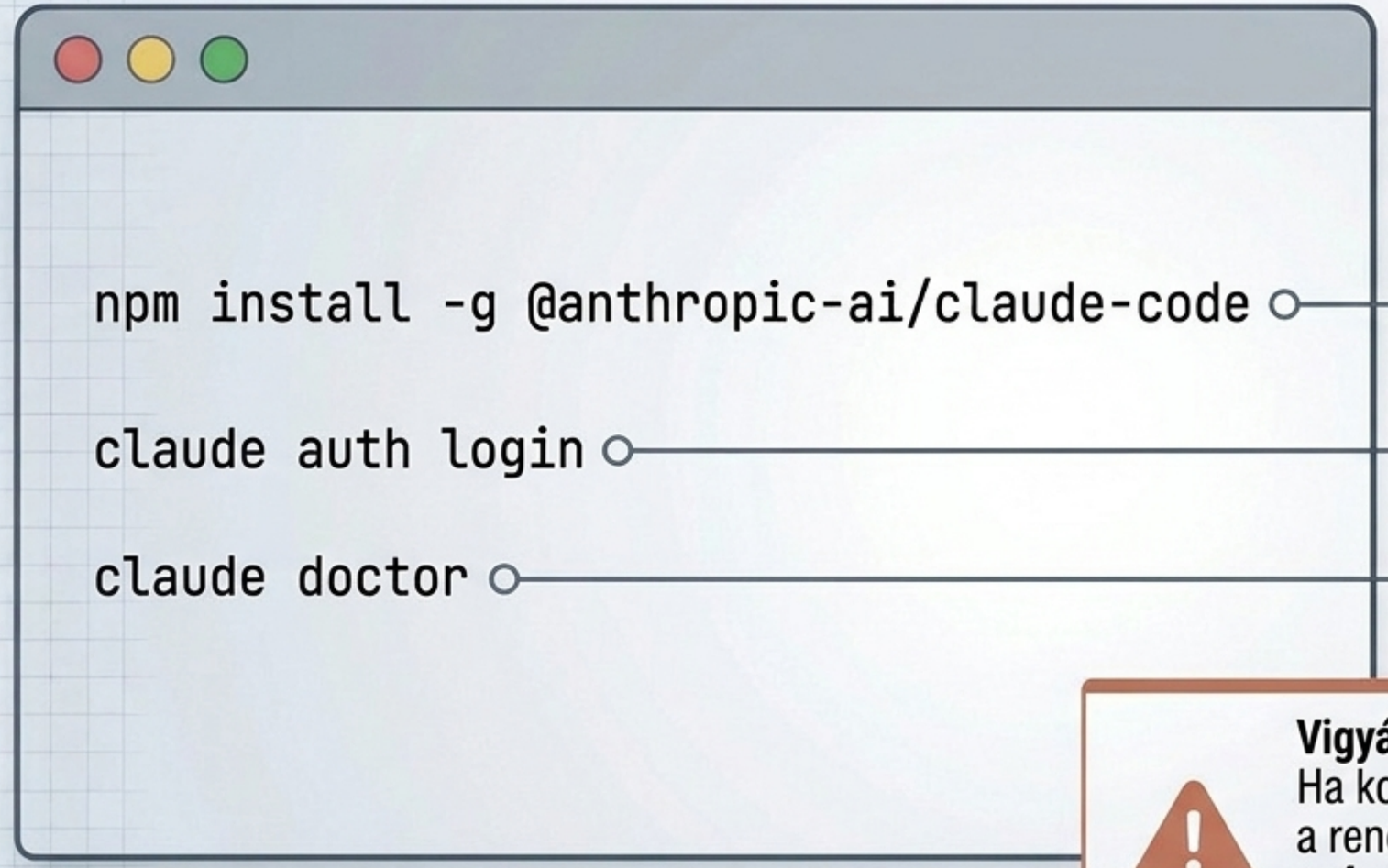


GitHub CLI Hitelesítés

- Bejelentkezés: `gh auth login` (Válaszd a HTTPS és böngészős opciót)
- Git összekötése: `gh auth setup-git`
- Állapotellenőrzés: `gh status`

Gemini Notebook

A központi intelligencia: A Claude Code CLI telepítése



```
npm install -g @anthropic-ai/claude-code
claude auth login
claude doctor
```

Globális telepítés npm segítségével.

