



How to Use ChatGPT

TO SUPERCHARGE YOUR
PROGRAMMING JOURNEY

How to Use ChatGPT to Supercharge Your Programming Journey

INTRODUCTION

If you're learning to code during this fast-paced time in tech, you've probably heard about the possibility that ChatGPT can help you with programming. While it would be overly optimistic to think that it can do everything for you, the truth is that ChatGPT can absolutely give you an edge in learning to code. But without proper guidance, it's easy to pick up unhelpful advice and bad habits – especially for a newcomer to programming.

It's critical, then, not to overuse this tool, despite its immense potential value. Ideally, you should combine ChatGPT with several other learning sources, such as the Programming QuickStart Guide Bundle. With the right mindset and guidance, you can use ChatGPT to get a leg up that developers just a few years ago did not have. Let's break down how you can get the most out of it, with real, *actionable* advice on getting started.

WHY DO YOU NEED TO USE CHATGPT AS A PROGRAMMER?

As programmers, we have the luxury of vast knowledge at our fingertips. The internet is a world of information built by programmers, for programmers. Despite this wealth, however, some content can still be prohibitively expensive, misguided, or just plain incorrect. Finding a trustworthy source is essential.

ChatGPT is just another tool in the vast sea of options you have as a student of programming. Not long ago, when you had a coding question, you would turn to a resource like Stack Overflow where you could post your query and await a helpful reply. Depending on how you phrased your question (or who looked at it first), this could end up going many different ways. Forum commenters can be a little hot-headed, or unwilling to deal with a newcomer who doesn't put in the proper legwork before posing a question. And almost every forum has at least one user who often posts unhelpful responses.

AI, meanwhile, has more patience and understanding than your average public forum. Using AI is more akin to hiring a private tutor and conversing with them via chat. You can ask dumb questions. You can repeat questions. You can go completely off-topic if you so choose, and you won't have to deal with an impatient human getting fed up with you for wasting their time.

ChatGPT can help a newcomer take the leap and jump in when comparative learning materials seem a bit intimidating. You can start from the absolute bottom

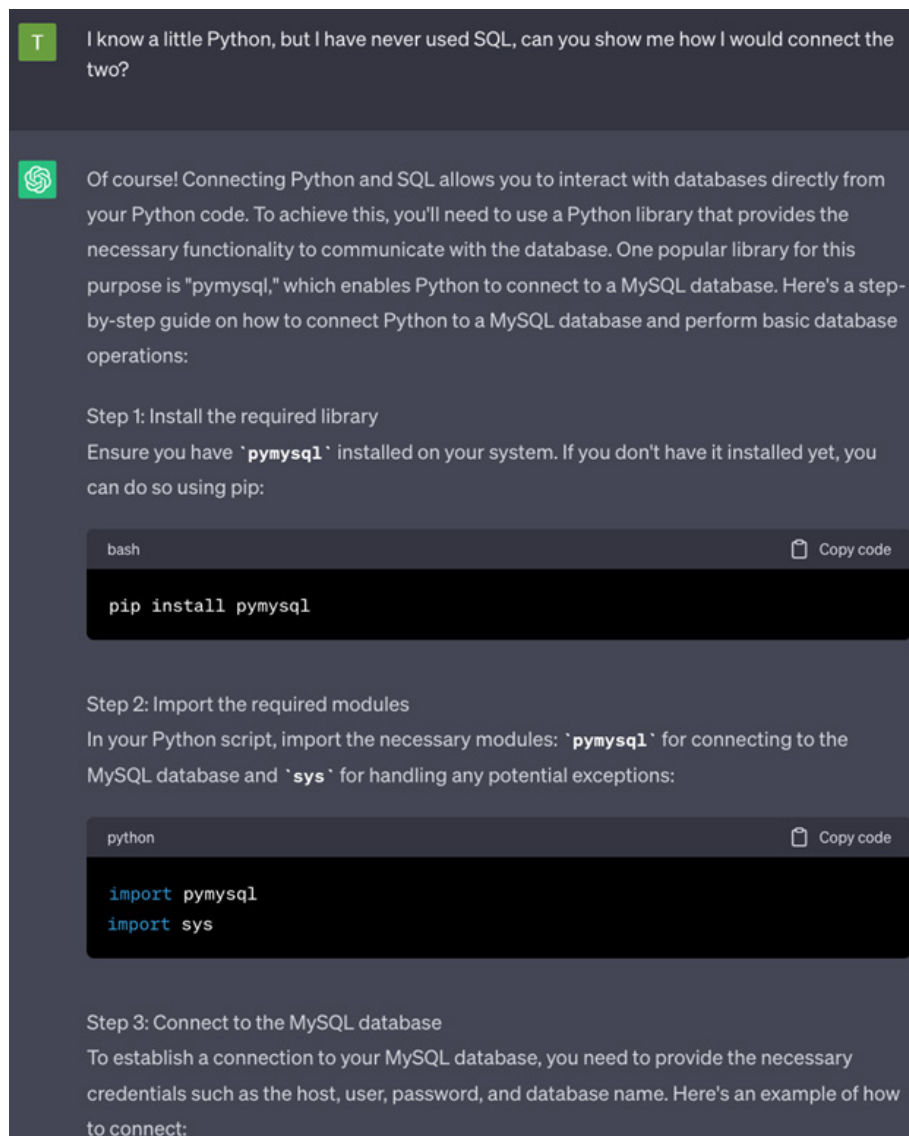
or somewhere in the middle. Or better yet, if you don't know where to start, you can ask ChatGPT.

