

## Copywriting / Product Descriptions

Turn raw product specifications into customer-focused benefits that explain why the product matters in real-world use.

Difficulty: Beginner → Intermediate

Model: GPT-4 / Claude / Gemini

Use Case: Ecommerce, Product Pages, Retail Descriptions

Updated: May 2026

Why This Prompt Exists

Most product descriptions fail because they list features without explaining why they matter.

You get:

- “10GB of RAM” (so what? What does that do for me?)
- “Stainless steel construction” (is that good? Why?)
- “500-thread count” (I don’t know what that means)
- specifications that impress engineers but confuse customers
- descriptions that inform but don’t persuade

But a feature is not a benefit.

It is a fact. The benefit is what that fact does for the customer.

- Feature: What it is (technical specification)
- Benefit: What it does for the customer (outcome)
- Translation: Feature + “which means...” + Benefit
- The best descriptions start with the benefit, then prove it with the feature

Without benefit translation, customers don’t know why they should care.

This framework forces AI to translate every feature into a customer outcome.

## The Prompt

Assume the role of a product copywriter who translates features into benefits that customers actually care about.

Your task is to write a product description that focuses on benefits, not just features.

Generate:

1. PRODUCT TITLE (benefit-driven, not feature-driven)

2. FEATURE-TO-BENEFIT TABLE

For each feature: Feature → "Which means..." → Benefit → "So you can..." → Deeper Benefit

3. BENEFIT-DRIVEN DESCRIPTION (150-250 words)

- Lead with the biggest benefit
- Prove each benefit with a feature
- End with the transformation the customer experiences

4. BULLET BENEFITS (5 items, under 15 words each)

- Feature + Benefit combined

5. "SO WHAT?" TEST PASSED

- Confirmation that every feature has been translated

## INPUTS:

Product Name:

[INSERT]

Raw Features or Specifications:

[LIST FEATURES]

Target Audience:

[WHO IS BUYING THIS?]

Primary Use Case:

[HOW WILL THEY USE IT?]

What's Most Important to This Customer:

[PRICE / QUALITY / DURABILITY / EASE OF USE / STATUS / OTHER]

## RULES:

- Every feature must be translated into a benefit
- No feature-only statements (always include the "which means")
- The benefit must be specific, not "better" or "improved"
- Lead with the benefit, support with the feature
- Avoid technical jargon unless your audience is technical

## How To Use It

- Start with the "So what?" test — if a feature doesn't have a benefit, cut it or find one.
- The deeper benefit (identity shift) is the most persuasive — find it for each feature.
- Benefit-driven titles convert better than feature-driven titles.
- Use the bullet benefits for scannable product grids.

- Save the feature-to-benefit table as a reference for customer support and sales teams.

Example Input

**Product Name:** Ultra-Slim Power Bank

**Raw Features or Specifications:** 10,000mAh capacity, 2 USB ports, 0.5-inch thickness, 8-hour charge time, LED battery indicator

**Target Audience:** Frequent travelers and commuters

**Primary Use Case:** Charging phone and tablet during long travel days

**What's Most Important to This Customer:** Portability and reliability

Why It Works

Most product descriptions fail because they assume customers understand why features matter.

This framework improves outcomes by forcing:

- feature-to-benefit translation (no orphans)
- “which means” bridge language (clarity)
- deeper benefit exploration (identity shift)
- benefit-driven titles (attention)
- “so what?” test verification (accountability)

Great product descriptions don't list what the product has — they describe what the customer gains.

# Build Better AI Systems

Subscribe for advanced prompt engineering, AI copywriting tools, product description frameworks, and practical strategies for writers and ecommerce professionals.

Carefully engineered prompts for people doing real work.

## Share this:

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)