

Polecenia ChatGPT do programowania

Oprócz umiejętności, o których czytałeś, ChatGPT całkiem dobrze radzi sobie z programowaniem. Może się to wydawać dziwne, ale ChatGPT to rozbudowany model językowy, a programy są pisane w różnych językach programowania, z których większość stosuje znacznie bardziej formalne struktury niż angielski i inne języki ludzkie, z którymi ChatGPT radzi sobie tak dobrze. Jeśli jesteś programistą lub pracujesz w branży oprogramowania, spróbuj zintegrować ChatGPT ze swoim procesem pracy i zobacz, co może dla Ciebie zrobić. Nawet jeśli Twoje zainteresowanie programowaniem jest sporadyczne i potrzebujesz jedynie szybko napisać makro VBA w Wordzie lub skrypt w Pythonie, aby wykonać jednorazowe zadanie, ChatGPT może zaoszczędzić Ci czasu i wysiłku.

Zacznijmy od pytania wartego 64 miliony dolarów...

W jakich językach programowania może Ci pomóc ChatGPT?

Krótką odpowiedź brzmi: w większości z nich. Python, Java, JavaScript, C, C++, C#, Ruby, PHP, Swift, Go, R, SQL, TypeScript, VBA, Rust i MATLAB — ChatGPT chętnie pomoże Ci w nauce tych języków i nie tylko.

Poza tymi językami programowania, ChatGPT może pomóc w tworzeniu skryptów powłoki za pomocą narzędzi takich jak Bash i PowerShell. Może również pomóc w HTML i CSS, które nie są tak naprawdę językami programowania, ale są powszechnie używane w witrynach internetowych. Jeśli pracujesz z nowszymi językami programowania, takimi jak Carbon (znany również jako Carbon-Lang, wprowadzony w 2022 roku) lub A Tensor Language (ATL, również nowy w 2022 roku), ChatGPT będzie miał mniej informacji, ale będzie mógł przeszukiwać sieć, jeśli korzystasz z modelu przeglądania sieci.

Jak ChatGPT może pomóc Ci w programowaniu?

ChatGPT może pomóc Ci w programowaniu na pięć głównych sposobów:

- Skonceptualizuj program lub cel kodu. Zanim zaczniesz pisać kod, zazwyczaj będziesz chciał opisać program lub inny obiekt kodu, który planujesz utworzyć. Jeśli program jest złożony, rozważ podzielenie go na komponenty. ChatGPT może pomóc Ci zdefiniować, co kod powinien robić.
- Napisz kod. Gdy jasno określisz cel kodowania, czas zacząć pisać — lub czas, aby ChatGPT zaczął pisać za Ciebie. ChatGPT może nie tylko pomóc Ci nakreślić strukturę programu lub obiektu, ale także napisać za Ciebie cały kod.
- Debuguj kod. Czasami początkowa wersja kodu ChatGPT — lub Twoja — nie działa. W takim przypadku możesz poprosić ChatGPT o debugowanie.
- Refaktoryzuj kod. Gdy kod działa, możesz zlecić ChatGPT restrukturyzację kodu, aby zmniejszyć jego złożoność i ułatwić jego czytanie i konserwację.
- Zoptymalizuj kod. Gdy kod jest skonstruowany w pożądanym przez Ciebie sposób, możesz polecić ChatGPT zoptymalizowanie kodu, aby działał szybciej, zużywał mniej zasobów lub spełniał oba te warunki.

Przyjrzyjmy się każdemu z tych pięciu kroków wraz z przykładami w kolejnych sekcjach.

Konceptualizacja celu programu lub kodu

Pierwszym krokiem jest zazwyczaj koncepcja programu lub kodu, który chcesz stworzyć. Oto przykłady monitów, z których możesz skorzystać, aby skorzystać z pomocy ChatGPT:

Jakiego języka programowania użyłbym do analizy dużych ilości danych?

ChatGPT zasugerował języki takie jak Python, R, SQL i Julia, wyjaśniając wady i zalety każdego z nich.

Napisz plan utworzenia skryptu w Pythonie, który będzie zachęcał użytkownika do wprowadzenia informacji, które skrypt następnie zapisze w arkuszu kalkulacyjnym Excel.

