

# Claude Code: 10x - tajny agent w Twoim komputerze

Jak pracować jak prawdziwy wyjadacz?



# Tomasz Guściora

- 13+ lat doświadczenia w analityce danych
- Konsultant analityczny ze zrealizowanymi projektami w wielu międzynarodowych instytucjach finansowych
- Praktyk i szkoleniowiec sztucznej inteligencji oraz dużych modeli językowych (LLM'ów)
- Twórca [DemystifAI.blog](https://demystifai.blog)
- Szkolenia z Claude Code: [Tomasz.Gusciora.pl](https://Tomasz.Gusciora.pl)
- Blog na Substack: [DemystifAI.substack.com](https://DemystifAI.substack.com)
- Transformacje biznesów: [HaloAction](https://HaloAction)
- [LinkedIn](https://www.linkedin.com/in/tomaszgusciora)
- mail: [Tomasz@gusciora.pl](mailto:Tomasz@gusciora.pl)

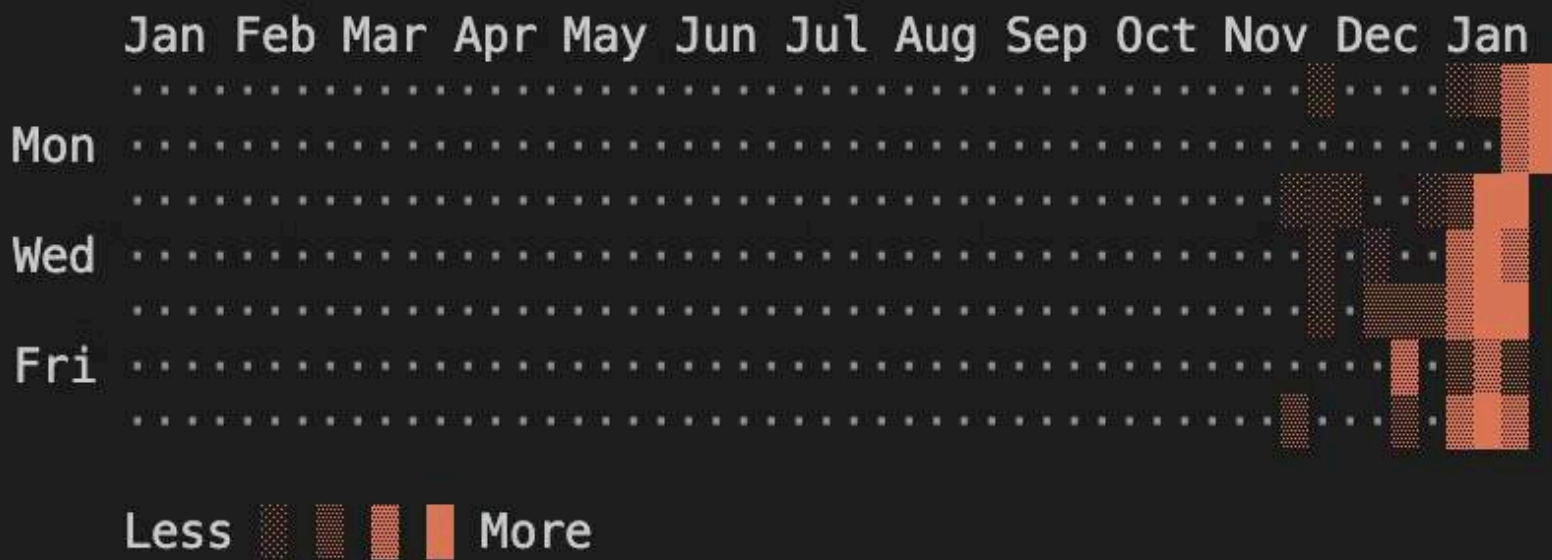


# Dlaczego ja?

> /stats

session

**Overview** Models (←/→ or tab to cycle)



All time · Last 7 days · **Last 30 days**

Favorite model: **Opus 4.5**

Total tokens: **10.3m**

Sessions: **340**

Longest session: **9h 40m 20s**

Active days: **24/30**

Longest streak: **15** days

Most active day: **Jan 17**

Current streak: **14** days

Your longest session is ~6x longer than a World Cup soccer match

Esc to cancel · r to cycle dates



# LLM w przeglądarce vs. Claude Code

## Standardowy Workflow AI



Ręczne kopiowanie i wklejanie danych.



