



AI-Powered Scripting with GitHub Copilot

Erika Dietrick - Developer Advocate
@erika_thedev
DEVNET-2159

Webex App

Questions?

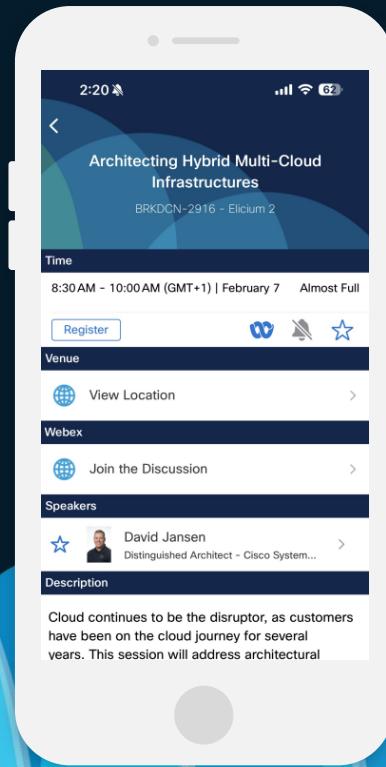
Use the Webex app to chat with the speaker after the session

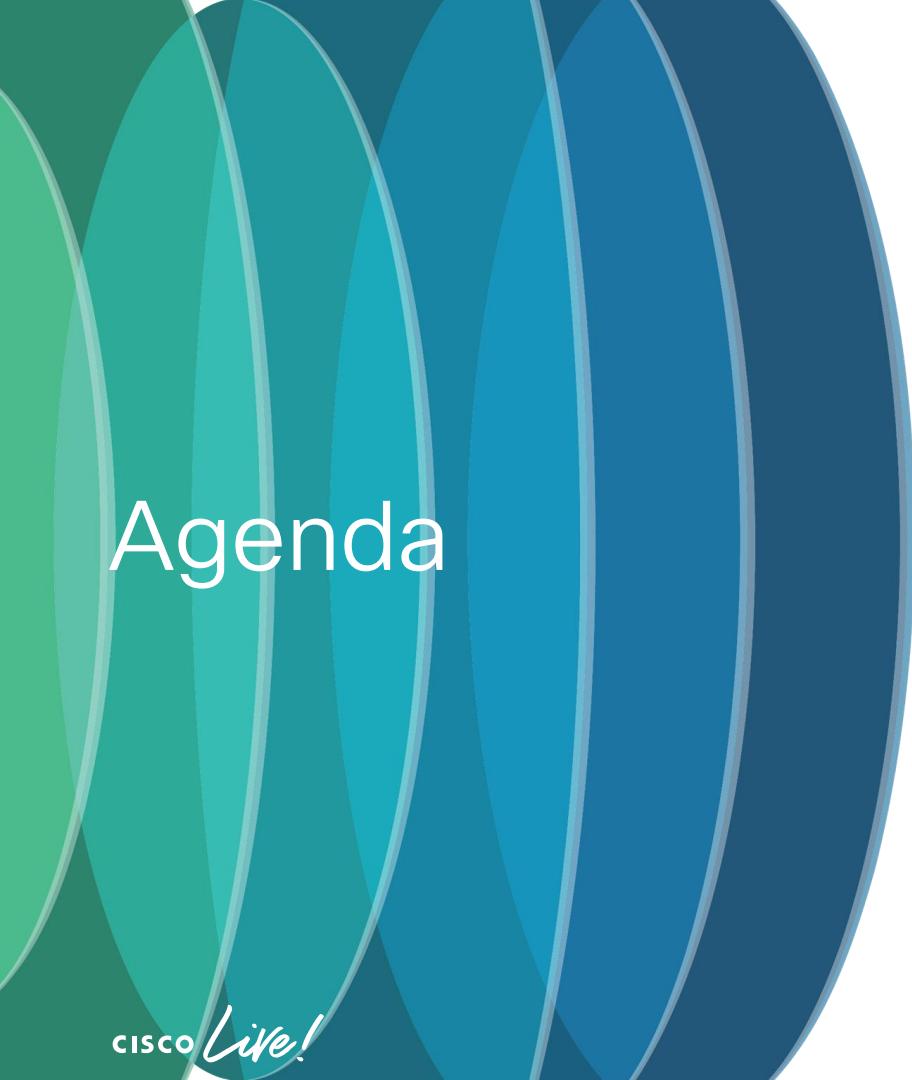
How

- 1 Find this session in the Cisco Events mobile app
- 2 Click “Join the Discussion”
- 3 Install the Webex app or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

CISCO *Live!*

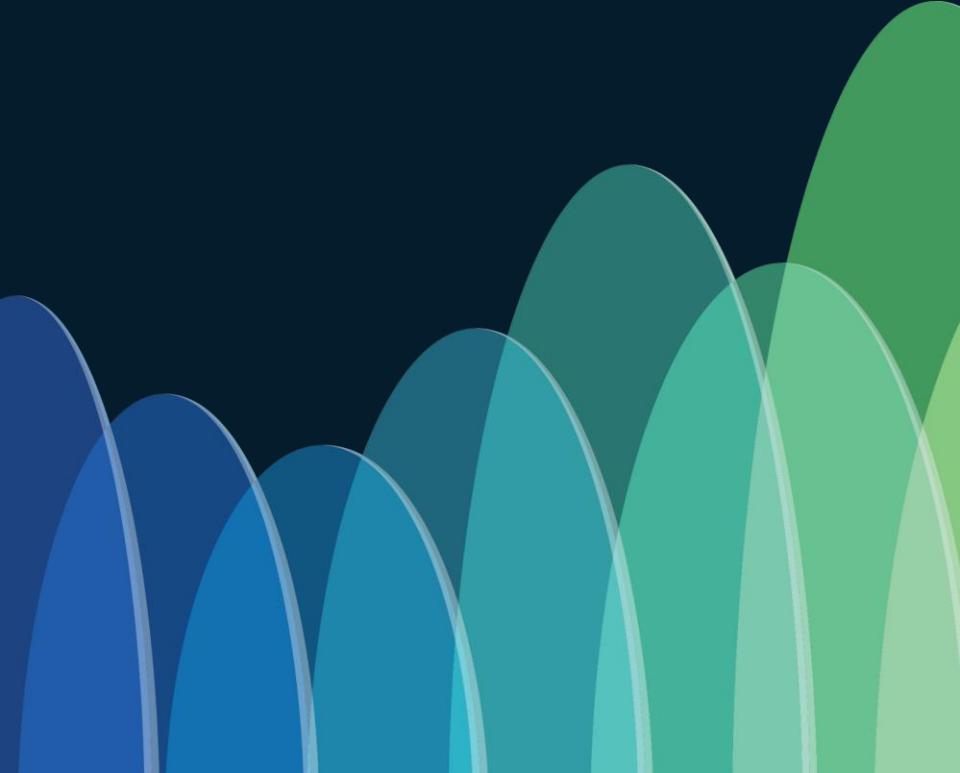


The background of the slide features a large, stylized graphic of overlapping circles in various shades of green and blue, creating a wave-like pattern. In the bottom left corner, the "CISCO Live!" logo is visible, with "CISCO" in a small sans-serif font and "Live!" in a larger, italicized, cursive font.

