

Terminal jako interfejs do AI: narzędzia, nie magia

Piotr Bolek
EuroBachTeX2026



Trzy drogi do AI

Three Ways to AI



Dostęp do modeli językowych

- 1 **Okienko czatu** w przeglądarce – **Chat box** in browser
ChatGPT, Claude.ai, Gemini, Openrouter...
- 2 **Wywołanie API** — skrypty, tokeny, JSON – **API calls**
curl, Python, własne narzędzia – custom tools
- 3 **Narzędzia wiersza poleceń** — w terminalu – **CLI tools**
Claude Code, OpenAI Codex CLI, Gemini CLI, Aider...

Terminal – naturalny interfejs do AI dla mnie



Terminalowe narzędzia do AI

CLI AI Tools



Główni gracze — AI CLI (2025/2026) / Main players

Claude Code (Anthropic)

Modele Claude Sonnet/Opus, okno 200k tokenów. Zadaje pytania przed działaniem. Rozszerzalny przez MCP i własne narzędzia.

OpenAI Codex CLI

Modele GPT. Trzy tryby pracy: suggest (tylko propozycja), auto-edit (edytuje pliki), full-auto (działa samodzielnie). Wbudowana piaskownica bezpieczeństwa.

Gemini CLI (Google)

Model Gemini 2.5 Pro z oknem **bardzo duży kontekst**. Integracja z Google Workspace i Drive.

Aider

Open-source, wiele backendów (OpenAI, Claude, Gemini, modele lokalne). git-aware



Jak to działa / How it works

- Uruchamiany w katalogu projektu — **zna kontekst** / Run in the working directory — **knows context**
- Opisywanie zadań w języku naturalnym / Natural language driven
- Narzędzie czyta pliki, proponuje zmiany, **pyta o zgodę** / Reads file, makes propositions, **ask for permission**
- Model pamięta historię rozmowy / Remembers the history of chat
- Tryby pracy / Modes of operation
 - interaktywny REPL — dialog jak w czacie / **interactive** — like chat
 - jednorazowe polecenie / **one shot command** — `claude -p "zrób X"`
 - potok — pipes — `cat plik.tex | claude "popraw styl"`

TEX

+ AI CLI



Dlaczego T_EX + AI CLI? / Why T_EX + AI CLI?

- **Czysty tekst** — pliki `.tex` to tekst, nie binarki / Plain text — `.tex` files are text, not binaries
łatwy do czytania, edycji, pipe'owania, diffowania / easy to read, edit, pipe, diff
- **AI zna T_EX-a** — ogromny korpus w danych treningowych / AI knows T_EX — huge corpus in training data
PlainT_EX, L^AT_EX, BibT_EX, Beamer, TikZ...
- **Formalna składnia** — AI generuje poprawny kod, nie zgaduje / Formal syntax — AI generates valid code, not guesses
- **CLI-native** — `latexmk`, `make`, tekstowe logi / CLI-native — *latexmk*, *make*, logs as text
kompilacja, błędy i poprawki za jednym zamachem / compile, errors, and fixes in one pipeline

$\text{\TeX}$ + AI CLI — generowanie makr / macro generation

Przykładowe polecenie / Example commands:

```
claude "napisz makro \RAMKA, które otacza tekst kolorową ramką z tytułem"
```

- Model rozumie składnię $\text{\TeX}$ / Model understands $\text{\TeX}$ syntax
- Generuje, reaguje na błędy kompilacji / Generates, reacts for compilation errors
- Zna dialekty – Plain $\text{\TeX}$, $\text{\LaTeX}$, Con $\text{\TeX}$ t / Understands dialects of $\text{\TeX}$
- Wyjaśnia co i dlaczego / Explains what and why
- Pułapki: halucynacje nazw pakietów, stare API / Pitfalls: packet name hallucination, outdated API

T_EX+ AI CLI — debugging

Wklejam log z błędem / Paste the log with errors:

```
cat compile.log | claude "wyjaśnij błędy i zaproponuj poprawki"
```

- Logi T_EX-a bywają enigmatyczne — AI czyta je sprawnie / T_EX logs are messy — no problem for AI
- `Undefined control sequence, issuing $ inserted`
- Wskazuje **prawdziwe źródło** błędu, nie tylko objaw / Finds the **real reason** not only symptom
- Pomaga z **trudnymi przypadkami**: kategorie znaków, grupy, rozwijanie makr w nieoczekiwanym momencie / Helps in **hard cases**: catcodes, grouping, macro expansion
- Daj **cały plik** jako kontekst / Give **whole file** as context
→ `claude --add-file main.tex "dlaczego \headheight za małe?"`