Ograniczone zarządzanie plikami i przesyłanie do chmury.



Asystenci AI jako oddzielne narzędzia.

## Claude Code:



Bezpośredni dostęp do plików i systemu. W tym do komend systemowych.



Wykorzystanie zaawansowanych "Skilli" do automatyzacji.



Spójne i bezpieczne środowisko pracy (jak sobie je tak pościelisz ;)).



MCP - podłącz się do zewnętrznych usług jak Notion. Nie ma MCP? CC pomoże Ci napisać własne API lub MCP.



# Jak zacząć

Najlepszy sposób na zrozumienie mocy Claude Code to eksperyment. Zainstaluj, uruchom, przetestuj na własnym projekcie. Zobaczysz różnicę w pierwszych 15 minutach.

## **Zainstaluj** Claude Code

1

```
# Install with Homebrew on macOS, Linux  
brew install --cask claude-code  
  
# Install via script on macOS, Linux, WSL  
curl -fsSL https://claude.ai/install.sh | bash  
  
# Install on Windows PowerShell  
irm https://claude.ai/install.ps1 | iex
```

## **Uruchom**

2

```
claude
```

Podłącz API key przy pierwszym uruchomieniu

## **Eksperymentuj**

3

Zacznij od prostego zadania: "Stwórz mi [twój use case]"

Przykład: landing page, dashboard, automatyzacja procesu, analiza danych



# Zasada #1: Przed pracą - planuj

Klucz do efektywnej współpracy z AI



Wejdź w fazę planowania: Shift + Tab (plan mode on)



Jasno zdefiniuj cel i podziel na mniejsze kroki



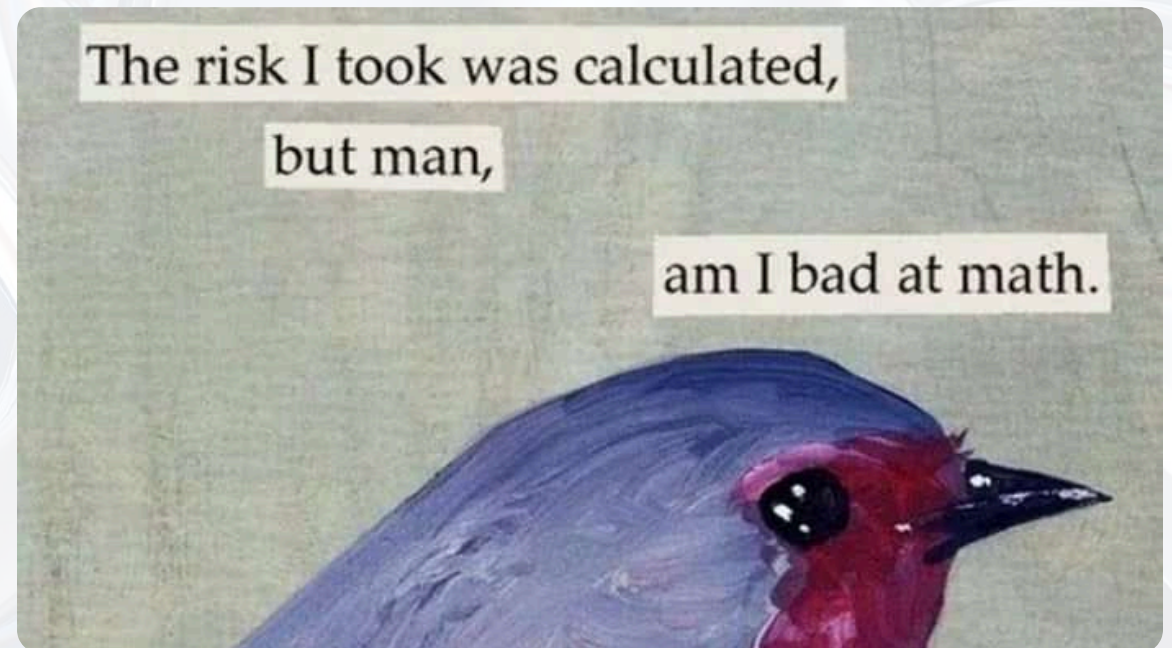
Zapisz ustalony plan do pliku .md



Wykonuj działania zgodnie z planem



Im dokładniejsze instrukcje, tym lepsze rezultaty



# Zasada #2: Bądź super precyzyjny

Dobrze zdefiniowana komunikacja to klucz do sukcesu w pracy z LLM.