Let's get into some actionable tips on getting started.

WHAT ARE THE DIFFERENT WAYS CHATGPT CAN HELP YOU ON YOUR CODING JOURNEY?

ChatGPT is great for providing general support as you learn to program. It can help you understand complex programming concepts by offering simple, easy-to-understand explanations. It can also suggest coding examples for you to study and experiment with.

Say you already know a little about Python, but you have never worked with SQL before. You can ask ChatGPT how to connect Python and SQL to your database and get thorough instructions that break down specifically how to install the required dependencies and set up your database in seconds.



The screenshot shows a chat interface with a dark background. On the left, a green square with a white 'T' represents the user's input. On the right, a green square with the ChatGPT logo represents the AI's response. The AI's response includes a detailed explanation of connecting Python to a MySQL database, followed by three numbered steps: 1. Installing the 'pymysql' library using pip, 2. Importing 'pymysql' and 'sys' in a Python script, and 3. Connecting to the MySQL database with credentials. Each step includes a code block with a 'Copy code' button.

T I know a little Python, but I have never used SQL, can you show me how I would connect the two?

ChatGPT Of course! Connecting Python and SQL allows you to interact with databases directly from your Python code. To achieve this, you'll need to use a Python library that provides the necessary functionality to communicate with the database. One popular library for this purpose is "pymysql," which enables Python to connect to a MySQL database. Here's a step-by-step guide on how to connect Python to a MySQL database and perform basic database operations:

Step 1: Install the required library

Ensure you have `pymysql` installed on your system. If you don't have it installed yet, you can do so using pip:

```
bash Copy code
```

```
pip install pymysql
```

Step 2: Import the required modules

In your Python script, import the necessary modules: `pymysql` for connecting to the MySQL database and `sys` for handling any potential exceptions:

```
python Copy code
```

```
import pymysql
import sys
```

Step 3: Connect to the MySQL database

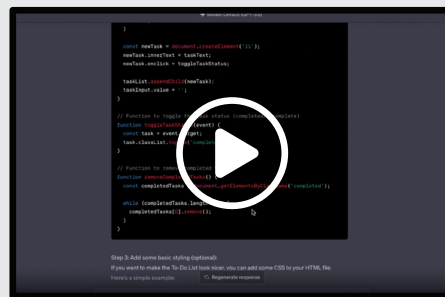
To establish a connection to your MySQL database, you need to provide the necessary credentials such as the host, user, password, and database name. Here's an example of how to connect:

PROJECT BRAINSTORMING

You've probably heard this advice echoed again and again: the best way to learn programming is by building your own projects. You can follow tutorials and watch videos all day, but putting the concepts to work in your own personal coding projects is the most reliable way to solidify your understanding of them.

