



AZURE OPENAI SERVICES

Codex – GitHub Copilot



The DevEx difference.

Faster Time to Revenue

Grow revenue by improving DevEx.

- Shorter time to market
- Increased profitability

Improved Operational Efficiency

Drive business performance through increased:

- Productivity
- Efficiency
- Product quality

Higher Talent Retention

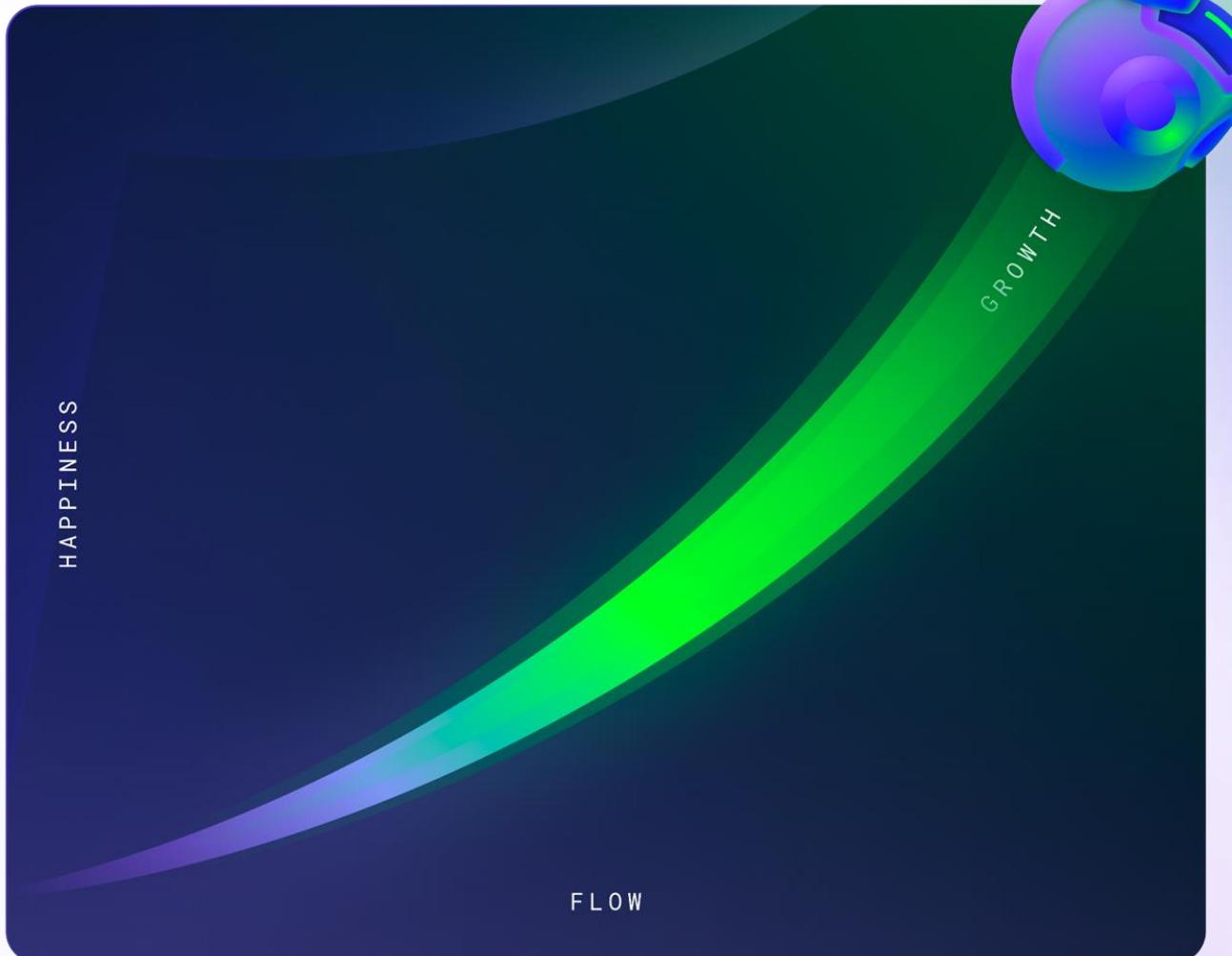
Attract and retain top engineering talent.

- High team morale
- Motivating and rewarding work



AI is the next frontier in DevEx.

A McKinsey study found that equipping developers to be their most productive also significantly improves the developer experience, and developers using generative AI-based tools were more than twice as likely to report overall happiness, fulfillment, and a state of flow.





The Copilot effect.