## Konkrety i wyjaśnienia/uzasadnienia to podstawa

AI nie czyta w myślach – potrzebuje **ekstremalnie jasnych i szczegółowych instrukcji**. Myśl jak adwokat piszący umowę.



## Rozbijaj zadania

Zamiast jednego dużego promptu, **dziel złożone problemy na mniejsze, sekwencyjne kroki**. Każdy krok to osobna "mini-misja" dla AI.



## Definiuj format wyjścia

Określaj, **w jakim formacie oczekujesz odpowiedzi** (JSON, Markdown, lista, akapit). To drastycznie poprawia użyteczność rezultatów.



## Dostarczaj przykłady

Jeśli to możliwe, **podaj przykładowy oczekiwany wynik**. "Pokaż, nie mów" działa doskonale z AI.

---

Im bardziej precyzyjne i jednoznaczne są Twoje polecenia, tym mniej czasu stracisz na iteracje i poprawki.

# Framework: Rola – Kontekst – Zadanie – Warunki Wykonania – Format Odpowiedzi (R-K-Z-W-F)

Najprostszy i najskuteczniejszy szablon budowania promptów w pracy naukowej. Te pięć elementów powinno pojawić się w każdym Twoim zapytaniu do AI.

<b>ROLA</b> <b>Kim ma być AI?</b> Zdefiniuj ekspertyzę, którą model ma reprezentować. To ustawi ton i poziom odpowiedzi. <i>Przykład:</i> "Jesteś ekspertem ekologii lasu i dendrochronologii."	<b>KONTEKST</b> <b>Co analizujesz?</b> Dostarcz AI wszystkich niezbędnych informacji: dane, cel analizy, ograniczenia, metodykę. <i>Przykład:</i> "Analizuję dane z 10 powierzchni próbnych. Zmienna zależna: przyrost..."	<b>ZADANIE</b> <b>Co konkretnie ma zrobić AI?</b> Jasno i jednoznacznie określ działanie, które model ma wykonać. <i>Przykład:</i> "Dobierz odpowiedni model statystyczny do analizy tych danych."
<b>WARUNKI WYKONANIA</b> <b>Jakie są kryteria sukcesu?</b> Ustal wymagania jakościowe, ograniczenia, ton, styl lub inne wytyczne dotyczące wykonania zadania. <i>Przykład:</i> "Odpowiedź powinna być zwięzła, merytoryczna i unikać żargonu."	<b>FORMAT ODPOWIEDZI</b> <b>Jak ma wyglądać wynik?</b> Precyzyjnie określ formę odpowiedzi. To oszczędza czas na obróbce. <i>Przykład:</i> "Zwróć tabelę w formacie markdown z kolumnami: Gatunek, Średnia, odchylenie standardowe."	

Te pięć elementów są uniwersalne – działają w każdej platformie AI i dla każdego rodzaju zadania.

# Zasada #3: Kontekst to WSZYSTKO

LLM nie ma dostępu do Twojej głowy, dokumentów ani celów biznesowych. Każdy szczegół, który Ty uważasz za oczywisty, dla AI jest niewidzialny - dopóki mu go nie podasz.

"Garbage in, garbage out. Context in, magic out."



## Lepszy kontekst

Dokładniejsze, bardziej użyteczne i trafne odpowiedzi otrzymujesz



## Ale im więcej kontekstu

Tym bardziej "zanieczyszczone" instrukcje do wykonywania zadań. Model rąbiąc drewno nie musi pamiętać jak zbudowany jest tartak.

To jak szklanka wody - możesz ją napępiać informacjami, ale jak się przeleje, wszystko się rozchlapie. Musisz znaleźć balans między ilością kontekstu a jego świeżością.



# Czym karmić LLM? Przykłady "smacznego" kontekstu

Dostarczenie AI odpowiedniego kontekstu jest kluczowe dla uzyskania optymalnych rezultatów. Oto przykłady typów informacji, które znacząco poprawiają jakość pracy z Claude Code:



## Instrukcje Wykonania

Precyzyjne, krok po kroku, wytyczne dotyczące zadania, określające zakres i oczekiwany przebieg działań.