ChatGPT dostarczył szczegółowy plan, od konfiguracji środowiska Python i utworzenia nowego skryptu, aż po jego przetestowanie i rozszerzenie w celu obsługi bardziej złożonych scenariuszy.

Używając języka Go, chcę utworzyć funkcję, która dodaje dwie liczby zmiennoprzecinkowe i konwertuje wynik na liczbę całkowitą. Czy możesz mi powiedzieć, jak zacząć?

Ponownie, ChatGPT podał szczegółowe informacje, w tym ostrzeżenie, że Go używa typowania statycznego, co oznacza, że musisz określić typy danych wejściowych i wyjściowych, z których korzysta funkcja.

Napisz kod

Aby ChatGPT napisał kod, wskaż mu język, którego ma używać i jaki kod ma utworzyć. Oto trzy przykłady:

Napisz program „Hello, Universe!” w C, dodając komentarze wyjaśniające każdą instrukcję.

ChatGPT zwrócił program z komentarzami wyjaśniającymi jego działanie. Możesz kliknąć przycisk Kopiuj kod, aby skopiować kod, wkleić go do kompilatora C i uruchomić.

Zaimplementuj algorytm sortowania bąbelkowego w Rust, który wyświetla każdą wykonaną operację, abym mógł zrozumieć, jak uzyskano wynik.

ChatGPT zwrócił „prostą implementację algorytmu sortowania bąbelkowego w języku Rust” z przykładem, który można skopiować, oraz szczegółowym wyjaśnieniem.

Jak mogę użyć biblioteki ruamel.yaml w Pythonie, aby uzyskać dwa słowniki: jeden dla par klucz-wartość tagu i tekstu oraz drugi dla par klucz-wartość tagu i komentarza dla wierszy tagów zawierających komentarze?

ChatGPT zalecił użycie obiektu CommentedMap w parserze ruamel.yaml i udostępnił przykład ze szczegółowym wyjaśnieniem.

UWAGA Możesz również poprosić ChatGPT o pomoc w zakresie konkretnych struktur programowania. Na przykład, możesz zapytać:

Jakich rodzajów wyrażeń pętli mogę używać w Rust?

Lub:

Pokaż mi, jak działa pętla while w Rust

Debugowanie kodu

Jeśli masz kod, który nie działa, sprawdź, czy ChatGPT może Ci pomóc w debugowaniu. ChatGPT sprawdza się w przypadku kodu, który w ogóle się nie uruchamia, kodu, który działa częściowo, ale generuje określone błędy, oraz kodu, który działa pozornie poprawnie, ale generuje nieprawidłowe dane wyjściowe z powodu błędów logicznych.

Debugowanie kodu, który się nie uruchamia

Jeśli ChatGPT utworzył lub pracował z kodem w tym czacie, możesz po prostu polecić ChatGPT jego użycie. Jeśli ChatGPT nie zna kodu, musisz go wkleić. Podaj monit, taki jak ten:

Poniższy kod nie działa. Sprawdź go i powiedz mi, co jest nie tak.

Naciśnij Shift+Enter, aby utworzyć nowy wiersz, a następnie wklej kod, naciskając Ctrl+V.

UWAGA Jeśli chcesz, możesz określić język programowania używany w kodzie, ale zazwyczaj nie jest to konieczne, ponieważ ChatGPT ustala, jaki to język.

Jeśli ChatGPT zidentyfikuje problem, zwróci krótkie wyjaśnienie jego istoty i przyczyny. Zwraca również wersję kodu, zaktualizowaną o zidentyfikowaną poprawkę, którą możesz wypróbować. Aby dać kodowi szansę na porządne przetestowanie, możesz poprosić ChatGPT o przypadki testowe. Oto przykładowy monit:

Podaj cztery przypadki testowe do przetestowania kodu, który właśnie naprawiłeś.

ChatGPT natychmiast zwrócił te przypadki testowe, wyjaśniając, co każdy z nich robi (np. „Test z pustym ciągiem”) i pokazując wynik, jaki powinien wygenerować każdy z nich.