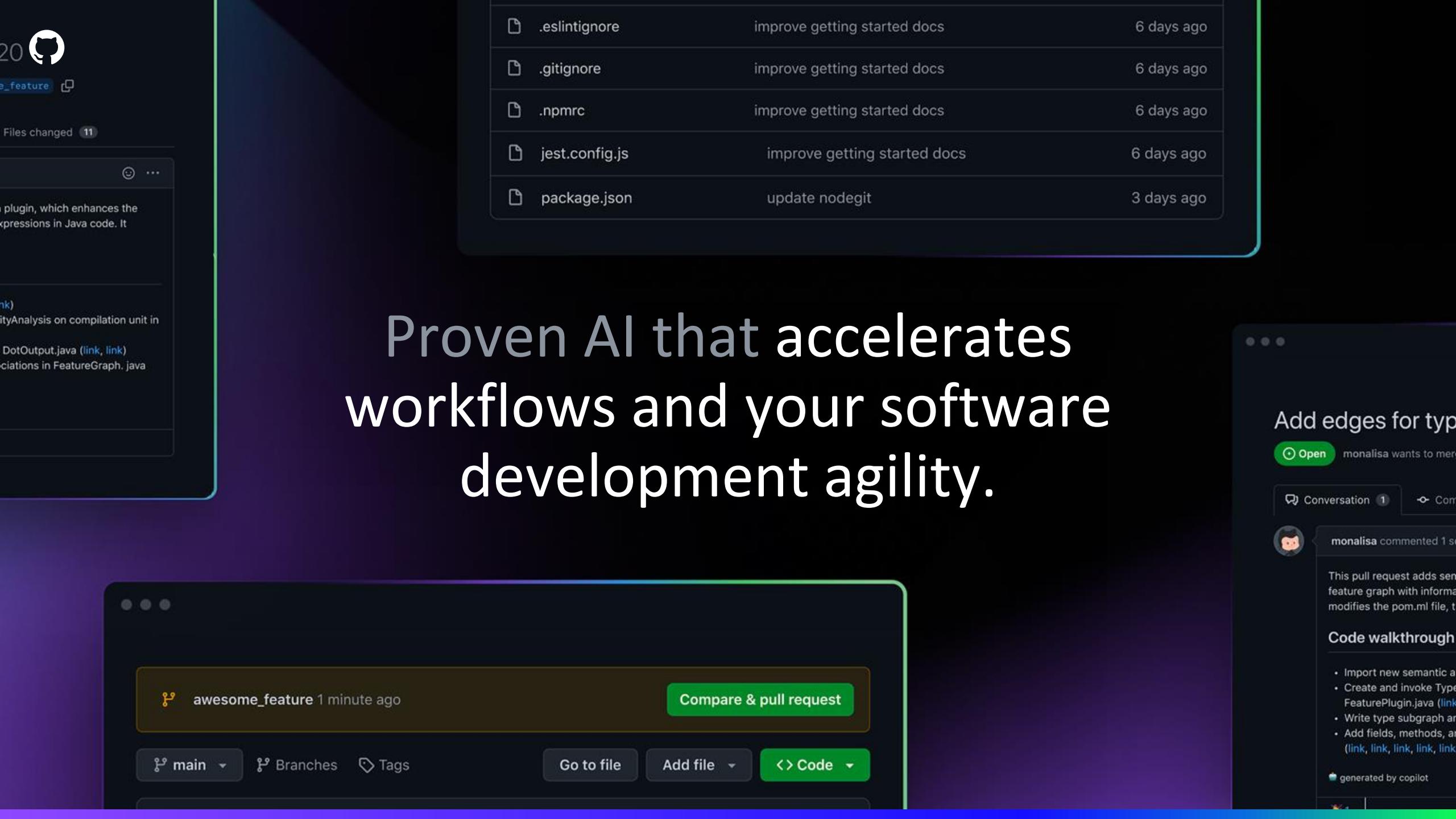
The world's most widely adopted AI developer tool.

Maximize developer happiness

Increase developer productivity

Accelerate software development





Proven AI that accelerates
workflows and your software
development agility.