T_EX+ AI CLI — praca z tekstem

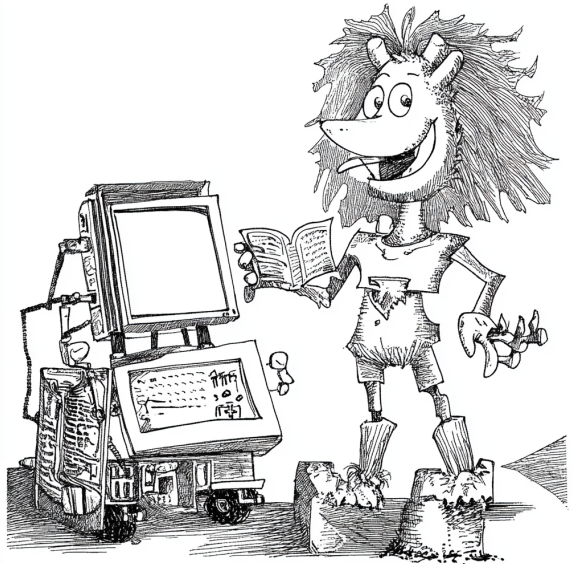
- Korekta językowa z zachowaniem znaczników / **Proofreading** without macro touching

```
claude "popraw literówki, nie ruszaj makr"
```

- **Przekształcenia struktury:** z artykuły na slajdy, z notatek na artykuł / **Structure transformation:** article to slides, notes to article
- Tłumaczenia, streszczenia, słowa kluczowe / **Translations, abstracts, keywords**
- **Zarządzanie bibliografią** – (nie tylko BibT_EX) / **Bibliography handling** – (not only BibT_EX)

Nie tylko T_EX

Not only T_EX



git + AI CLI

- Tworzenie i zarządzanie repozytorium / Repo creating and managing
- Commity, wersje i logi / Commits, versions and logs
- Wspieranie obsługi gałęzi i merge / Branch and merge support
- Zarządzanie zmianami / Change management
- Wspieranie dokumentowania / Documentation supporting
- Przeglądanie i analiza kodu / Code review and analysis
- Wsparcie „w tle” / “Background” support

make + AI CLI

- **Makefile jako klej** — łączy kompilację, uruchamianie skryptów, czyszczenie /
Makefile as glue — compilation, scripts, cleanup
- **AI generuje reguły i wyjaśnia zależności** / AI writes rules and explains dependencies

```
claude "dodaj regułę, która konwertuje wszystkie SVG do PDF"
```

- **Największa korzyść:** standardowy sposób używania generowanego kodu bez pamiętania szczegółów / **Key benefit:** standard you use **generated code** without knowing the details
make compile, make run, make clean — jeden interfejs do wszystkiego / one interface for everything

python + docker + AI CLI

- **Pomocne skrypty:** konwersje plików, przetwarzanie tekstu, parsowanie logów, przetwarzanie grafiki i filmów / Helper scripts: file conversion, text processing, log parsing, image and video manipulation
- **Zaawansowane programy:** korzystanie z API, automatyzacja, przetwarzanie danych / Advanced programs: API usage, automation, data processing
AI zna biblioteki, generuje testy, refaktoryzuje / AI knows libraries, generates tests, refactors
- **Docker = izolacja i przenośność** / Docker = isolation and portability
Zapewnia zgodność wersji pakietów i zależności / Ensures package version and dependency consistency
Działa na każdej maszynie bez dodatkowej konfiguracji / Runs anywhere without extra setup

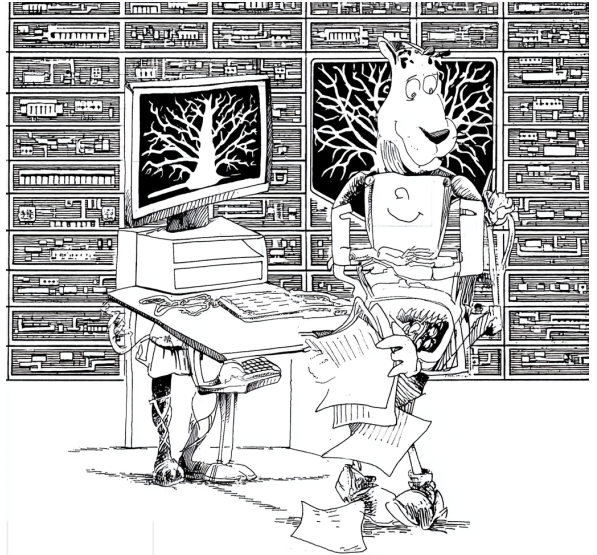


html + js + AI CLI

- Szybkie generowanie interaktywnych narzędzi z GUI / Quick generation of interactive GUI helper tools
formularze, wizualizacje, kalkulatory, edytory — bez frameworka / forms, visualizations, calculators, editors — no framework needed
- Uruchamianie bez instalacji / Serving without installation:
 - `python -m http.server 8080` (statyczne pliki / static files)
 - `flask run` (z logiką w Pythonie / with logic in Python backend)
 - `docker run -p 8080:80 ...` (izolacja, przenośność)

