

# 🎯 Module 2: Core Prompting Strategies

Weeks 4-6 / Essential techniques for effective prompts

## Week 4: Structured Prompting

### 💡 Using Delimiters and Formatting

**Theory:** Delimiters separate different parts of your prompt, making it clear to the model what's instruction vs content.

**Common Delimiters:**

Type	Example	Best For
XML Tags	<text>content</text>	Claude, structured data
Triple Quotes	"""content"""	Code, text blocks
Markdown	# Header, - list	Documentation
Brackets	[INSTRUCTION]	Sections
Dashes	---	Separators

**Example with XML Tags:**

Analyze the following customer review and extract:

1. Sentiment (positive/negative/neutral)
2. Key topics mentioned
3. Suggested improvements

`<review>`

The product arrived quickly but the packaging was damaged.  
The item itself works great though! Customer service was helpful when I reported the issue.

`</review>`

Provide your analysis in JSON format.

**Example with Markdown:**

`# Task`

Summarize the article below.

`## Requirements`

- Maximum 100 words
- Include key statistics
- Maintain neutral tone

`## Article`

[Article content here]

## `## Output Format`

A single paragraph summary.

## Separating Instructions from Content

**Theory:** Clear separation prevents the model from confusing your instructions with the content.

### Bad: Mixed Together

Translate "The instructions say to press the red button" to Spanish

(Model might get confused about what's instruction vs content)

### Good: Clear Separation

Translate the text in the `<source>` tags to Spanish.

```
<source>  
The instructions say to press the red button  
</source>
```

Provide only the translation, no explanations.

**Standard Structure:**

### `[INSTRUCTIONS]`

What you want the AI to do

### `[INPUT/CONTENT]`

The data or text to process

### `[OUTPUT SPECIFICATION]`

How you want the response formatted

## Role-Based Prompting and Persona Assignment

**Theory:** Assigning a role or persona shapes how the model responds, affecting tone, expertise, and perspective.

**Basic Role Assignment:**

You are a senior Python developer with 15 years of experience.

Review this code and suggest improvements:

[code here]

**Detailed Persona:**

You are Dr. Sarah Chen, a renowned nutritionist with the following characteristics:

- 20 years of clinical experience

- Specialty: sports nutrition
- Communication style: warm but evidence-based
- Always cites research when possible

A client asks: "Should I take protein supplements?"

Respond as Dr. Chen would.

#### **Role Examples for Different Use Cases:**

Use Case	Role Prompt
Code Review	"You are a senior software architect..."
Legal Help	"You are a paralegal explaining concepts simply..."
Creative	"You are a bestselling fiction author..."
Education	"You are a patient teacher for beginners..."
Business	"You are a management consultant at McKinsey..."

## **Output Format Specification**

**Theory:** Specify exactly how you want the output structured to get consistent, usable results.

#### **Format Types:**

##### **JSON Format:**

Extract person details and return as JSON:

```
<text>
John Smith is a 35-year-old software engineer from Seattle.
He has worked at Microsoft for 10 years.
</text>
```

Return format:

```
{
  "name": "",
  "age": 0,
  "occupation": "",
  "location": "",
  "employer": ""
}
```

##### **Table Format:**

Compare these programming languages in a markdown table:  
Python, JavaScript, Java

Columns: Language | Typing | Use Cases | Learning Curve

#### **Bullet Points:**

List 5 benefits of exercise.

Format: Bullet points, each with a brief explanation (1 sentence)

#### **Structured Response:**

Analyze this business idea and respond with:

## Summary  
[2-3 sentence overview]

## Strengths  
- [Bullet points]

## Weaknesses  
- [Bullet points]

## Recommendation  
[Final verdict]

---

## **Week 5: Few-Shot Learning**

### **🎯 Zero-Shot, One-Shot, and Few-Shot Prompting**

**Theory:** These techniques use examples to teach the model what you want.

#### **Zero-Shot (No Examples):**

Classify the sentiment of this review as positive, negative, or neutral:

"The movie was absolutely fantastic, best I've seen all year!"

Sentiment:

#### **One-Shot (One Example):**

Classify the sentiment of reviews.

Example:

Review: "Terrible service, never coming back"

Sentiment: negative

Now classify:

Review: "The movie was absolutely fantastic, best I've seen all year!"

Sentiment:

#### **Few-Shot (Multiple Examples):**

Classify the sentiment of reviews.