Agenda

- Intro to Coding Assistants
- Getting Started: GitHub Copilot
- Scripting Use Cases & Workflows
- Effectiveness of Copilot
- Caveats & Concerns
- Updates/Looking Forward
- Conclusion

Coding Assistants
improve developer
productivity
using GenAI.



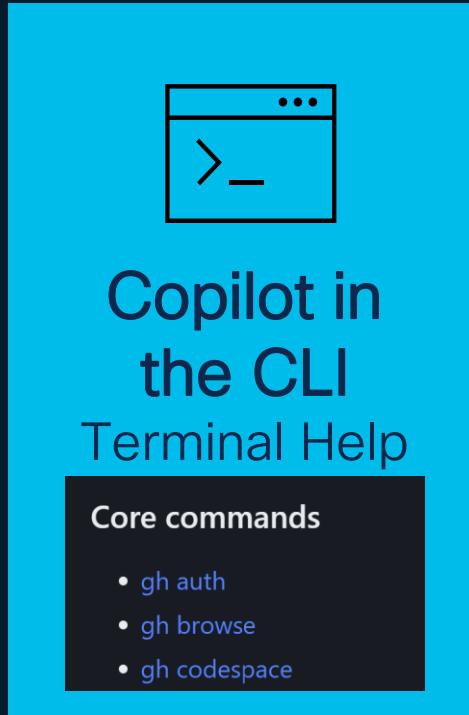
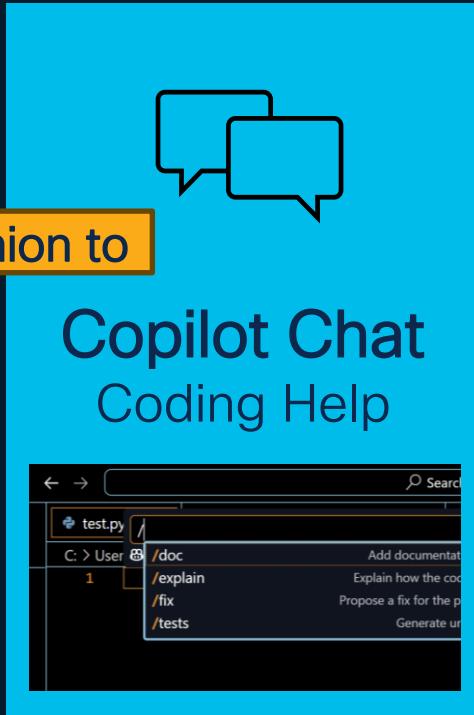
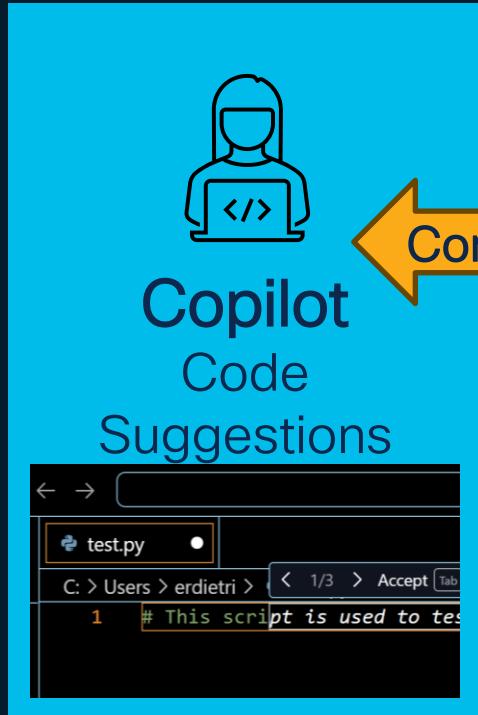
There is GenAI, and there is GenAI tooling.

You don't need to "know AI" to use an AI tool!

You need to know **how to use the tool**, which includes
knowing **how to create effective prompts**.

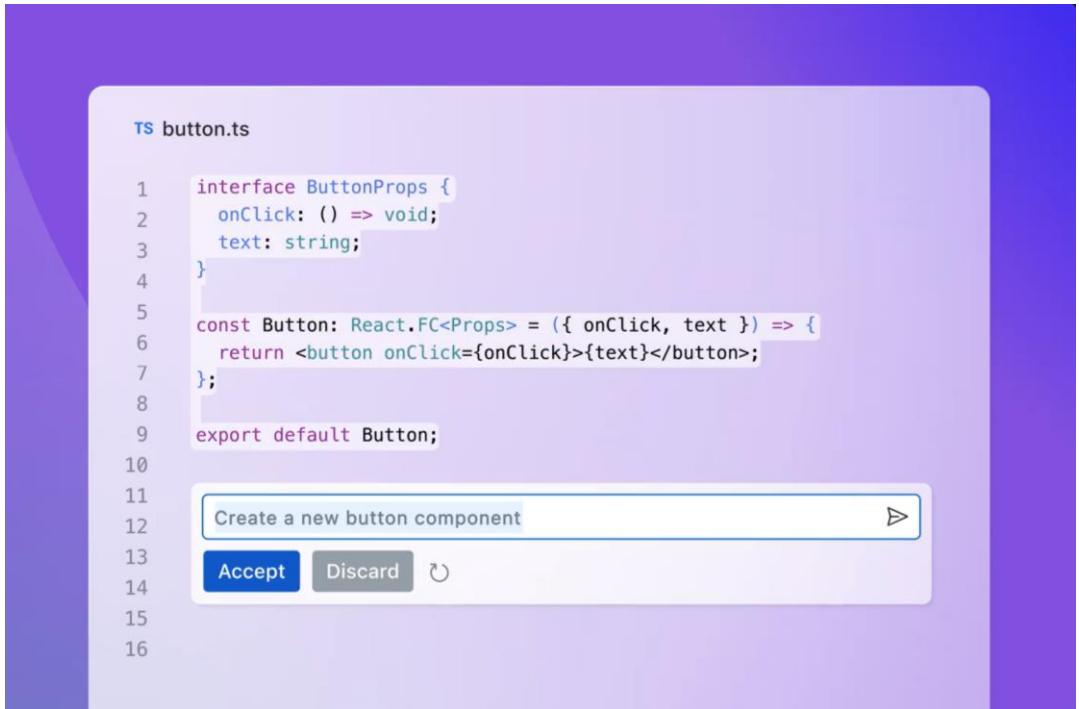
Getting Started: GitHub Copilot

The GitHub Copilot Suite



GitHub Copilot is the most widely used coding assistant in the world.

