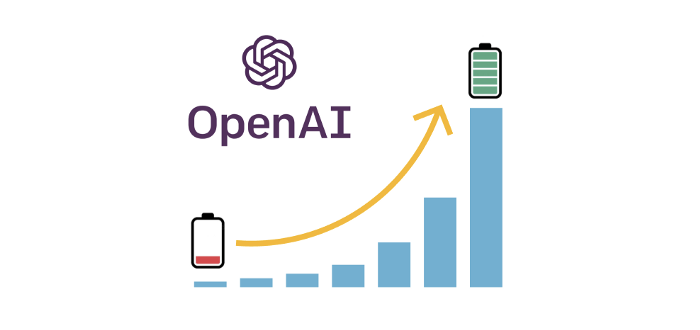
**5 ChatGPT features to boost your daily work**

And how to enhance your code quality using it



Self-made image.

**ChatGPT has completely changed the way to develop code.** However, most software developers and data professionals still do not use ChatGPT to improve — and ease — their work.

This is why I am outlining here 5 different features to increase both velocity and quality in your daily work.

You can use them in your day-to-day work. Let’s discover together how 👇🏻

*⚠️ Warning: You should never use critical code or information with ChatGPT.*

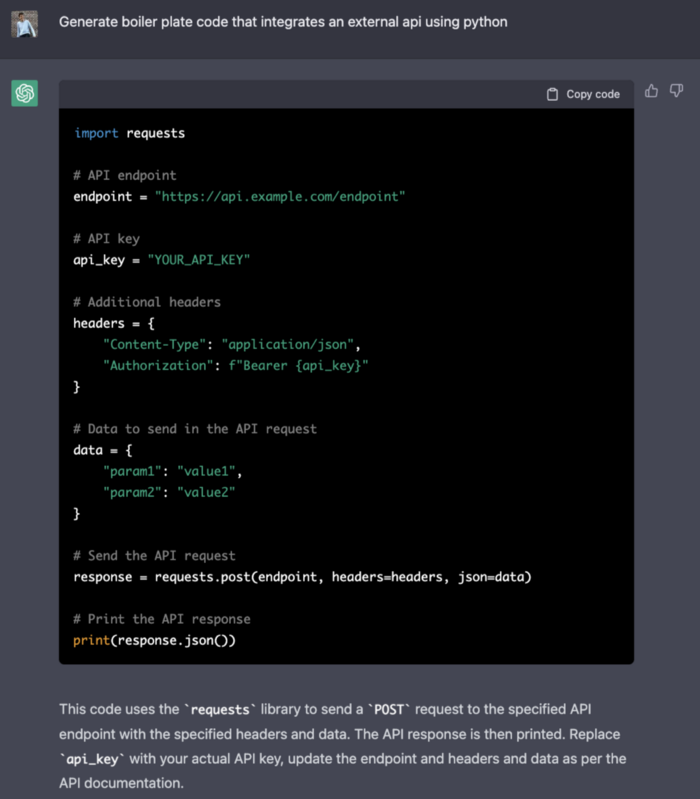
**#1. Generate your coding skeleton**

ChatGPT is my secret weapon when it comes to building new projects from scratch. With just a few prompts, it can generate a skeleton of the code I need, complete with my chosen technologies, frameworks, and versions. **Not only does it save me at least an hour or so of work each time**, but it also helps to keep my documentation — or team’s one — organized and up-to-date. *It’s like having an extra set of hands on the team!*

For instance, imagine I want to use some external API. I can ask ChatGPT to draft an initial code structure for me.

*Generate boiler plate code that integrates an extern API using Python.*

And ChatGPT answers right away:



Screenshot ChatGPT chat. ChatGPT gives me a code skeleton.

Or I can even ask for a skeleton [to scrape LinkedIn using Selenium](https://medium.com/dev-genius/how-to-build-a-scraping-tool-for-linkedin-in-7-minutes-tool-data-science-csv-selenium-beautifulsoup-python-a673f12ac579), and ChatGPT gives me the following output.

from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
import time  
  
# Initialize the web driver  
driver = webdriver.Firefox()  
  
# Navigate to LinkedIn  
driver.get("https://www.linkedin.com/")  
  
# Find the email and password input fields  
email = driver.find\_element\_by\_id("username")  
password = driver.find\_element\_by\_id("password")  
  
