



—blexin.
CON NOI È SEMPLICE

GITHUB COPILOT: THE AGENT AWAKENS

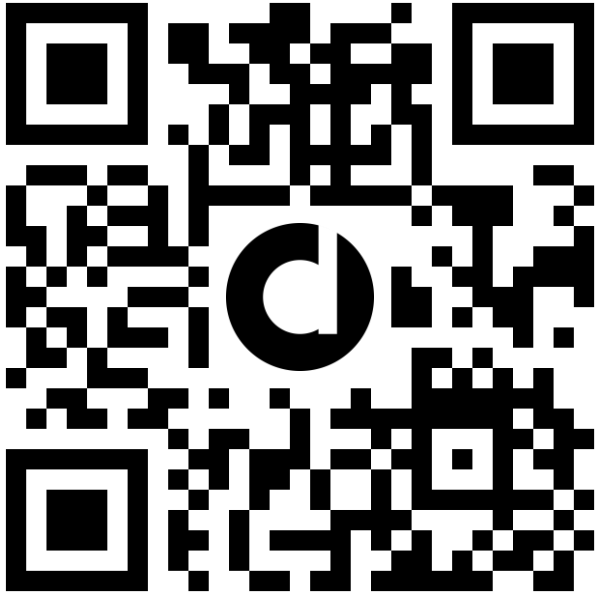


GERARDO GRECO
*SENIOR DEVELOPER &
DEVOPS ENGINEER @ BLEXIN*
GERARDO.GRECO@BLEXIN.COM



ANTONIO LICCARDI
*CHIEF TECHNOLOGY OFFICER @ BLEXIN
MICROSOFT MVP FOR DEV TECHNOLOGIES*
ANTONIO.LICCARDI@BLEXIN.COM

Repository del workshop



[https://git.new/
e2fzHVk](https://git.new/e2fzHVk)



Come è organizzato il codice?

- Il workshop è strutturato in moduli
- I moduli che includono codice sono organizzati nelle cartelle **Esercizio/Soluzione**
- Per gli esercizi è possibile selezionare il codice in **C#**, **TypeScript** o **JavaScript**
- Ogni modulo include il file **Readme.md** con le istruzioni

Quanto mi costa?

	Copilot Free	Copilot Pro	Copilot Pro+	Copilot Business	Copilot Enterprise
Pricing	Not applicable	\$10 USD per month, or \$100 USD per year (free for some users)	\$39 USD per month, or \$390 USD per year	\$19 USD per granted seat per month	\$39 USD per granted seat per month
Premium requests	50 per month	300 per month	1500 per month	300 per user per month	1000 per user per month
Purchase additional premium requests at \$0.04/request	X	✓	✓	✓	✓

Standard Models	
✓ GPT-4.1	Included
GPT-4o	Included
Premium Models	
Claude Sonnet 3.5	1x
Claude Sonnet 3.7	1x
Claude Sonnet 4 (Preview)	1x
Gemini 2.5 Pro (Preview)	1x
o4-mini (Preview)	0.33x
Manage Models...	

Copilot Usage	
Code completions	Included
Chat messages	Included
Premium requests	4,7%
Additional paid premium requests disabled.	
Manage paid premium requests	
Allowance resets July 1, 2025.	

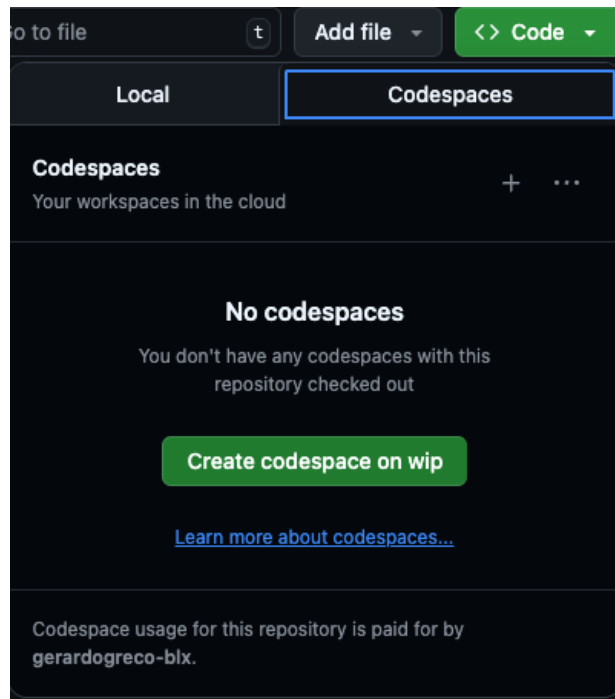
<https://docs.github.com/en/copilot/about-github-copilot/plans-for-github-copilot>

Github Codespace

GitHub Codespaces è un ambiente di sviluppo cloud-based che consente di scrivere, eseguire e fare il debug del codice direttamente dal browser, senza la necessità di configurare nulla sul proprio computer. È uno strumento ideale per iniziare rapidamente a lavorare su un progetto, con un setup già pronto all'uso.

