

Is ChatGPT or Gemini an AI chatbot that writes better code?

If you're stuck in a programming project, you might be looking for a tool to help you brainstorm ideas, write clear code, or explain a complex concept.

Which AI chatbot do you choose: the agile and informative Gemini or the comprehensive and powerful ChatGPT?

Language support

When it comes to language support, ChatGPT is superior to Gemini. While Gemini officially supports around 22 popular programming languages ??- including Python, Go and TypeScript - ChatGPT's language capabilities are much more extensive.

Unlike Gemini, ChatGPT does not have an official list of supported languages. However, not only can it handle the popular languages ??supported by Gemini, but it can also handle dozens of additional languages, from newer ones like TypeScript and Go to older languages ??like Fortran , Pascal and BASIC.

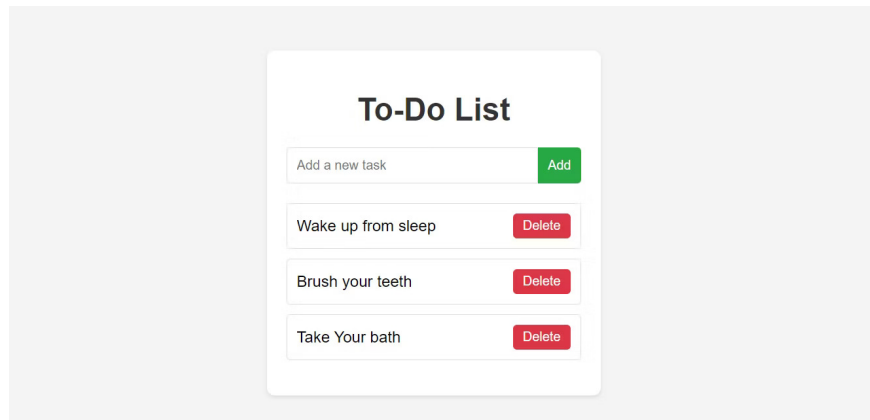
To test their language capabilities, the author tried simple coding tasks in languages ??like PHP, JavaScript, BASIC and C++. Both Gemini and ChatGPT work well with common languages, but only ChatGPT can convincingly put together programs in older languages ??like BASIC.

Accuracy and code quality

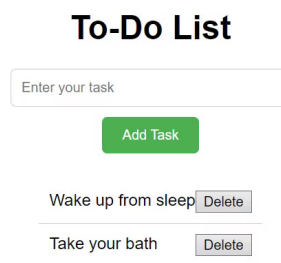
You're behind on a project deadline and need some boilerplate code. When asking ChatGPT and Gemini to generate code to implement that functionality, both tools generated dozens of lines of code.

But which tool's code can you trust to use for the functionality you require? To compare the accuracy and quality of the code generated by two AI chatbots, the author assigned them a simple task. The author asked Gemini and ChatGPT to create a simple to-do list application using HTML, CSS, and JavaScript, without providing any primer information; The goal was to see how well both chatbots could perform with limited information at work.

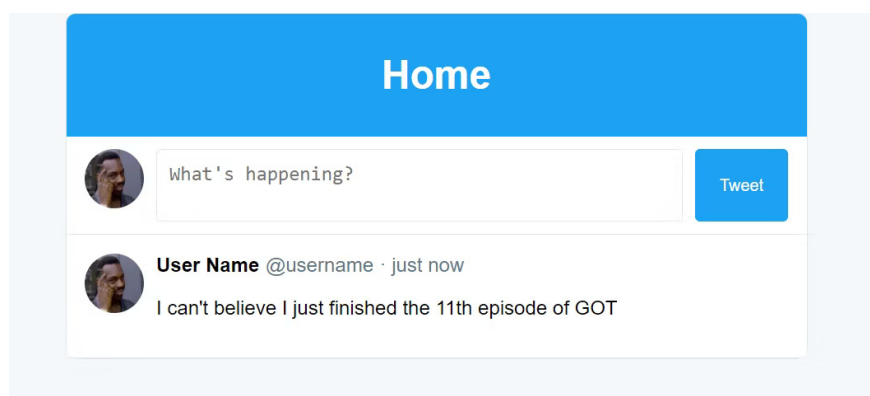
ChatGPT (GPT-4o) created functional code that was aesthetically "good enough". Using ChatGPT's code, you can add or remove tasks. This is what we get after running the results of ChatGPT in the browser:



