

The Educational Conference For Oracle Technology Users

ODTUG  
Kscope23  
aurora, co    june 25-29

# GitHub Copilot: My first year with AI Co-Programming

Summit 1 - Wed, Jun 28, 2023 (03:15 PM - 04:15 PM)



**Robert Marz**  
**DATABEE**  
Die IT-Architekten



Interested in volunteering?  
Scan the code to sign up



Don't Forget To  
Fill Out Your Evals



Robert Marz – Independent Consultant

## Primary Role

Senior Technical Architect  
with database centric view of the world

## DOAG (German Oracle User Group)

Active Member of Database Community  
Responsible for Cloud Topics



@RobbieDatabee



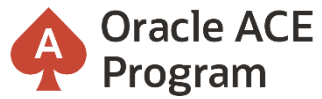
<https://robbie.databee.org>



[robert.marz@databee.org](mailto:robert.marz@databee.org)



Oracle ACE  
Pro



## 500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community



### 3 membership tiers



For more details on Oracle ACE Program:  
[ace.oracle.com](https://ace.oracle.com)



### Nominate

yourself or someone you know:

[ace.oracle.com/nominate](https://ace.oracle.com/nominate)

Connect: [aceprogram\\_ww@oracle.com](mailto:aceprogram_ww@oracle.com)

[Facebook.com/OracleACEs](https://Facebook.com/OracleACEs)

[@oracleace](https://twitter.com/oracleace)





# GitHub Copilot: Intro



# GitHub Copilot: Whats that?

## GitHub Copilot

- is a cloud-based artificial intelligence tool developed by GitHub and OpenAI to assist users [...]by autocompleting code. [[Wikipedia](#)]

## Autocomplete on Steroids

- watches you typing and suggests constantly the next lines [[RobbieDatabee](#)]

## Your AI pair programmer

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





# GitHub Copilot: History

June 29, 2021

- GitHub Copilot technical preview VS Code development environment.

October 29, 2021

- GitHub Copilot was released as a [plugin](#) on the JetBrains marketplace on.

May 27, 2022

- [@RobbieDatabee](#) was accepted for technical trial

October 27, 2021

- GitHub released the GitHub Copilot Neovim plugin as a public repository.

March 29, 2022

- GitHub announced Copilot's availability for the Visual Studio 2022 IDE on.

June 21, 2022

- GitHub announced that Copilot was out of "technical preview", and is available as a subscription-based service for individual developers.



# GitHub Copilot: Installation & Prerequisites

## Supported Editors / IDEs:

- VS Code
- NeoVim
- Visual Studio
- JetBrains IDEs

## Installation:

- Download the Plugin from the Marketplace

## Prerequisite:

- IDE needs internet connection
- Github Copilot Hosts must be whitelisted on firewall
- GitHub Account
- GitHub Copilot Subscription



©vlarvix - stock.adobe.com



# GitHub Copilot: Invocation

## How to invoke

Just type - works like normal autocomplete

Press Ctrl-<Enter>

Use Context Menu

Invoke Copilot Labs



@Andreas Prott - stock.adobe.com

# GitHub Copilot: Usecases





# Usecase: Auto Complete

Copilot continuously reads your Input

suggestions:

next line

Algorithms /  
Functions  
(from comments)



```
def get_primes(self):  
    return self.primes  
  
def get_prime_factors(self, num):  
    factors = []  
    for prime in self.primes:  
        while num % prime == 0:  
            factors.append(prime)  
            num //= prime  
    return factors  
  
def get_factors(self, num):  
    factors = []  
    for divisor in range(1, num + 1):  
        if num % divisor == 0:  
            factors.append(divisor)  
    return factors  
  
self.primes = self.generate_primes()
```