Per iniziare:

- Effettuate un **fork** del repository.
- Una volta effettuato il fork, seguite i **passaggi** indicati nel Readme del vostro repository.





La prima regola per un prompt efficace

Quality In = Quality Out

Un prompt iniziale ben formulato — chiaro, completo e contestualizzato — permette di ottenere risposte più pertinenti, accurate ed efficaci.

Tecniche per prompt efficaci

Contesto dettagliato

Fornisci informazioni sul linguaggio, framework e obiettivo specifico del codice

Vincoli e requisiti

Elenca limiti, standard di codifica e requisiti di performance

Custom instructions

Fornire istruzioni e regole che Copilot dovrà seguire

Prompt generico

Crea un metodo per ordinare

Troppo vago, Copilot non sa cosa ordinare, come e in che linguaggio

Esempi di Input/Output

Specifica esempi concreti di dati in ingresso e risultati attesi

Stile e convenzioni

Specifica lo stile di codifica e le convenzioni da seguire

Prompt specifico

Crea un metodo per ordinare in ordine crescente una lista di oggetti utente in base all'età.

Input: List<User> con proprietà Name e Age. Usa LINQ e le feature moderne di C# 13.

Esempio di output:

```
[ { Name: "Anna", Age: 25 }, { Name: "Marco", Age: 33 } ]
```

Chiaro, specifico e con esempio concreto

Copilot Chat: slash commands & participants

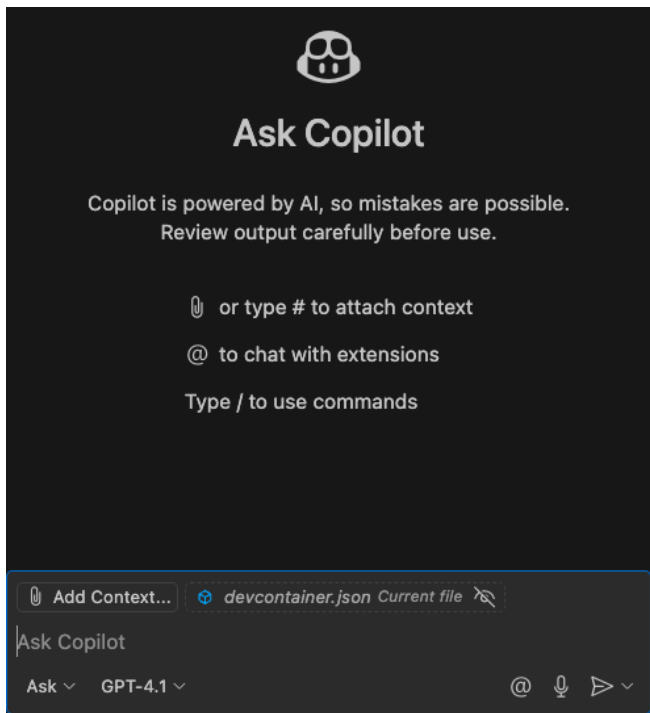
Slash commands: comandi per guidare e limitare le azioni di Copilot su un contesto specifico.

/fix	Permette di correggere errori nel codice
/test	Permette di generare unit test
/doc	Permette di generare documentazione
/new	Permette di impostare un nuovo progetto
/clear	Permette di cancellare il contesto corrente della chat, utile per iniziare una nuova conversazione

Chat participants: entità specializzate che forniscono a Copilot competenze o contesto su uno specifico dominio.

