



GitHub Copilotをもっと活用！ ～コードレビューの紹介からTipsまで～ 2025年7月版

ファインディ株式会社 様 主催

GitHub Copilot 活用術

～MCP・レビュー・issue作成からCoding Agent活用まで！～



自己紹介

名前：いしかず

所属：トヨタコネクティッド株式会社 先行企画部 新技術開発室

※本日は個人として発表しており、所属している組織の活動とは無関係となります。

略歴：SESでの客先常駐→GMOメディアで媒体開発運用・情シス・AI推進 → 現職（2025/07～）

趣味：ドライブ/ドローン/ゲーム/某テーマパークに行くこと 🏰 🎆

Twitter(X): [@k-ishii1020](https://twitter.com/k-ishii1020)（いしかず） GitHub: [k-ishii1020](https://github.com/k-ishii1020)





今回のお話について

主にGitHub CopilotにおけるレビューについてのおさらいとちょっとしたTipsのご紹介。
基本的にはVS Codeを利用しているユーザ向けです。

2025年7月時点での情報なので、都度最新情報をご確認ください。

※今回出てこないお話

- GitHub Copilot Agent Mode
- GitHub Copilot Coding Agent



資料は後ほど共有します！

駆け足で進みますが、資料は後ほどconnpassで共有しますので
ざっくり流し見程度にご覧ください。

なお、設定方法なども記載したリンクの内容をご確認ください。

※リンク先へ遷移したい場合は右上のQRコードからご確認ください。

プレミアムリクエストが始まった



GitHub Copilot でプレミアムリクエストが始まった

本題の前にちょっとだけニュース…

2025年6月からGitHub Copilotの課金形態に「プレミアムリクエスト」という単位の概念が加わりました。

今後は契約プランやコーディング支援のシナリオごとに上限が発生します。



GitHub Copilot プレミアムリクエスト とは

高度な支援機能を使う際にはプレミアムリクエストがカウント（積算）されていきます

- ・ ベースモデル（GPT-4.1およびGPT-4o）**以外**でのコード補完（ClaudeとかGeminiなど）
- ・ チャット/チャットエージェント/コーディングエージェント/レビューなどの利用

Understanding and managing requests in Copilot

<https://docs.github.com/ja/copilot/concepts/copilot-billing/understanding-and-managing-requests-in-copilot>



GitHub Copilot プレミアムリクエスト 上限を超えると ... ?

- Freeプランユーザ… プレミアムリクエスト対象の機能が使用不可に😭
 - 有償プランへアップグレードするしかない。
※月間最大2,000件のコード補完リクエストと最大50件のプレミアムリクエスト
- 有償プランの場合…
 - 上限の緩いさらに上位プランへグレードアップする（Businessプラン→Enterpriseプラン）
上限はプランによる。有償プランでは毎月月初にリセットされる模様。余剰枠の翌月繰越不可。
 - あるいは、従量課金にする（上限金額の設定が可能）💰💰💰

予算を使用した従量制課金製品での支出の管理

<https://docs.github.com/ja/billing/managing-your-billing/using-budgets-control-spending>