Next, ask Google's Gemini to repeat the same task. Gemini can also create a functional to-do list app. You can also add and remove tasks but the overall design isn't as attractive:



The author performed a second test, this time asking both chatbots to recreate the Twitter (X.com) feed. ChatGPT has created a classic style Twitter feed with a functional tweet feature. You can type into the text box, send a tweet and upload it to the page in a flexible way. It's not the expected Twitter feed, but since most of ChatGPT's training data is filled with old Twitter code, the results are understandable.



Unfortunately, in this round, Google's Gemini was unable to provide functional code. It generates hundreds of lines of JavaScript code, but there are too many placeholders that need to be filled in without logic. If you are in a hurry, coding so many placeholders will not be useful, as it still requires heavy development work. In such cases, writing the code from scratch may be more effective.

The author tried several other basic encryption tasks and in all cases, ChatGPT's solution was clearly the better choice.

Troubleshoot and detect errors

Errors are like puzzles that programmers hate. They will drive you crazy, but fixing them is also quite satisfying. So, when you encounter an error in your code, should you turn to Gemini or ChatGPT for help? The answer depends on the type of error you are trying to avoid.

The author gave both AI chatbots 2 problems to debug. First, the author asked both chatbots to resolve logic errors in some simple PHP code. Logic errors are notoriously harder to detect than syntax errors because they depend on the purpose of the code:

```
php Copy code  
  
$a = 10;  
$b = 5;  
$c = 2;  
  
if ($a > $b && $a > $c) {  
    echo "a is the largest number.";  
} elseif ($b > $a && $b > $c) {  
    echo "b is the largest number.";  
} else {  
    echo "c is the largest number.";  
}
```

The code in this screenshot runs and even produces correct results in many cases. However, it has some logic errors that are not immediately apparent; Can you recognize them? With the help of Gemini, unfortunately, the chatbot could not find the logic error in the code:

◆ No, the code you provided has no functional or logical errors. It correctly determines the largest number and prints the corresponding message based on the assigned values.

As you pointed out, the conditions within the if-else statements accurately identify the largest number. The code functions as intended without any unexpected behavior.

All three of Gemini's attempts to solve the problem were incorrect. Tried a similar problem 6 months ago with similarly disappointing results; It seems that Gemini is not yet more advanced in this area.

The author then asked ChatGPT for help and it immediately detected the logic error.

🌀 **ChatGPT**

The given code has a logical flaw in the conditional checks. Specifically, it fails to correctly handle cases where two or more variables have the same value and should be considered the largest. The conditions only account for each variable being strictly greater than the other two, which can result in incorrect conclusions when variables are equal.

Gemini also rewrote the code to fix the error:

```
$a = 10;
$b = 5;
$c = 2;

if ($a >= $b && $a >= $c) {
    if ($a == $b && $a == $c) {
        echo "a, b, and c are equal and are the largest numbers.";
    } elseif ($a == $b) {
        echo "a and b are equal and are the largest numbers.";
    } elseif ($a == $c) {
        echo "a and c are equal and are the largest numbers.";
    } else {
        echo "a is the largest number.";
    }
} elseif ($b >= $a && $b >= $c) {
    if ($b == $c) {
        echo "b and c are equal and are the largest numbers.";
    } else {
        echo "b is the largest number.";
    }
} else {
    echo "c is the largest number.";
}
```

After trying several other debugging and debugging tasks, ChatGPT clearly did a better job. However, Gemini is not completely useless. It can fix a lot of syntax errors, but has difficulty with complex errors, especially logic errors.