The Programming QuickStart Guide Bundle contains a few easy hands-on projects to get you started with a project-based approach to coding. But once you've completed them all, branching out on your own can help you practice important coding concepts even further and build on what you're learning as a beginner.

Of course, as a beginner, it can be a little intimidating to consider all the different possibilities for projects. ChatGPT can be used as a brainstorming tool to help you come up with ideas for projects or features. Prompting it for beginner-friendly project ideas will help you narrow down your options and get started coding. Plus, you can ask clarifying questions if you get stuck, and ChatGPT will help you sort out any errors.



Watch a Quick Clip on brainstorming with ChatGPT



To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.

or

www.QuickClips.io/ChatGPT-Brainstorm

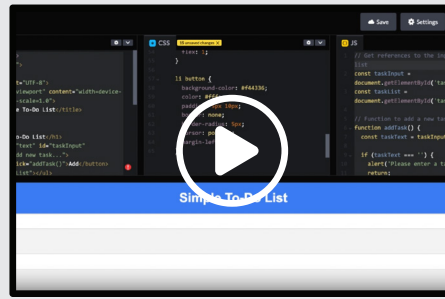


CODE SUGGESTIONS

ChatGPT can provide code snippets based on a given prompt, which can help you learn how to implement different features.

For example, if you want to find the best way to style your website but aren't sure how to create the CSS for it, you can prompt ChatGPT for some help. Inputting your existing HTML and asking it to suggest appropriate CSS can work wonders, generating all the CSS you need in a few seconds. You can ask ChatGPT what specific chunks mean, and even ask it to modify the code to fit your tastes.

If you want to learn a little more about modifying CSS to suit your needs, our *HTML & CSS QuickStart Guide* is an excellent place to start.



Watch a Quick Clip on code suggestions with ChatGPT



To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.



www.QuickClips.io/ChatGPT-Suggestion



ERROR EXPLANATION

ChatGPT is great at analyzing short pieces of code and picking out things that a human developer might not notice at first glance. This makes it a good instructor if you are trying to figure out whether you have any bugs or syntax errors in simple functions, or if you're trying to figure out why a certain line of code is not working as intended.

T can you generate code for a more complex python program and hide 1 error in there for me to find?

G Certainly! Here's a more complex Python program with a hidden error for you to find:

```
python Copy code

def calculate_average(numbers):
    total = sum(numbers)
    average = total / len(numbers)
    return average

grades = [90, 85, 92, 78, 88]
average_grade = calculate_average(grades)
print("The average grade is:", average_grade)

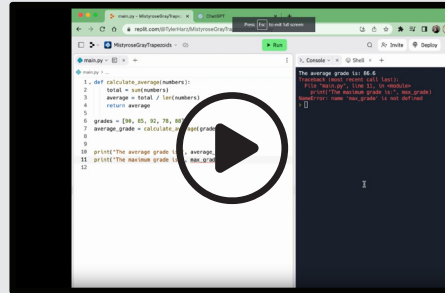
# Hidden Error: Uncomment the line below to introduce an error
# print("The maximum grade is:", max_grade)
```

In this code, we have a function called `calculate_average` that calculates the average of a list of numbers. We then have a list of grades, and we calculate the average grade using the `calculate_average` function. Finally, we print the average grade.

However, there is a hidden error in the code. To find the error, uncomment the line `print("The maximum grade is:", max_grade)` by removing the `#` at the beginning of the line. This line tries to print the value of `max_grade`, but `max_grade` is not defined anywhere in the code.

Your task is to identify and fix the error. Once you've found the error, you can share your observations, and I'll guide you on how to fix it.

And if you're trying to teach yourself how to read code more efficiently and become a better debugger, ChatGPT makes an excellent coach. Try asking ChatGPT to generate a short snippet of code containing a hidden error. Copy the sample code and paste it into your coding environment. Once you find the error, you can attempt to fix it yourself, and if you have any trouble, you can turn to ChatGPT for help.



Watch a Quick Clip on error analyzing using ChatGPT



To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.