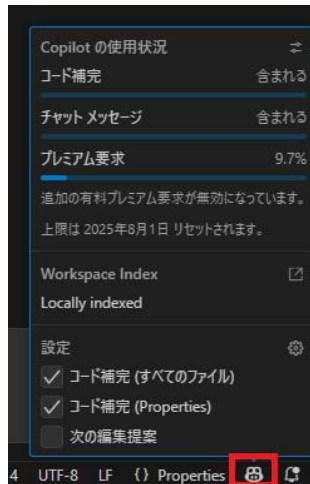


GitHub Copilot 使用状況の確認方法

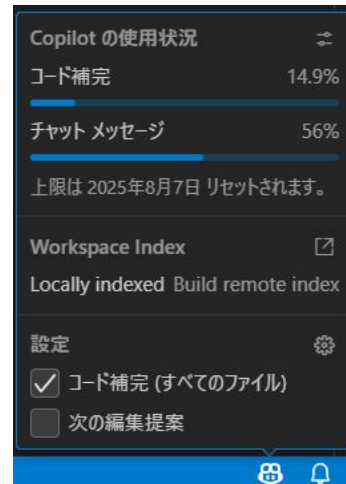
VSCodeの右下 Copilot君
のアイコンを押すと確認ができます。

有償版GitHubCopilot
プレミアム要求→プレミアムリクエスト

無償版GitHubCopilot
チャットメッセージ→プレミアムリクエスト
が含まれる模様？



有償版契約



Free版契約

GitHub Copilot コードレビュー

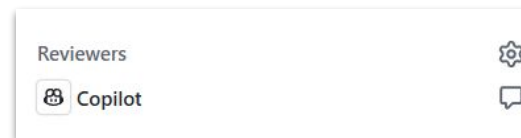
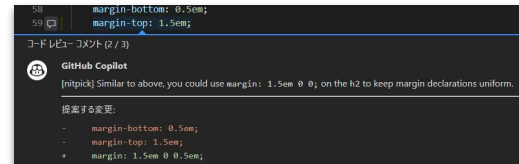
には大きく分けて 2 種類ある



GitHub Copilot コード レビュー

GitHub Copilot コード レビューには大きく 2 つの機能がある。

- IDE (VSCode・VisualStudio) 上でのコードレビュー
 - コード数行レベルでのレビュー機能
 - コミット前（ステージング）単位でのレビュー**Freeを含むすべてのGitHub Copilot プランで利用可能**
- プルリクエスト全体を通してレビューをしてくれる機能
 - ブラウザ上やVSCode上で確認が可能**有償プランのGitHub Copilot プランでのみ利用可能**



コード レビュー

<https://docs.github.com/ja/copilot/how-tos/agents/copilot-code-review>

GitHub Copilot コードレビュー IDE編

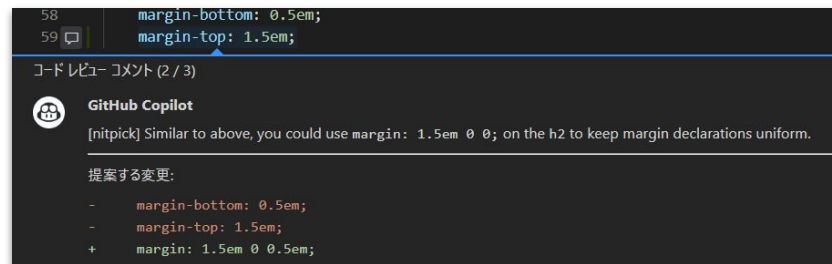
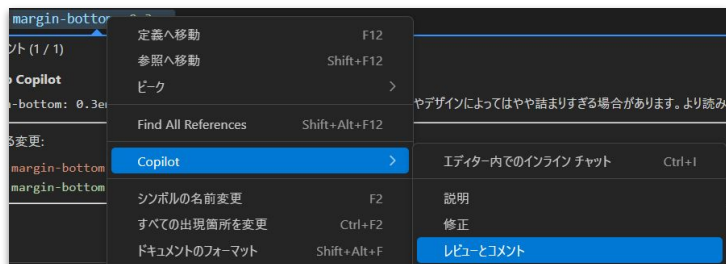


GitHub Copilot コード レビュー (IDE・範囲選択)

IDE上でのコードレビュー機能には、以下の2つの機能がある。

①カーソルで選択された範囲のコードをレビューする機能

1. レビューしてほしいコードを選択
2. 右クリック→Copilot→Review and Comment
or コマンドパレットを開いて、「GitHub Copilot: Review and Comment」を選択

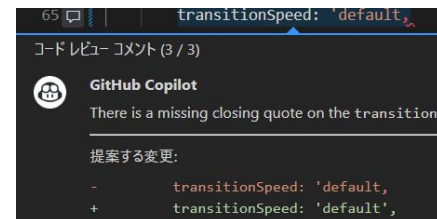
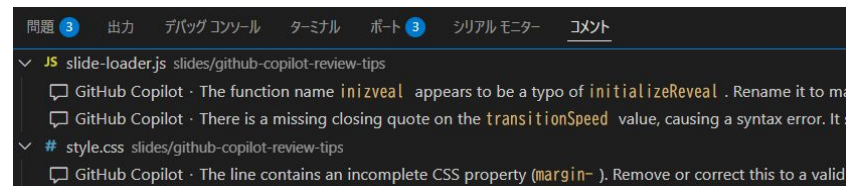
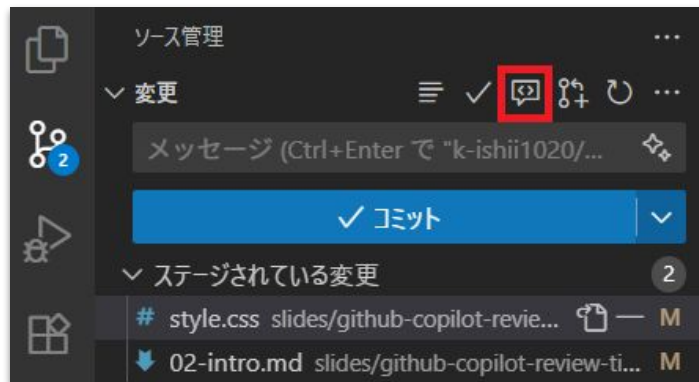