- Developed by GitHub and OpenAI in 2021
- Supports Python, JavaScript, TypeScript, Ruby, Go, C# and C++
- Developers are using it to **write code 55% faster**



The screenshot shows a code editor window for a file named `button.ts`. The code defines a TypeScript interface for a button component and a corresponding implementation:

```
TS button.ts

1  interface ButtonProps {
2    onClick: () => void;
3    text: string;
4  }
5
6  const Button: React.FC<Props> = ({ onClick, text }) => {
7    return <button onClick={onClick}>{text}</button>;
8  };
9
10 export default Button;
```

Below the code, a modal window is displayed with the text "Create a new button component" and two buttons: "Accept" and "Discard".

"Why use Copilot when I can use ChatGPT?"



"Why use Copilot when I can use ChatGPT?"

GitHub Copilot	ChatGPT
Specialized Purpose	General Purpose
More Accurate Code	Less Accurate Code

"Why use Copilot when I can use ChatGPT?"

GitHub Copilot	ChatGPT
Specialized Purpose	General Purpose
More Accurate Code	Less Accurate Code
More Contextual Awareness	Less Contextual Awareness

"Why use Copilot when I can use ChatGPT?"

GitHub Copilot	ChatGPT
Specialized Purpose	General Purpose
More Accurate Code	Less Accurate Code
More Contextual Awareness	Less Contextual Awareness
Can Include Internal Knowledge Base	Publicly Available Information Only

"Why use Copilot when I can use ChatGPT?"

GitHub Copilot	ChatGPT
Specialized Purpose	General Purpose
More Accurate Code	Less Accurate Code
More Contextual Awareness	Less Contextual Awareness
Can Include Internal Knowledge Base	Publicly Available Information Only
Subscription-Based	Free for v3.5
More Technical	Familiar/Friendly Interface

How useful or reliable is Copilot **really**?

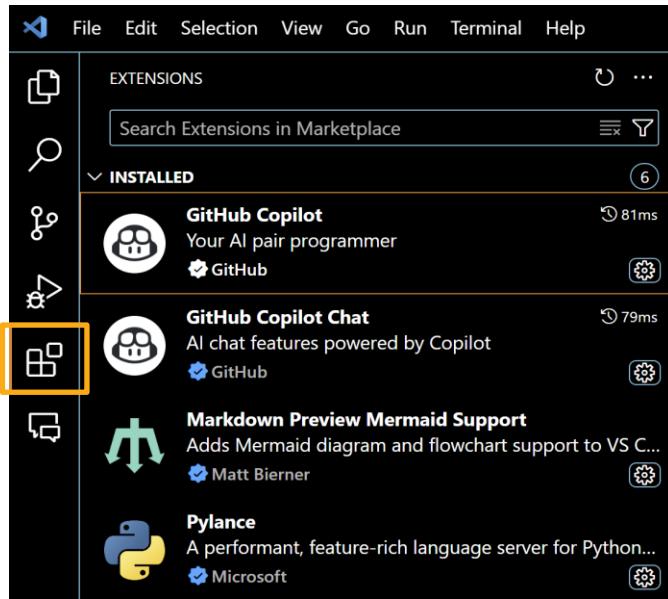
Prerequisites To Using GitHub Copilot

- Available as an IDE extension:
 - VS Code, Azure Data Studio, JetBrains IDEs, Xcode, Vim, Neovim, Visual Studio
- **BIG NEWS:** Free Tier now available! (It has limitations)

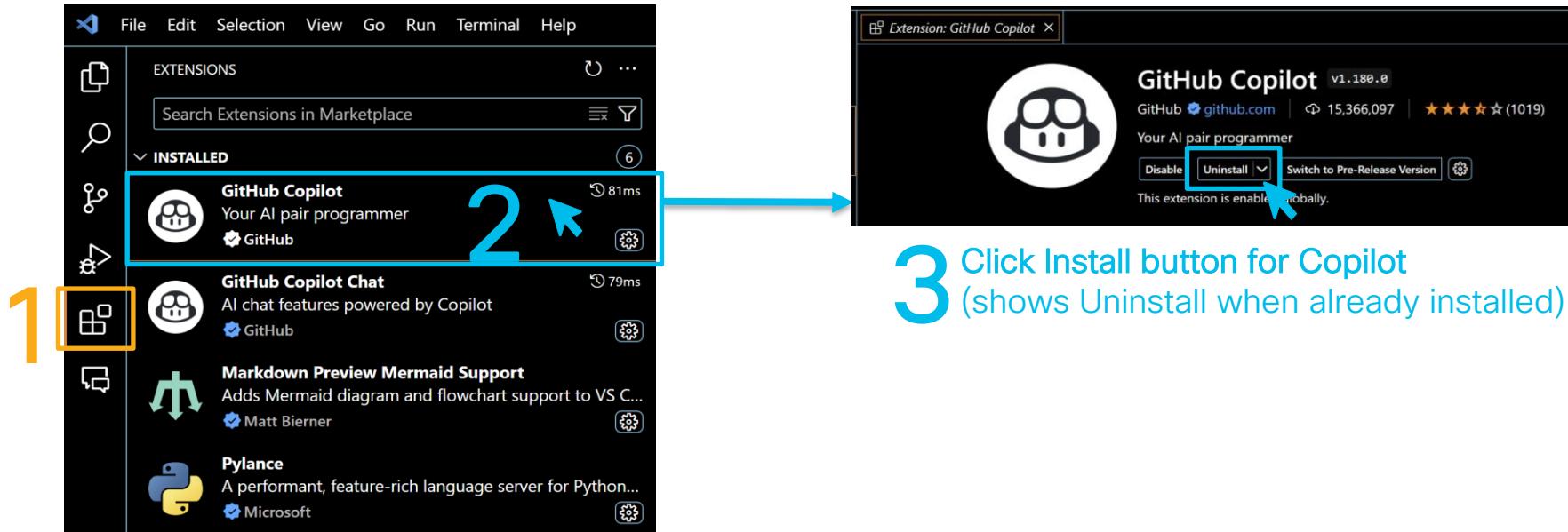
The image shows the GitHub Copilot pricing page with four tiers: Free, Pro, Business, and Enterprise. Each tier has a 'Get started' button and a 'Contact sales' button.