Context awareness

One of the biggest challenges when using AI chatbots for coding is their relatively limited context-aware capabilities. They can create separate pieces of code for clearly defined tasks but have difficulty building a code base for a larger project.

For example, let's say you are building a web application with an AI chatbot. You tell it to code the registration and login HTML page, then it does it perfectly. You then ask the chatbot to create a server-side script to handle the login logic. This is a simple task, but due to limited context awareness, it can end up creating a login script with new variables and naming conventions that don't match the rest of the code.

Which chatbot is better at maintaining contextual awareness? The author of the article assigned both tools the same programming task: Create a chat application.

Since GPT-4 Turbo arrived with its 128k context window, ChatGPT's ability to retain more context over long periods of time has increased significantly. When first building a chat application with ChatGPT using the GPT-4 4k context window, it ran relatively smoothly, with the only minor issues with context redirects.

Recreating the same project in November 2023 with GPT-4 Turbo 128k showed a clear improvement in context awareness. 6 months later, in May 2024, there has not been any significant change in context awareness but there has also been no decline.

Unfortunately, when trying Gemini (then called Bard) for the first time on the same project, it lost track of the project's context and couldn't complete the application. A few updates later, when testing Gemini again on the same project, it seemed to have degraded further. So, once again, in terms of context awareness, ChatGPT wins.

Problem solving

At this point, Google's Gemini is lacking in a lot of ways. But can Gemini win in the end? Let's test its problem-solving capabilities. Sometimes you have a problem but aren't sure how to programmatically represent it, let alone how to solve it.

In these situations, chatbots like Gemini and ChatGPT can be helpful. The author asked both of them to "Write a JavaScript code that counts the number of times a specific word appears in the text".

Here are the results from Google's Gemini:

```
function countWord(text, word) {
  // Convert both text and word to lowercase for case-insensitive counting
  text = text.toLowerCase();
  word = word.toLowerCase();

  // Split the text into words using a regular expression for whitespace
  const words = text.split(/\s+/);

  // Filter the words array to count only occurrences of the target word
  const wordCount = words.filter(w => w === word).length;

  return wordCount;
}

// Example usage
const text = "This is a sample text to count words. This sentence is repeated."
const wordToCount = "This";

const wordCount = countWord(text, wordToCount);

console.log(`The word "${wordToCount}" appears ${wordCount} times in the text.`);
```

And here is the result from ChatGPT:

```
function countWordOccurrences(text, word) {
  // Convert the text and word to lowercase to make the search case-insensitive
  const lowerCaseText = text.toLowerCase();
  const lowerCaseWord = word.toLowerCase();

  // Split the text into an array of words using a regular expression
  const wordsArray = lowerCaseText.split(/\W+/);

  // Initialize a counter for the occurrences of the word
  let count = 0;

  // Iterate through the array and count the occurrences of the word
  for (const w of wordsArray) {
    if (w === lowerCaseWord) {
      count++;
    }
  }

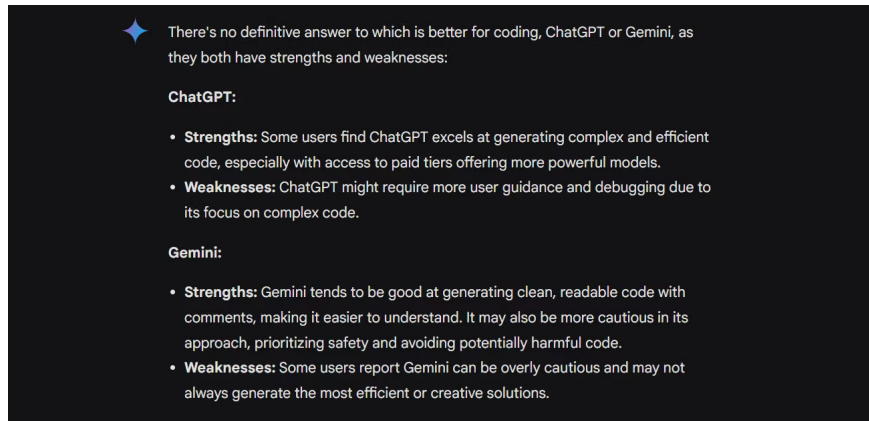
  return count;
}

// Example usage
const text = "The quick brown fox jumps over the lazy dog. The dog was not amused.";
const word = "the";
const occurrences = countWordOccurrences(text, word);
```

