

# PROMPT ENGINEERING: BEGINNER-FRIENDLY GUIDE FOR WEB DEVELOPERS

Presented by Sewak Dhakal



# 💡 First, What Is Prompt Engineering and why you need it as a developer?

---

Prompt engineering is the skill of writing clear, structured, and optimized instructions to get the best responses from AI models like me (ChatGPT, Claude, Gemini, etc.).

If you know how to ask, you can unlock code generation, testing, debugging, UI design help, content creation, even documentation—and much more.

As a web developer, this skill gives you:

🚀 **Speed:** AI writes skeleton code, snippets, even full components.

💡 **Ideas:** Stuck? Ask for suggestions.

🧠 **Learning:** Treat the AI like a tutor.

🔧 **Tools:** Build tools with AI and using AI.

# THE 3 CORE PRINCIPLES (AS A DEV)

## 1. Be Specific — Clarity Is Code

✗ “Make a navbar”

✓ “Create a responsive navbar using Flexbox that includes a logo on the left and 3 links on the right. Hide the links and show a hamburger menu on screens <768px.”

✓ Tip: Use HTML/CSS/JS/React explicitly. Don’t make AI guess what stack you’re using.

## **2. Break It Down — Think Like a Dev, Not a Writer**

**You don't just write "Build an app."**

**Instead, structure like this:**

**"I'm building a React-based portfolio site. Here's the context:**

- I want a hero section with text and animated background**
- Then a projects section showing 3 cards**
- Then a contact form using Web3Forms.**

**Give me code for the hero section first, using Tailwind CSS."**

**✅ This is how you speak AI's language as a developer. Context first, then request.**

### 3. Iterate Like a Developer

Prompt engineering is not a “one-shot” game. It’s an iterative workflow, like debugging code.

Think in cycles:

-  Ask →  Get output →  Test it →  Tweak prompt →  Repeat

# MUST-LEARN PROMPT TEMPLATES (TAILORED FOR YOU)

## ◆ Code Helper

“I’m trying to [do X] using [Y tech]. Here’s my code: [paste code]. Why isn’t it working and how can I fix it?”

## ◆ Design with Dev in Mind

“Generate a mobile-first responsive layout using HTML and Tailwind CSS for a product card with image, title, description, price, and ‘Add to Cart’ button.”

### ◆ Debugging Partner

**“This JavaScript isn’t behaving as expected. I expected [behavior], but instead I get [error/output]. What’s wrong with my logic?”**

### ◆ Learning Mode

**“Teach me how event delegation works in JavaScript. Keep it beginner-friendly and show one real-world example.”**

### ◆ Documentation Generator

**“Write developer documentation for this function/component. Explain what it does, its parameters, return value, and example usage.”**

# TOP USE CASES FOR PROMPT ENGINEERING (AS A WEB DEV)

Use Case	Example Prompt
Code Generation	"Write a React component that renders a list of blog posts with title, excerpt, and read more link."
Bug Fixing	"This fetch call doesn't return the data I expect. Here's the code and error."
CSS Styling	"Style this login form using Tailwind. Make it centered, with spacing, and a glowing button on hover."
Documentation	"Document this JS function that handles form validation."
Learn Concepts	"Explain how closures work in JS with real world analogy."
Git Help	"Write a <code>.gitignore</code> file for a React + Node.js project."



# TOOLS FOR PROMPT ENGINEERING

To supercharge your prompt engineering, these are must-haves:

Tool	Use
ChatGPT (GPT-4)	General coding, troubleshooting, design ideas
Cursor IDE	AI pair-programmer right in your VS Code
Github Copilot	Auto-completion powered by prompt tuning
Claude / Gemini	Long context + fast ideation
AI-powered Docs	Ask: "How does <code>useEffect</code> work in React?" on MDN, DevDocs, etc.

# TIPS FOR BETTER PROMPTS

- Give context first (e.g. 'I'm making a React site')
- Be clear about the technology (HTML, JS, React, etc.)
- Ask one question at a time
- Always test the AI's answer yourself

# **PRO TIP: TALK LIKE A DEVELOPER, NOT A USER**

**The more your prompt sounds like a developer's train of thought, the better the AI will perform. Think out loud in code logic.**

**Instead of:**

**“How to make my website nice?”**

**Say:**

**“Suggest layout and animation ideas for a responsive tech blog homepage built with Tailwind CSS and Next.js.”**

# PRO TIP: TALK LIKE A DEVELOPER, NOT A USER

 "Create a component that..."

 "Explain this bug/error: ..."

 "Style this using Tailwind to look modern/clean/neumorphic/retro..."

 "Refactor this layout to be responsive and mobile-first"

 "Generate test cases for this function"

 "Build a reusable form component with validation"

 "Summarize this code and what it does"

 "Turn this into a React/Vanilla JS reusable module"



**FINAL ADVICE:**

# **PROMPT ENGINEERING IS A DEV SKILL NOW**

**In 2025 and beyond, AI isn't optional. You don't have to be an AI engineer—but you MUST know how to speak to AI.**

**You already learned HTML, CSS, JS.**

**Now add prompt engineering as your 4th language—because it's your shortcut to mastering the first three.**

**Keep practicing, keep building, and use AI as your coding partner.**