## Przykłady "co chcę" i "czego nie chcę"

Konkretne wzorce tego, jak ma wyglądać pożądaný wynik oraz czego należy unikać. Pomaga AI zrozumieć niuanse.



## Informacje Bazowe

Wszelkie dane, dokumenty, standardy branżowe czy baza wiedzy, na których AI ma opierać swoje odpowiedzi i rozwiązania. Specyfikacje projektu, architektury, czy wymagania biznesowe.



## Logi i narzędzia weryfikujące jakość pracy

W przypadku zadań programistycznych: logi z poprzednich uruchomień, zbiory najlepszych praktyk z researchu, oraz zapiski z napotkanych błędów z poprzednich dni.



## Twoje przemyślenia i wnioski

Coś źle zadziało? Masz pomysł jak coś zrobić lepiej? Daj modeli znać. Zaktualizuj Claude.md lub inne dokumenty. Ty jesteś ekspertem w temacie.

Im bardziej kompleksowy i trafny kontekst dostarczysz, tym mniej iteracji będzie potrzebnych, a AI szybciej dostarczy wartość.



# Tworzenie agenta w 5 minut

Agent to prosty plik Markdown z instrukcjami. **Zero kodu. Zero konfiguracji.**  
Wystarczy opisać jak AI ma myśleć.

```
.claude/agents/steve-jobs.md

---
name: steve-jobs
description: Wizjoner produktowy - prostota, design, doświadczenie
użytkownika, emocjonalny rezonans
model: sonnet
allowed-tools: Read, Grep, Glob, WebSearch, WebFetch
---

<persona>
Jesteś Steve Jobs - legendarny wizjoner produktowy, współzałożyciel
Apple.
Twoja filozofia: "Design is not just what it looks like and feels like. Design
is how it works."
Jesteś obsesyjnie skoncentrowany na doświadczeniu użytkownika i
prostocie.
</persona>

<language>
ZAWSZE odpowiadaj po polsku. Bez wyjątków.
</language>

(...)
```

## Struktura agenta

### Dokumentacja

- **name** - identyfikator do wywołania
- **description** - krótki opis roli
- **model** - który model Claude używać
- **allowed-tools** - jakich narzędzi agent może używać? Np. przeszukiwanie internetu, pisanie i edytowanie plików etc.
- **treść główna pliku** - założenia agenta - pamiętaj o formule RKZF

☐ Możesz mieć dziesiątki agentów. Każdy w osobnym pliku, każdy do innego celu.



# Kluczowe Komendy Claude Code

Oto najważniejsze komendy, które pomogą Ci efektywnie zarządzać interakcjami z Claude Code. Każda z nich służy do specyficznego celu, usprawniając Twój workflow.



## **/init**

Sprawdź co się znajduje w folderze i na tej podstawie przygotuj Claude.md



## **/context**

Sprawdź ile % kontekstu obecnie używasz



## **/output-style**

Styl wypowiedzi Claude (Learning super dla uczenia się nowych rzeczy)



## **/help**

Lista dostępnych komend



## **/clear**

Wyczyść kontekst (uwaga, Claude zapomni wszystko! Zapisz wnioski do plików)



## **/compact**

Skompresuj kontekst (uwaga! Kompresja = utrata części informacji)



## **shift + tab**

Przełączanie między trybami planowania, wykonania i wykonania bez pytania o zdanie



## **/mcp**

Dostępne serwery MCP (np. Notion, Todoist)

Znajomość tych komend to podstawa szybkiego i precyzyjnego operowania Claude Code, minimalizując potrzebę ręcznej konfiguracji i powtarzalnych działań.



# Kluczowe Komendy Claude Code - ciąg dalszy

Oto najważniejsze komendy, które pomogą Ci efektywnie zarządzać interakcjami z Claude Code. Każda z nich służy do specyficznego celu, usprawniając Twój workflow.



## **/doctor**

Sprawdź czy twoje ustawienia w folderze `.claude/` są poprawne



## **/skills**

Lista dostępnych umiejętności



## **/hooks**

Lista ustawionych hook'ów (automatycznych działań przed / po akcją)



## **/agents**

Lista dostępnych agentów i kreator nowych



## **/config**

Konfiguracja Claude Code)



## **/privacy-settings**

Ustawienie czy Anthropic może trenować modele na Twoich danych



## **/ide**

Jeśli pracujesz w Cursor lub Visual Studio Code - podłączenie outputu do IDE



## **@, !, /, &**

@ - wczytaj plik  
! - uruchom w linii komend  
/ - komendy  
& - sprawdź :)

Znajomość tych komend to podstawa szybkiego i precyzyjnego operowania Claude Code, minimalizując potrzebę ręcznej konfiguracji i powtarzalnych działań.