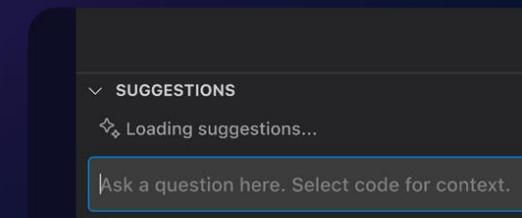
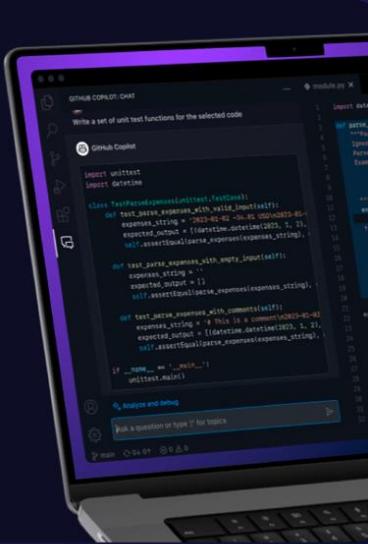
1st overall pick among developers.



75% of
developers want
to try it next
year

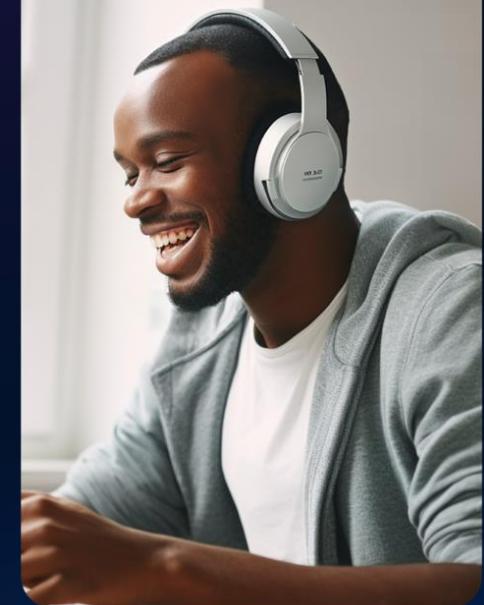


55% of developers
using AI coding
tools choose
GitHub Copilot



Copilot enables
faster coding
by 55%

75% of
developers are
more fulfilled



More than
1.5M developers



Pioneer in enterprise adoption of AI.

Over 1 in 5 Fortune 500 companies use Copilot
(<1 yr since launch)



37,000+ businesses building software with Copilot

```
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'
    expected_output = [(datetime.datetime(2023, 1, 2),
    self.assertEqual(parse_expenses(expenses_string), expected_output))

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

Analyze and debug

Ask a question or type '/' for topics



Software is hard



Technical debt

Burden imposed to manage application code continues to accumulate



Knowledge management gaps

Existing codebases not as well understood



App and data silos

Complex system integrations that are difficult to dissect



Benefits of AI in software development



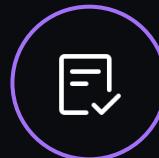
Reduce monotonous tasks

Project scaffolding, unit testing, code duplication



Employ idiomatic techniques

Write idiomatic code from the outset



Understand existing code

Gain a better understanding of existing codebases and documentation



What value does it represent?

Proof Points

More than 2,000 developers surveyed

We recruited
 **95**
developers, and split them randomly into two groups.

We gave them the task of writing a web server in JavaScript

 **45 Used**

GitHub Copilot

 **78%**

finished

 **1 hour, 11 minutes**

average to complete the task



 **50 Did not use**

GitHub Copilot

 **70%**

finished

 **2 hours, 41 minutes**

average to complete the task



Results are statistically significant ($P=.0017$) and the 95% confidence interval is [21%, 89%]



Developer productivity with Copilot goes beyond speed.

87%

74%

73%

less mental effort on repetitive tasks

focus on more satisfying work

say they are more in the flow



Developer productivity with Copilot means developers focus on what matters most.



More time on

Designing

Brainstorming

Collaborating

Iterating

Planning



Less time on

Writing Tests, Repetitive Code, & Boilerplate

Debugging

Searching Documentation

Finding Vulnerabilities

Deciphering Existing Code

Correcting Syntax

Summarizing Changes and Comments

Learning Git Commands



Developer productivity with Copilot benefits enterprises.



Release
cycle agility

55%

Faster completion



Developer
onboarding

57%

More upskilling



Developer
retention

41%

Less burnout

Developers are finding joy with Copilot

More in the flow

Less frustrated when coding

More fulfilled with my job

Focus on more satisfying work

Less mental effort on repetitive tasks

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%



GitHub Copilot



John Skoubourdis
@scoubourdis

GitHub Copilot is like giving a programmer a super power 🤖

11:41 PM · Jun 2, 2022

1,579 Retweets 137 Quote Tweets 22.6K Likes



Alex MacCaw ✅
@maccaw

One of the best uses of GitHub CoPilot is autocompleting your tests.

At this point it's writing the majority of my tests.

5:55 PM · Oct 29, 2022

20 Retweets 3 Quote Tweets 307 Likes



swyx ✨
@swyx

By far the greatest benefit of using @Github Copilot so far is I now don't have to be forced to document my code.

I actively *want* to write great comments, because when I do, I get the dopamine hit of a good Copilot suggestion.

7:51 PM · Oct 12, 2022 from Puerto Vallarta, Jalisco

21 Retweets 3 Quote Tweets 338 Likes



@willman@xoxo.zone
@willmanduffy

You win this round Copilot

updateMyselfMutationVari
ame: 'Bob',
ame: 'Builder',

5:01 PM · Jun 14, 2022

50 Retweets 3 Quote Tweets 780 Likes



Danny Postma
@dannypostmaa

AI is doing 80% of my coding – the future looks brighter than ever! 😊