Tier	Price	Description
Free	\$0 USD	For developers looking to get started with GitHub Copilot. Includes up to 2,000 completions and 50 chat requests per month.
Pro	\$10 USD per month. First 30 days free.	For developers who want unlimited access to GitHub Copilot. Free for verified students, teachers, and maintainers of popular open source projects.
Business	\$19 USD per user / month	For teams ready to accelerate their workflows with GitHub Copilot.
Enterprise	\$39 USD per user / month	For organizations seeking a fully customized GitHub Copilot experience.

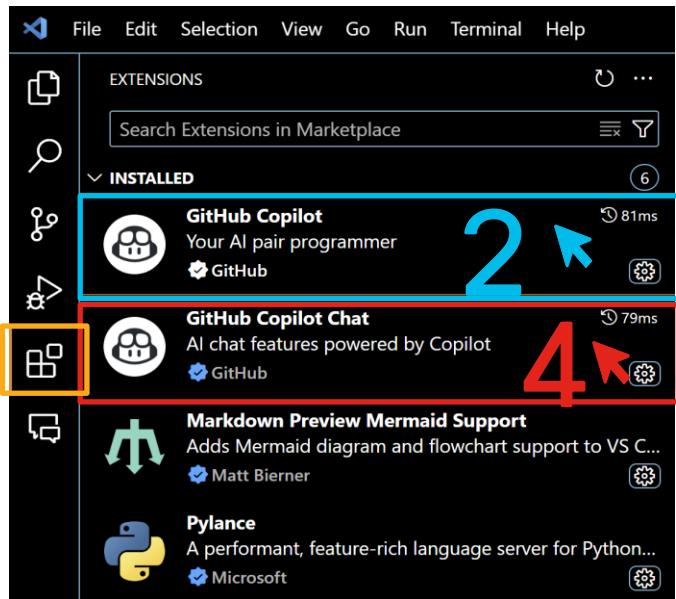
Install GitHub Copilot with Chat in VS Code 2022



Install GitHub Copilot with Chat in VS Code 2022



Install GitHub Copilot with Chat in VS Code 2022



1



3 Click Install button for Copilot
(shows Uninstall when already installed)



5 Click Install button for Copilot Chat
(shows Uninstall when already installed)

Using Copilot Inside VS Code

Press Ctrl+I to pull up Copilot

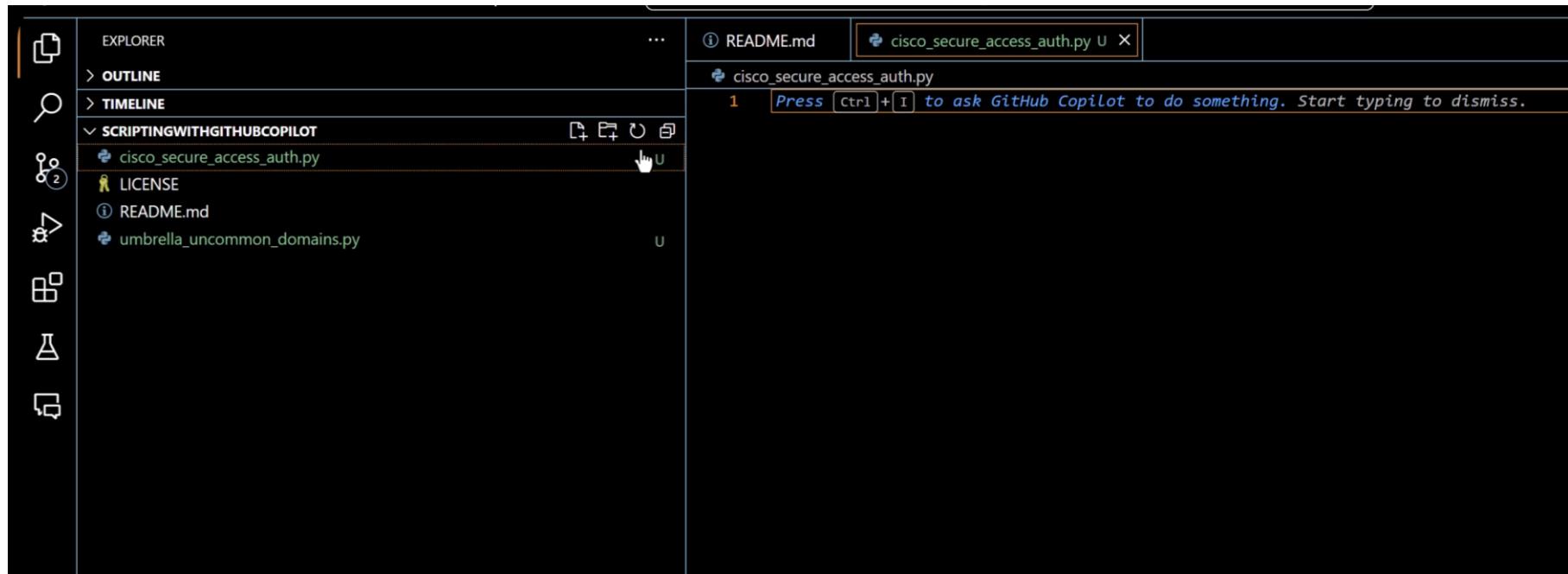
To view Copilot Chat window

After typing, can click the sparkle to pull up Copilot

Start a chat →

Scripting Use Cases with Copilot

Accept predictive code suggestions as you type.



Copilot can generate comments for code and predict the next comment based on context.

```
git cisco_secure_access_reporting.py > ...
1  import requests
2  import json
3  import requests
4
5
6
7  # Authenticate to the Cisco Secure Access Reporting API
8  def authenticate():
9
10 url = "https://api.sse.cisco.com/auth/v2/token"
11
12 payload = None
13
14 headers = {
15     "Content-Type": "application/x-www-form-urlencoded",
16     "Accept": "application/json"
17 }
18
19 response = requests.request('POST', url, headers=headers, data = payload)
20
21 return(response.json()['access_token'])
22
23 # Get the list of all the devices
```

A decently written comment can be enough for GitHub Copilot to suggest accurate code.

After using various files in a script, at the end of the script, I wanted to remove all unnecessary files.

Copilot correctly understood which files I would consider "unnecessary" based on context.