# Usecase: Learn a new Language

## Suggestions

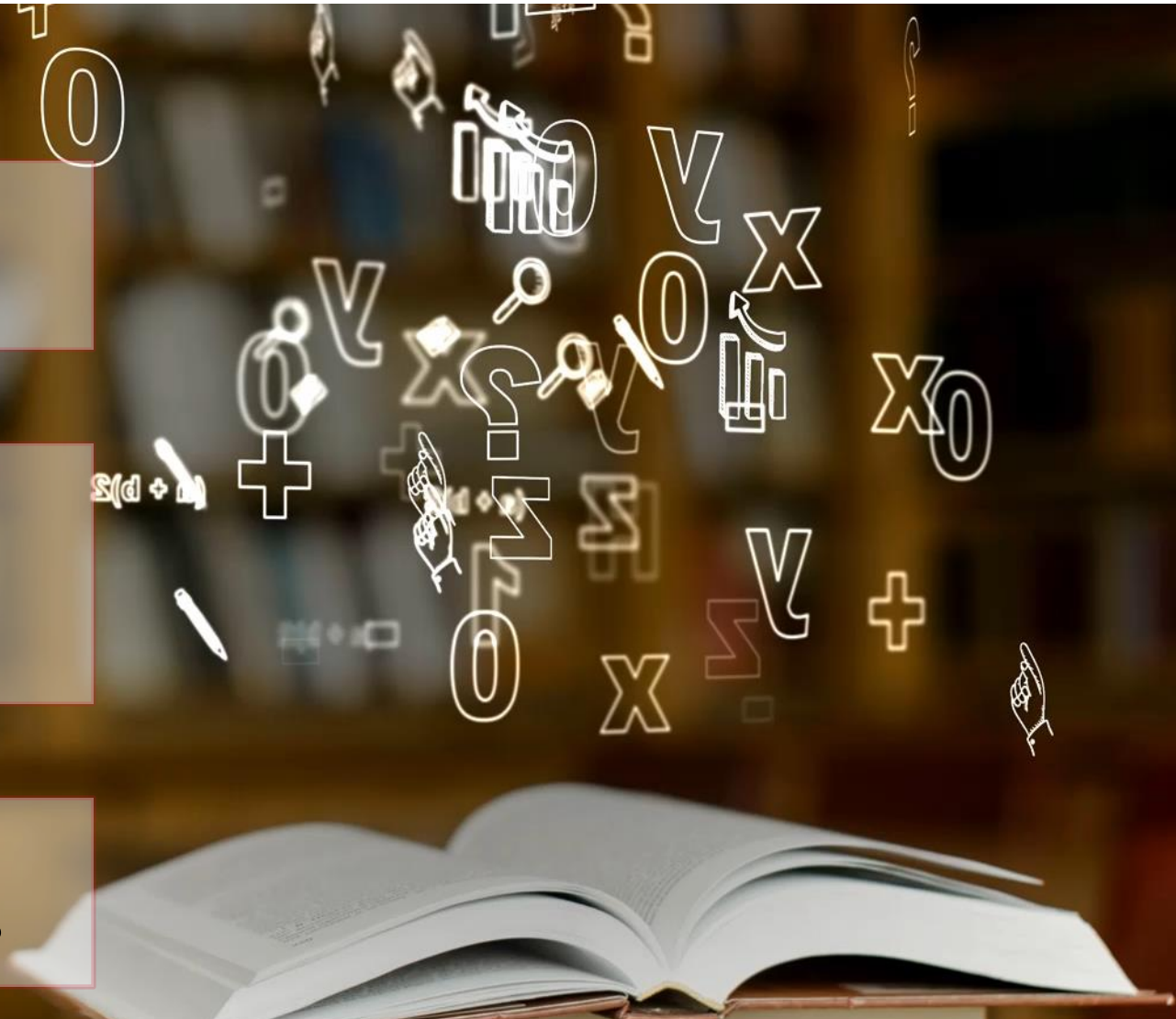
- are syntactically correct

## Code snippets

- are in style
- e.g., „Pythonic“

## Examples

- Explains how code works





# Usecase: Find the Docu you really need - Copilot for Docs

# Copilot for Docs



Search engine for Documentation

[githubnext.com/projects/copilot-for-docs](https://githubnext.com/projects/copilot-for-docs)

Preview with waitlist

(Unfortunately, I'm still waiting...)

The screenshot displays the GitHub Copilot for Docs interface. At the top, it says 'GitHub Copilot for Docs' and shows the user 'Amelia Wattenberger' with a 'Sign out' link. The main content is divided into two sections: 'Docsets' and 'Recent threads'. The 'Docsets' section contains four cards for different documentation sources: GitHub, React.js, MDN, and Azure. Each card provides a brief description and a 'Start a new thread' button. The 'Recent threads' section shows a list of recent discussions, including questions about running Node.js scripts, OAuth/GitHub App usage, and creating pull requests. At the bottom of the interface, there is a search bar with the placeholder text 'Ask a follow-up question about React.js' and a 'Search' button. Below the search bar are three sliders for user preferences: 'I am an intermediate developer', 'I know this material well', and 'I want a balanced response'.



# Usecase: Discuss Challenges with CopilotChat

## GitHub Copilot Chat



- Companion Extension to Copilot Nightly
- In Preview Mode / Waitlist to join
- Offers Chat to discuss / ask questions
- Restricted to coding topics
- 

concurrent-ki.py - beemirror [WSL: OracleLinux\_8\_7] - Visual Studio Code

CHAT: GITHUB COPILOT

This JSON file defines a function that takes two arguments (x and y) and returns their sum. The `co_varnames` field is a list of the function's argument names, `co_argcount` is the number of arguments, `co_consts` is a list of constants used in the function (in this case, there are none), and `co_code` is a list of bytecode instructions that implement the function's logic.

Note that this is just a simple example, and in practice, you would likely want to use a more complex data structure to represent a function in JSON.

**rmarz**  
does python code instead of bytecode work?

**GitHub Copilot**  
Yes, Python code can be loaded from a JSON file instead of bytecode. You can use the `compile` function to compile the code and then use the `exec` function to execute it. Here's an example:

```
import json

def load_function_from_json(file_path):
    with open(file_path, 'r') as f:
        data = json.load(f)
        code = compile(data['code'], '<string>', 'exec')
        exec(code, globals())
        return globals()[data['name']]
```

This function takes a file path as an argument and returns a function object loaded from the

[How can I save a Python function to a JSON file?](#)

Ask Copilot a question or type '/' for topics

© You, 4 days ago Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.11.2 ('robbiedev': venv) Spell



# Usecase: Generate Testdata

```
emptab.sql > ...
1  create table emps(emp_id number not null,
2  emp_name varchar2(20),
3  emp_salary number(10,2),
4  emp_hiredate date,
5  );
6
7  /* insert 10 rows into table emps */
8  insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
9  values(1, 'John', 10000, '01-JAN-2010');
10 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
11 values(2, 'Smith', 20000, '01-FEB-2010');
12 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
13 values(3, 'Peter', 30000, '01-MAR-2010');
14 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
15 values(4, 'Mary', 40000, '01-APR-2010');
16 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
17 values(5, 'David', 50000, '01-MAY-2010');
18 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
19 values(6, 'Paul', 60000, '01-JUN-2010');
20 insert into emps(emp_id, emp_name, emp_salary, emp_hiredate)
21 values(7, 'James', 70000, '01-JUL-2010');
22
23
```



# Usecase: Write Unittests

```
27  
28  
29 def test_get_co  
30  
60  
61 ...  
62 | I  
63 ...
```



# Usecase: Write Unittests (Preview in Copilot Labs)

The screenshot displays the GitHub Copilot Labs interface for 'TEST GENERATION'. It features a dark-themed editor with a sidebar on the left containing navigation icons. The main editor area is split into two panes. The left pane shows the source code for a JavaScript function named 'plural'. The right pane shows the same function code with a 'Test Pilot' overlay, which includes a 'Suggest a new test' button and a list of suggested tests. The status bar at the bottom indicates the current file is 'index.js' at line 72, column 1, with a UTF-8 encoding and LF line endings.

```
function plural(word, num) {
  var i
  , rule

  if (num !== 1 || num === undefined) {
    for (i = 0; i < rules.length; i++) {
      rule = rules[i]

      if (type(rule[0]) === 'RegExp' && rule[0].test(word)) {
        return type(rule[1]) === 'Function' ? rule[1](word) : rule[1]
      }
      if (type(rule[0]) === 'String' && rule[0] === word) {
        return type(rule[1]) === 'Function' ? rule[1](word) : rule[1]
      }
    }
  }
}
```