At first, both approaches look quite reliable. Gemini's approach even seems brief. However, ChatGPT's code has a more powerful and accurate approach to counting the number of word occurrences in text. It considers word boundaries and is case sensitive, handles punctuation properly, and gives more reliable results. Once again, ChatGPT comes out on top.

ChatGPT's approach breaks the input text into words in a way that can handle all non-word characters such as punctuation and special characters as word delimiters. Meanwhile, Gemini only considers spaces as delimiters. This approach may fail if the text contains punctuation or other non-word characters within words, or if words are not separated by space characters.

Since Google Gemini lost pretty much every metric used for comparison, the author decided to give it a chance to redeem itself. The author asked the chatbot "Which is better at viewing code? ChatGPT or Gemini?" Here is its answer:



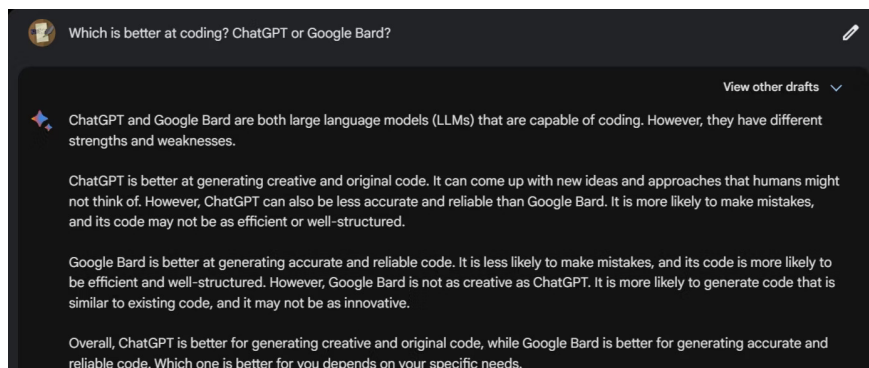
The author then asked ChatGPT what it thought about this review and it agreed:

Agreement and Perspective

The assessment seems accurate and reflects the different design philosophies and strengths of both models. Here's a synthesis of the comparison:

- **ChatGPT** is highly capable of generating complex and efficient code, making it suitable for users who need powerful coding solutions and are comfortable with guiding and refining the output. Its versatility across various programming scenarios is a significant advantage.
- **Gemini** excels in producing clean and readable code, with an emphasis on safety and understandability. This can be especially useful for educational purposes, code maintainability, and for users who prefer straightforward solutions.

While this seems pretty normal, there's an interesting twist here. Over the last year, Gemini (then Bard) has always confidently asserted that he can create good, efficient code with fewer errors. Here is a screenshot from one of the tests in November 2023:



It seems like Gemini is a bit more self-aware and humble now!

Programming features

Neither ChatGPT nor Gemini have major programming-specific features. However, both chatbots have features that can greatly enhance your programming experience if you know how to use them effectively.

ChatGPT offers a series of features that can streamline the programming process when using chatbots. Useful additions like Memory and Custom GPT let you customize ChatGPT for your specific programming needs.

For example, the Custom GPT feature can help create small specialized ChatGPT versions for specific projects by uploading relevant files. This makes tasks like debugging code, optimizing, and adding new features much simpler. Overall, compared to Google's Gemini, ChatGPT includes more features that can enhance the programming experience.

You finished reading the article "**Is ChatGPT or Gemini an AI chatbot that writes better code?**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.