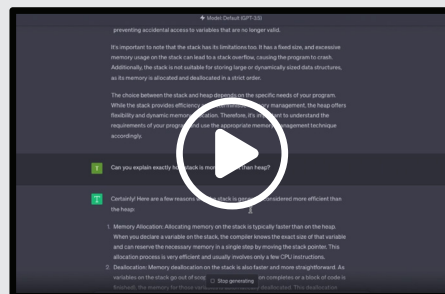
or

www.QuickClips.io/ChatGPT-Error



THEORY CLARIFICATION

ChatGPT can explain complex computer science theories and concepts in a simplified manner, making learning more accessible. Here is a quick example: Say you don't understand the difference between data structures like the stack and the heap. Query ChatGPT, and it will come to the rescue with a thorough explanation. If you want further clarification, you can prompt it to come up with a simple analogy to help you understand.



Watch a Quick Clip on using ChatGPT to explain theories and concepts







To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.

or

www.QuickClips.io/ChatGPT-Support



T I am trying to understand the difference between stack and heap, can you tell me and give me a simple analogy to help me visualize it?

 Certainly! In computer programming, the stack and the heap are two regions of memory used for different purposes. Let me explain the difference between them and provide you with a simple analogy.   

Stack:

The stack is a region of memory that is organized in a Last-In-First-Out (LIFO) manner. It is used for storing local variables, function call information, and other temporary data. When a function is called, a new stack frame is created to store its local variables and function call information, and when the function completes, the stack frame is removed.

Analogy: Think of the stack as a stack of plates in a cafeteria. When you enter the cafeteria, you take a new plate and place it on top of the stack. Similarly, when a function is called, a new stack frame is created and placed on top of the stack. When the function completes, the stack frame is removed, just like taking a plate from the top of the stack in the cafeteria.

Heap:

The heap is a region of memory used for dynamic memory allocation. It is not organized in any particular order and allows for the allocation and deallocation of memory blocks at runtime. The heap is typically used for storing objects or data that need to persist beyond the lifetime of a single function or for dynamically growing data structures like arrays or linked lists.

Analogy: Imagine the heap as a large open space where you can place objects freely. In a park, you might have an area where people can gather and place objects such as chairs, tables, or even build sandcastles. Similarly, in the heap, you can dynamically allocate memory and create objects or data structures as needed. It offers more flexibility, like having a wide open space to put objects wherever you like.

HOW TO USE CHATGPT PRUDENTLY AS A BEGINNER

Your ultimate goal should be to enhance your coding skills and your understanding of key concepts, not just to find quick solutions. To help you do that, here are a few more tips.

CRAFTING CLEAR, SPECIFIC PROMPTS

When it comes to the actual language, or *prompts*, you use to interact with your AI teacher, ChatGPT is more forgiving than your typical Stack Overflow forum and smarter than your average search engine. Unlike with some search engine interactions, adding detail to your ChatGPT prompt can make the results better rather than worse. The days of needing to carefully select your keywords and hope you land on the right result are over. Contrary to what “professional prompt engineers” might tell you, ChatGPT can usually pick up your gist even if your prompt is rambling and unclear.

But that doesn’t mean a more well-thought-out and coherent prompt won’t yield better results.

The clarity and specificity of your prompts *largely* influence the quality of responses you receive from ChatGPT. In contrast to some search engines, ChatGPT might return an answer that is confusing or irrelevant

if your question is too ambiguous or overly broad. When you're stuck on a programming problem, try to be as precise as possible, detailing the language you're using, the specific problem you're facing, and any attempts you've made to solve it. For example, you might be tempted to ask, "How do I sort data?" But a better question is "How can I sort a list of dictionaries by a specific key in Python?"

UNDERSTANDING OVER IMITATION

Keep in mind that relying heavily on AI for programming solutions might lead to a superficial understanding of the subject. While ChatGPT can provide code snippets, it's your responsibility to dissect, understand, and then implement them into your work. Merely copying and pasting code without comprehending its function won't enhance your skill set.

Used correctly, however, ChatGPT isn't a crutch. Instead, it becomes your greatest ally. You can always ask for further clarification on a given code snippet. Besides helping you gain a deeper understanding of the material, ChatGPT can guide you through each part of the code and even improve it and help you put your own touch on it.

The key to mastering programming is practice and experimentation. Use ChatGPT as a guide when you get stuck, but strive to solve problems yourself. This approach helps you better understand the underlying logic and concepts, rather than receiving ready-made solutions that may not apply to future problems.