Suggest a new test

```
describe('test plural', function() {
  it('test plural.plural', function(done) {
    assert.equal(plural('zebra', 1), 'zebra');
    assert.equal(plural('zebra', 2), 'zebras');
    assert.equal(plural('zebra'), 'zebras');
    assert.equal(plural('stop'), 'stoppes');
  });
});
```

Ln 72, Col 1 Spaces: 2 UTF-8 LF {} JavaScript



# Usecase: Explain Code (Preview in Copilot Labs)

The screenshot shows a code editor with the following TypeScript code:

```
TS function buildInvertedIndex(s: string): Untitled-1 •  
1 function buildInvertedIndex(s: string): { [key: string]:  
2   const words = s.split(/\W/);  
3   const index: { [key: string]: number[] } = {};  
4   for (let i = 0; i < words.length; i++) {  
5     const word = words[i];  
6     if (word) {  
7       if (index[word]) {  
8         index[word].push(i);  
9       } else {  
10        index[word] = [i];  
11      }  
12    }  
13  }  
14  return index;  
15 }  
16
```

The right-hand side of the image shows the 'GITHUB COPILOT LABS' interface. It features a dropdown menu with the text 'Code does following (English)'. Below this is an 'Advanced' section with a prominent teal button labeled 'Ask Copilot'. At the bottom of the interface, there is a 'LANGUAGE TRANSLATION' section.

At the bottom of the code editor, the status bar shows: Ln 16, Col 1 Spaces: 2 UTF-8 LF {} TypeScript Insert Prettier



# Usecase: Do my Job



@Golden\_hind - stock.adobe.com

## Copilot knows nothing

- about your goals
- the environment
- the ecosystem
- the company rules
- the business rules