```
90 methods: () =>  
91   async fetch () =>  
92     // ...  
93  
94     this.test = await this.$axios.$get('/admin/analytics', {  
95       params: {  
96         startDate: this.date.start,  
97         endDate: this.date.end  
98       }  
99     })  
100     this.isLoading = false  
101   }  
102   0:17 493.5K views
```

4:37 PM · Dec 1, 2022

1,161 Retweets 187 Quote Tweets 9,969 Likes



JD Ross ✅
@justindross

Some of our engineers just told me they'd estimate 40% of the lines of code they produce are now written by the GitHub CoPilot AI

6:13 PM · Oct 19, 2022

386 Retweets 159 Quote Tweets 5,220 Likes





GitHub Enterprise

Mercado Libre frees developers' minds to focus on their mission with GitHub.

~50%

reduction in time
spent writing code
with GitHub Copilot

9,000+

developers using
Copilot

100,000

pull requests
merged per day



GitHub Enterprise

Itaú delivers software faster with GitHub Enterprise.

93%

speed increase in
time to commit code
with GitHub Actions

68%

increase in
deployment
speed

75%

increase in code
reuse



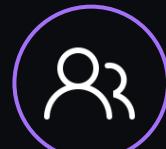
GitHub Copilot For Business

Business controls



Policy controls

At the Enterprise and Organization Level
No code snippets are used to train the model



Easy to assign users

Migration support built-in



Filtering public code matches

Allow or block suggestions for your business



GitHub Copilot

Your AI pair programmer

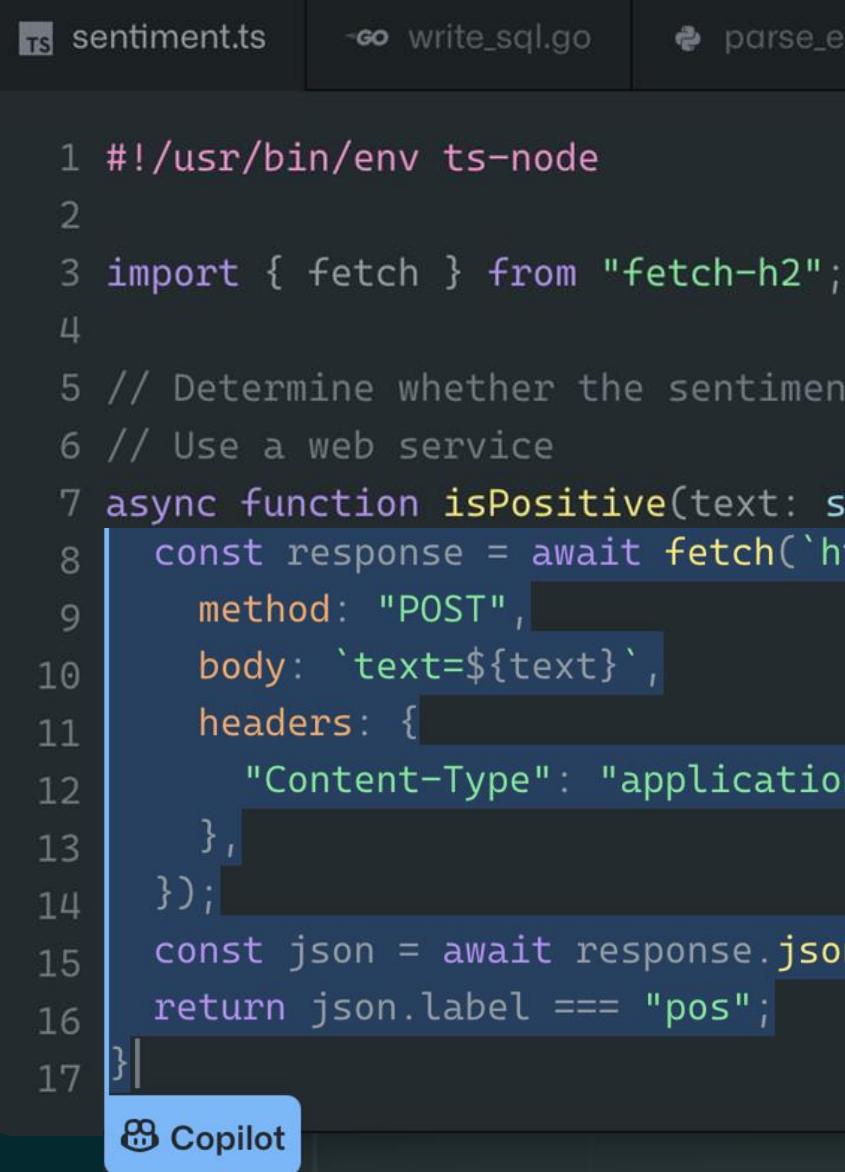
**Uses the context you've provided
and synthesizes code to match**

Convert comments to code

Autofill for repetitive code

Autosuggest tests

Show alternatives



A screenshot of a code editor interface showing GitHub Copilot in action. The editor has tabs for 'sentiment.ts' (selected), 'write_sql.go', and 'parse_ex...'. The code in 'sentiment.ts' is:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment
6 // Use a web service
7 async function isPositive(text: string): boolean {
8     const response = await fetch(`https://api.fetch-h2.com/sentiment?text=${text}`);
9     const json = await response.json();
10    return json.label === "pos";
11}
12
13
14
15
16
17}
```

