

# Markdown & YAML Prompt Patterns - Cheatsheet

## General Tips (applies to both)

Do:

- Name the format explicitly in the system/first user message.
- Add boundary markers (fenced code blocks or --- lines) so parsers can slice output cleanly.
- One item per line/key - predictable tokenization means fewer hallucinations.
- Post-prompt validation - e.g. output MUST parse with PyYAML or rendered Markdown must have an H1.

Avoid:

- Relying on the model to 'guess' the format from context.
- Mixing human instructions inside the fenced block.
- Long free-form sentences after a YAML key or list bullet.
- Vague statements like 'format nicely please'.

## Markdown-Only Patterns

- Heading hierarchy: For docs/notes. Prompt: 'Return H1 title, H2 Overview, H3 bullets.'
- Checklist: Task/rubric. Prompt: 'Produce a markdown checklist [ ]/[x] with 5 items.'
- Table: Comparisons. Prompt: 'Give a Markdown table with columns: Concept | One-liner | Difficulty'
- Inline code vs fenced: 'Use inline back-ticks for filenames; fenced block for code.'
- Footnotes: Citations. Prompt: 'Add numeric footnotes at end of doc.'
- Word/char limits: 'Respond with exactly 50 words in Markdown.'

## YAML-Only Patterns

- Flat key-value: Configs. 'Return YAML with keys title, summary, difficulty (int).'
- Nested objects: Hierarchy. 'Make nested YAML: blog > title/body/tags'
- Sequences (list): 'Return list of 5 strings under key bullets.'
- Anchors & aliases: 'Use YAML anchors so "common" appears twice.'
- Front-matter header: 'Give YAML front-matter only (no more text).'
- Parsing guarantee: 'Your answer must start with --- and end with ..., valid PyYAML.'

## Combined / Multi-Format Prompts

Recipe:

1. Lead-in: 'You are a formatting engine...'
2. Explicit menu:  
Please output the same content in five sections, each fenced:  
1) RAW 2) MARKDOWN 3) HTML 4) YAML 5) CSV
3. Use triple back-ticks with language tags (``md, ``yaml, etc.)
4. Optional: 'The YAML must parse and the CSV must have 4 columns.'

## Escape-Hatch Techniques

- Model adds commentary after YAML? End prompt with: 'Do NOT add any prose outside the YAML block.'
- Markdown bullets collapse? Pre-seed one bullet: '- ' then <<CONTINUE>>.
- Word-limit violations? Reinforce: 'Stop immediately at 75 words - you may truncate mid-sentence.'
- Undesired quotes in YAML? Add: 'Avoid using > or | multiline scalars; keep plain style.'

## Quick-Reference Prompt Templates

Markdown Section Template:

SYSTEM: You are a precise markdown generator.

USER: Produce a Markdown doc with:

- # Title (provided)
- ## Overview (<= 40 words)

- ### Key Points (5 bullets, <= 10 words each)

Title = "{topic}"

Return only the markdown.

YAML Config Template:

SYSTEM: Return valid YAML only - no explanation.

USER: Build configuration for a micro-service:

service:

name: "{name}"

version: "{version}"

ports: [80, 443]

env:

LOG\_LEVEL: "INFO"

RETRIES: 3

Dual Output (Markdown + YAML):

SYSTEM: You are an LLM that outputs two fenced blocks.

USER: First block - Markdown one-paragraph summary (<= 60 words).

Second block - YAML with keys short, long. No other text.

## Verification Snippets (Python)

```
import markdown, yaml, textwrap, textstat
```

```
# Parse Markdown -> HTML
```

```
html = markdown.markdown(md_output)
```

```
# Validate YAML
```

```
data = yaml.safe_load(yaml_block)
```

```
assert isinstance(data, dict)
```

## Memorise the mantra

Name it, fence it, bound it, test it.

If you follow those four steps, structured-output headaches disappear. Happy prompting!