DIFFERENCES BETWEEN FREE AND PAID VERSIONS

There are distinct differences between the free and paid versions of ChatGPT. The most notable distinction is that the free version allows you to access only the GPT 3.5 model, whereas the paid version offers both GPT 3.5 and GPT 4. The latter is a substantially more powerful language model, with a more extensive training set and a higher token limit.

However, this matters much more for creative writing than for programming. GPT 4 might have better prose than GPT 3.5, but both of them display sufficient competence when it comes to entry-level programming knowledge. If you're learning programming, the free version is enough to get you started. Since OpenAI has scaled up its cloud-computing resources, we don't have to deal with frequent outages like we did when the service first launched. This means it's no longer necessary to upgrade to the paid plan to ensure constant access.

Despite this, there is still a reason to consider the paid version of ChatGPT: plugins. Since the most recent update, the paid plan offers a growing library of third-party plugins with a wide range of features. Perhaps the most helpful plugin for a coding novice is a search engine integration, allowing you to send your prompt to Google for a more fact-checked result.

For more advanced programmers, another welcome feature lies within ChatGPT's paid version: Code Interpreter. Although it's still in beta, this addition to the chatbot lets you upload your own files for data analysis, data visualization, and even file conversion.

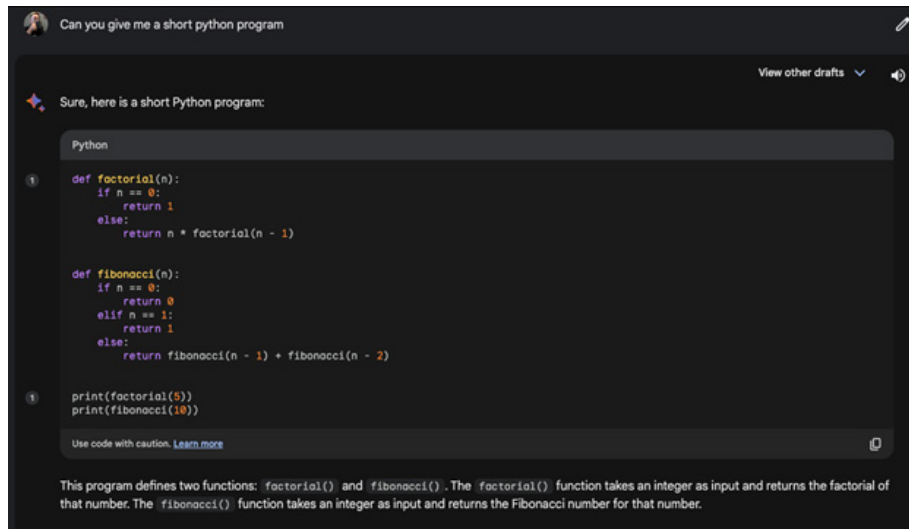
CHAT GPT VS. COMPETITORS LIKE BARD, CLAUDE, OR GITHUB COPILOT

ChatGPT is not alone in the wave of new technologies we're seeing every day. In fact, it wasn't even the first. There are a few competitors that can give ChatGPT a run for its money, especially when it comes to coding.

GOOGLE BARD

The most prominent competitor to ChatGPT in the current landscape is definitely [Google Bard](#). Prompting it with "Can you give me a short Python program" returned a different, more complex program than ChatGPT provided when given the same prompt.

Bard also goes on to provide specific directions for executing the program, by saving it as a .py file and executing it from the command line.