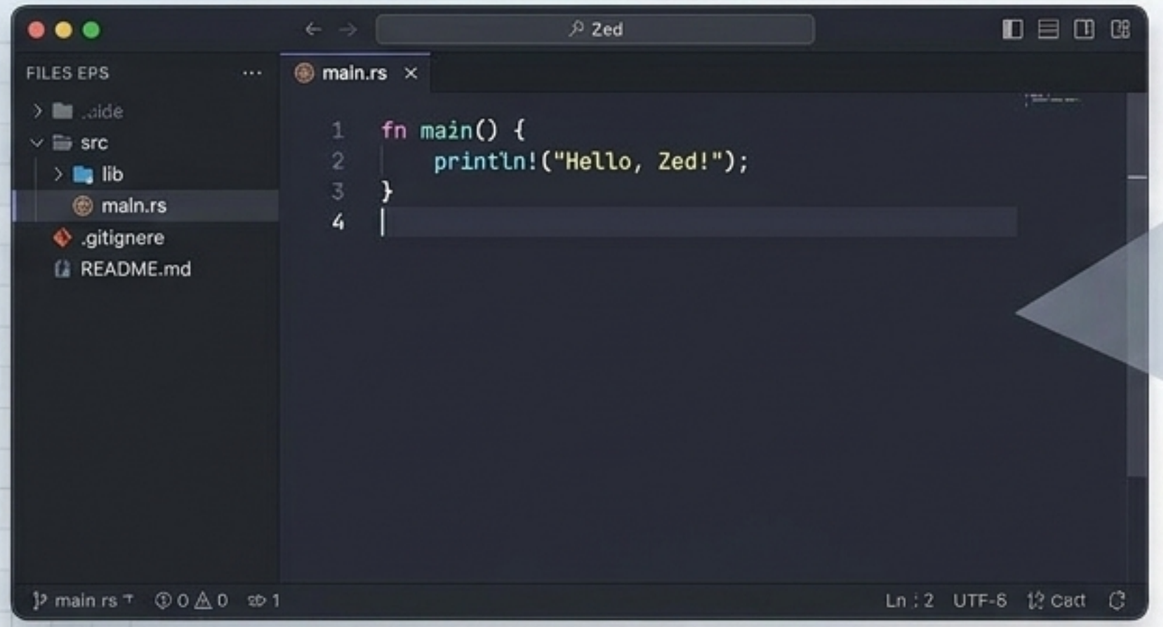
Jelentkezz be a Pro előfizetéssel rendelkező fiókoddal.

Rendszerdiagnosztika futtatása hiba esetén.

Vigyázat a számlázással!
Ha korábban már állítottál be API kulcsot, a rendszer automatikusan **API-alapú számlázásra** válthat az előfizetés helyett. Ellenőrizd a státuszt a /status paranccsal!

Gemini Notebook

A kódszerkesztő: Claude összekötése a Zed-del





Miért a Zed?

- Villámgyors, letisztult, és ingyenes.
- Nem kell minden apró módosításért új AI beszélgetést nyitni; ideális a kézi finomhangoláshoz.

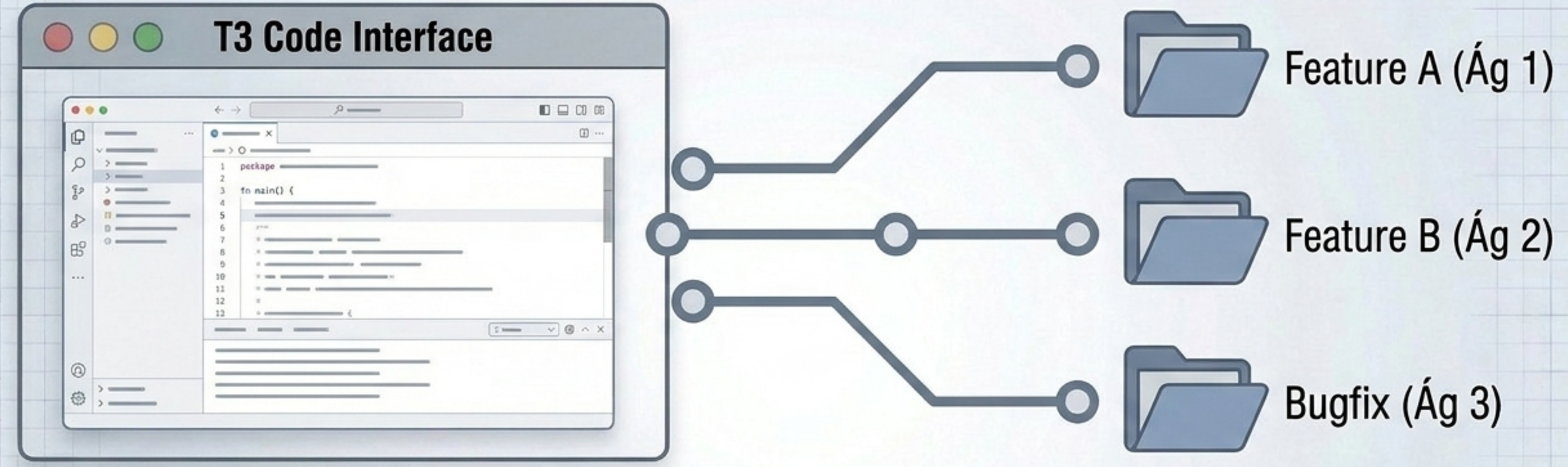
Az ACP (Agent Client Protocol) Integráció