# Enter your credentials  
email.send\_keys("YOUR\_EMAIL")  
password.send\_keys("YOUR\_PASSWORD")  
  
# Click the login button  
driver.find\_element\_by\_xpath("//button[@type='submit']").click()  
  
# Wait for the page to load  
time.sleep(5)  
  
# Navigate to the profile of the desired user  
driver.get("https://www.linkedin.com/in/USER\_PROFILE\_NAME")  
  
# Extract the information you want from the page  
name = driver.find\_element\_by\_xpath("//span[@class='inline t-24 t-black t-normal break-words']").text  
location = driver.find\_element\_by\_xpath("//span[@class='t-16 t-black t-normal inline-block']").text  
  
print(name)  
print(location)  
  
# close the browser  
driver.quit()

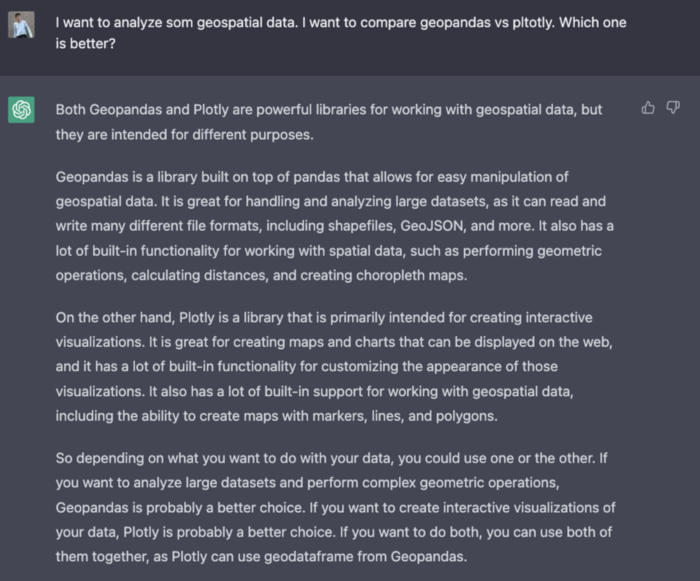
*Easy, right? Try to challenge ChatGPT with any project you can imagine.*

**#2. Research and compare**

Making decisions on how to implement something can be tough, **especially when there are multiple options to choose from**. My go-to method is to create a basic proof of concept for each approach and then compare them. But, with the help of ChatGPT, *this process just got a lot easier.*

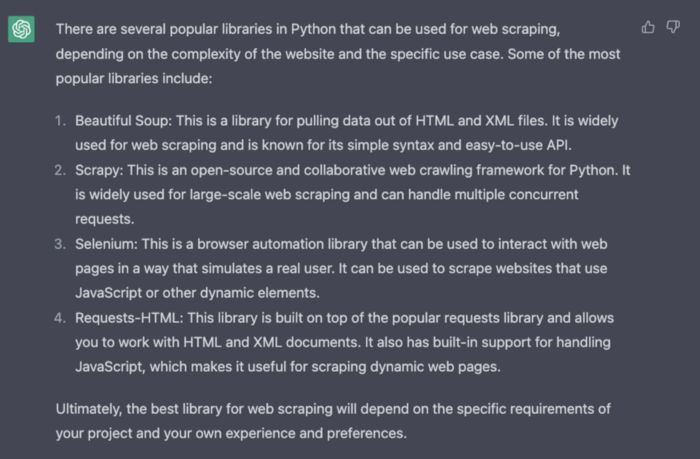
I can now directly ask it for its expert opinion on which option or library is best for my code development. **This saves me time and effort in the decision-making process and ensures that I am using the best tools for the job.**

Let’s imagine I want to work with geospatial data but I am not sure whether I should use Geopandasor a Plotly. I can ask ChatGPT to compare for me — with a type included ;) — and it answers right away the main differences between both libraries.



Screenshot ChatGPT chat. ChatGPT explains to me the differences between geopandas and plotly.

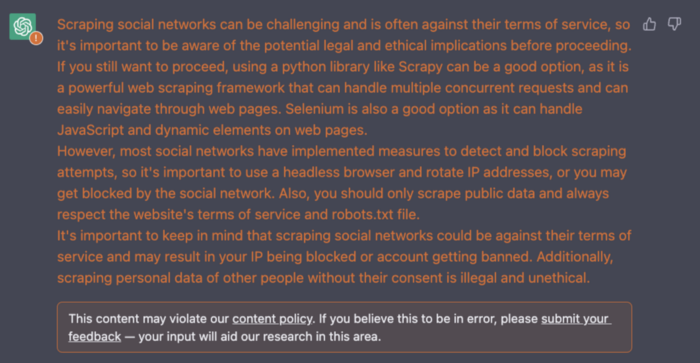
If now I want to scrape a website, I can ask what’s the best library to do so. ChatGPT answers with the most popular web-scraping libraries in Python.