Original Comment

```
128
129 # Clean up files used to determine uncommon domains.
130
```

Accepted Code Suggestion

```
129 # Clean up files used to determine uncommon domains.
130 def clean_up_files():
131     os.remove('top-1m.csv.zip')
132     os.remove('top-1m.csv')
133     os.remove('top_destinations.csv')
134     os.remove('top_1million_csv')
```

GitHub Copilot provides you with code options.

Option 1

Original Comment

```
128
129  # Clean up files used to determine uncommon domains.
130
```

```
128
129 < 1/3 > Accept Tab Accept Word Ctrl + RightArrow ...
130 def clean_up_files():

    os.remove('top-1m.csv.zip')
    os.remove('top-1m.csv')
    os.remove('top_1million_csv')
    os.remove('top_destinations.csv')
```

Option 2

```
128
129 < 2/3 > Accept Tab Accept Word Ctrl + RightArrow ...
130 def clean_up_files():

    os.remove('top-1m.csv.zip')
    os.remove('top-1m.csv')
    os.remove('top_1million_csv')
    os.remove('top_destinations.csv')

    print("Files have been cleaned up.")
```

Option 3

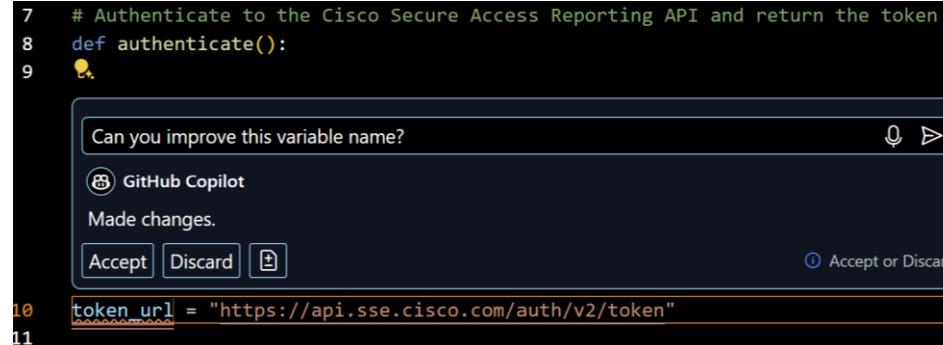
This alternative became unavailable to me while creating this example! Alternatives suggested often change.

Copilot can suggest better variable and function names.

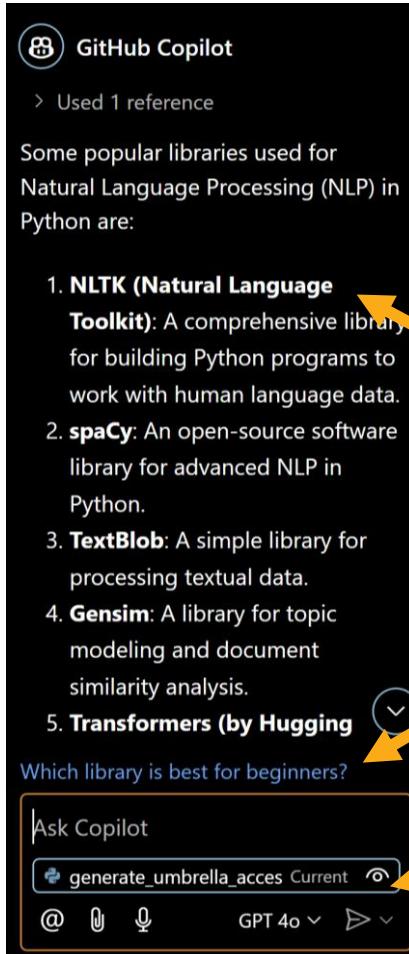
Original Variable Name

```
7  # Authenticate to the Cisco Secure Access Reporting API and return the token
8  def authenticate():
9
10 url = "https://api.sse.cisco.com/auth/v2/token"
11
```

Suggested Variable Name



Copilot chat can be used for learning.
(i.e. understanding a new library)



“What libraries are used for NLP in Python?”

List of libraries with short descriptions

Suggests a follow-up question

Indicates files referenced and LLM used

Get suggestions around code security.

```
2 import requests
3 import json
4 import os
5 import pandas as pd
6 from dotenv import load_dotenv
7
8 load_dotenv()
9
10 # Set the Cisco Umbrella Investigate API key
11 api_key = 'fjdsklewoiafewiaeji'
12
13 ✨
14
15
```

“How can I secure my API endpoints in Python?”



GitHub Copilot

› Used 1 reference

Securing API endpoints in Python involves several steps, including authentication, authorization, input validation, and using HTTPS. Below is a step-by-step guide to secure your API endpoints using Flask, a popular web framework for Python.

Step 1: Install Required Packages

First, ensure you have Flask and Flask-JWT-Extended installed:

```
pip install Flask Flask-JWT-Extended
```

Step 2: Set Up Your Flask Application

Create a basic Flask application with JWT authentication.

```
from flask import Flask, jsonify, request
from flask_jwt_extended import JWTManager, create_access_token, jwt_required, get_jwt_identity

app = Flask(__name__)
app.config['JWT_SECRET_KEY'] = 'your_jwt_secret_key' # Change this to a secure key
jwt = JWTManager(app)
```

“How can I secure my API endpoints in Python?”

 GitHub Copilot
Used 1 reference

Helpful!!

Securing API endpoints in Python involves several steps, including authentication, authorization, input validation, and using HTTPS. Below is a step-by-step guide to secure your API endpoints using Flask, a popular web framework for Python.

Step 1: Install Required Packages

First, ensure you have Flask and Flask-JWT-Extended installed:

```
pip install Flask Flask-JWT-Extended
```

Step 2: Set Up Your Flask Application

Create a basic Flask application with JWT authentication.

```
from flask import Flask, jsonify, request
from flask_jwt_extended import JWTManager, create_access_token, jwt_required, get_jwt_identity

app = Flask(__name__)
app.config['JWT_SECRET_KEY'] = 'your_jwt_secret_key' # Change this to a secure key
jwt = JWTManager(app)
```

“How can I secure my API endpoints in Python?”



GitHub Copilot

› Used 1 reference

Helpful!!

Securing API endpoints in Python involves several steps, including authentication, authorization, input validation, and using HTTPS. Below is a step-by-step guide to secure your API endpoints using Flask, a popular web framework for Python.

