

□---

description: "Defines the C.R.A.F.T. framework (Context, Role, Action, Format, Tone/Audience) and provides templates, examples, and an author checklist for crafting prompts."

applyTo: "**"

The C.R.A.F.T. Framework

All prompt generation, analysis, and refactoring must be performed through the lens of the C.R.A.F.T. framework.

- ****Context:**** Background, inputs, and constraints the model needs to understand the task.
- ****Role:**** The persona or skill the model should adopt (expert, teacher, analyst, etc.).
- ****Action:**** A single, clear imperative describing what the model should do.
- ****Format:**** The exact output structure (schema, file type, or layout) required.
- ****Tone / Audience:**** The writing style and the intended reader (e.g., "concise for executives").

```instructions

---

description: "Defines the C.R.A.F.T. framework (Context, Role, Action, Format, Tone/Audience) and provides templates, examples, and an author checklist for crafting prompts."

applyTo: "\*\*"

---

## ## The C.R.A.F.T. Framework

All prompt generation, analysis, and refactoring must be performed through the lens of the C.R.A.F.T. framework.

- **\*\*Context:\*\*** Background, inputs, and constraints the model needs to understand the task.
- **\*\*Role:\*\*** The persona or skill the model should adopt (expert, teacher, analyst, etc.).
- **\*\*Action:\*\*** A single, clear imperative describing what the model should do.
- **\*\*Format:\*\*** The exact output structure (schema, file type, or layout) required.
- **\*\*Tone / Audience:\*\*** The writing style and the intended reader (e.g., "concise for executives").

## ## Purpose and scope

This file provides a prescriptive template and practical guidance for writing prompts used in ``.prompt.md``, ``.chatmode.md``, and other instruction-oriented Markdown files. Use the minimal template for short one-off prompts and the extended template for complex or reusable prompts.

## ## Minimal CRAFT prompt template (copy & fill)

Context: \${short context – 1–2 sentences describing input, environment, constraints}  
Role: \${persona – e.g., "Senior Data Scientist"}  
Action: \${single imperative verb + brief object – e.g., "Summarize the report into 5 bullet points"}  
Format: \${output form – e.g., "Markdown numbered list" or "JSON: {summary, details}"}  
Tone / Audience: \${tone and audience – e.g., "plain English for product managers"}

### Example (minimal):

Context: You are given a 10-page product requirements document about a new payments feature.  
Role: Product manager summarizer.  
Action: Extract the top 6 functional requirements and the 3 main risks.  
Format: Markdown with H2 sections "Requirements" and "Risks" and numbered lists underneath.  
Tone / Audience: Executive-level, concise.

## ## Extended CRAFT template (for complex or reusable prompts)

Include the minimal template plus these fields when the task is non-trivial or will be reused.

- Inputs: Named inputs expected by the prompt (e.g., `file: report.md`, `variables: {start\_date, end\_date}`).
- Constraints: Hard limits and rules (e.g., "max 150 words", "no invented facts", "must include citations").
- Examples: 1–2 minimal input – output examples showing exact format and level of detail.
- Verification: How outputs will be checked (simple rubric, unit test, or follow-up verification prompts).
- Metadata: Optional changelog, author, and `applyTo` recommendations for `.instructions.md` files.

### Extended example:

Context: You have a CSV of customer support tickets with columns {id, created\_at, category, resolution\_time, text}.  
Role: Senior analyst who writes reproducible summaries.  
Action: Produce an executive summary of trends for the prior quarter and recommend 3 operational changes.  
Format: 1) A 3-paragraph executive summary (max 150 words). 2) A Markdown table of top 5 categories and their avg resolution\_time. 3) A numbered list of 3 recommendations.  
Tone / Audience: Non-technical VP of Support; use plain English and define any domain terms.

Inputs: `file: tickets\_q3.csv`

Constraints: Do not extrapolate outside the data. Include the exact SQL used to compute the table (one-line code fence). Max 150 words for the executive summary.

Examples:

- Input: (small sample rows) - Output: (example summary)

Verification: Check that the table rows match results from the provided SQL.

### ## Quick author checklist (pre-submit)

- [ ] Context gives necessary background and scope (who, what, when, where).
- [ ] Role is an actionable persona (one sentence) and matches desired expertise.
- [ ] Action is a single, clear imperative (avoid compound verbs like "analyze and decide").
- [ ] Format is machine- or human-readable and unambiguous (include a schema when needed).
- [ ] Tone/Audience is specified and consistent with examples.
- [ ] Inputs and Constraints are explicit for any data-driven task.
- [ ] Examples (if present) are minimal, representative, and follow the expected output format.
- [ ] Verification steps or acceptance criteria are stated (e.g., "3 bullets, each <= 20 words").

### ## Common failure modes and mitigations

- Failure: Model ignores provided inputs or tools.  
Mitigation: Make Role explicitly state "You must use the provided inputs/tools and must not hallucinate" and add a Verification step.
- Failure: Output format drift (e.g., prose instead of JSON).  
Mitigation: Provide a strict schema and a minimal example JSON instance; instruct "Return only valid JSON".
- Failure: Overly long responses.  
Mitigation: Add a hard constraint like "Max 150 words" and show a short Format example.

### ## Small reusable prompt snippets

Use these snippets to speed prompt authoring and reduce errors. Replace variables in braces.

- "If you cannot answer from the inputs, respond with exactly: 'I don't know based on the provided data.'"
- "Return only valid JSON conforming to this schema: {\"summary\":string, \"items\": [ {\"id\":int, \"note\":string} ] }"
- "Cite the source line or file for any factual claim in the format: (source: <filename>:<line-range>)"

### ## Evaluation rubric (prompt quality)

Score the prompt before reusing it.

- 1 – Poor: Missing components; likely to produce ambiguous results.
- 2 – Fair: Most components present but missing constraints or examples.
- 3 – Good: Complete CRAFT, includes constraints and short examples.
- 4 – Excellent: Complete CRAFT, includes verification rules, example outputs, and explicit anti-hallucination language.

## Implementation notes for `.instructions.md` files

- Keep YAML frontmatter (`description`, `applyTo`) accurate and concise.
- `applyTo` should be as specific as practical (e.g., `"/\*\*/\*.prompt.md"` or `"/docs/\*\*")`).
- If this instruction file is also a template, add a short example prompt at the bottom of the file inside a fenced block and maintain a changelog comment at the top.

## References and source materials

This guidance draws on research and practitioner summaries in `Training Guides/Updating CRAFT/` and the CRAFT (toolset) paper by Yuan et al. (ICLR 2024). Use those materials for deeper background when creating complex, reusable prompts.

## Contact and iteration

When you iteratively improve a prompt template, add a one-line changelog at the top with date and reason. Small iterative changes are encouraged.

□