The screenshot shows a chat window with the prompt "Can you give me a short python program". The response is a Python program with two functions: factorial() and fibonacci(). The factorial function uses a recursive approach, and the fibonacci function uses a recursive approach as well. The program also includes print statements to demonstrate the functions. Below the code, there is a warning to use code with caution and a link to learn more.

```
Python

def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

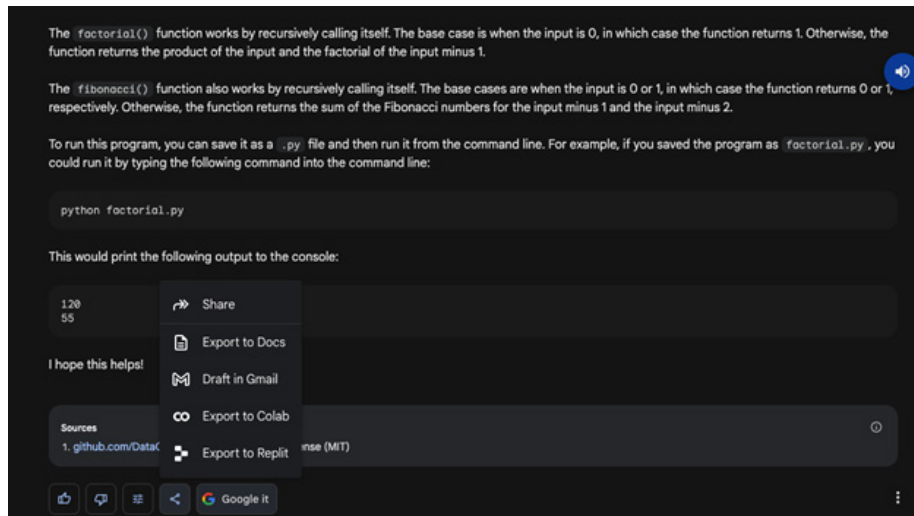
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

print(factorial(5))
print(fibonacci(10))
```

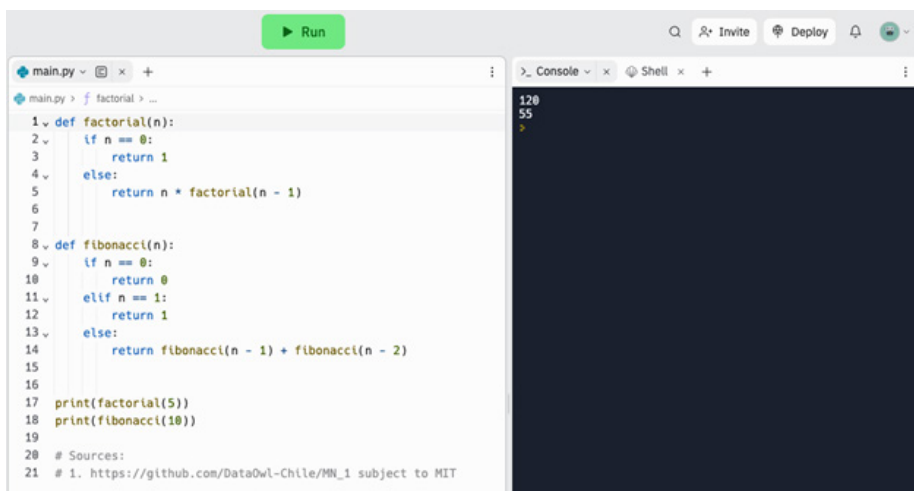
Use code with caution. [Learn more](#)

This program defines two functions: `factorial()` and `fibonacci()`. The `factorial()` function takes an integer as input and returns the factorial of that number. The `fibonacci()` function takes an integer as input and returns the Fibonacci number for that number.

What really sets Bard apart is its **Export to Replit** feature. You can share Bard's code output to [Replit](#), a browser-based IDE, where you can then compile and run the code.

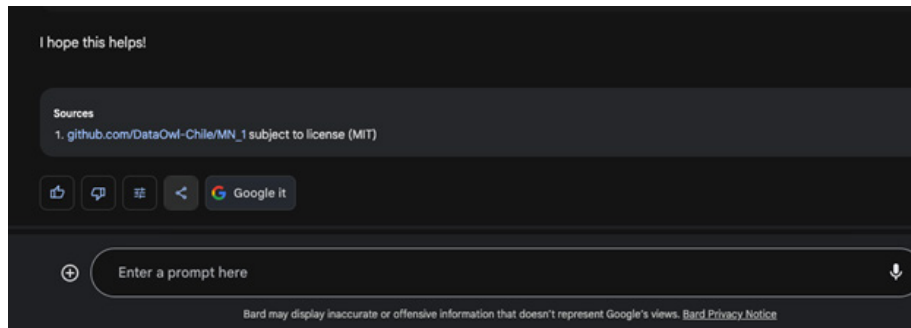


As we can see here, the simple Python program Bard generated for us works just fine in Replit.