Step 1: Install Required Packages

First, ensure you have Flask and Flask-JWT-Extended installed:

```
pip install Flask Flask-JWT-Extended
```

Step 2: Set Up Your Flask Application

Create a basic Flask application with JWT authentication.

```
from flask import Flask, jsonify, request
from flask_jwt_extended import JWTManager, create_access_token, jwt_required, get_jwt_identity

app = Flask(__name__)
app.config['JWT_SECRET_KEY'] = 'your_jwt_secret_key' # Change this to a secure key
jwt = JWTManager(app)
```

Less helpful

(Referencing the previous code)

“Can you tell me if this code is secure?”

 GitHub Copilot

➤ Used 1 reference

The code snippet you provided contains a potential security risk because the JWT secret key is hardcoded directly in the source code. Hardcoding sensitive information like secret keys can lead to security vulnerabilities, especially if the code is shared or stored in a version control system.

Recommendations for Securing the JWT Secret Key

1. **Environment Variables:** Store the secret key in an environment variable.
2. **Configuration Files:** Use a configuration file that is not included in version control.
3. **Secrets Management Services:** Use a secrets management service like AWS Secrets Manager, Azure Key Vault, or HashiCorp Vault.

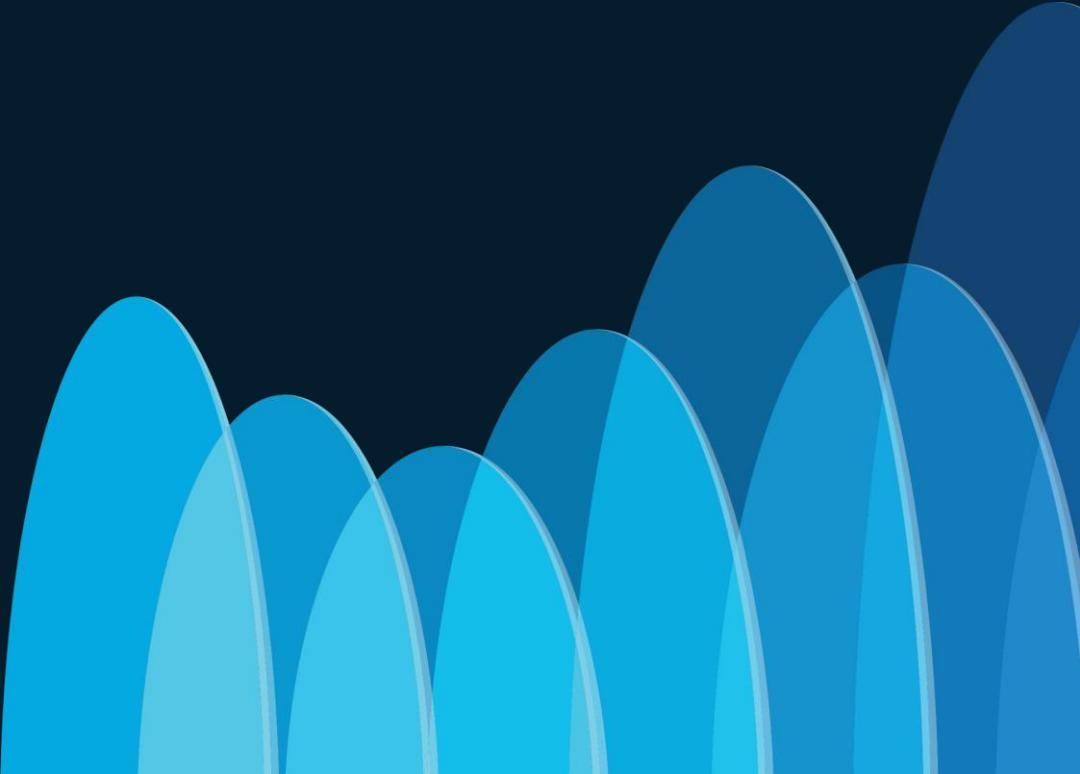
Example Using Environment Variables

1. **Set the Environment Variable:**

- On Windows:

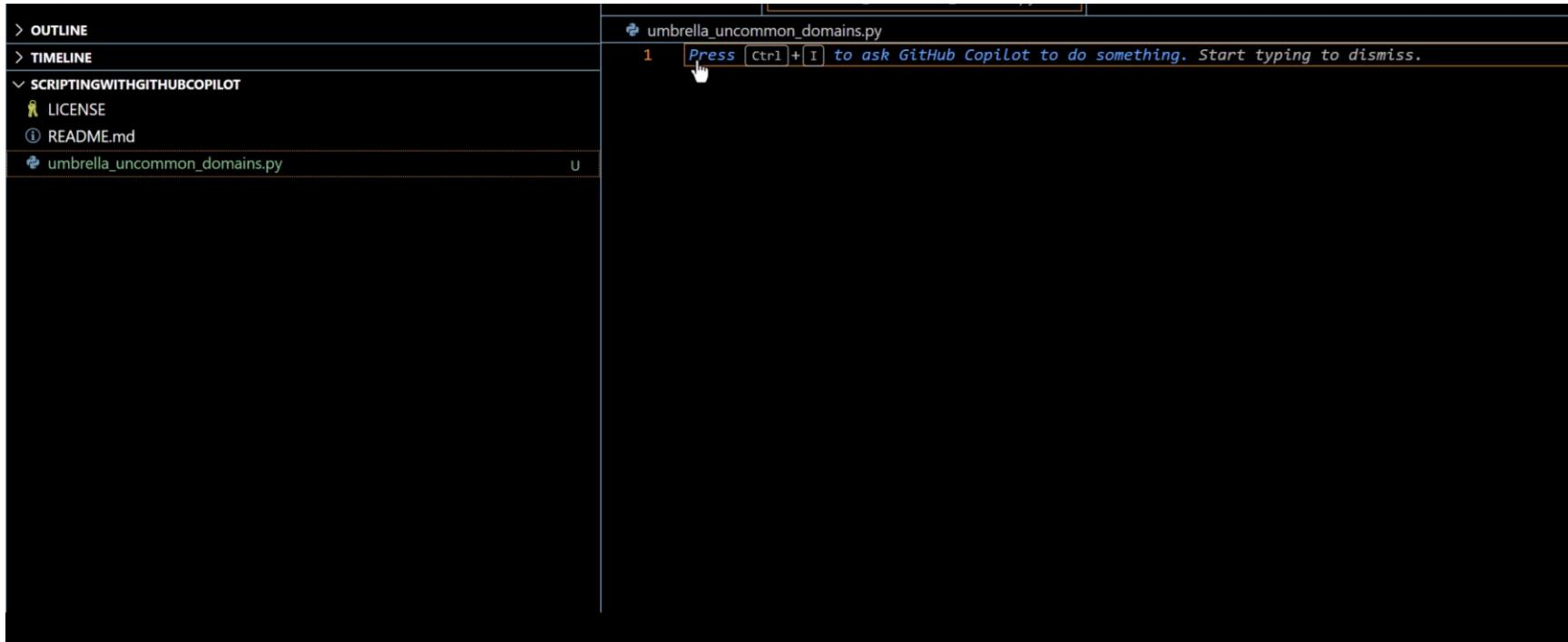
```
set JWT_SECRET_KEY=your_jwt_secret_key
```

Scripting Workflows with Copilot



Option 1: Copilot-Informed Scripting

Code as you normally would, but with Copilot enabled.



