

Prompt engineering – jak pisać skuteczne prompty dla LLM

Admin Główny 11.10.2026 447 wyświetleń 0 komentarzy

Dostępne w językach:

PL

EN

[Start](#) / [Blog](#) / [Sztuczna Inteligencja](#)

/ Prompt engineering – jak pisać skuteczne prompty dla LLM

[← Wróć do bloga](#)



Pobierz PDF

Czym jest prompt engineering?

Prompt engineering to sztuka formułowania zapytań do modeli językowych tak, żeby dostać przewidywalne, dokładne i użyteczne wyniki.

Dobry prompt = jasny kontekst + konkretne zadanie + format odpowiedzi.

Złe vs dobre prompty

✗ Złe — zbyt ogólne:

Napisz kod PHP.

✓ Dobrze — konkretne:

Napisz funkcję PHP 8.3, która:

- przyjmuje tablicę stringów
- filtruje elementy dłuższe niż 5 znaków
- zwraca nową tablicę z wynikami
- używa `array_filter` i `fn()`

Dodaj przykład użycia z `var_dump()`.

Techniki

1. System prompt — nadaj rolę

Jesteś senior PHP developerem z 10-letnim doświadczeniem. Odpowiadaj zwięźle. Kod pisz zgodnie z PSR-12. Jeśli pytanie jest niejasne, dopytaj przed odpowiedzią.

2. Few-shot — pokaż przykłady

Zamień tytuły na slugi:

"Hello World"	→ "hello-world"
"Jak się masz?"	→ "jak-sie-masz"
"PHP 8.3 Tutorial"	→ "php-8-3-tutorial"
"Zażółć gęślą jaźń"	→ ?

3. Chain of thought — myśl krok po kroku

Rozwiąż ten problem wydajnościowy krok po kroku:

1. Opisz co robi ten kod
2. Zidentyfikuj wąskie gardło
3. Zaproponuj optymalizację
4. Pokaż poprawiony kod

Kod:

[wklej kod]

4. Output format – określ format odpowiedzi

Odpowiedz w formacie JSON:

```
{
  "problem": "...",
  "solution": "...",
  "code": "..."
}
```

Narzędzia

Narzędzie	Zastosowanie
Anthropic Console	Testowanie promptów, porównywanie modeli
Claude Code	Asystent kodowania w terminalu
GitHub Copilot	Uzupełnianie kodu w IDE
LangChain	Budowanie aplikacji z LLM

Prompt dla code review

Przeanalizuj poniższy kod PHP jako senior developer:

Szukaj:

- błędów bezpieczeństwa (SQL injection, XSS, CSRF)
- problemów z wydajnością
- naruszeń zasad SOLID
- braku obsługi błędów

Format:

Krytyczne

- [opis problemu] → [sugerowana poprawka]

Ważne

- ...

Sugestie

- ...

Kod:

```
```php
```

```
[twój kod]
```

---

### **## Ograniczenia LLM**

- **\*\*halucynacje\*\*** – model może zmyślać fakty; zawsze
- **\*\*odcięcie wiedzy\*\*** – model nie zna najnowszych b
- **\*\*kontekst\*\*** – długie konwersacje mogą „gubić” wc
- **\*\*deterministyczność\*\*** – ten sam prompt może dać

🔖 Tagi:

#Python

#AI

🔗 Udostępnij:



Facebook



X



LinkedIn



Kopiuj link

## Komentarze (0)

---

Bądź pierwszy — zostaw komentarz!

[Zaloguj się](#), aby dodać komentarz.

## FactoryCode

Projekt demonstracyjny zbudowany na Symfony 8, PHP 8.4 i PostgreSQL 16.  
Tłumaczenia zarządzane przez panel admina z AI.

### NAWIGACJA

-  Start
-  Blog
-  Projekty
-  Kontakt
-  Polityka prywatności

### KONTAKT

-  [kontakt@factorycode.pl](mailto:kontakt@factorycode.pl)
-  [GitHub](#)