Image Generation / Midjourney

Determine –c value based on desired output diversity — calibrates variation scientifically, not randomly.
Difficulty: Intermediate
Model: GPT-4 / Claude / Gemini
Use Case: Variation Control, Exploration vs. Consistency
Updated: May 2026
Why This Prompt Exists
–chaos (–c) controls how much variation you get between generations. Most users leave it at 0 (identical outputs) or crank it to 100 (unusable randomness) — missing the productive middle range.

You get:

- chaos = 0: every generation looks the same (no exploration)
- chaos = 100: outputs are unrecognizably different (no consistency)
- no understanding of what different chaos values actually produce
- wasted generations because chaos is set wrong for the task
- inability to explore variations while keeping brand identity

But chaos values have predictable effects:

- 0-10: Very similar outputs, minor variations in details
- 11-30: Noticeable variations in composition and framing
- 31-60: Significant variations, different interpretations of the prompt
- 61-80: High variation, may change subject interpretation
- 81-100: Extreme variation, unpredictable results

Without calibration, chaos is a guessing game.

This prompt recommends chaos values by intended variation level.

The Prompt

```
Assume the role of a Midjourney variation scientist who calibrates
chaos parameters.

Your task is to recommend --c values based on desired output
diversity.

Generate:

1. VARIATION GOAL CLASSIFICATION
   - Desired consistency: [IDENTICAL / SIMILAR / MODERATE VARIATION /
HIGH VARIATION / EXPLORATION]
   - Use case type: [BRAND ASSET / CONCEPT EXPLORATION / SERIES / ONE-
OFF]

2. CHAOS VALUE RECOMMENDATION TABLE

| Variation Goal | Recommended --c | Expected Output | Best For |
|----------------|-----------------|-----------------|----------|
| Identical | 0 | Same image, same seed | Reproducible assets |
| Very similar | 1-10 | Minor detail changes | Logo variants, small
adjustments |
| Noticeable variation | 11-30 | Composition changes, different
framing | Multiple options for same brief |
| Significant variation | 31-60 | Different interpretations | Concept
exploration |
```

| High variation | 61-80 | May change subject interpretation | Brainstorming, early ideation |
| Extreme exploration | 81-100 | Unpredictable, experimental | Creative inspiration only |

3. USE CASE MAPPING

Use Case	Recommended --c	Rationale
Brand logo exploration	30-50	Need variety but maintain recognizability
Character concept art	40-70	Explore different looks for same character
Product photography	0-10	Consistent product representation
Social media content	10-30	Fresh looks, same style
Print ad series	5-20	Consistent but not identical
Mood board creation	50-80	High variation for inspiration

4. CHAOS COMBINATION WITH OTHER PARAMETERS
   - --c + --seed: fixed seed + chaos = variations on same theme
   - --c + high --stylize: unpredictable (use lower --c)
   - --c + --no: chaos may override negative prompts at high values

5. CHAOS CALIBRATION TEST PROTOCOL
   - Step 1: Run 4 generations at --c 0 (baseline)
   - Step 2: Run 4 generations at --c 25 (moderate variation)
   - Step 3: Run 4 generations at --c 50 (high variation)
   - Step 4: Select best value based on output

6. COMMON MISTAKES TO AVOID
   - Using --c 0 for exploration (wasted generations, no variety)
   - Using --c 100 for brand assets (inconsistent, unusable)
   - Expecting chaos to create variation within one generation (it affects multiple runs)
   - Combining high chaos with high stylize (too unpredictable)

INPUTS:

Desired consistency level:
[IDENTICAL / SIMILAR / MODERATE / HIGH / EXTREME]

Use case:
[E.G., "Brand logo exploration", "Product photos", "Concept art"]

Number of variations needed:
[E.G., "10-20 options to choose from"]

Subject type:
[E.G., "Abstract", "Realistic product", "Character", "Landscape"]

RULES:
- --c controls variation across multiple generations, not within one image
- Lower chaos (1-10) for brand consistency where identity matters
- Medium chaos (30-50) for concept exploration where variety is valuable
- High chaos (70-100) for brainstorming only — not final assets
- Test chaos at 25, 50, and 75 to understand your specific prompt's

```
sensitivity
- Some prompts are more chaos-sensitive than others (abstract prompts
vary more)
```

How To Use It

- –c controls variation across multiple generations, not within a single image.
- Lower chaos (1-10) for brand consistency where identity matters.
- Medium chaos (30-50) for concept exploration where variety is valuable.
- High chaos (70-100) for brainstorming only — not for final assets.
- Test chaos at 25, 50, and 75 to understand your specific prompt's sensitivity.
- Some prompts are more chaos-sensitive than others (abstract prompts vary more).

Example Input

Desired consistency level:
"MODERATE — want different options but same character"

Use case:
"Character concept art for a game protagonist"

Number of variations needed:
"20-30 options to choose from"

Subject type:
"Character — fantasy warrior"

Why It Works
Most users treat chaos as a binary choice — off or maximum — missing the graduated scale that makes it useful.

This framework improves outcomes by forcing:

- variation goal classification (identical to extreme exploration)
- chaos value mapping (specific ranges for specific outcomes)
- use case alignment (brand assets vs. concept exploration)
- parameter interaction awareness (chaos + stylize, chaos + no)
- calibration test protocol (finding the right value for your prompt)

Failure modes this prevents:

- Chaos = 0 for exploration (all outputs identical, no variety)
- Chaos = 100 for brand assets (outputs unrecognizable, unusable)
- Expecting chaos to vary within one generation (it affects multiple runs)
- Combining high chaos with high stylize (too unpredictable)

This improves on: Random chaos guessing. Calibrated variation produces useful diversity without unusable outputs.

Related to: MJ-01 (Parameters) for chaos syntax; MJ-06 (Remix) for variation patterns.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

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

See also Parameter Classifier