When comparing different AI chatbots for the purpose of learning to program, you want to look at one factor in particular: factual accuracy. You don't want to deal with a language model that makes nonsensical errors disguised as fact, a phenomenon known as "hallucinating." This is a serious issue wherein an AI will fill in any missing information with a guess or an outright lie, all while packaging it as fact. Luckily, this problem is growing rarer by the day as models continuously improve.

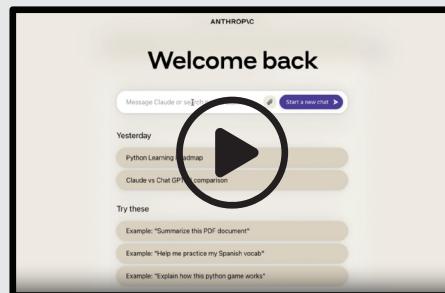
While Bard has been criticized for generating excessive hallucinations, Google offers a unique feature allowing the user to combat this directly. The "Google It" button is featured prominently above the search bar, so you can check the internet to determine for yourself if the output is accurate.



CLAUDE

[Claude](#) is another popular alternative to ChatGPT. It was first introduced by Anthropic in early 2023 and has since undergone a major overhaul to become Claude 2, which was released in July.

Claude is similar to ChatGPT with a few noticeable differences. It has [tested slightly better](#) than ChatGPT on simple coding tests, and it offers more up-to-date information, given that its training set cuts off in early 2023 instead of 2021. Beginners will notice a similar level of coding knowledge and language abilities.



Watch a Quick Clip on Claude



To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.

or

www.QuickClips.io/ChatGPT-Claude



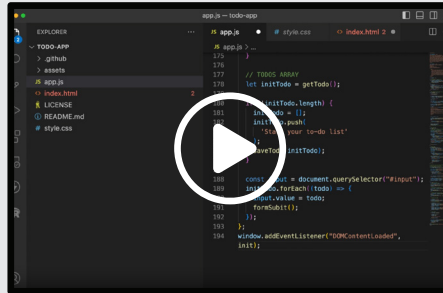
GITHUB COPILOT

GitHub [Copilot](#) is notably different from ChatGPT and Bard in that it is not a chatbot but rather an AI coding assistant that you install as an extension in your development environment. You can use this AI without ever leaving the confines of your favorite IDE.

It is designed less as a tool for beginners and more as a workflow upgrade for intermediate developers looking to increase their productivity. Copilot won't teach you basic programming concepts, and it won't hold your hand as

you ask questions and work through a tutorial. However, it will point you in the right direction if you are stuck on determining the specific function or class to define when writing your code.

To interact with Copilot, you can start commenting, and you'll see its generative AI prowess come to life as you type. Copilot's greatest strength is filling in common blocks of code, such as defining a class, calling a function, or just generating some boilerplate.



Watch a Quick Clip on Copilot



To watch the Quick Clip, use the camera on your mobile phone to scan the QR code or visit the link below.

or

www.QuickClips.io/ChatGPT-Copilot



THE FUTURE OF AI IN THE PROGRAMMING WORLD

Many developers who are just now entering the field are worried that AI will advance to the point of replacing human programmers and eliminating the jobs they worked so hard to get. The truth is that at this point we can't say anything with certainty.

But one thing that seems likely to remain is the need for a human driver to lead the computer. The software development life cycle is highly complex, and the intricacies of system design and architecture, coupled with dealing with human communication and emotions, is a tall order for current AI.

Still, AI is changing the landscape of programming. Newer programming languages and frameworks have come along and improved upon older ones and removed the drudgery and hassle of performing manual or repetitive tasks. The same holds true with AI.

Language models like ChatGPT are just a tool to speed up workflows and abstract away some of the difficult or mind-numbingly repetitive tasks. The optimistic programmer knows that AI won't replace humans but instead will make them even more productive and efficient.

Programmers who adapt and learn how to use AI to their advantage will be in the best position to succeed. And using ChatGPT to learn programming is just one way in which you're already ahead of the game.