Screenshot ChatGPT chat. ChatGPT explains the most popular scraping website

You can even ask what’s the best option for the website you want to scrape — even though ChatGPT will most likely warn you that it will be against that website’s content policy — so just be careful.

*What’s the best option to scrape a social network?*



Screenshot ChatGPT chat. ChatGPT explains the best option to scrape a social network.

**#3. Understanding code**

We’ve all been there, **struggling to understand a codebase that wasn’t created by us.** Navigating through a complex and poorly-organized code — also known as *spaghetti code —*can be a frustrating and time-consuming task.

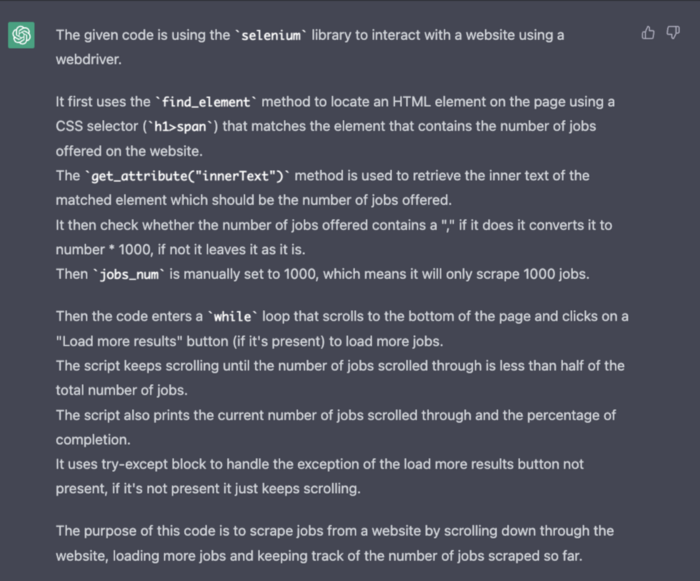
But, with ChatGPT, understanding a new codebase just got a lot easier. I can now simply ask it to explain the functionality of the code and understand it in no time. **No more wasting valuable time and effort trying to decipher poorly-written code.**

Let’s imagine I am trying to scrape Linkedin and I found a random code on the internet that is supposed to scroll down the Linkedin job offers website.

*What does the following code do? [insert code here]*

#We find how many jobs are offered.  
jobs\_num = driver.find\_element(By.CSS\_SELECTOR,"h1>span").get\_attribute("innerText")  
if len(jobs\_num.split(',')) > 1:  
 jobs\_num = int(jobs\_num.split(',')[0])\*1000  
else:  
 jobs\_num = int(jobs\_num)  
  
jobs\_num = int(jobs\_num)  
  
#Here I choose manually a number of jobs, so it wont take that long:  
jobs\_num = 1000;  
  
  
#We create a while loop to browse all jobs.   
i = 2  
while i <= int(jobs\_num/2)+1:  
 #We keep scrollind down to the end of the view.  
 driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")  
 i = i + 1  
 print("Current at: ", i, "Percentage at: ", ((i+1)/(int(jobs\_num/2)+1))\*100, "%",end="\r")  
 try:  
 #We try to click on the load more results buttons in case it is already displayed.  
 infinite\_scroller\_button = driver.find\_element(By.XPATH, ".//button[@aria-label='Load more results']")  
 infinite\_scroller\_button.click()  
 time.sleep(0.1)  
 except:  
 #If there is no button, there will be an error, so we keep scrolling down.  
 time.sleep(0.1)  
 pass

And ChatGPT answers right away with a really detailed explanation of the code above. You can even ask for some more details in a given part of the code —*just play a bit with ChatGPT and you’ll discover all the wonders it can make.*



Screenshot ChatGPT chat. ChatGPT explains the code I didn’t understand.

**#4. Adding comments to our code**

ChatGPT can also help us to improve the quality and maintainability of our own code. By asking it to add comments line by line, we can ensure that our code is properly documented before shipping. This eliminates the need for uncommented code and makes it easier for others to understand and work with our code.

**ChatGPT can significantly improve readability, maintainability and collaboration with others.**

Let’s imagine the previous code had no comments at all. We can just ask ChaGPT to add comments directly to it.

