

ChatGPT Prompts for Programmers

Code Generation

1. create a function in [language] to [objective]
2. create a model in [language] to represent an entity with these attributes: [attributes]
3. [code in some language]: convert this code to [another language name] code
4. act as an expert [language / library / framework / stack] developer. now give me step by step instructions to [list of objectives]
5. write a function in [language] to solve [list of objectives]
6. [desired output in plain words]: write a SQL query to get this output
 - example

[list all students from students table who got grades above 50 and below 80 and for each student who meets this criterion, retrieve their assignment grade for history from the Assignments table]: write a SQL query to get this output
7. [SQL query in some flavor e.g. SQLite, MySQL, PostgreSQL]: convert this SQL query into [another flavor of SQL e.g. SQLite, MySQL, PostgreSQL]
8. [objective]: write a [language] program for this. the program should follow these constraints: [list of constraints]
9. write a program in [language] using [functional / object oriented / aspect oriented] style to [objective]
10. write a program in [language] to [objective] in the style of [famous programmer]

Documentation

11. search official documentation for [technology] and explain [topic] with analogies for a 5 year old
12. search official documentation for [framework / language / library] and explain [topic] with analogies for a [profession]
13. [snippet from documentation]: explain this with analogies for a 5 year old / [profession / persona]
14. list [number] ways to create technical documentation that is clear, concise, and easy to understand for [target audience]
15. provide [number] techniques for writing technical documentation that helps people with disabilities
16. list [number] tips for developers to ensure that their documentation is easy to navigate and search
17. list the top [number] tools and formats used to create technical documentation
18. provide industry best practices for managing and updating technical documentation

19. list [number] strategies for writing technical documentation that a 5 year old can understand
20. list [number] best practices for documenting API's and SDK's

Debugging

21. [error / warning message]: provide 5 tips to solve this error
22. [code] [error]: debug code for error with step-by-step instructions to resolve it
23. suggest [number] best practices for debugging [language] code.
24. list [number] techniques for debugging [language / library / framework / stack] code in [IDE / code editor]
25. give me [number] best practices for debugging [type of app] apps
26. tell me [number] ways to use logging and error messages in [language / library / framework / stack] to diagnose and troubleshoot problems
27. I want to learn about code profiling and tracing to identify performance issues in software. tell me the top [number] techniques used for this.
28. list [number] ways for web developers to debug issues related to network requests, cookies, and session state
29. act as an expert [language / library / framework / stack] developer who is a master of debugging. now help me debug this code [code]
30. You're a developer working on a web application that is experiencing intermittent issues with session state. Users are reporting that they are occasionally logged out of the application and lose their progress. List [number] debugging techniques you would use to identify and fix the issue. Also list [number] strategies for preventing similar issues from occurring in the future.

Code Refactoring

31. [code]: simplify this code
32. [code]: calculate cyclomatic complexity for this code
33. [code]: refactor this code to change cyclomatic complexity to [desired cyclomatic complexity]
34. [code]: refactor this code using [coding convention]
35. share 5 best practices for refactoring large and complex codebases
36. act as an expert coder in [language]. now list 5 common code smells and for each code smell, list 3 best practices to solve them
37. act as an expert agile developer. now create 10 best practices for integrating agile workflows into a [language / library / framework / stack] project
38. list step by step instructions for refactoring converting [type of architecture] into [another type of architecture]
39. provide 5 ways in which developers can measure the impact of their refactoring efforts
40. list 10 tips for refactoring spaghetti code in [language / library / framework / stack]

Algorithms

41. explain [algorithm] using analogies for a 5 year old
42. explain [algorithm] using analogies for [language / library / framework] developer
43. explain [algorithm] using analogies to a genetic engineer
44. write step by step instructions to explain [algorithm] to [persona / profession]
45. provide [number] differences between [algorithm 1] and [algorithm 2]. explain with analogies to a 5 year old / [persona, profession]
46. provide the top [number] algorithms to be used in [scneraio / problem statement]
47. list [number] algorithms commonly used in [scenario / field of study / subject]
48. provide a bullet list of techniques for analyzing time and space complexity of algorithms. explain each technique using analogies for a [5 year old / profession / persona]
49. give me [number] ways to implement [algorithm] in [language]
50. create a new algorithm based on [list of algorithms] to solve [scenario / problem statement] given these constraints; [list of constraints]