Kod debugowania generujący określone błędy

Jeśli Twój kod generuje określony błąd, taki jak błąd składni w Pythonie, prześlij ChatGPT kod i zapytaj o przyczynę wystąpienia tego błędu. Oto przykład:

Dlaczego ta instrukcja w Pythonie generuje błąd składni?

Naciśnij Shift+Enter, aby utworzyć nowy wiersz, a następnie wklej błąd, naciskając Ctrl+V. ChatGPT zwrócił wyjaśnienie, takie jak poniższe, oraz poprawiony kod.

Błąd składni występuje, ponieważ brakuje nawiasów wokół ciągu, który próbujesz wydrukować. W Pythonie 3 instrukcja print została zastąpiona funkcją print().

Debugowanie kodu generującego błędy logiczne

Czasami kod działa bez błędów, ale zwraca wynik, który można uznać za błędny. Zazwyczaj jest to wynikiem błędów logicznych, czyli błędów związanych z logiką działania kodu, a nie ze składnią. Gdy kod generuje błąd logiczny, możesz zapytać ChatGPT, co jest nie tak. Na przykład:

Dlaczego poniższy kod zwraca liczbę dodatnią zamiast ujemnej?

ChatGPT przeanalizował kod, przedstawił szczegółowe wyjaśnienie i przekazał mi poprawioną wersję kodu do wypróbowania.

Refaktoryzacja kodu

W tym momencie kod powinien działać. O ile nie tworzysz szybkiego rozwiązania problemu, który nigdy więcej się nie powtórzy, nie czas jeszcze spocząć na laurach. Zamiast tego powinieneś prawdopodobnie dokonać refaktoryzacji kodu, zmniejszając jego złożoność i ułatwiając czytanie i utrzymanie. A jeszcze lepiej, zleć refaktoryzację kodu ChatGPT.

UWAGA Refaktoryzacja kodu i jego optymalizacja w pewnym stopniu się pokrywają, ale zazwyczaj refaktoryzację kodu przeprowadza się przed jego optymalizacją. Oprócz ułatwienia czytania i utrzymania kodu, refaktoryzacja może również zidentyfikować obszary wymagające optymalizacji.

Oto cztery przykłady podpowiedzi dotyczących refaktoryzacji kodu:

Proszę o refaktoryzację tego kodu, aby był czytelny i łatwy w utrzymaniu.

Chcę, aby ten kod był łatwiejszy w utrzymaniu. Czy możesz znaleźć istniejącą bibliotekę, która wyodrębni dane tak, jak ja to robię?

W poniższym kodzie proszę zastąpić łańcuch if-then-else instrukcją case.

Przeprowadź refaktoryzację tego kodu, przenosząc dyskretną funkcjonalność do zestawu funkcji, aby zmniejszyć zagnieżdżanie.

W każdym z tych przykładów należy przekazać ChatGPT kod, chyba że ChatGPT ma już do niego dostęp w aktywnym czacie.

Optymalizacja kodu

Po refaktoryzacji kodu do preferowanego celu, można zlecić ChatGPT optymalizację kodu. Oto dwa przykłady monitów, których można użyć, aby poprosić ChatGPT o optymalizację kodu. Jak zwykle, należy wkleić kod, chyba że ChatGPT ma już do niego dostęp w tym czacie.

Zidentyfikuj wszelkie prawdopodobne problemy z wydajnością w poniższym kodzie C# i powiedz mi, jak je naprawić.

Chciałbym, abyś pomógł mi zidentyfikować operacje wymagające największych nakładów obliczeniowych w tym kodzie i powiedział, jak przyspieszyć jego działanie.

Możesz również poprosić ChatGPT o ogólne porady dotyczące optymalizacji, nie podając swojego kodu. Oto przykład:

Używam Pythona z lxml i próbuję pobrać wszystkie elementy z atrybutem „persist”. Mój kod iteruje po każdym elemencie i sprawdza go pod kątem atrybutu „persist”. Kod działa wolno. Czy istnieje lepszy sposób na osiągnięcie tego celu?

ChatGPT zalecił użycie języka XPath do bezpośredniego wybierania elementów XML z atrybutem „persist”.