*Add comments to the following code [insert code here]*

# Find the number of jobs offered on the website  
jobs\_num = driver.find\_element(By.CSS\_SELECTOR,"h1>span").get\_attribute("innerText")  
  
# Check if the number of jobs contains a ',' and if so, convert it to number \* 1000  
if len(jobs\_num.split(',')) > 1:  
 jobs\_num = int(jobs\_num.split(',')[0])\*1000  
else:  
 jobs\_num = int(jobs\_num)  
  
# Convert jobs\_num to an integer  
jobs\_num = int(jobs\_num)  
  
# Set the number of jobs to scrape to 1000  
jobs\_num = 1000;  
  
# Initialize a counter variable to scroll through the jobs  
i = 2  
  
# Start a while loop to scroll through the jobs  
while i <= int(jobs\_num/2)+1:  
 # Scroll to the bottom of the page  
 driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")  
 # Increment the counter variable  
 i = i + 1  
 # Print the current number of jobs scrolled through and the percentage of completion  
 print("Current at: ", i, "Percentage at: ", ((i+1)/(int(jobs\_num/2)+1))\*100, "%",end="\r")  
 try:  
 # Try to locate the "Load more results" button and click on it  
 infinite\_scroller\_button = driver.find\_element(By.XPATH, ".//button[@aria-label='Load more results']")  
 infinite\_scroller\_button.click()  
 # Sleep for 0.1 seconds  
 time.sleep(0.1)  
 except:  
 # If the button is not present, sleep for 0.1 seconds and continue scrolling  
 time.sleep(0.1)  
 pass

*Impressive right? No more code without comments! :D*

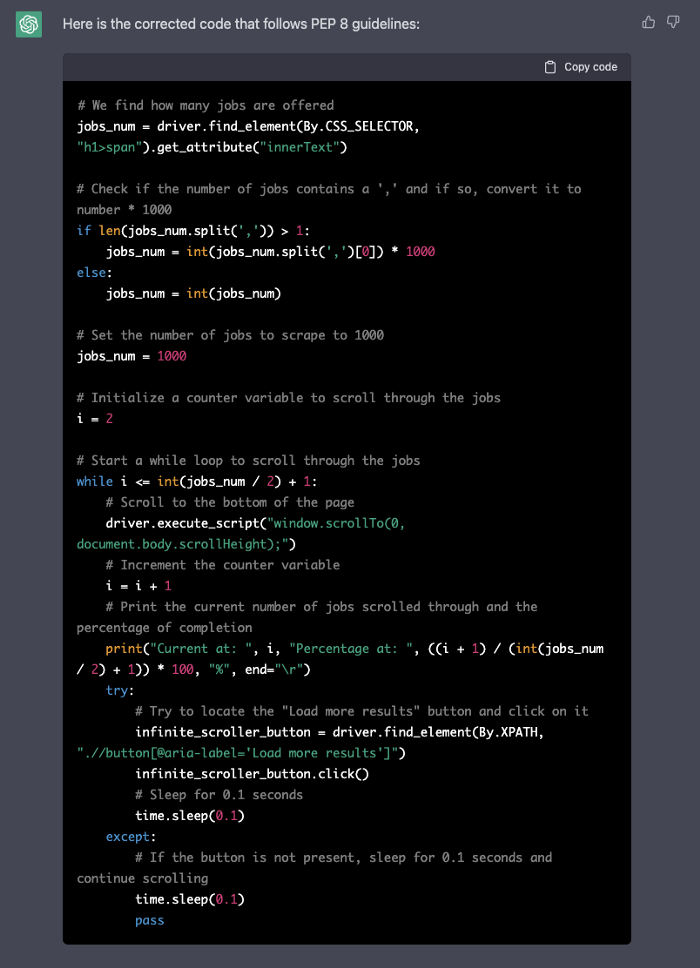
**#5. Rewriting our code using some style**

ChatGPT is not only a valuable tool for understanding unfamiliar code,**but it can also help us to ensure our own code follows industry standards and conventions.** By asking it to correct our code to conform with the Pep-8 convention — or even create a custom convention for our coding style — we can avoid the need for costly and time-consuming refactoring when merging code from different repos or teams.

**This helps to streamline the collaboration process and make it more efficient.** Overall, ChatGPT is a versatile tool that can improve the quality and maintainability of our codebase.

Is we ask ChatGPT to write the previous code using Pep-8 standard, it will directly gives us the refactorized code.

*Can you rewrite the following code using Pep8 standard [Insert code here]*



Screenshot ChatGPT chat. ChatGPT giving our code following Pep8 standard.

**Main Conclusions**

I hope after this article you realize that ChatGPT can **help us to be more productive and create even higher quality output.**I know it can be easy to fall into the trap of thinking that AI may eventually take over our jobs, **but the right kind of AI can be a powerful asset that can be used in our behalf.**

However, **it’s important to remember that critical thinking is still key when working with AI**, just like it is when working with our human colleagues.

So, before you rush to implement AI-generated responses, make sure to take the time to review and assess them first. Trust me, it’s worth it in the end!

*Let me know if ChatGPT surprises you with some other good features. I will read you in the comments! :D*

**Data always has a better idea — trust it.**