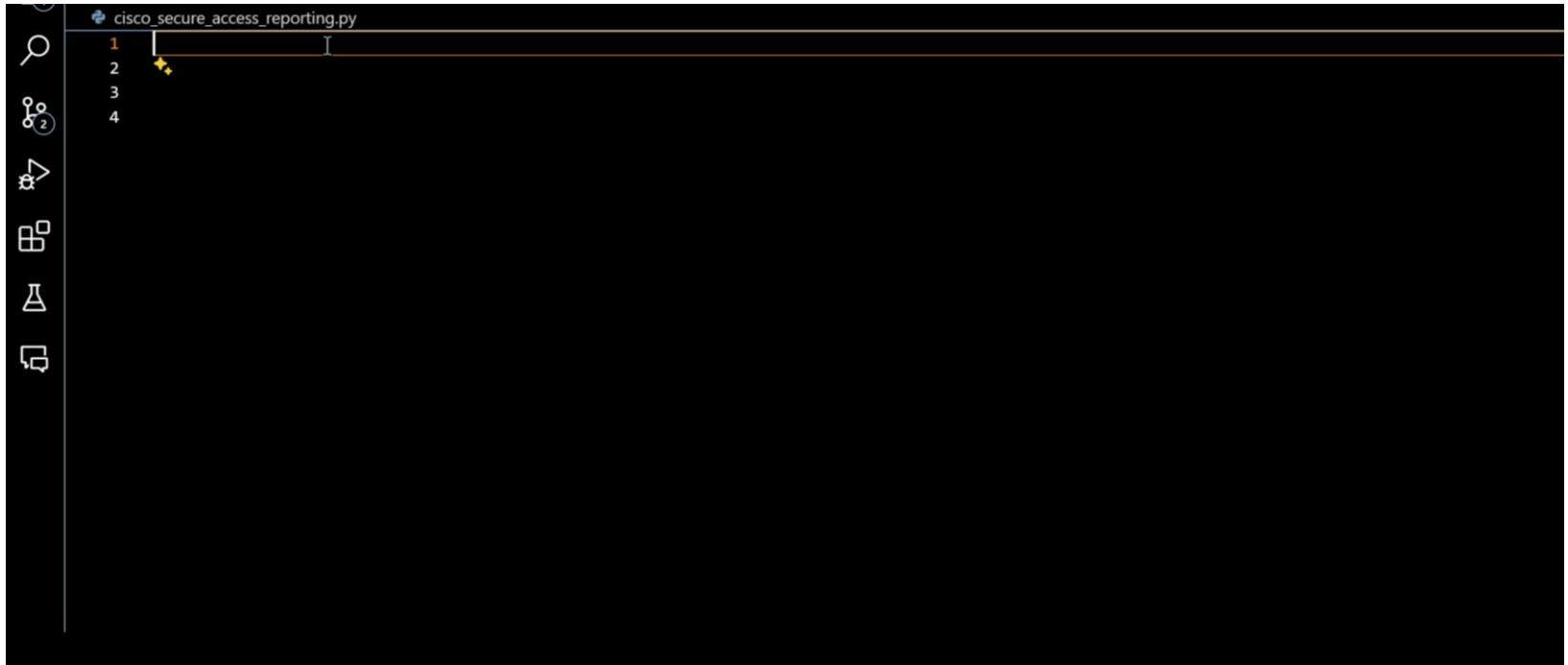
The screenshot shows a dark-themed code editor interface. On the left, there's a sidebar with 'OUTLINE', 'TIMELINE', and a expanded section 'SCRIPTINGWITHGITHUBCOPilot' containing 'LICENSE' and 'README.md'. The main workspace shows a file named 'umbrella_uncommon_domains.py'. A tooltip in the top right corner provides instructions: 'Press [Ctrl] + [I] to ask GitHub Copilot to do something. Start typing to dismiss.' A cursor is hovering over the tooltip. The right side of the interface has a vertical color bar.

Ways To Be “Copilot-Informed”

- Accepting predictive text as convenient
- Asking Chat questions as needed
- Getting code security suggestions
- Getting refactoring suggestions
- Generating comments
- Generating code documentation

Option 2: Comment-Driven Scripting

Generate code from descriptive comments ("effective prompts")



```
cisco_secure_access_reporting.py
1
2
3
4
```

How To Be “Comment-Driven”

- Generate code from descriptive comments ("effective prompts")
- Being "Comment-driven" also requires...
 - Knowing how to implement suggested logic
 - Evaluation of all code generated
 - Willingness to make code changes where needed

Effectiveness of Copilot



How useful or reliable is Copilot **really**?

Highly reliable

- Variable naming
- Function naming
- Answering questions about libraries
- Teaching concepts

Moderately reliable

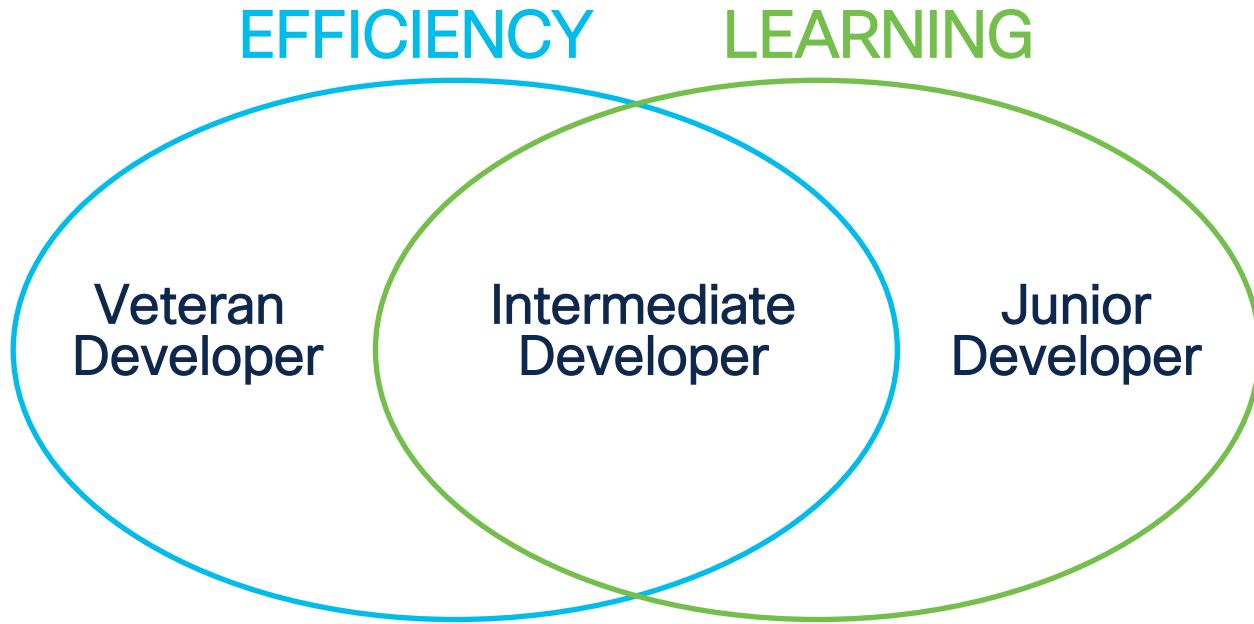
- Comment generation
- Code generation from comments
- Specific code generation
- Secure code suggestions