## It's your code

→ your responsibility

## You need to understand and review

- every generated line of code

## Copilot generates

- Code
- Bugs
- Security Issues



# GitHub Copilot: Demo

---



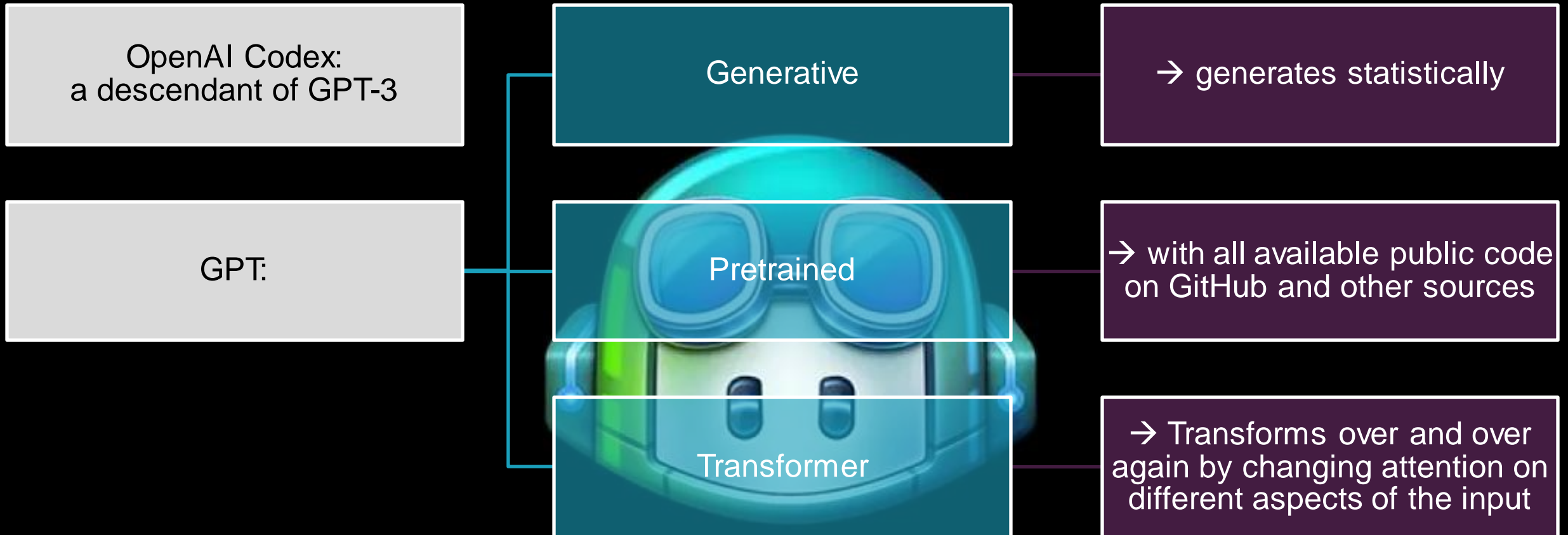
©iStockworld - iStockphoto.com

# GitHub Copilot: How it works





# GitHub Copilot – How it works I



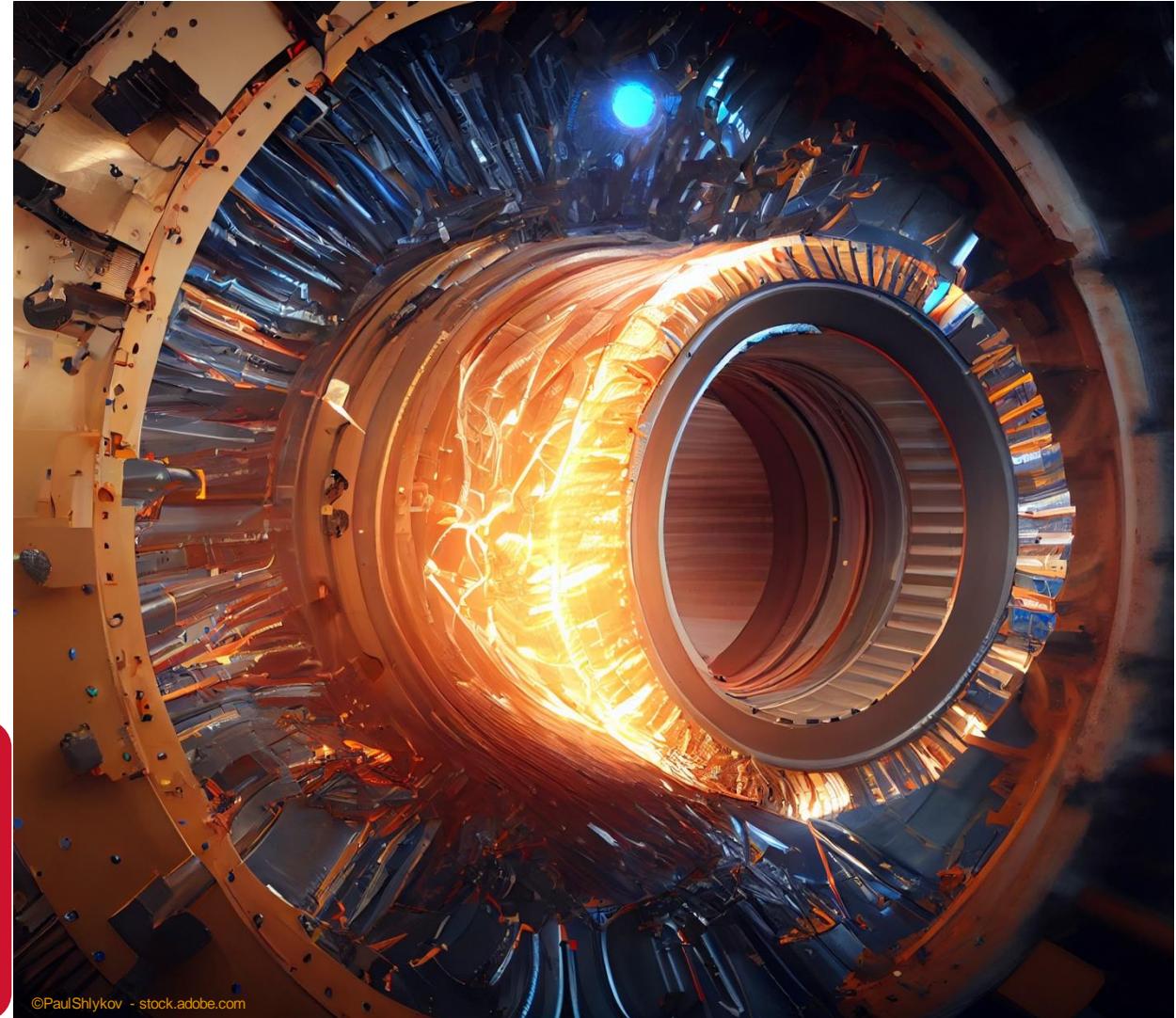


# GitHub Copilot – How it works II

Copilot uploads your prompt to the cloud.

openAI Codex does ist magic

Results are displayed in your editor





# GitHub Copilot: What is a prompt?

Copilot prompts everytime,  
you stop typing

## User Engagement Data

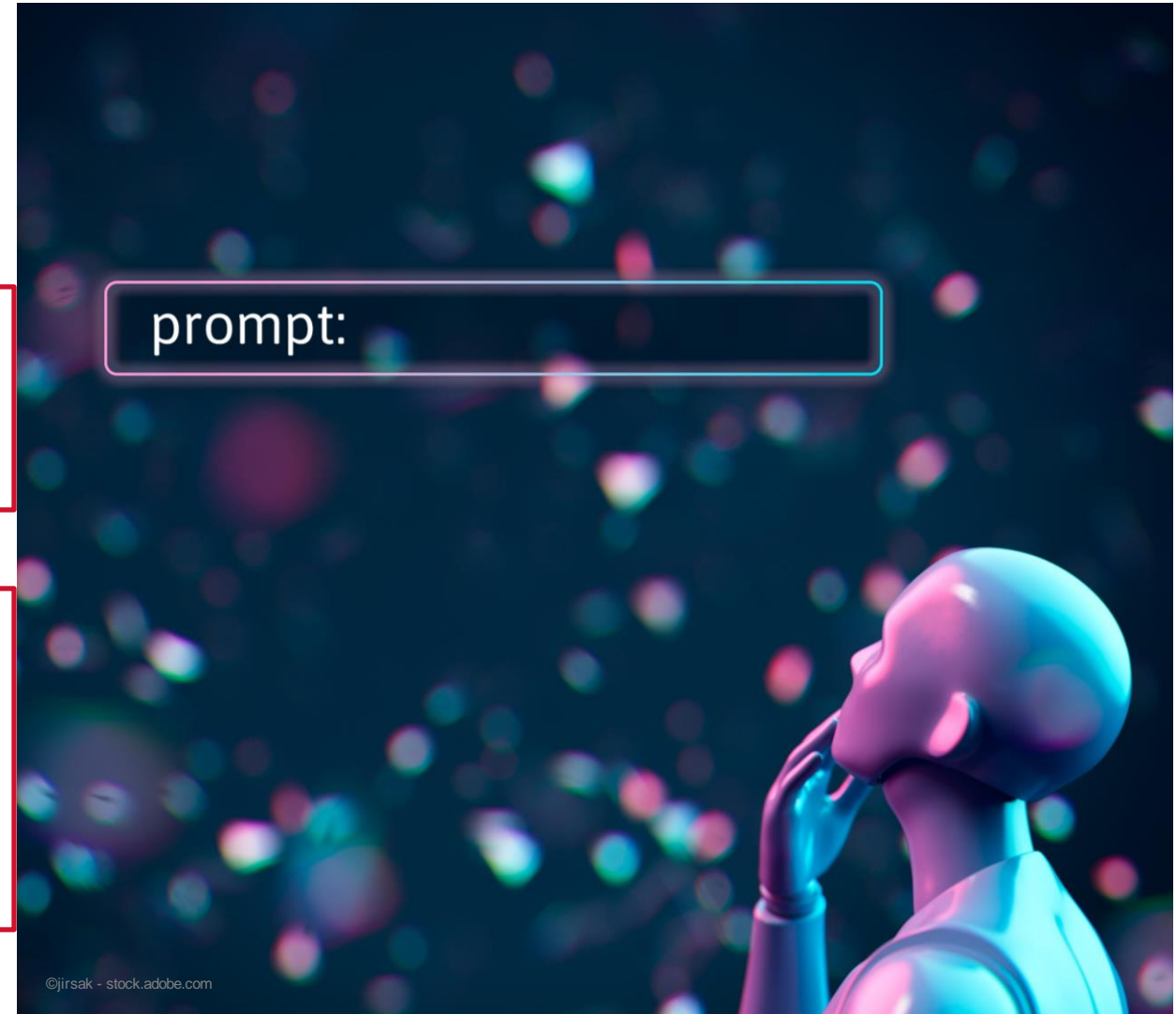
- edit actions
- completions accepted and dismissed