**Plusy, minusy
i problemy**

**Pros, cons
and pitfalls**



Mocne strony AI CLI / Strengths

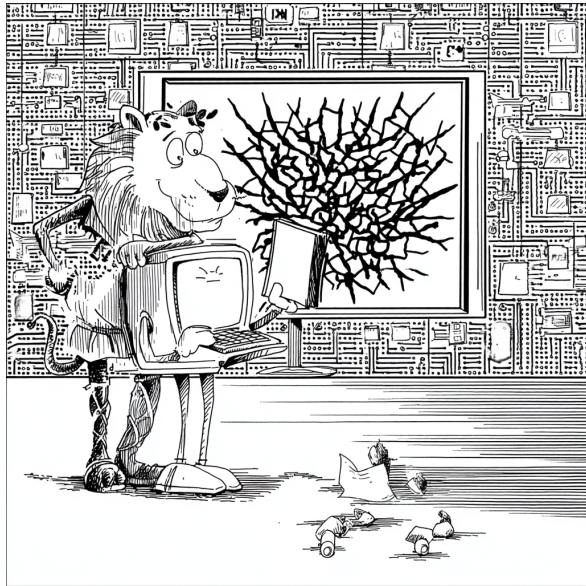
- **Kontekst projektu** — model ma dostęp do plików, nie trzeba kopiować / Project context — model has file access, no copy-paste
- **Znajomość poleceń** — pipe, xargs, skrypty shellowe / Shell fluency — pipes, xargs, shell scripts
- **Automatyzacja powtarzalnych zadań** — szablony, konwersje / Automation of repetitive tasks — templates, conversions
- **Niski koszt „zabawy”** — szybkie sprawdzenie pomysłów / Low cost of “fiddling” — quick testing of ideas
- **Metoda na uczenie się** — prośby o wyjaśnienia / Learning method — ask for explanation
- **Praca z wieloma/dużymi plikami** (duży kontekst) / Working with **multiple/long files** (large context window)
- **Refaktoryzacja przy zmieniających się wymaganiach** / Refactoring under changing requirements

Pułapki i ograniczenia / Pitfalls and limits

- **Halucynacje** — nieistniejące pakiety, parametry, API / Hallucinations — nonexistent packages, parameters, APIs
→ zawsze warto sprawdzić; nie ufaj ślepo / always verify; do not trust blindly
- **Wiedza ma datę** — nowe wersje narzędzi mogą być nieznane / Knowledge has a cutoff — new tool versions may be unknown
- **Koszty** — tokeny kosztują; zwłaszcza duże konteksty / Cost — tokens aren't free; especially large contexts
- **Bezpieczeństwo** — co wklejasz do modelu? dane wrażliwe? / Privacy — what are you pasting into the model?
- **Uzależnienie i brak rozumienia kodu** / Over-reliance — easy to lose understanding of the code — (Vibe coding)

Pokaz na żywo

Demo live



Szczegółowa wiedza o konstrukcji wewnętrznej **wzmacniacza inteligencji** nie jest nam dostępna. Być może jest ona całkiem zbędna. Być może wystarczy potraktować ów wzmacniacz jako „czarną skrzynkę”, jako urządzenie, o którego planie wewnętrznym i kolejnych stanach nie mamy najbłedszego wyobrażenia, natomiast interesować nas będą wyłącznie końcowe rezultaty działania. Wzmacniacz ów posiada, jak każde szanujące się urządzenie cybernetyczne, „wejścia” i „wyjścia”. Pomiędzy nimi rozpościera się strefa naszej ignorancji, ale co to szkodzi, jeżeli tylko będzie ta maszyna rzeczywiście zachowywać się jak intelekt o ilorazie inteligencji rzędu 10 000?

Stanisław Lem/Summa technologiae/1964

Detailed knowledge of the internal construction of an **intelligence amplifier** is not available to us. Perhaps it is entirely unnecessary. Perhaps it is sufficient to treat this amplifier as a “black box”, a device about whose internal design and successive states we have not the faintest idea, while what interests us are solely the final results of its operation. This amplifier possesses, like any self-respecting cybernetic device, “inputs” and “outputs.” Between them stretches a zone of our ignorance, but what does that matter, as long as the machine truly behaves like an intellect with an IQ on the order of 10,000?

**Dziękuję
za uwagę**

**Thank you
for your attention**