GitHub Copilot コード レビュー(IDE・ステージング)

②コミット前（ステージング中）のファイルまとめてレビューする機能

1. コミットしたいファイルをステージング状態にし、赤い枠のアイコンをクリック



GitHub Copilot レビュー プルリクエスト編



GitHub Copilot コード レビュー (プルリクエスト編)

プルリクエスト全体に対するレビュー機能

- PRの要約 (Overview)
- 指摘のコメントなど

初期状態だと英語でレビューしてしまうため
後述のTipsを活用すると良い。





GitHub Copilot コード レビュー(プルリクエスト編)

プルリクエスト全体に対するレビュー機能

- PRの要約 (Overview)
- 指摘のコメントなど

初期状態だと英語でレビューしてしまうため
後述のTipsを活用すると良い。

```
main.py
...    ...    @@ -17,7 +16,7 @@
17    16
19    18
20    - load_dotenv()
19    + load_dottenv()
```

Copilot AI 1 minute ago Preview ...

[must] 環境変数の読み込み関数名にタイプミスがあります。'load_dotnet()'ではなく、正しくは'load_dotenv()'であることを確認してください。

Copilot is powered by AI, so mistakes are possible. Review output carefully before use.

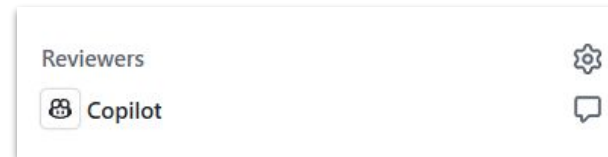
GitHub Copilot Tips集



Tips① 自動レビューを有効にする

PRのオープン時に自動でCopilot君をレビューワーにアサインできます。

- ・ 個人ユーザ… 自身で作成した全てのPRに自動アサインが設定可
- ・ リポジトリ所有者… 当該リポジトリにPR作成時に自動アサインが設定可
- ・ 組織の所有者（管理者）… 全てor任意のリポジトリにおいて自動アサインが設定可



Copilot を使って自動コード レビューを構成する

<https://docs.github.com/ja/copilot/how-tos/agents/copilot-code-review/configuring-automatic-code-review-by-copilot>



Tips② PRレビューのルール（指示）を有効にする

.github/copilot-instructions.mdに書かれたプロンプトを元にPRのレビューが可能です。

※元々はコード補完などをGitHub Copilotに指示をするためのmd

（以前はEnterpriseプランのみ利用可能）

しかし！2025年6月から**すべての有料プラン**で上記mdファイルに書かれたプロンプトもレビューに反映されるようになりました🎉🎉🎉

※現在はパブリックレビューなので管理者による設定が必要です。

General settings

Use custom instructions when reviewing pull requests

Copilot references [custom-instructions.md](#) when reviewing pull requests.

On ☒

Copilot code review: Customization for all

<https://github.blog/changelog/2025-06-13-copilot-code-review-customization-for-all>



Tips② PRレビューのルール（指示）を有効にする

ちなみに以前はEnterpriseプラン以外でもレビューに対するルールを適用したかったので、[.github/PULL_REQUEST_TEMPLATE.md](https://github.com/PULL_REQUEST_TEMPLATE.md)にこのようなプロンプトを仕込んでレビューに反映させていました。