## Code Snippets Data

- source code currently edited
- related files
- other files open in same IDE or Editor
- URLs of repositories and files paths

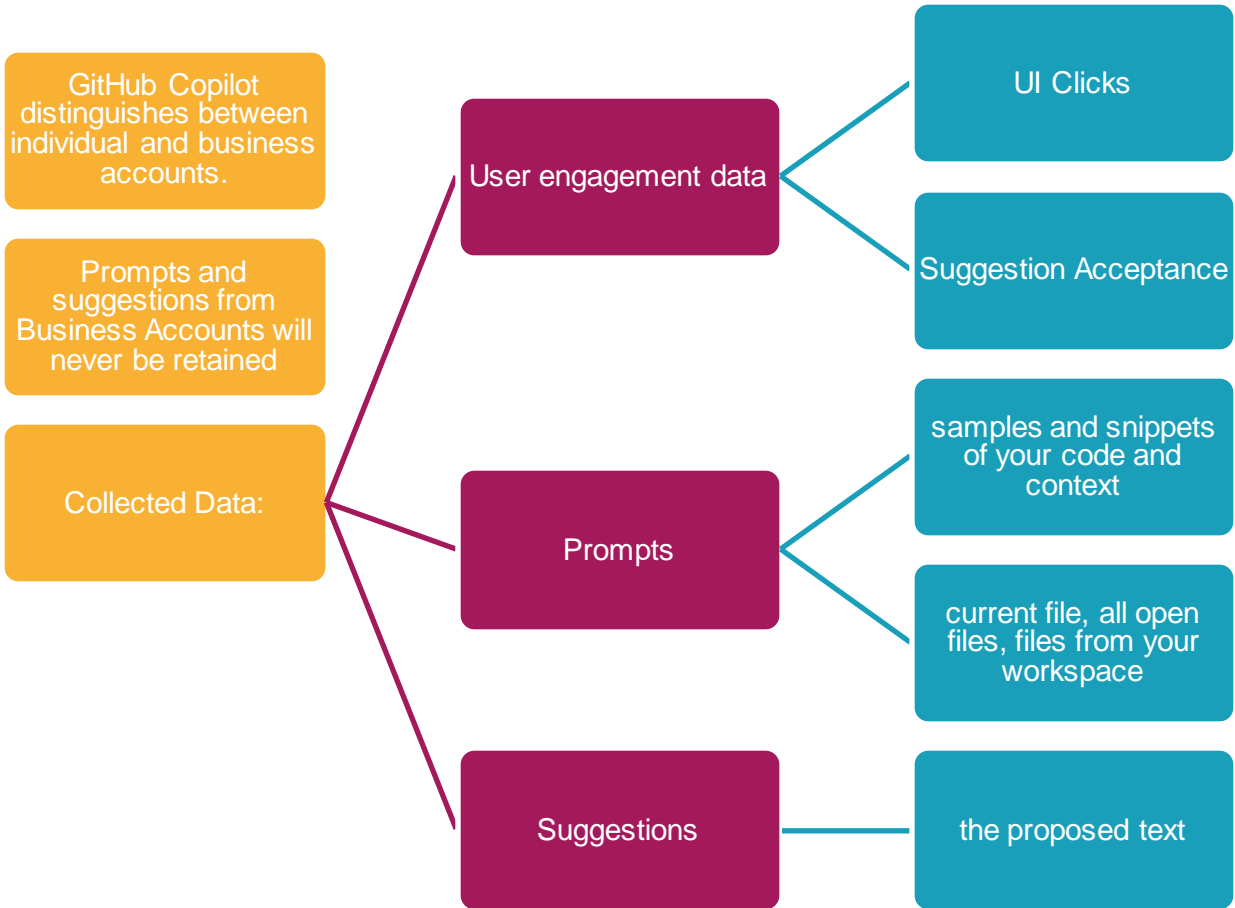
Source:

<https://web.archive.org/web/20220917002449/https://github.com/features/copilot>





# GitHub Copilot: Data privacy – collected data



©yod67 - stock.adobe.com



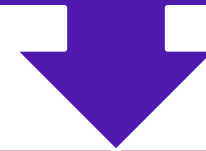
# GitHub Copilot: Copyright & License Issues



Copilot is trained with code available in public repos regardless of their license.



Copilot generates every answer based on the prompt – it is unlikely, that it reproduces copyrighted code.

