Santiago
@svpino

11 ways ChatGPT saves me hours of work every day, and why you'll never outcompete those who use AI effectively.

A list for those who write code:

1 of 16

11.9k Likes **2.4k** Retweets **425** Replies

Santiago
@svpino

1. Explaining code

Take some code you want to understand and ask ChatGPT to explain it.

I've found explanations are very detailed. This is much quicker than trying to figure out convoluted code.

2 of 16

Santiago
@svpino

2. Improve existing code

Ask ChatGPT to improve existing code by describing what you want to accomplish.

It will give you instructions about how to do it, including the modified code.

3 of 16

I want to raise an exception if the authentication is incorrect

To raise an exception if the authentication is incorrect, you can check the `status_code` attribute of the `response` object. If the `status_code` is `401 Unauthorized`, it means that the authentication was incorrect.

Here is an example of how you can modify the function to raise an exception in this case:

Copy code

```
def get_issue(issue, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/issues/{issue}"
    response = requests.request(
        "GET", url, headers=headers, auth=(GITHUB_USERNAME, GITHUB_TOKEN)
    )

    if response.status_code == 401:
        raise Exception("Authentication failed")

    response_json = json.loads(response.text)

    title = response_json["title"]
    body = response_json["body"]

    return {
        "title": title,
        "body": body,
    }
```

This will raise an exception with the message "Authentication failed" if the `status_code` is `401`. You can customize the exception message and type as needed.

Santiago
@svpino

3. Rewriting code using the correct style

This is great when refactoring code written by non-native Python developers who used a different naming convention.

Notice how ChatGPT not only gives you the updated code; it also explains the reason for the changes.

4 of 16

Rewrite this function following Python's naming conventions:

```
def GetISSueFromGitHub(Issue, Owner, Repo):
  Url = f"https://api.github.com/repos/{owner}/{repo}/issues/{issue}"
  response = requests.request(
    "GET", Url, headers=headers, auth=(GITHUB_USERNAME, GITHUB_TOKEN)
  )

  ResponseJson = json.loads(response.text)
  return {
    "title": ResponseJson["title"],
    "body": ResponseJson["body"],
  }
```

Here is the function rewritten to follow Python's naming conventions:

Copy code

```
def get_issue_from_github(issue, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/issues/{issue}"
    response = requests.request(
        "GET", url, headers=headers, auth=(GITHUB_USERNAME, GITHUB_TOKEN)
    )

    response_json = json.loads(response.text)
    return {
        "title": response_json["title"],
        "body": response_json["body"],
    }
```

In Python, it is recommended to use lowercase letters and underscores for function and variable names. It is also recommended to use meaningful names that accurately describe the purpose of the function or variable. In this case, the function has been renamed to `get_issue_from_github`, and the variables have been renamed to `issue`, `owner`, and `repo`. The constants `GITHUB_USERNAME` and `GITHUB_TOKEN` have not been changed, as they are already in the recommended style.

