Best practice of Prompting cheat sheet for Coding

1

Refactor using classes and typehint.

Design a class based on the best design pattern.

Design a class based on the best architecture.

2

Propose an algorithm that satisfies the following constraints.

3

Code review.

Give an example.

4

Eliminate duplication and seek ways to consolidate into a single function or class if possible.

5

Check for appropriate comments. If not, fix them.

Generate Docstrings.

6

Check for potential vulnerabilities. If not, fix them.

Check for proper exception handling using LBYL-Style. If not, fix them.

Check for logical integrity. If not, fix them.

7

Output a sample proposal.

Output a configuration diagram using diagrams.

Convert the schema to DBML.

8

Generate code that satisfies the test code.

common prompt

Think Step By Step.

No explanation needed, just the code.

You should always add a citation with a valid URL to support each your statement when there.

If you don't know, please state that you don't know.

Return full script (I don't have Fingers).

Also make sure to add the confidence of your answer.

Make it a specialized article and explain the analysis in detail.

There is no need for beginner-friendly definitions of terms, UI explanations, introductions, or conclusions.

Include multiple topics with practical content at the academic research level.

Based on the purpose of the technology and the technical problems it wants to solve.

specific libraries

import pytest, functools, signal, traceback, pydantic, starlette, tailwind, nextjs

abstruct shot

Code with robustness, effectiveness, high-performance, security in mind.

specific usecases

Check for proper signal handling. If not, fix them.

use user-defined types, subtypes, and protocols.

Include references to established and latest popular frameworks, if any.

Incorporates other services or tools. For example: OSINT

Explain in Storytelling, narrative.

Create a curriculum for MIT's Mastery class of Architecture program

specific word pool

Code with the following in mind:

Type annotations

formatting and style

MVC

Invariants

Anemic Domain Model

Transaction Script

Object Oriented

Self-protection Responsibility

Immutable

Mutator

Complete Constructor

Value Object

Strategy

Policy

First-Class Collection

Single Responsibility Principle

Composition

Goal-Driven Naming Design

Encapsulation

Modeling

Command-Query Separation

Refactoring

Ease of Modification

Code Metrics

Core Domain

Low Cohesion

Tight Coupling

Duplicate Code

Half-Baked Objects

Multiple Nesting

Technology-Driven Naming

Anemic Domain Model

Primitive Type Attachment

Flag Argument

Smart

God Class

Exception Suppression

Technology-Driven Packaging

Degenerate Comments

Double Meaning

Technical Debt

Security Checklist

Are filenames reflected back on the page? If so, are they HTML Entity encoded (XSS via file names)?

Does it accept .zip files? Try a ZipSlip

If it processes an image, check for Image Tragick (CVE-2016-3714)

Can you bypass file type restrictions by changing the content-type value?

Can you bypass file type restrictions by forging valid magic bytes?

Can you upload a file with a less-common extension (such as .phtml)?

Try playing with the filename in the request, a potential vector for traversal or SQL injection.

Check for the acceptance of double extensions on uploaded files.

Test for null-byte injection.

Is the server windows? Try adding a trailing . to bypass extension blacklists, this dot will be removed automatically by the OS.

Can you upload an SVG for XSS?

If supported by the webserver, can you upload .htaccess files?

Does the backend process the image with the PHP GD library?

Is the app vulnerable to the infamous ffmpeg exploit?

Can custom polyglots be developed to bypass specific filters?

Does the app pass the file name to some sort of system function? If so, can you achieve RCE via code injection within the file name?

Does the application run the uploaded file through exiftool? If so, can you get RCE via the djvu exploit?

Can you bypass extension filters by using varied capitalization?