@workspace	Permette di accedere al contesto del workspace corrente
@terminal	Permette di eseguire comandi direttamente nel terminale
@vscode	Permette di interagire con l'IDE
@github	Permette di ottenere risposte basate sulla ricerca sul web, sul codice e sul knowledge base aziendale

Copilot modes: Ask

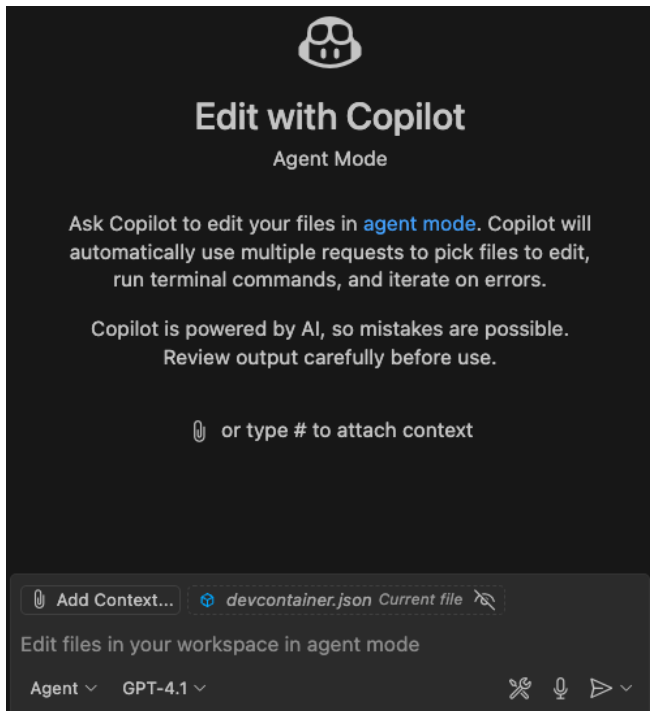


Modalità pensata per interagire con l'AI in modo semplice e diretto.

Funziona come una chat: scrivi in linguaggio naturale e Copilot ti risponde nel contesto del tuo codice.

Utile per capire, migliorare e velocizzare lo sviluppo anche quando non si sa da dove iniziare.

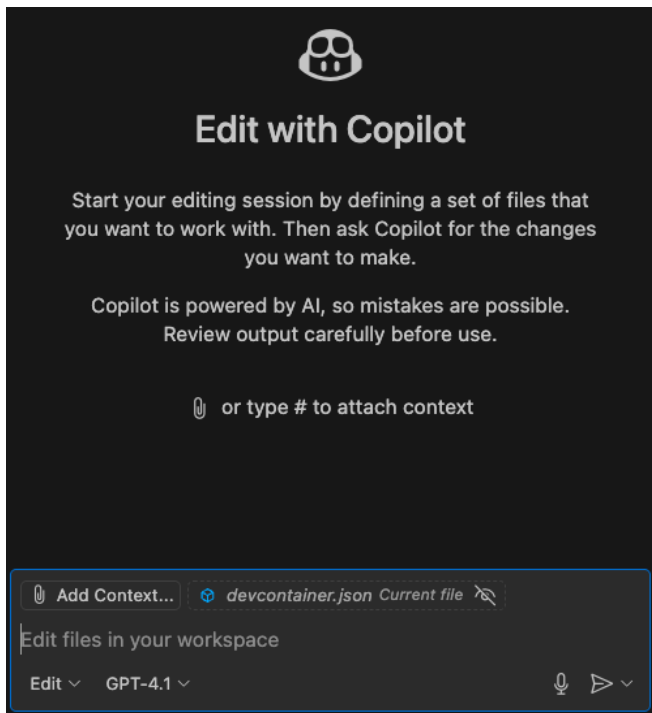
Copilot modes: Agent



Modalità pensata per automatizzare attività complesse, come se avessi un assistente che lavora al tuo fianco. È possibile:

Copilot analizzerà il contesto e proporrà modifiche multiple, guidandoti passo-passo.

Copilot modes: Edit



Modalità pensata per indicare un set **limitato** di file sui quali Copilot può intervenire, applicando le modifiche che chiedi, come:

Il focus è locale: Copilot lavora solo sulla selezione, rendendolo utile per migliorare o trasformare parti di codice mirate.