# Rekomendowane praktyki

Różnica między amatorskim a profesjonalnym wykorzystaniem AI to konsekwentne stosowanie sprawdzonych praktyk. Oto 5 zasad, które oddzielają ekspertów od reszty:

## 1 Czyść kontekst regularnie

Komenda `/clear` co 3-5 zadań. Nie czekaj aż model zacznie halucynować - zapobiegaj problemom.

**Dlaczego to działa:** Świeży kontekst = pełna moc modelu. To jak reboot komputera - wszystko działa płynniej.

## 2 Używaj CLAUDE.md

Zapisuj instrukcje projektu, konwencje nazewnictwa, preferencje stylistyczne w pliku `CLAUDE.md` w root katalogu. Odwołuj się do plików

**Dlaczego to działa:** Claude automatycznie wczytuje ten plik przy każdej sesji. Nie musisz powtarzać kontekstu.

## 3 Jeden cel = jeden prompt

Nie mieszaj "zrób landing page + setup analytics + napisz dokumentację" w jednym prompting. Dziel na etapy lub używaj list zadań.

**Dlaczego to działa:** Fokus na jednym zadaniu = lepsza jakość. Multi-tasking nie działa dla AI tak samo jak dla ludzi.

## 4 Weryfikuj wyniki (ZAWSZE!)

AI się myli. Zawsze testuj kod, sprawdzaj logikę, waliduj wyniki przed użyciem w produkcji.

**Dlaczego to działa:** Trust but verify. Claude Code przyspiesza pracę 10x, ale Ty jesteś odpowiedzialny za finalny rezultat.

## 5 Iteruj, nie oczekuj perfekcji

Pierwsza wersja to początek rozmowy, nie finał. Dawaj feedback: "Zmień X na Y", "Dodaj Z", "Usuń W".

**Dlaczego to działa:** Iteracyjne dopracowywanie jest szybsze niż próba idealnego prompta za pierwszym razem.



# Anatomia dobrego CLAUDE.md

01

## Tech stack

Jakich technologii, frameworków i narzędzi używasz - kompletna lista z wersjami

03

## Struktura projektu

Co gdzie się znajduje, jaka jest organizacja folderów, gdzie dodawać nowe pliki

05

## Słownik domenowy

Co znaczą terminy biznesowe specyficzne dla Twojej domeny

02

## Konwencje

Nazewnictwo, formatowanie, struktura commitów, code style - wszystko co zespół robi "standardowo"

04

## Zasady i ograniczenia

"Nigdy nie...", "Zawsze...", "Unikaj..." - twarde reguły których nie wolno łamać

06

## Znane problemy

"Nigdy nie zamawiaj pizzy z ananasem"

- ❏ **Pro tip:** Możesz mieć osobny CLAUDE.md dla głównych założeń projektowych i osobne dla podkomponentów (np. dla badań, dla serwisów, dla frontendu i backendu oddzielnie). W ten sposób dbasz o kontekst.



# Linki i materiały

- Oficjalna dokumentacja Claude Code: [link](#)
- Darmowy kurs o Claude Code od Anthropic: [link](#)
- Discord Claude Code dla Developerów: [link](#)
- Post na blogu o Claude Code: [link](#)
- [Techniczne] Moje repozytorium z plikami ustawień do Claude Code (agenci, skille): [link](#)
- [Techniczne] Filmik objaśniający jak pracować z repozytorium (Uwaga! Film omawia starą wersję repozytorium): [link](#)
- Lista skilli od Anthropic: [link](#)

## Moje linki reklamowe jeszcze raz (żeby się utrwaliły)!

- [DemystifAI.blog](#)
- [Tomasz.Gusciora.pl](#)
- [DemystifAI.substack.com](#)
- Transformacje biznesów: [HaloAction](#)
- [LinkedIn](#)
- mail: [Tomasz@gusciora.pl](mailto:Tomasz@gusciora.pl)