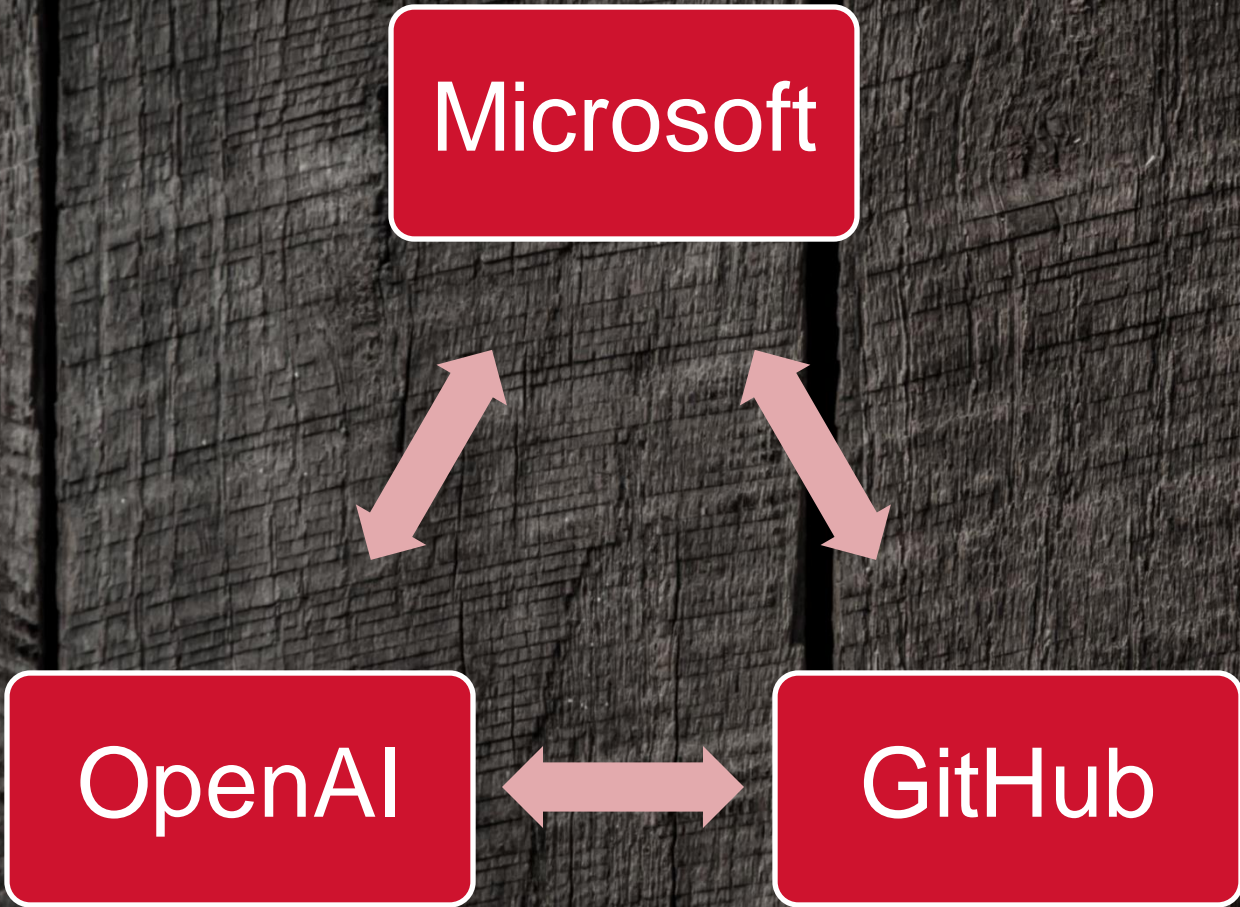


Let's leave those disputes to the lawyers

©STOATPHOTO - stock.adobe.com



# GitHub Copilot – Companies involved





# GitHub Copilot: Costs

## 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
- ✓ Blocks suggestions matching public code

[Start a free trial >](#)

## Copilot for Business new

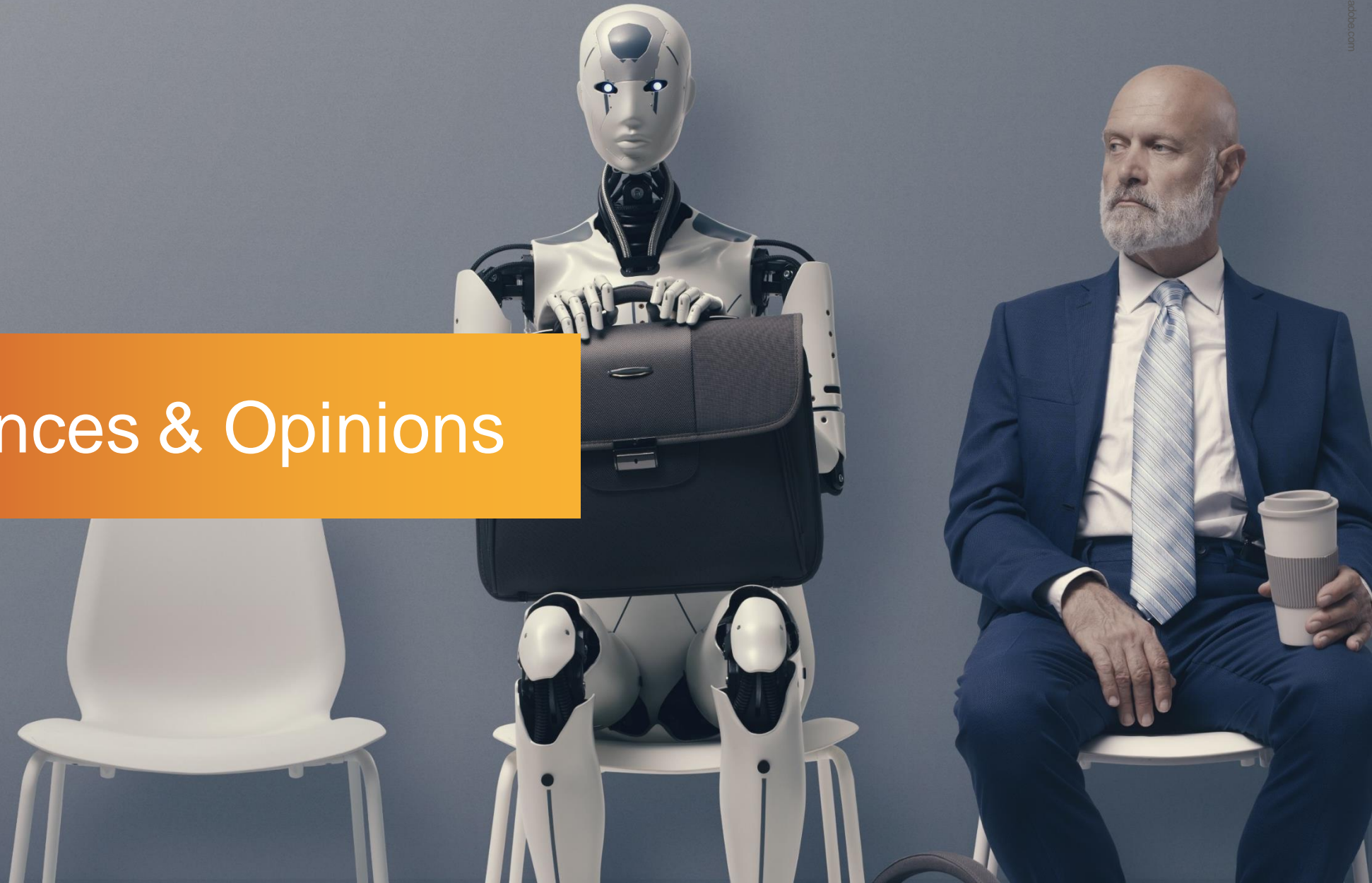
\$ **19** per user  
per month

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

[Buy Now >](#)

[Contact Sales >](#)

# My Experiences & Opinions





# How Do I use GitHub Copilot?



©Charnchai saeheng - stock.adobe.com

## My usage

I keep typing and enjoy the autogenerated suggestions.