GitHub Copilot / Code Generation

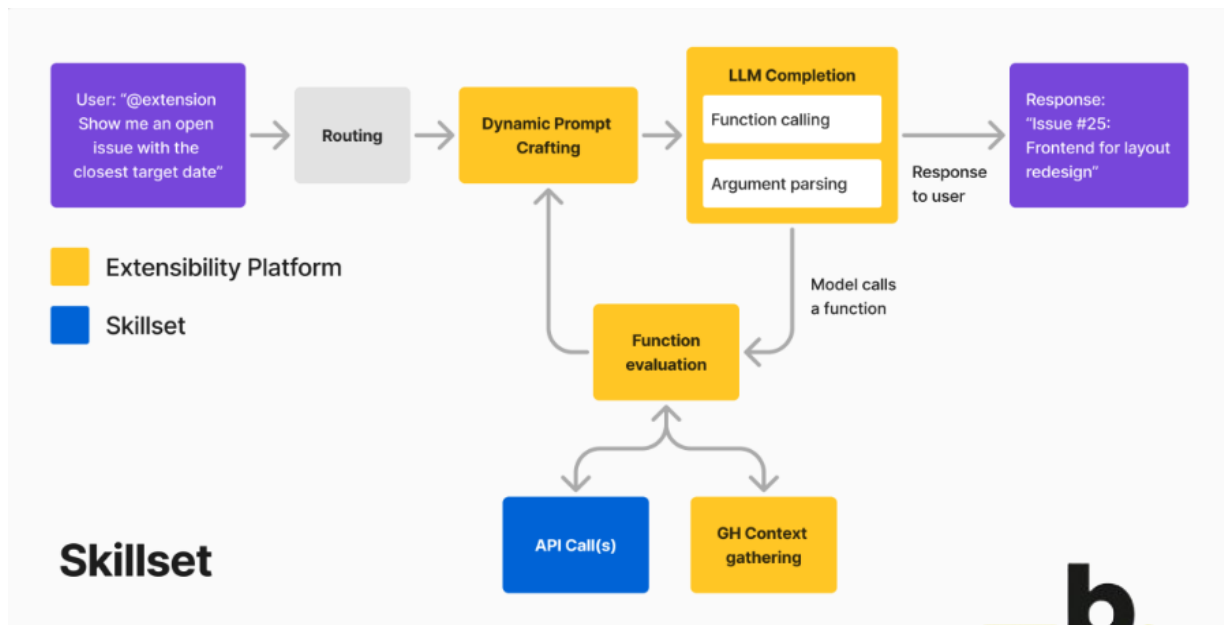
	Completions	Chat/Edits	Agent Mode	Copilot coding agent
Scope Of Changes	Next Few Lines	Multi-File Edits	Complete tasks	Entire issues
Frequency of Interaction	Hundreds of Milliseconds	Seconds	Minutes	Tens of minutes
Inner/Outer Loop	Inner Loop	Inner Loop	Inner Loop	Outer Loop
Developer Canvas	VS Code (Editor)	VS Code (Chat)	VS Code (Chat)	GitHub.com

Github Copilot: Skillset

Una Skillset è un insieme di funzionalità specializzate che estendono le capacità di GitHub Copilot, permettendogli di risolvere compiti specifici integrando API, contesto GitHub, e funzioni personalizzate

Quando utilizzarla?

Operazioni semplici, API calls, validazioni, trasformazioni dati

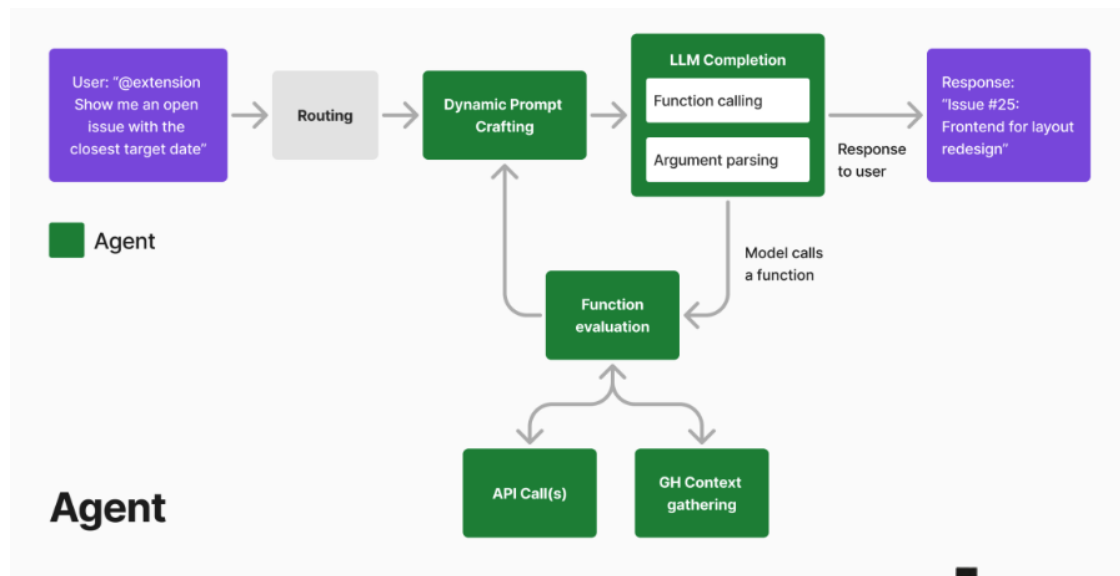



Github Copilot: Agent (extension)

Un Agent (extension) è un'estensione intelligente che lavora in sinergia con GitHub Copilot per rispondere a richieste complesse. Gli agenti combinano prompt dinamici, funzioni personalizzate e integrazione con contesto GitHub per fornire risposte precise e utili.