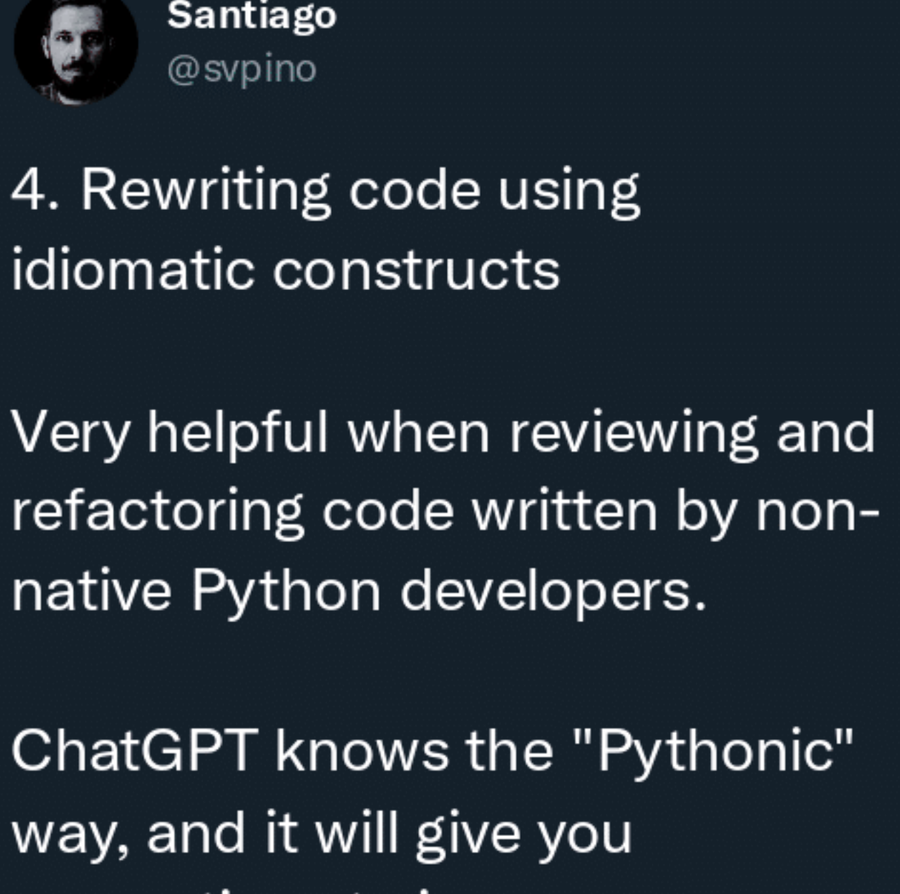
Santiago
@svpino

4. Rewriting code using idiomatic constructs

Very helpful when reviewing and refactoring code written by non-native Python developers.

ChatGPT knows the "Pythonic" way, and it will give you suggestions to improve your code and make it much more readable.

5 of 16



Santiago
@svpino

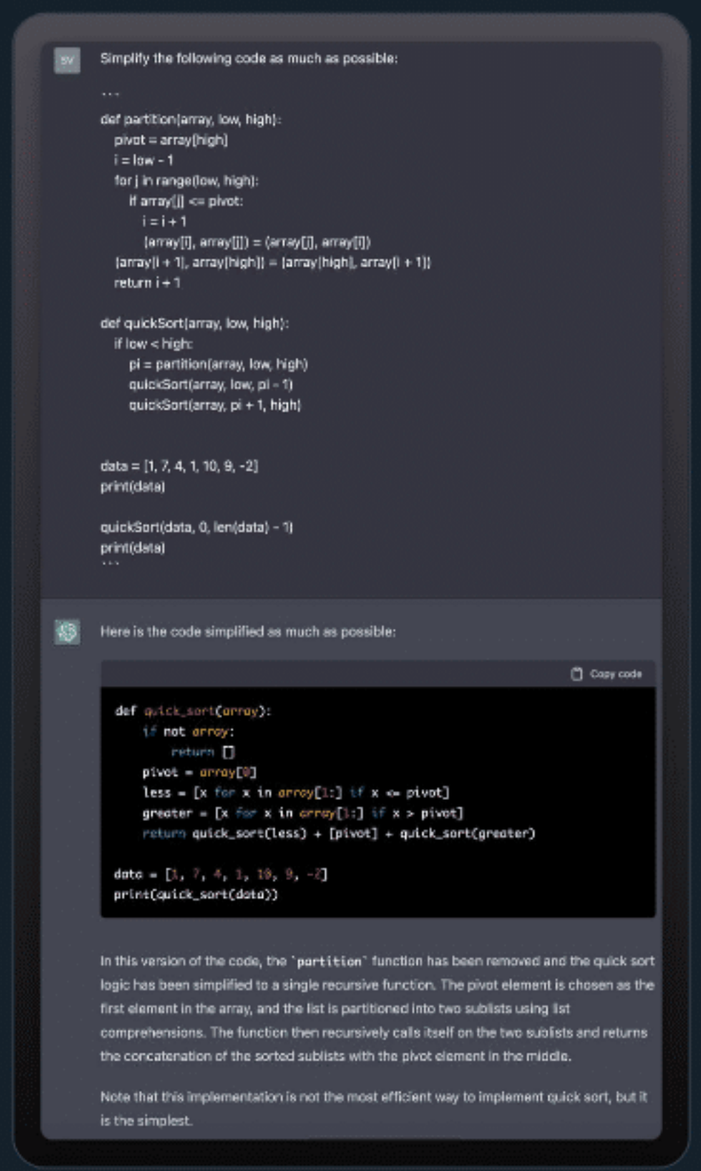
5. Simplifying code

This is one of my favorite tricks: Ask ChatGPT to simplify complex code.