Rarely, I write a comment and get inspired by generated functions. I almost never accept functions as a whole

I use the explain feature a lot, when writing documentation

I explore & enjoy the CopilotChat Beta Feature, since I got accepted two weeks ago



# Acceptance Rate of Suggestions



## GitHub Copilot FAQ

- In a recent evaluation, we found that users accepted on average 26% of all completions shown by GitHub Copilot. We also found that on average more than 27% of developers' code files were generated by GitHub Copilot, and in certain languages like Python that goes up to 40%. However, GitHub Copilot does not write perfect code.

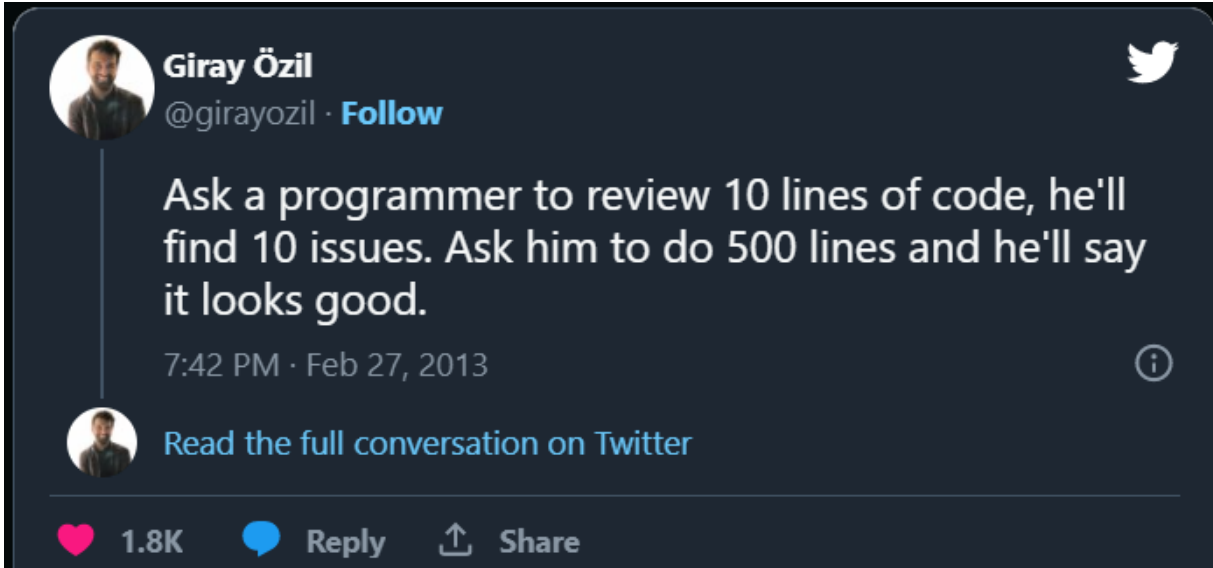
Source: <https://github.com/features/copilot>

## My personal acceptance rate

- Suggestions roughly 5-10%, Python slightly higher
- I never accept complete files