The line 'const response = await fetch(`https://api.fetch-h2.com/sentiment?text=\${text}`);' is highlighted in blue, indicating it's being generated by Copilot. A blue button at the bottom right of the code area says 'Copilot'.





Once enabled...



OpenAI GPT-3.5
Turbo
Derivative



Context

Suggestions

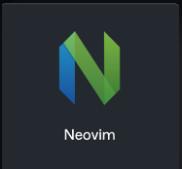
A screenshot of a code editor interface. At the top, there are tabs for "runtime.go", "course.rb", "time.js", and "IsPrimeTest.java". The "runtime.go" tab is active. The code editor displays the following Go code:

```
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

The code editor has a dark theme with syntax highlighting. A purple double-headed arrow labeled "Context" connects the OpenAI GPT-3.5 logo to the GitHub Copilot logo. A blue double-headed arrow labeled "Suggestions" connects the GitHub Copilot logo to the code editor interface.



Visual Studio



Neovim



VS Code



JetBrains IDEs





CODE EDITOR

PROXY

MODEL

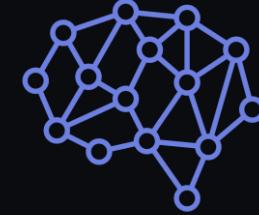
Toxicity
Code classifier



CODE EDITOR



PROXY



MODEL

PII
Toxicity
Code classifier
Code quality
Duplicate detection



Pillars for the vision

Ubiquitous across every tool developers use and integrated into every task that developers perform. Offering suggestions wherever you write and providing insights wherever you read. All with a radically-usable experience.

Conversational by default, so natural language can be used to achieve anything. Offering an informal and iterative interface that empowers developers to create more fluidly by thinking out loud and then reacting to emergent insights.

Personalized to the context and knowledge of the individual, project, team, organization, and community. Offering a "second brain" to developers, which understands your intent, history, and helps translate information into meaning (code, docs, etc.)

Safe, Trustworthy, and Reliable so our customers can take a bet not only on the technology and product but on GitHub as a partner and a contributor to responsible AI practices and goals.

GitHub Copilot Chat

Chat experience in the IDE

ChatGPT-like experience in editor

- Generic questions
- Generate unit tests
- Explain the code
- Propose a fix for bug
- Make code robust / secure

Support VS Code and Visual Studio

Voice-to-code with GitHub Copilot Voice

The screenshot shows the GitHub Copilot Chat interface integrated into a dark-themed IDE. On the left, there's a sidebar with icons for file operations, search, and GitHub. The main area has a title 'GITHUB COPILOT' and a message from 'GitHub Copilot' welcoming the user (@martinwoodward) and stating its purpose: to help with code-related tasks like generating unit tests, explaining code, and proposing fixes. Below this, there's a section for learning more about capabilities and a search bar at the bottom labeled 'Ask a question or type '/' for topics'. On the right, the code editor displays a Python file named 'add_elements.py' with code for parsing expense strings. A tooltip is shown over line 17, highlighting a syntax error: 'ValueError: not enough values to unpack (expected 3, got 1)'. The status bar at the bottom provides details about the current file: 'Ln 3, Col 1 (685 selected) Spaces: 4 UTF-8 LF {} Python 3.11.2 64-bit'. There are also tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', and 'DEBUG CONSOLE'.