The result will be a much more compact version of the original code.

Notice the explanation and how it tells us this is simpler but not the most efficient.

6 of 16

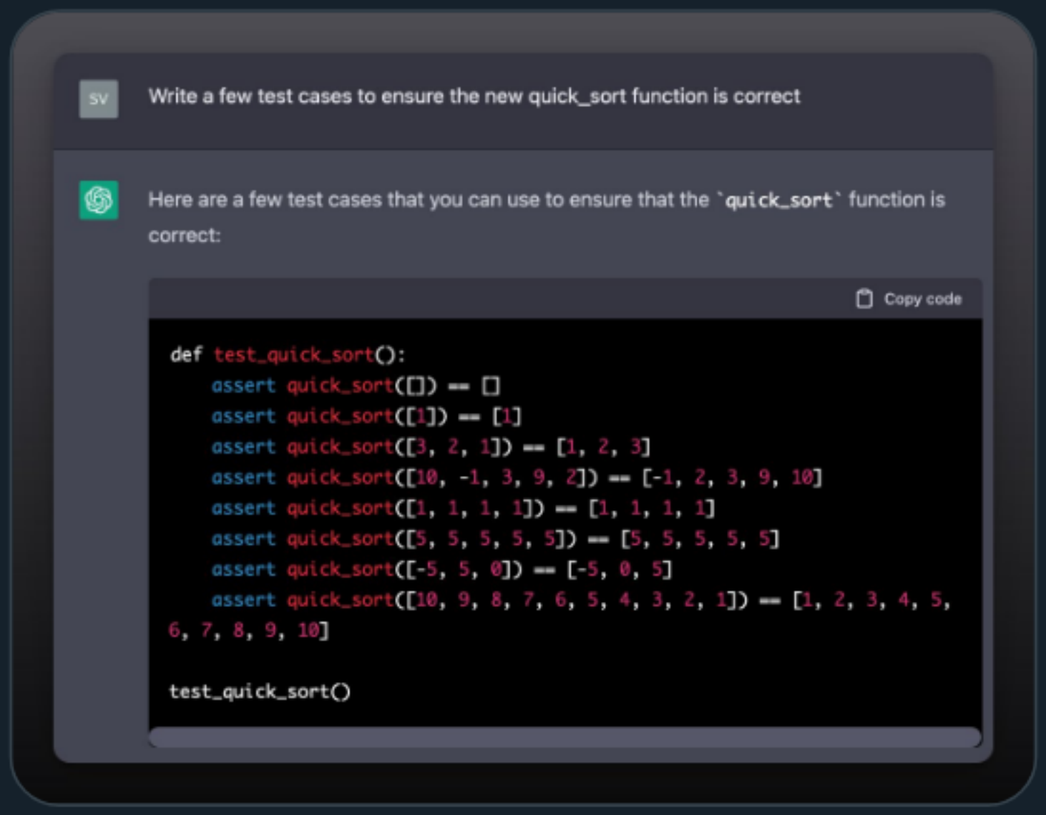


Santiago
@svpino

6. Writing test cases

This has become one of my favorite ChatGPT abilities: Ask it to help you test a function, and it will write test cases for you.

This example focuses on the quick_sort function from the previous example.



Santiago
@svpino

7. Exploring alternatives

ChatGPT told me its Quick Sort implementation wasn't the most efficient, so I asked for an alternative implementation.

This is great when you want to explore different ways to accomplish the same thing.

I've learned a ton from this!