# How Do I get the best results?



## Best results by

accepting only little amounts of code

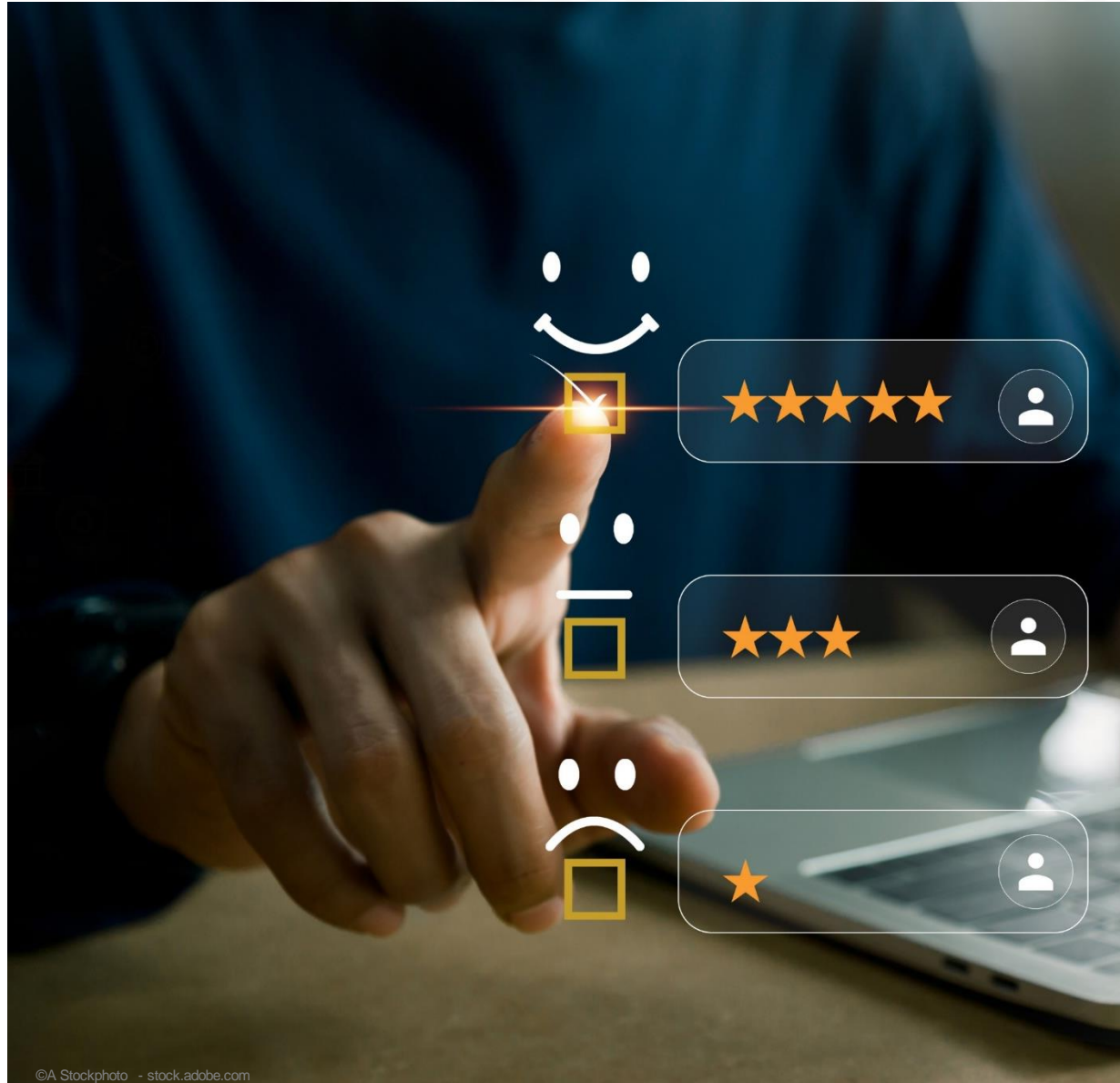
carefully checking, what's suggested

never accept code, I don't understand

divide code in small functions with meaningful names and parameters



# What languages work best?



## GitHub Copilot works

- with most computer languages.

## Results are better

- when more training data (aka Repositories) is available.

## Best Results come for

- JavaScript
- TypeScript
- Python
- Ruby
- Go
- C#
- C++



# Copilot & Oracle SQL

```
SELECT  
FROM  
WHERE  
Employee-  
Employees  
HireDate BETWEEN  
SELECT  
WHERE  
City IN ('Seattle', 'Tacoma')  
EmployeeID, FirstName, Last  
Employees  
HireDate NOT BETT  
SELECT  
FROM  
WHERE  
EmployeeID, FirstNa
```

@Proxima Studio - stock.adobe.com

GitHub Copilot works well with SQL

Mixes SQL dialects

- e.g. mySQL & pgSQL

In most cases easy to adapt



# Copilot & Oracle PL/SQL

# ORACLE®

## DATABASE



## PL/SQL

Only little amount  
of PL/SQL  
Trainingdata

Wrong suggestions  
quite common

Copilot can't distinguish  
between Oracle PL/SQL  
and PL/pgSQL

Easy to adapt

→ create more PL/SQL  
repositories on GitHub!



# GitHub Copilot: Pitfalls & Caveats

Copilot generates „B\*\*\*Sh\*\*\*“

(B\*\*\*Sh\*t is an AI technical term, similar to hallucinating)  
generated code looks good, but isn't understood by Copilot  
→ Be careful, its easy to drift away from your path

## Check

- every line
- every variable name
- every function call

## Copilot doesn't generates the best code

- rather the average of all repos in GitHub
- with bugs and security issues

## It's your code. You are in charge

- There won't be a label
- „Generated by an AI“



# Is it worth the money?

---



It worth the money?

Definitely!

It spares multiple hours a week googeling for examples

Types the boring repetitions

Sometimes it even helps, when I'm stuck.

I'll keep my rubber ducky, though.



# GitHub Copilot Alternatives

Product	Trained on	Self hosting option	Fine tuned to company codebase	Status	Open source	Pricing
<a href="#">Tabnine</a>	Open source code with permissive licenses.	Yes (enterprise plan)	Yes (enterprise plan)	Production	No	Free + Paid
<a href="#">GitHub Copilot</a>	Publicly available code (regardless of license)	No	No	Production	No	Paid*
<a href="#">Replit Ghostwriter</a>	Publicly available code (regardless of license)	No	No	Production	No	Paid
<a href="#">Amazon CodeWhisperer</a>	Amazon and open-source code: but can select to not generate licensed open-source code	No. But professional version collects no data.	No	General availability	No	Free + Paid**
<a href="#">Codeium</a>	Own model. Trained on permissive licenses.	Yes (enterprise plan)	Yes	Production	<a href="#">Some plugins</a>	Free + Paid
<a href="#">SourceGraph Cody</a>	Users can select: OpenAI or Anthropic currently. Future plans for other models.	Yes for Sourcegraph. Cody: not yet; it talks with third-party LLM providers.	No	Experimental	Yes	Free + Paid***
<a href="#">CodeComplete</a>	Open source code with permissive licenses	Yes	Yes	Private beta	No	N/A
<a href="#">FauxPilot</a>	Salesforce CodeGen: open source but includes code with non-permissive license	Yes	No	Production	Yes	Free
<a href="#">Tabby</a>	Unclear	Yes	Yes	Production	Yes	Free

\* GitHub Copilot is free for verified students, teachers and maintainers of popular open-source repositories on GitHub. There is no way to request this, though, GitHub has decided which ones these “popular” projects are.

\*\* Amazon CodeWhisperer is free for personal use, forever

\*\*\* Cody is free for open source projects. It’s “free” as part of a paid Sourcegraph package.

[pragmaticengineer.com](https://pragmaticengineer.com)

## Alternatives

- [Tabnine](#) (2019)
- [GitHub Copilot](#) (2021)
- [Replit Ghostwriter](#) (2022)
- [Amazon CodeWhisperer](#) (2022)
- [Codeium](#) (2022)
- [SourceGraph Cody](#) (2023)
- [CodeComplete](#) (2023)
- [FauxPilot](#) (2023)
- [Tabby](#) (2023)

Source: [blog.pragmaticengineer.com/github-copilot-alternatives/](https://blog.pragmaticengineer.com/github-copilot-alternatives/)

# Conclusion





Boosts your Coding Speed

Types the boring repetitions for you

Really helpful when you're stuck

Won't take your job

**PLEASE**

**DO  
TRY THIS  
AT HOME**

The Educational Conference For Oracle Technology Users

ODTUG  
Kscope23  
aurora, co    june 25-29



Interested in volunteering?  
Scan the code to sign up



Don't Forget To  
Fill Out Your Evals