1. Nyisd meg a Command Palette-et (Cmd+Shift+P)
2. Keresd meg és futtasd: `zed: acp registry`
3. Telepítsd a Claude Agent kiegészítőt.

Fontos: A külső Claude agenst válaszd, ne a Zed saját, külön előfizetést igénylő AI szolgáltatását!

Gemini Notebook

T3 Code: Vizuális feladatkezelés és izolált munkaterületek



- **Szerepe:** Ingyenes, nyílt forráskódú asztali alkalmazás, amely grafikus felületet ad a Claude CLI köré.

- **Beállítás:** A Providers menüben engedélyezd a Claude-ot. Biztonsági okokból válaszd a ‘Supervised Permissions’ opciót.

- **Work Tree Koncepció**: A rendszer minden új feladatnak saját git ágat (branch) és elkülönített mappát hoz létre. Az AI anélkül dolgozhat, hogy felülírná a fő kódbázist.

Gemini Notebook

Dedikált AI kliensek összehasonlítása

T3 Code (Ajánlott)

Ingyenes (Open Source)

- Git fókusz és egyszerű diff nézet.
- Izolált Work Tree-k alapértelmezetten.
- **Kinek**: Kezdőknek és a havi 20 dolláros limiten belül maradóknak.

Bambood

$20/hó

- Építész és Munkások koncepció.
- Vizuális feladattábla és csapatkoordináció.
- **Kinek**: Haladóknak, akik komplex, többszereplős AI rendszereket irányítanak.