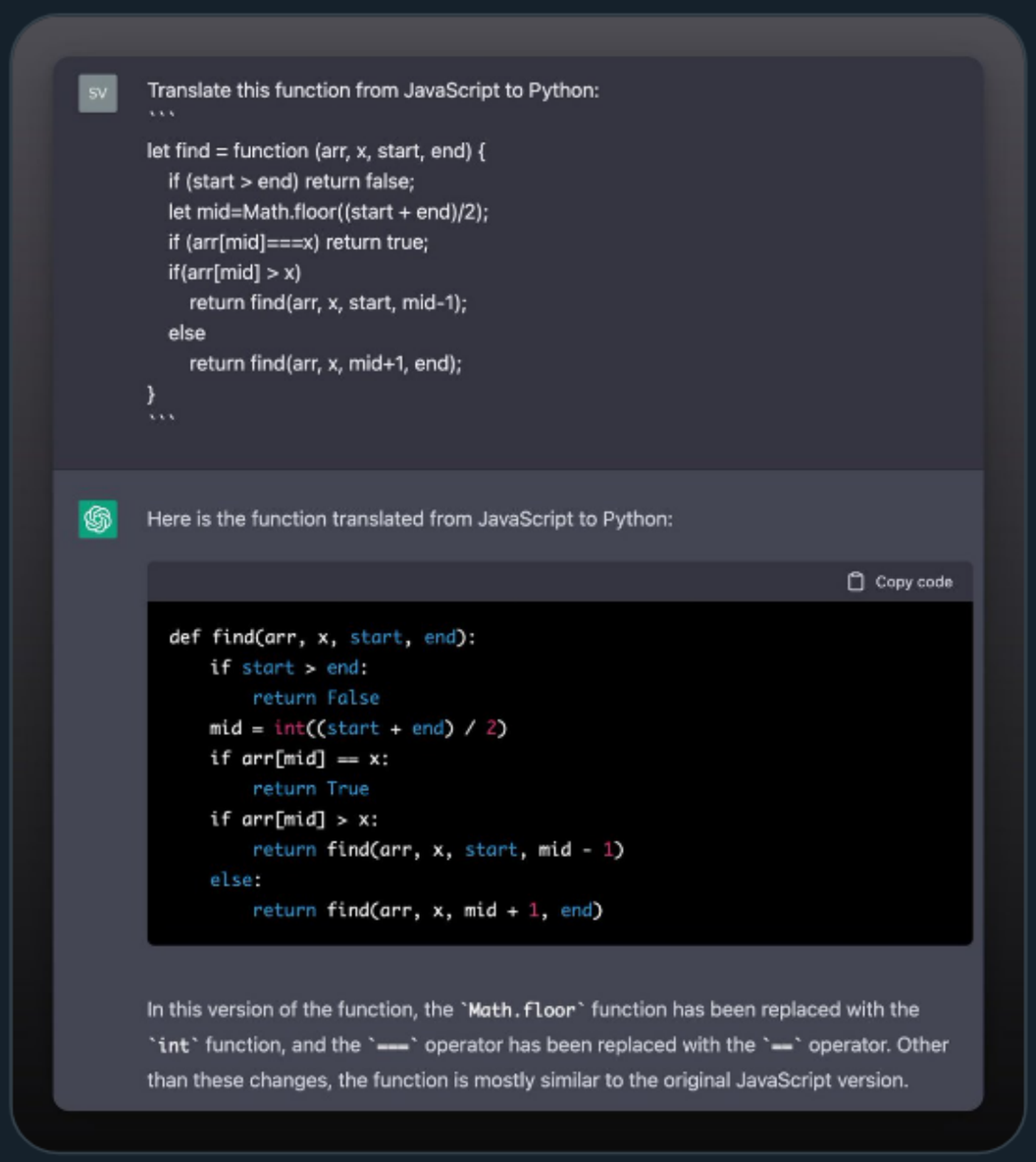
8 of 16

Santiago
@svpino

8. Translating code

Anytime you want to port some code from one language to another, ask ChatGPT to help you.

