

A PROMPTS

Prompt Template for Extracting Code Questions from issues

You are given a GitHub issue from the {REPOSITORY_NAME} repository. Extract or rewrite it into one or more **short, clear, concise questions** about understanding the {REPOSITORY_NAME} codebase, APIs, or system design.

Rules:

1. Only include questions answerable by code, documentation, or logic.
2. Ignore bug reports, environment issues, or problems that require fixing code.
3. Each question should ideally be <= 20 words.

IMPORTANT:

- Be STRICT in quality control: if the issue doesn't contain meaningful questions about code understanding, return an empty questions array.
- It's better to return no questions than to generate low-quality or irrelevant questions.
- Only extract questions that genuinely help understand the {REPOSITORY_NAME} codebase, APIs, or system design.

GitHub issue from {REPOSITORY_NAME} repository:

Title:
 {ISSUE_TITLE}

Body:
 {ISSUE_BODY}

Output JSON format:

```
{
  "issue_number": <number>,
  "questions": [
    {"question": "<string>"},
    ...
  ]
}
```

Prompt Template for Generating Repository-level Code Questions

You are an expert software research assistant.

Given:

1. A function/class description extracted from a software repository.
2. A list of seed questions as candidates from the {CATEGORY} category.

Task:

1. Based on the seed questions and the function/class description, generate **one single question** that is:
 - As difficult and complex as possible,
 - Requires multi-hop reasoning or deep technical understanding,
 - Not answerable by simple retrieval or direct lookup (i.e., not solvable by basic RAG methods),
 - Clearly related to the function/class description,
 - Technically precise and detailed,
 - Reflects the style and intent of the original seed questions but goes significantly deeper.
 - **Must not be a compound question** (e.g., no use of "and", "or", or comma-based subquestions),
 - **Must be not too long and syntactically simple**
 - **Must be specific to the {CATEGORY} category**
2. The question should encourage advanced analysis, integration of multiple concepts, or insight beyond surface-level information.
3. Output only the single refined question without additional explanation or commentary.

Input:

1. Function/Class Description:
 {DESCRIPTION}
2. Seed Questions from {CATEGORY}:
 {SEED_QUESTIONS}

Prompt Template for Generating Reference Answers

You are a software research assistant specialized in Python code.
Working directory: Python repository {REPO_NAME}

Task:

1. Search and analyze only the code, documentation, and knowledge inside this repository related to the following question.
2. Do not use or assume external knowledge unless explicitly required.
3. Summarize findings into a concise, direct answer strictly based on repository content.
4. Output only the final synthesized answer, without code snippets, explanations, or commentary.

Question: {QUESTION}

Prompt for LLM-as-Judge

You are a professional evaluator. Please rate the candidate answer against the reference answer based on five criteria.

- Evaluation Criteria and Scoring Guidelines (each scored 0 to 10):
1. Correctness:
 - 9-10 – Completely correct; core points and details are accurate with no ambiguity.
 - 7-8 – Mostly correct; only minor details are slightly inaccurate or loosely expressed.
 - 5-6 – Partially correct; some errors or omissions, but main points are generally accurate.
 - 3-4 – Several errors or ambiguities that affect understanding of the core information.
 - 0-2 – Serious errors; misleading or fails to convey key information.
 2. Completeness:
 - 9-10 – Covers all key points from the reference answer without omission.
 - 7-8 – Covers most key points; only minor non-critical information missing.
 - 5-6 – Missing several key points; content is somewhat incomplete.
 - 3-4 – Important information largely missing; content is one-sided.
 - 0-2 – Covers very little or irrelevant information; seriously incomplete.
 3. Relevance:
 - 9-10 – Content fully focused on the question topic; no irrelevant information.
 - 7-8 – Mostly focused; only minor irrelevant or peripheral information.
 - 5-6 – Topic not sufficiently focused; contains considerable off-topic content.
 - 3-4 – Content deviates from topic; includes excessive irrelevant information.
 - 0-2 – Majority of content irrelevant to the question.
 4. Clarity:
 - 9-10 – Fluent language; clear and precise expression; easy to understand.
 - 7-8 – Mostly fluent; some expressions slightly unclear or not concise.
 - 5-6 – Expression somewhat awkward; some ambiguity or lack of fluency.
 - 3-4 – Language obscure; sentences are not smooth; hinders understanding.
 - 0-2 – Expression confusing; very difficult to understand.
 5. Reasoning:
 - 9-10 – Reasoning is clear, logical, and well-structured; argumentation is solid.
 - 7-8 – Reasoning generally reasonable; mostly clear logic; minor jumps.
 - 5-6 – Reasoning is average; some logical jumps or organization issues.
 - 3-4 – Reasoning unclear; lacks logical order; difficult to follow.
 - 0-2 – No clear reasoning; logic is chaotic.

INPUT:

Question:{QUESTION}
Reference Answer:{REFERENCE}
Candidate Answer:{CANDIDATE}

OUTPUT:

Please output ONLY a JSON object with 5 integer fields in the range [0,10], corresponding to the evaluation scores:

```
{
  "correctness": <0-10>,
  "completeness": <0-10>,
  "relevance": <0-10>,
  "clarity": <0-10>,
  "reasoning": <0-10>
}
```

REQUIREMENT:

No explanation, no extra text, no formatting other than valid JSON