



Microsoft Build



Accelerating your Development with AI

Using GitHub Copilot X and using Azure OpenAI Services

Allan Pead

GitHub Copilot

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

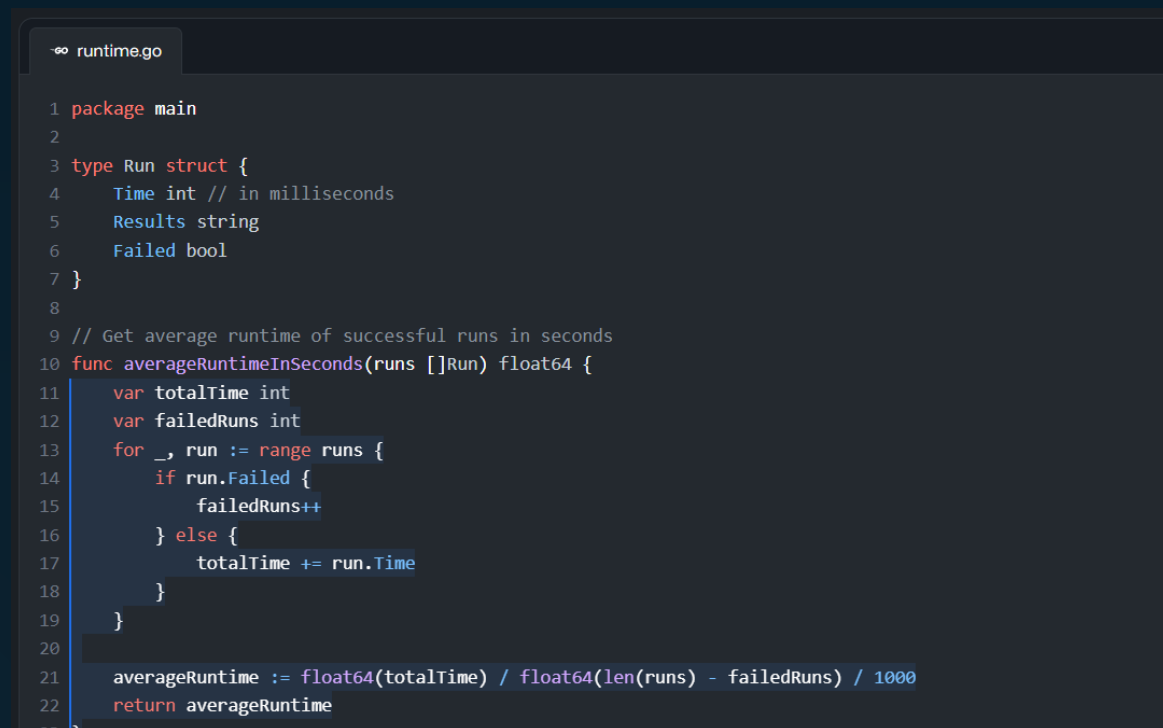
```
TS sentiments.ts
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

<https://github.com/features/copilot/>

GitHub Copilot

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.



The screenshot shows a code editor window with a tab labeled "runtime.go". The code is written in Go and includes a struct definition for "Run" and a function "averageRuntimeInSeconds". GitHub Copilot suggestions are shown as blue-shaded lines of code. The function "averageRuntimeInSeconds" is highlighted, showing the logic for calculating the average runtime by summing successful run times and ignoring failed runs.

```
runtime.go

1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9 // Get average runtime of successful runs in seconds
10 func averageRuntimeInSeconds(runs []Run) float64 {
11     var totalTime int
12     var failedRuns int
13     for _, run := range runs {
14         if run.Failed {
15             failedRuns++
16         } else {
17             totalTime += run.Time
18         }
19     }
20
21     averageRuntime := float64(totalTime) / float64(len(runs) - failedRuns) / 1000
22     return averageRuntime
23 }
```

GitHub Copilot X

Introducing GitHub Copilot X

Your AI pair programmer is leveling up

With chat and terminal interfaces, support for pull requests, and early adoption of OpenAI's GPT-4, GitHub Copilot X is our vision for the future of AI-powered software development. Integrated into every part of your workflow.

<https://github.com/features/preview/copilot-x>

GitHub Copilot Chat

GITHUB COPILOT: CHAT

monalisa

Write a set of unit test functions for the selected code

GitHub Copilot