9 of 16



Santiago
@svpino

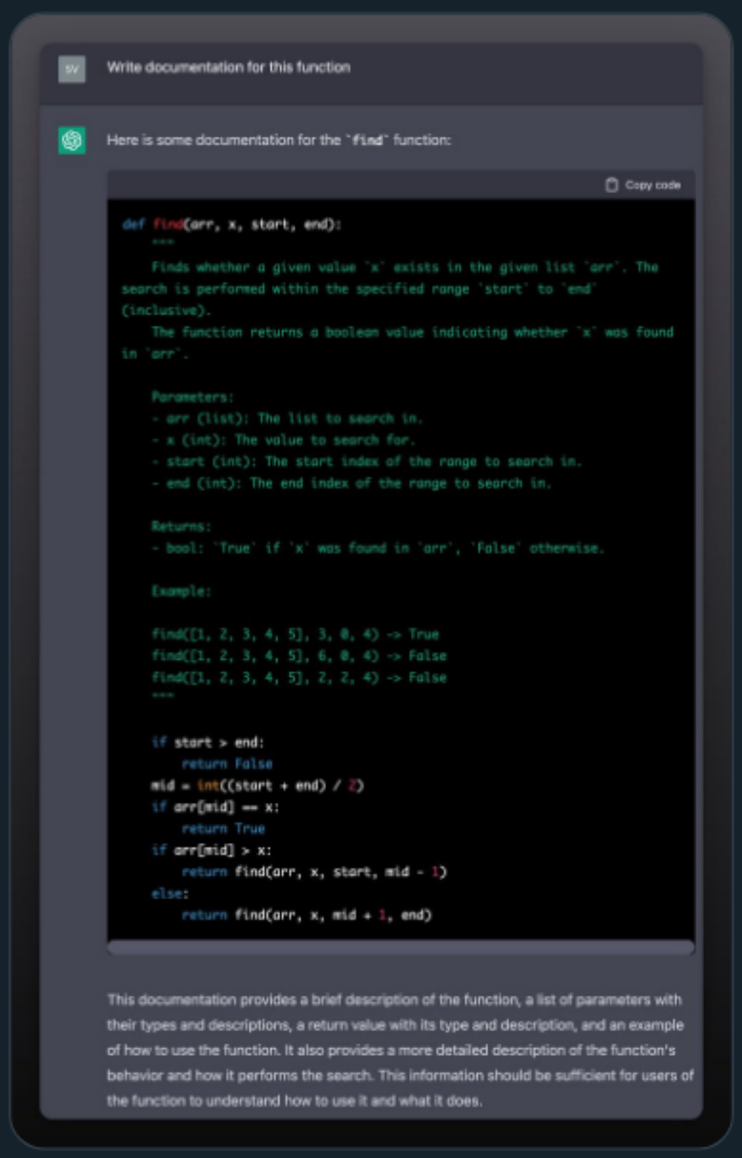
9. Writing documentation

This is another one of my favorite tricks.

Ask ChatGPT to write the documentation for a piece of code, and it usually does a great job.

It even includes usage examples as part of the documentation!

10 of 16



Santiago
@svpino

10. Tracking down bugs

If you are having trouble finding a bug in your code, ask ChatGPT for help.

It took ChatGPT seconds to find the bug in the attached example. I don't know about you, but it would have taken me much longer than that.

11 of 16

SV Find the bug in the following function that finds whether a supplied value x exists in an array.

```
def find(arr, x, start, end):
  if start > end:
    return False
  mid = int((start + end) / 2)
  if arr[mid] == x:
    return True
  if arr[mid] > x:
    return find(arr, x, start, mid)
  else:
    return find(arr, x, mid + 1, end)
```

The bug in the `find` function is that it does not update the end index of the range when calling itself recursively with the `arr`, `x`, `start`, and `mid` arguments. This can cause the function to enter an infinite loop if the value of `mid` is not updated.