Quando utilizzarla?

Workflow complessi, analisi codice, debugging assistito, code review

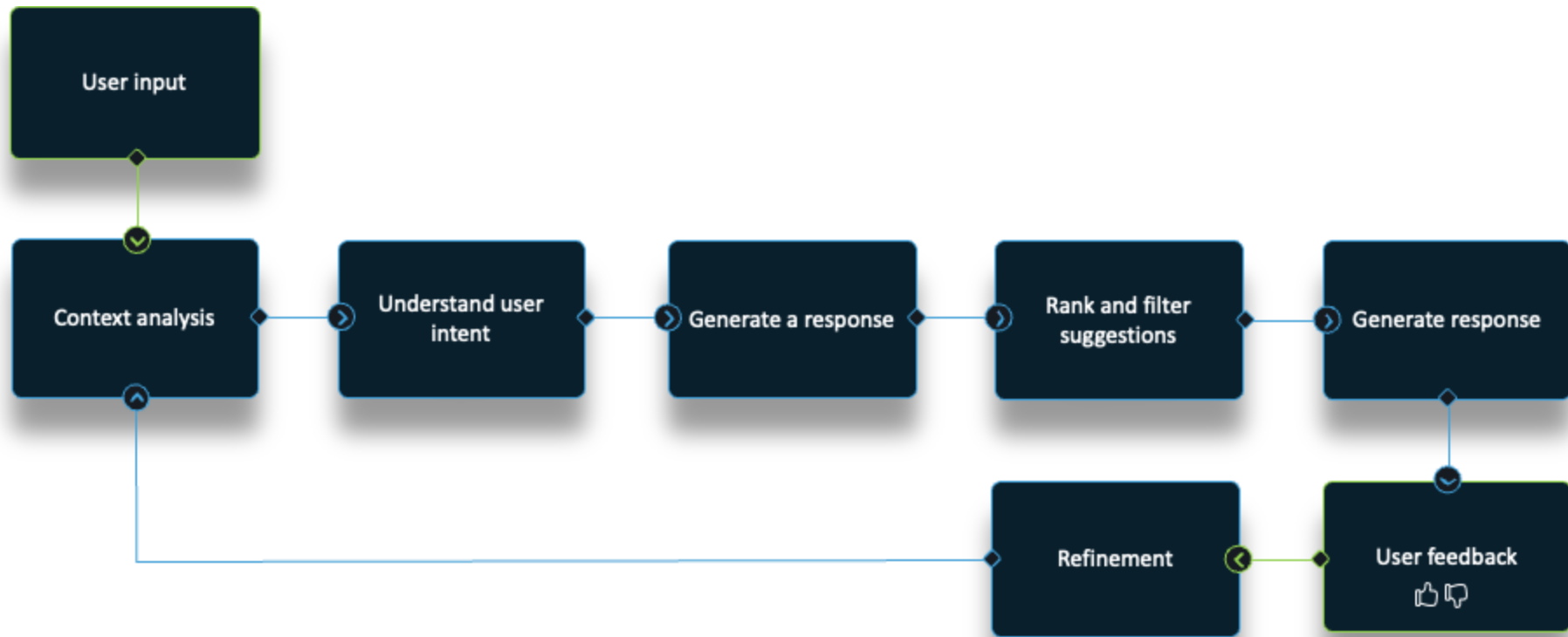


 Model Context Protocol

Bring in outside context with MCP servers



The Copilot request-response cycle



Instructions to provide guidance

- Repository-wide with copilot-instructions.md
- Task specific with instructions files

```
# Tailspin Toys Crowd Funding website
```

```
Website for crowd funding for games.
```

```
## Backend
```

```
The backend is written using:
```

- Flask for the API
- SQLAlchemy for the ORM

```
## Frontend
```

```
The frontend is written using:
```

- Astro for routing
- Svelte for the components and interactivity
- Tailwind CSS for styling

 **b.** THANK YOU