日本語でレビューさせたい場合、このようにプロンプトを囲んであげると有効です。

詳細はこちらのZenn記事をご覧ください。

<https://zenn.dev/gmopedia/articles/copilot-japanese-review-tips>

```
1 <!-- I want to review in Japanese. -->
2 ## 要件
3 xxxの改修をしました。
4 ## テスト項目
5 - [x] xxxxxx
6
7 ## レビュー希望日
8 mm/dd(曜日) までにレビューをお願いします！
9
10 ▼ ## レビューに関して
11 レビューする際には、以下のprefix(接頭辞)を付けましょう。
12 <!-- for GitHub Copilot review rule -->
13 [must] → かならず変更してね
14 [imo] → 自分の意見だとうけど修正必須ではないよ(in my opinion)
15 [nits] → さいいな指摘(nitpick)
16 [ask] → 質問
17 [fyi] → 参考情報
18 <!-- for GitHub Copilot review rule-->
19
20 ▼ ## PRのルール
21 - まずはDraftでPRを作成する。
22 - レビューに出せる状態になったらOpenにする。(レビュワーがランダムにアサインされます)
23 - レビューなしでのmainブランチへのマージは原則禁止
24 <!-- I want to review in Japanese. -->
```



Tips② PRレビューのルール（指示）を有効にする

つまり・・・

GitHub CopilotのPRのレビュー機能にレビュールールを設定する方法は2つあります。

①.github/copilot-instructions.mdに記載する方法（正攻法）

②.github/PULL_REQUEST_TEMPLATE.mdに記載する方法（裏技的）

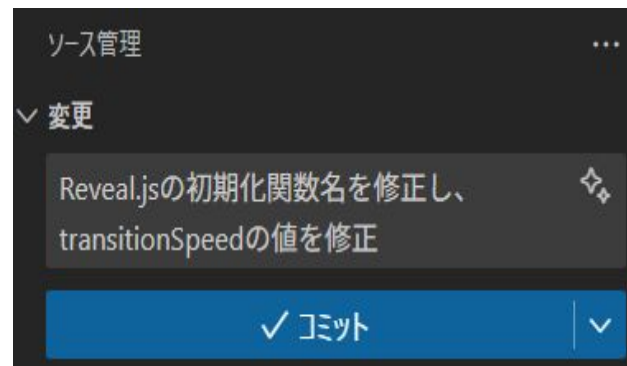
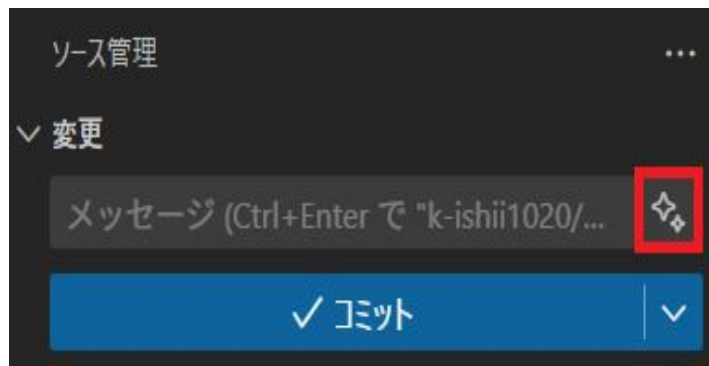
※PULL_REQUEST_TEMPLATE.mdは元々PRのテンプレート用のmdファイルです。

チームでの運用に合わせてどちらが効果的か、保守しやすいかを踏まえてぜひお試しください。



Tips③ 生成されるコミットメッセージにルールを設定

VSCodeのコミットメッセージ欄のキラキラマーク✨を押すと
自動でコミットメッセージが生成されます。





Tips③ 生成されるコミットメッセージにルールを設定

このコミットメッセージにもルールを設けることができます。（プレビュー段階です）
.vscode/settings.jsonに以下の内容を追記すればOKです。
jsonに直接指示を書いてもOK、ファイルを指定するのもOK。

```
{
  "github.copilot.chat.commitMessageGeneration.instructions": [
    {
      "text": "日本語でコミットメッセージを生成してください。",
    },
    {
      "file": ".vscode/copilot-commit-message-prompt.md" //任意のファイル名でOK
    },
  ]
}
```




その他

GitHub CopilotやVSCodeに関するカスタマイズは以下に記載されています。

GitHub Copilotに限らず、生成AI系のアップデートは目まぐるしい頻度で行われているため、定期的に公式ドキュメントを確認しておく良さそうです。

Customize AI responses in VS Code

<https://code.visualstudio.com/docs/copilot/copilot-customization>



最後までご覧いただき
ありがとうございました✨