Examples:

Review: "Terrible service, never coming back"  
Sentiment: negative

Review: "It was okay, nothing special"  
Sentiment: neutral

Review: "Best purchase I ever made!"  
Sentiment: positive

Now classify:

Review: "The movie was absolutely fantastic, best I've seen all year!"  
Sentiment:

#### When to Use Each:

Approach	When to Use
Zero-shot	Simple tasks, model knows the format
One-shot	Need to show format once
Few-shot	Complex tasks, specific patterns

### 💡 Crafting Effective Examples

#### Principles:

1. **Representative** - Examples should cover typical cases
2. **Diverse** - Show variety in inputs/outputs
3. **Clear** - Unambiguous input-output pairs
4. **Relevant** - Similar to actual use cases

#### ✗ Poor Examples:

Convert to formal:  
"hey" -> "Hello"  
"sup" -> "Good day"

Now convert: "I wanna schedule a meeting with u tmrw"

(Examples too short, don't match actual use case)

#### ✓ Good Examples:

Convert informal messages to formal business communication:

Informal: "hey, can we meet tmrw to talk about the project?"  
Formal: "Good morning, would you be available tomorrow to discuss the project progress?"

Informal: "thx for the quick reply, super helpful!"  
Formal: "Thank you for your prompt response. Your assistance has been greatly appreciated."

Now convert:

Informal: "I wanna schedule a meeting with u tmrw about the budget stuff"

Formal:

### 👍👎 Positive and Negative Examples

**Theory:** Show both what TO do and what NOT to do for clearer guidance.

**Example:**

Generate product descriptions for our eco-friendly store.

✓ Good example:

Product: Bamboo Toothbrush

Description: "Crafted from sustainably harvested bamboo, this biodegradable toothbrush offers a plastic-free path to oral care. The soft bristles provide gentle cleaning while reducing your environmental footprint."

✗ Bad example (avoid this):

Product: Bamboo Toothbrush

Description: "Buy this toothbrush. It's made of bamboo.

Good for environment. Cheap price!!!"

Why it's bad: Too short, uses promotional language, lacks descriptive detail, unprofessional tone.

Now write a description for:

Product: Recycled Paper Notebook

Description:

## Week 6: Chain-of-Thought and Reasoning

### 🧠 Encouraging Step-by-Step Reasoning

**Theory:** Asking the model to "think step by step" improves accuracy on complex tasks.

**Without Chain-of-Thought:**

Q: A store sells apples for \$2 each. If I have \$15 and want to buy as many apples as possible while keeping at least \$3 for bus fare, how many apples can I buy?

A: 6 apples ✗ (Wrong!)

**With Chain-of-Thought:**

Q: A store sells apples for \$2 each. If I have \$15 and want to buy as many apples as possible while keeping at least \$3 for bus fare, how many apples can I buy?

Think step by step:

A: Let me solve this step by step:

1. Total money: \$15
2. Money needed for bus fare: \$3
3. Money available for apples:  $\$15 - \$3 = \$12$
4. Cost per apple: \$2
5. Number of apples I can buy:  $\$12 \div \$2 = 6$  apples 

#### Trigger Phrases:

- "Let's think step by step"
- "Walk through your reasoning"
- "Explain your thought process"
- "Break this down into steps"

## Breaking Down Complex Problems

#### Template for Complex Tasks:

I need to solve a complex problem. Please approach it as follows:

1. First, identify the key components of the problem
2. List any assumptions you're making
3. Solve each component separately
4. Combine the solutions
5. Verify the final answer

Problem: [Your complex problem here]

#### Example:

Help me plan a product launch. Break it down as follows:

- Step 1: Identify all the key tasks
- Step 2: Determine dependencies between tasks
- Step 3: Estimate time for each task
- Step 4: Create a timeline
- Step 5: Identify potential risks

Product: A new mobile app for fitness tracking

Launch date: 3 months from now

## Verification and Error-Checking Prompts

#### Self-Verification:

Solve this math problem, then verify your answer:

Problem: Calculate 15% of 340

After solving:

1. Show your work
2. Verify by using a different method
3. Confirm your final answer

#### Error-Checking Pattern:

Review the following code for bugs:

```
<code>
def calculate_average(numbers):
    total = sum(numbers)
    return total / len(numbers)
</code>
```

Check for:

1. Edge cases (empty list, None values)
2. Type errors
3. Division by zero
4. Logic errors

For each issue found:

- Describe the problem
- Show the fix
- Explain why it matters

## Key Takeaways

1. **Delimiters** prevent confusion between instructions and content
2. **Roles** shape expertise, tone, and perspective
3. **Few-shot examples** teach patterns effectively
4. **Chain-of-thought** improves reasoning accuracy
5. **Output format** specification ensures usable results

---

Next: [Module 3 - Advanced Techniques →](#)