To fix this bug, you can update the end index of the range to `mid - 1` when calling the function recursively with the `arr`, `x`, `start`, and `mid` arguments, like this:

Copy code

```
def find(arr, x, start, end):
    if start > end:
        return False
    mid = int((start + end) / 2)
    if arr[mid] == x:
        return True
    if arr[mid] > x:
        return find(arr, x, start, mid - 1)
    else:
        return find(arr, x, mid + 1, end)
```

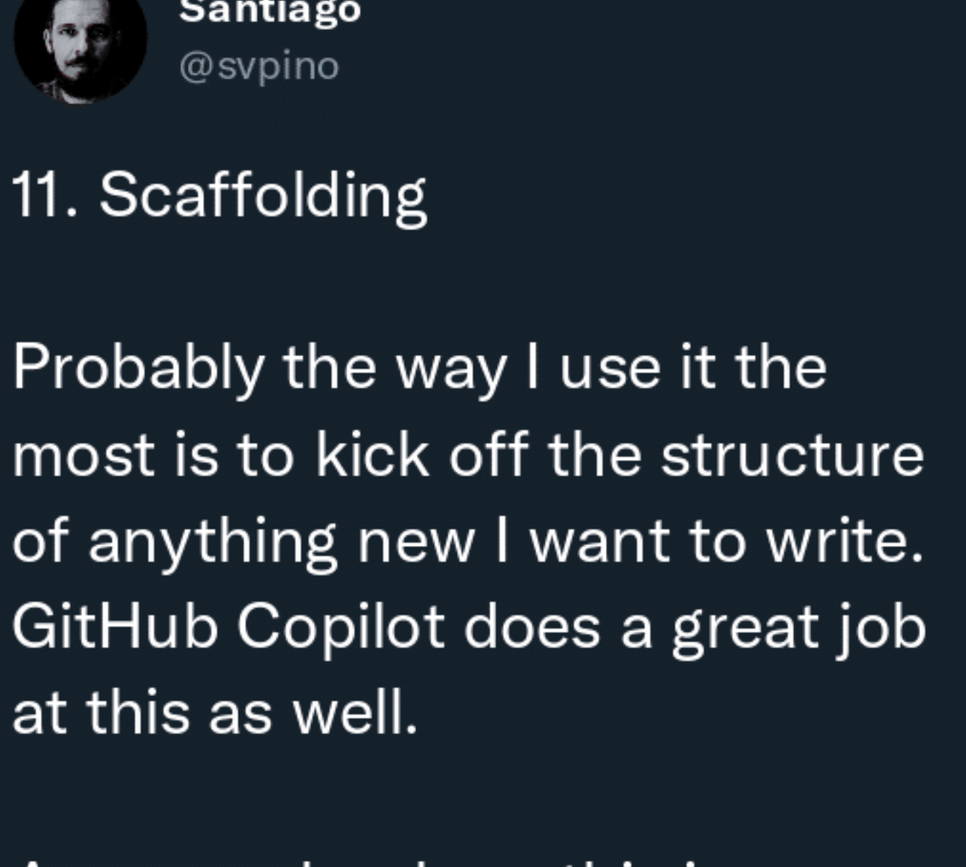
This fixes the bug by correctly updating the end index of the range when the function is called recursively. The function will now terminate correctly and return the correct result.

Santiago
@svpino

11. Scaffolding

Probably the way I use it the most is to kick off the structure of anything new I want to write. GitHub Copilot does a great job at this as well.

An example where this is very useful is when interacting with a RESTful API.



Santiago
@svpino

Something to keep in mind:

I have 2+ decades of programming experience. I like to think I know what I'm doing.

I don't trust people's code (especially mine,) and I surely don't trust ChatGPT's output.

13 of 16

321 Likes **14** Retweets **3** Replies

Santiago
@svpino

This is not about letting ChatGPT do my work. This is about using it to 10x my output.

ChatGPT is flawed. I find it makes mistakes when dealing with code, but that's why I'm here: to supervise it.

Together we form a more perfect Union. (Sorry, couldn't help it)

14 of 16

458 Likes **32** Retweets **12** Replies

Santiago
@svpino

Developers who shit on this are missing the point.

The story is not about ChatGPT taking programmers' jobs. It's not about a missing import here or a subtle mistake there.

The story is how, overnight, AI gives programmers a 100x boost.

Ignore this at your own peril.

15 of 16

485 Likes **45** Retweets **14** Replies