Not so reliable

- Code scaffolding
- API requests
- Complex code generation

The smaller and
more specific the task,
the more reliable Copilot is.

Where can Copilot **provide the most value** when scripting?



Caveats

Coding assistants
aren't always correct.

Faster code
doesn't equal
better code.

Coding assistants
are not a replacement
for knowing how to code.

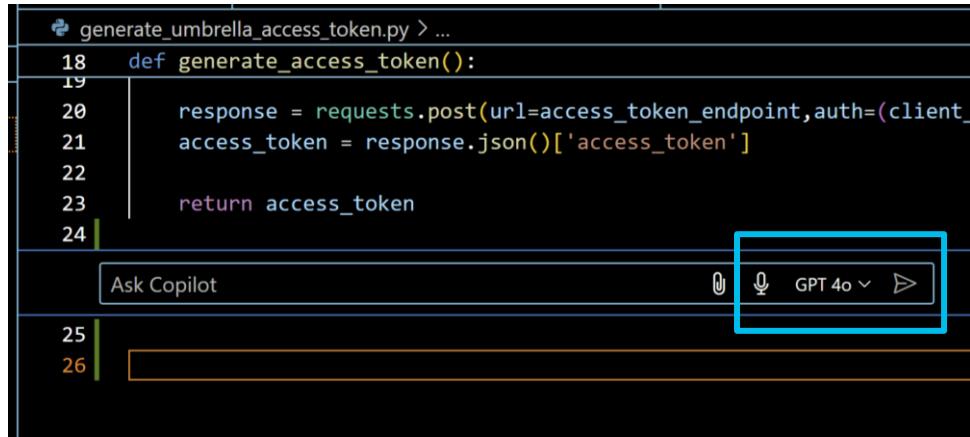
“What is Copilot doing with my data?”

- Prompts are encrypted (in-transit and at rest) and transmitted to Azure, where Copilot’s model is hosted
- If using the Copilot extension, Copilot will discard your prompt, suggestion, and supporting context.
- GitHub doesn’t train on code from Business/Enterprise customers, but Individuals must opt-out

Updates/ Looking Forward

Copilot is now offering developer choice in large language model.

- Anthropic's Claude 3.5 Sonnet
- Google's Gemini 1.5 Pro
- OpenAI's GPT-4o
- OpenAI's o1-preview
- OpenAI's o1-mini



```
generate_umbrella_access_token.py > ...
18  def generate_access_token():
19
20      response = requests.post(url=access_token_endpoint, auth=(client
21          access_token = response.json()['access_token']
22
23      return access_token
24
25
26
```

Ask Copilot

Ask Copilot

GPT 4o

The next big thing: Workspaces

Guidelines for teaching junior engineers to code in the age of AI.

Conclusion

AI coding assistants
are helpful assistants only;
you are the brain.

How will **you** start using an AI coding assistant in your workflow?

cisco *Live!*



GO BEYOND

“What is Copilot doing with my data?”

- Prompts are encrypted (in-transit and at rest) and transmitted to Azure, where Copilot’s model is hosted
- If using the Copilot extension, Copilot will discard your prompt, suggestion, and supporting context.
- GitHub doesn’t train on code from Business/Enterprise customers, but Individuals must opt-out

Webex App

Questions?

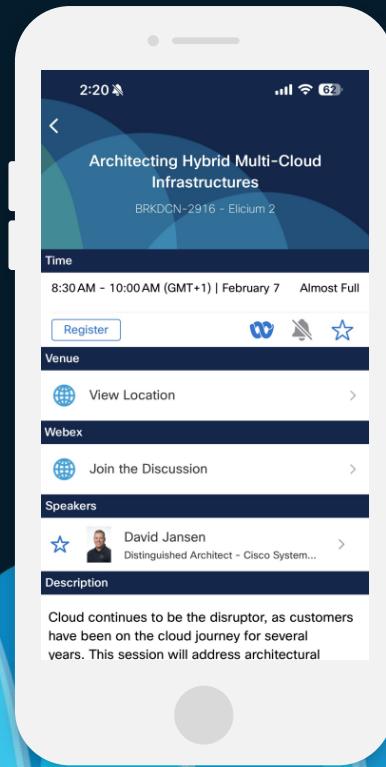
Use the Webex app to chat with the speaker after the session

How

- 1 Find this session in the Cisco Events mobile app
- 2 Click “Join the Discussion”
- 3 Install the Webex app or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

CISCO *Live!*



Fill Out Your Session Surveys



Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)



All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the 'Participant Dashboard'



Content Catalog

A dark blue background featuring a series of overlapping, semi-transparent blue wave-like shapes that create a sense of depth and motion.

Continue your education

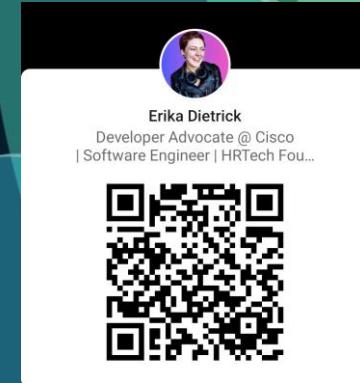
CISCO *Live!*

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Thank you

cisco *Live!*

LinkedIn



X, TikTok, Instagram:
@erika_thedev



cisco *Live!*



GO BEYOND