```
import unittest
import datetime

class TestParseExpenses(unittest.TestCase):
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n2023-01-03 2.59 DKK\n2023-01-03 -2.72 EUR'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'), (datetime.datetime(2023, 1, 3), 2.59, 'DKK'), (datetime.datetime(2023, 1, 3), -2.72, 'EUR')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_comments(self):
        expenses_string = '# This is a comment\n2023-01-02 -34.01 USD\n2023-01-03 2.59 DKK\n2023-01-03 -2.72 EUR'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'), (datetime.datetime(2023, 1, 3), 2.59, 'DKK'), (datetime.datetime(2023, 1, 3), -2.72, 'EUR')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

if __name__ == '__main__':
    unittest.main()
```

The code assumes that the datetime module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.

Analyze and debug

Ask a question or type '/' for topics

module.py X

```
1 import datetime
2
3 def parse_expenses (expenses_string):
4     """Parse the list of expenses and return the list of triples (date, amount, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2023-01-02 -34.01 USD
9         2023-01-03 2.59 DKK
10        2023-01-03 -2.72 EUR
11    """
12    expenses = []
13
14    for line in expenses_string.splitlines():
15        if line.startswith("#"):
16            continue
17        date, value, currency = line.split(" ")
18        expenses.append((datetime.datetime.strptime (date, "%Y-%m-%d"),
19                        float (value),
20                        currency))
21
22    return expenses
23
24 expenses_data = '''2023-01-02 -34.01 USD
25                 2023-01-03 2.59 DKK
26                 2023-01-03 -2.72 EUR'''
27
28
29
30
31
```

Ln 17, Col 3 Spaces: 2 UTF-8 LF {} TypeScript

Demo

GitHub Co-Pilot X in Visual Studio





GitHub Copilot

Using Voice to Code

Join the GitHub Copilot chat Waitlist

Admission to the private beta for GitHub Copilot chat is limited and requires an active subscription to [GitHub Copilot](#). Signing up does not guarantee access.

A ChatGPT-like experience in your editor with GitHub Copilot chat

We are bringing a chat interface to the editor that's focused on developer scenarios and natively integrates with VS Code and Visual Studio. This does far more than suggest code. GitHub Copilot chat is not just a chat window. It understands what code a developer has typed, what error messages are shown, and it's deeply integrated into the IDE. A developer can get in-depth analysis and explanations of what code blocks are intended to do, generate unit tests, and even get proposed fixes to bugs.

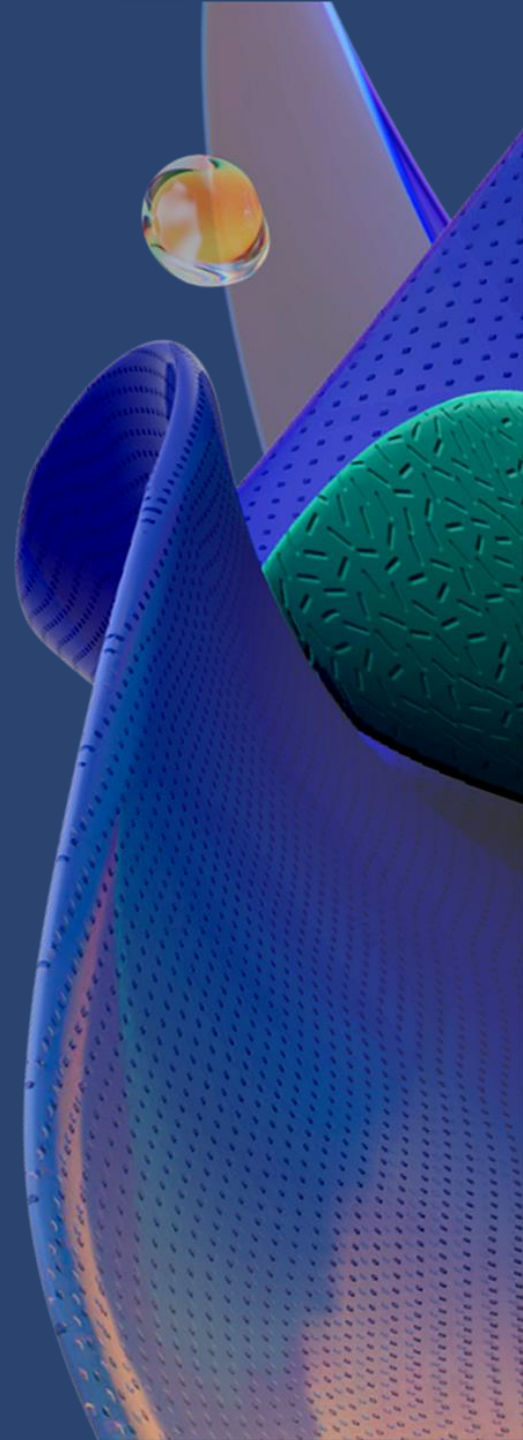
Where do you plan to use GitHub Copilot chat? *

- ☐ Visual Studio Code [Learn More](#)
- ☐ Visual Studio [Learn More](#)

This is a Beta Preview under [GitHub's Terms of Service](#).

[Join the waitlist](#)

https://github.com/github-copilot/chat_waitlist_signup/join



GitHub Next

GitHub Next investigates the future of software development.

We are a team of researchers and engineers at GitHub, exploring things beyond the adjacent possible. We prototype tools and technologies that will change our craft. We identify new approaches to building healthy, productive software engineering teams.

<https://githubnext.com/>

GitHub Copilot

Don't fly solo

Developers all over the world use GitHub Copilot to code faster, focus on business logic over boilerplate, and do what matters most: building great software.