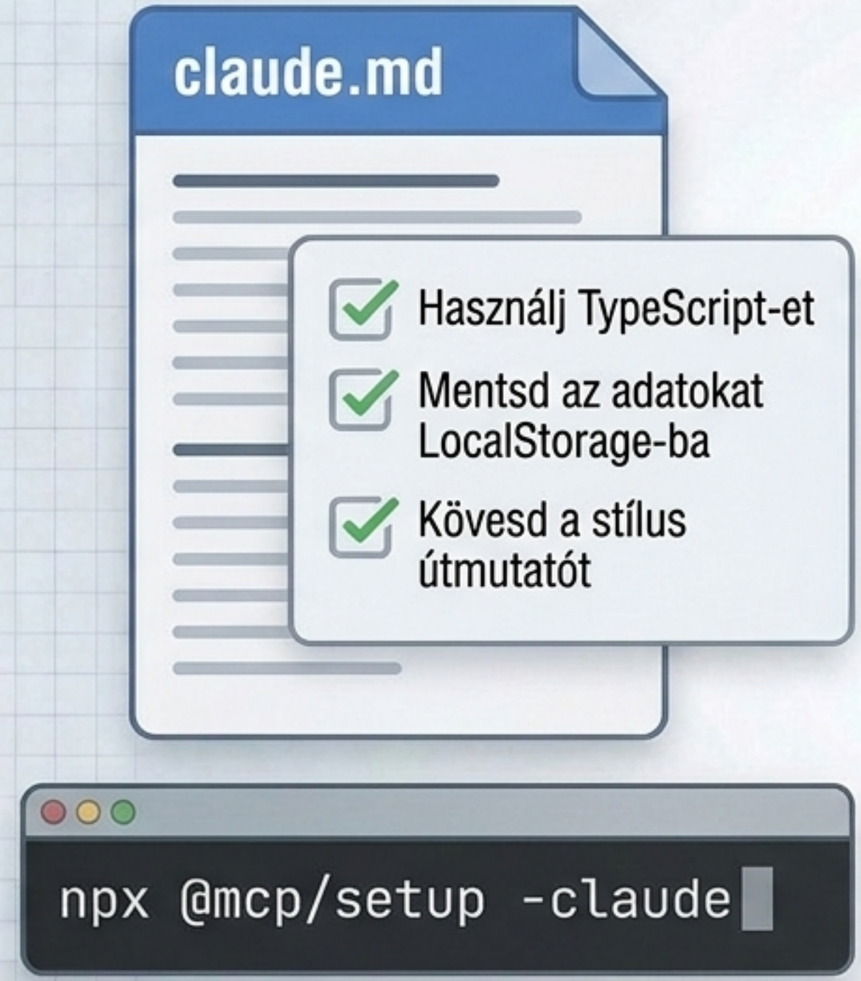
Delta

$10/hó (Pro béta)

- **Delta DB**: egyedi verziókövetés, amely összeköti a kódot a beszélgetéssel.
- **Hátrány**: A direkt Claude CLI integráció még fejlesztés alatt áll.

Gemini Notebook

A projekt inicializálása: A játékszabályok rögzítése



A /init parancs

Hozd létre a claude.md fájlt a projekt gyökerében. Ez tartalmazza az alapvető utasításokat (pl. ‘Használj TypeScript-et’, ‘Mentsd az adatokat LocalStorage-ba’).

.gitignore beállítása

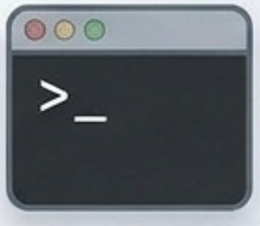
Zárd ki a fordítási fájlokat (build output) és a titkos kulcsokat, hogy ne kerüljenek be a verziókövetésbe.

Dokumentáció elérése (Context MCP)

Futtasd a parancssorban az **npx @mcp/setup** majd a **-claude** parancsot. Ezzel az AI közvetlenül olvashatja a használt könyvtárak aktuális dokumentációját.

Gemini Notebook

Vizuális tesztelés a böngészőben



1. CLI Indítása

Futtasd a **claude --chrome** parancsot és a helyi fejlesztői szervert.



2. Hozzáférés

A **/chrome** paranccsal engedélyezd az AI-nak a lokális weboldal olvasását.



3. Önálló Javítás

A Claude végigkattintja a felületet és automatikusan kijavítja a vizuális hibákat.

Gemini Notebook

A napi AI kódoló ciklus



1. Tervezés (Plan)

Kérj implementációs tervet (Shift+Tab), mielőtt a kódolás elkezdődik.



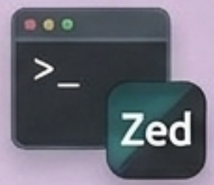
2. Generálás (Scaffold)

Hagyd, hogy az AI felépítse az alapstruktúrát és telepítse a függőségeket.



3. Tesztelés (Test)

Használd a böngészős integrációt a funkciók automatikus validálására.



4. Felülvizsgálat (Review)

Nyisd meg a módosított fájlokat a Zed-ben. Értsd meg, mit írt az AI!



5. Mentés (Commit)

Rögzítsd az új, működő funkciót, mielőtt új feladatot adnál.

Gemini Notebook

Esettanulmány: Egy Olvasónapló alkalmazás felépítése



1. Kérés & Korlátok

Készíts egy React/Vite appot. Korlátok: Nincs üres könyvcím, az adatok frissítés után is maradjanak meg, és a mobilnézet legyen tökéletes.



2. Építés & Önellenőrzés

A Claude legenerálja a kódot, majd a böngésző integráció segítségével maga kattintja végig az appot, hogy letesztelje az ‘üres cím’ validációt.



3. Emberi Döntés

A fejlesztő a T3 Code-ban átnézi a Work Tree fájljait, jóváhagyja a változtatásokat, és a Zed-ben elvégzi az utolsó finomhangolásokat.

Gemini Notebook

A sikeres AI-vezérelt fejlesztés 3 pillére



Gemini Notebook