```
def parse_expenses(expenses_string):  
    """Parse the list of expenses and return the list of triples  
    (date, value, currency).  
    Ignore lines starting with #.  
    Parse the date using datetime.  
    Example expenses_string:  
        2023-01-02 -34.01 USD  
        2023-01-03 2.59 DKK  
        2023-01-03 -2.72 EUR  
    """  
    expenses = []  
    for line in expenses_string.splitlines():  
        if line.startswith("#"):  
            continue  
        date, value, currency = line.split("#")  
        expenses.append((float(value),  
                        currency,  
                        datetime.datetime.strptime(date,  
                            "%Y-%m-%d")))  
    return expenses  
  
expenses_data = '''2023-01-02 -34.01 USD  
2023-01-03 2.59 DKK'''
```

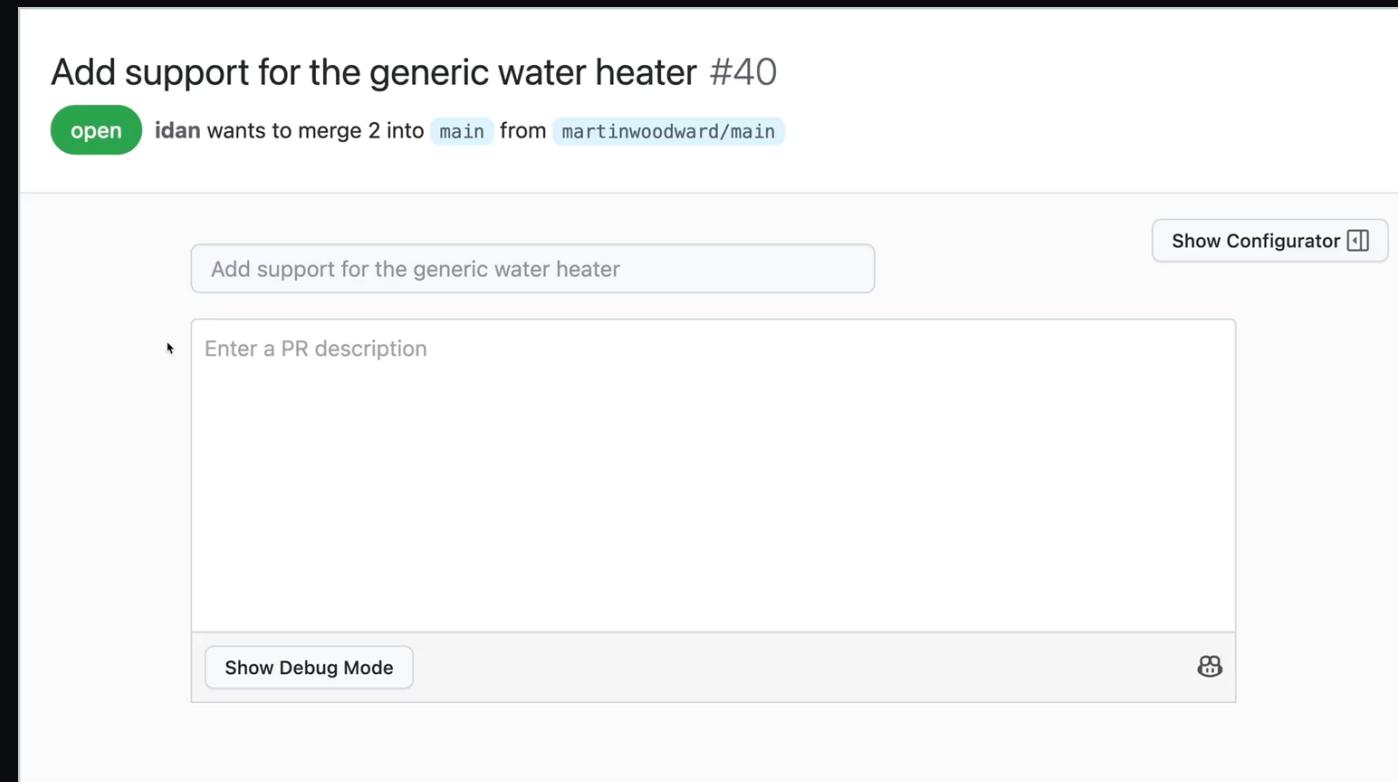
GitHub Copilot for Pull Requests

the future of AI-powered software development

AI powered tags in Pull Requests

Automatically suggest sentences and paragraphs as developers type based on PR code changes

Automatically warn and suggest tests where needed



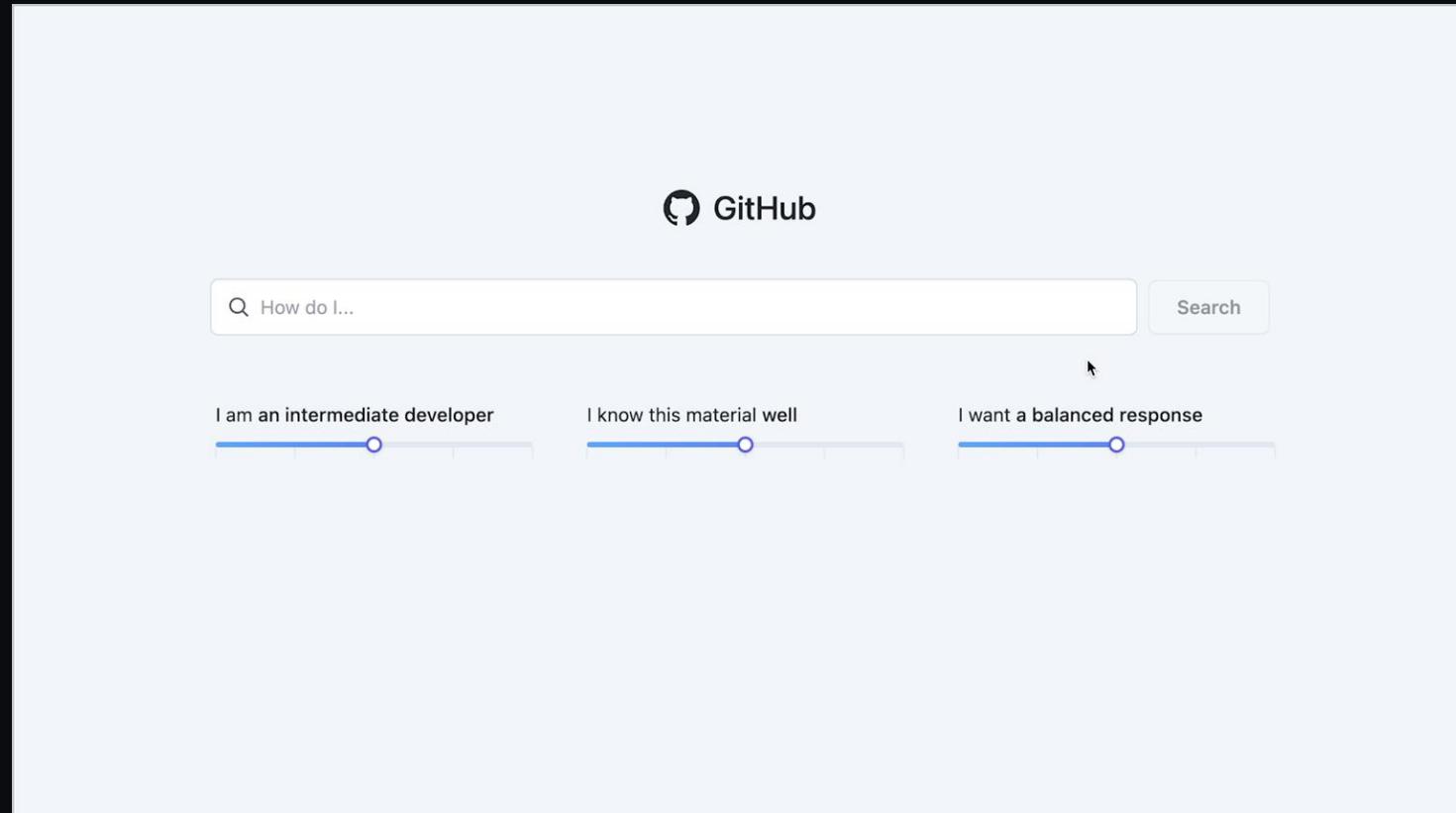
GitHub Copilot for Docs

the future of AI-powered software development

AI-generated responses to questions
based on documentation

Starting with docs for:

- GitHub
- React
- Azure
- MDN
- In future, your in-house docs and codebases



GitHub Copilot for the CLI

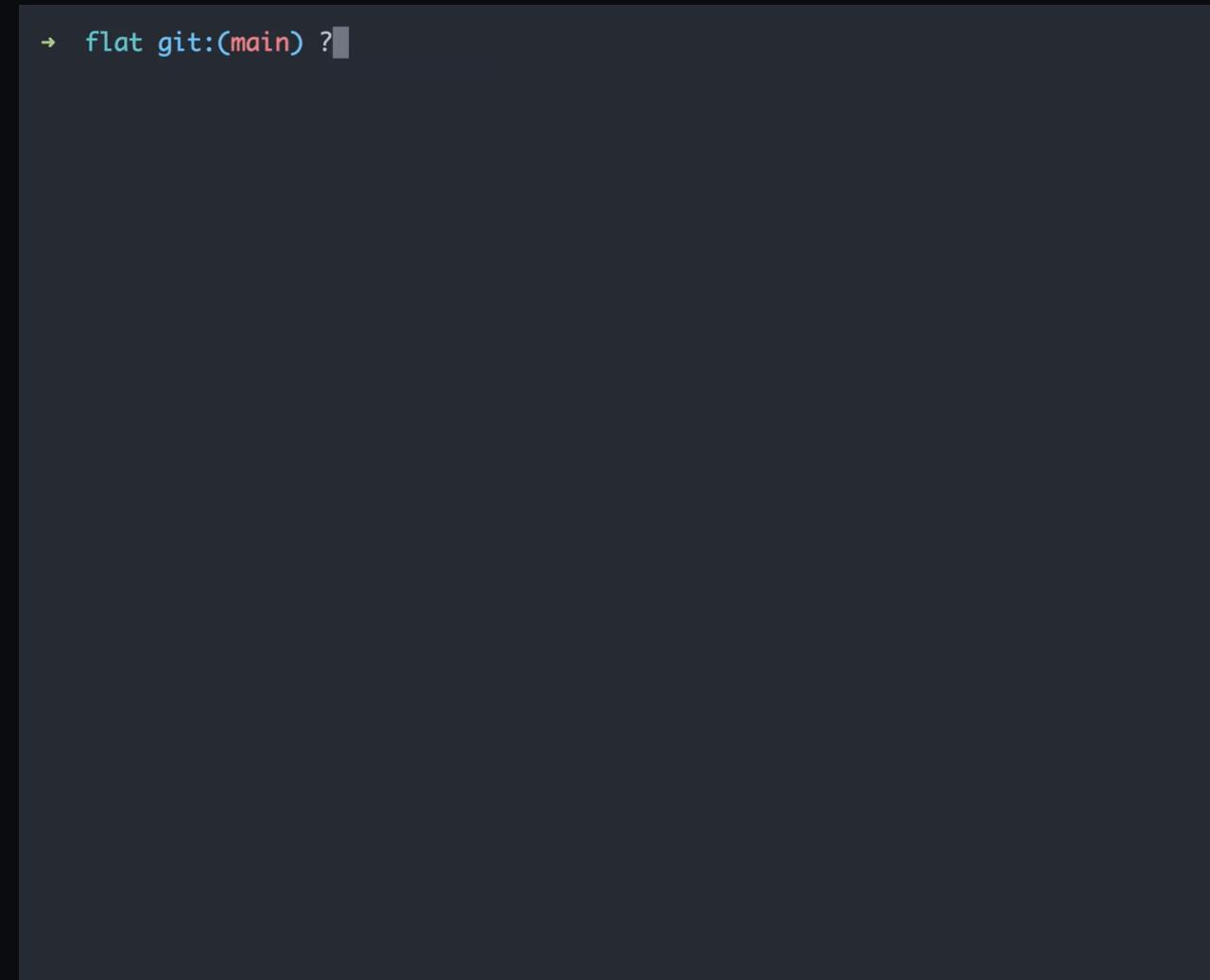
the power of Copilot in the terminal

Suggest CLI commands in the terminal window

Three shell commands:

- ?? – General CLI
- git? – git specific CLI
- gh? – GitHub specific CLI

Revise & Execute



A screenshot of a terminal window with a dark background. The text "flat git:(main) ?" is displayed at the top, followed by a question mark icon. The terminal is empty below this line.



Demo



Get the
advantages of
AI today



Increase developer productivity

And satisfaction by focusing on real problems



Accelerate innovation

Prototype & innovate more rapidly



Bridge skill gaps

Learn new languages and techniques





GitHub Copilot tomorrow
AI for Pull Request
Copilot for your Docs
Copilot Security

GitHub Copilot Chat
Unit testing
Finding code errors
Debugging
Code review
Refactoring code (code translate)
Reviewing code (code explain)
Documentation

GitHub Copilot today
Convert comments to code
Autofill for repetitive code
Show alternatives







THANK YOU