Copilot for Individuals

\$ **10** per month /
\$100 per year

- ✓ Plugs right into your editor
- ✓ Turns natural language prompts into code
- ✓ Offers multi-line function suggestions
- ✓ Speeds up test generation
- ✓ Filters out common vulnerable coding patterns
- ✓ Blocks suggestions matching public code

[Start a free trial >](#)

Copilot for Business new

\$ **19** per user
per month

- ← Everything included in Copilot for Individuals, plus...
- ✓ Simple license management
- ✓ Organization-wide policy management
- ✓ Industry-leading privacy
- ✓ Corporate proxy support

[Buy Now >](#)

[Contact Sales >](#)

GitHub Copilot for Students



Student Developer Pack

Get the best developer tools

There's no substitute for hands-on experience, but for most students, real-world tools can be cost prohibitive. That's why we created the Pack with some of our partners and friends.

Individuals

Students

Learn using real-world development tools

- ✓ **FREE** [GitHub Pro](#) while you are a student
- ✓ Valuable [GitHub Student Developer Pack](#) partner offers
- ✓ [GitHub Campus Expert training](#) for qualified applicants

Teachers

Teach your students with the industry-standard tools

- ✓ **FREE** [GitHub Team](#) for courses, coding clubs, and nonprofit research
- ✓ [GitHub Classroom](#) for managing assignments

Azure AI

Applied AI Services



Bot Service



Cognitive Search



Form Recognizer



Video Indexer



Metrics Advisor



Immersive Reader

Cognitive Services



Vision



Speech



Language



Decision



Azure OpenAI Service

Azure Machine Learning



Prepare & Preprocess | Build, Train & Consume | Deploy & Scale | Manage & Monitor

AI Infrastructure

Azure OpenAI Service

GPT-3

Codex

DALL·E (preview)

ChatGPT (preview)

GPT-4 (preview)



Deploy on your
own data



Provisioned
Throughput
SKU



Plugins for
Azure OpenAI
Service

<https://azure.microsoft.com/en-in/products/cognitive-services/openai-service/>

Demo

Azure OpenAI Services



Azure OpenAI Service

Request Access to Azure OpenAI Service

Prerequisites

- An Azure subscription - [Create one for free](#).
- Access granted to Azure OpenAI in the desired Azure subscription.

Currently, access to this service is granted only by application. You can apply for access to Azure OpenAI by completing the form at <https://aka.ms/oai/access>. Open an issue on this repo to contact us if you have an issue.

- An Azure OpenAI Service resource with either the `gpt-35-turbo` or the `gpt-4`¹ models deployed. For more information about model deployment, see the [resource deployment guide](#).

¹ GPT-4 models are currently only available by request. To access these models, existing Azure OpenAI customers can [apply for access by filling out this form](#).

<https://aka.ms/oai/access>.



- Twitter: @adpead
 - About.me: https://about.me/allan_pead
 - LinkedIn: <https://www.linkedin.com/in/adpead/>
 - Blog: <https://explorationspace.co.za>
-
- Cape Town MS Developer User Group
 - <https://www.meetup.com/Cape-Town-Ms-Dev-User-Group/>

Thank you!!

© Copyright Microsoft Corporation. All rights reserved.

Resources

GitHub Copilot

<https://github.com/features/copilot/>

Github Copilot X

<https://github.com/features/preview/copilot-x>

Getting Started with Azure OpenAI

<https://learn.microsoft.com/en-us/azure/cognitive-services/openai/overview>

