The github.dev web-based editor

You can use the github.dev web-based editor to edit files and commit your changes.

> **Note:** The github.dev editor is currently in beta preview. You can provide feedback in our Discussions.

About the github.dev editor 🔗

The github.dev editor introduces a lightweight editing experience that runs entirely in your browser. With the github.dev editor, you can navigate files and source code repositories from GitHub, and make and commit code changes. You can open any repository, fork, or pull request in the editor.

The github.dev editor is available to everyone for free on GitHub.com.

The github.dev editor provides many of the benefits of Visual Studio Code, such as search, syntax highlighting, and a source control view. You can also use Settings Sync to share your own VS Code settings with the editor. For more information, see "Settings Sync" in the VS Code documentation.

The github.dev editor runs entirely in your browser's sandbox. The editor doesn't clone the repository, but instead uses the GitHub Repositories extension to carry out most of the functionality that you will use. Your work is saved in the browser's local storage until you commit it. You should commit your changes regularly to ensure that they're always accessible.

You must be signed in to GitHub.com to use the github.dev editor.

Opening the github.dev editor 🔗

You can open any GitHub repository in github.dev in either of the following ways:

- To open the repository in the same browser tab, press . while browsing any repository or pull request on GitHub.

  To open the repository in a new browser tab, press > .

- Change the URL from "github.com" to "github.dev".

- When viewing a file, select the ⏷ dropdown menu and click **github.dev**.

Codespaces and github.dev 🔗

Both github.dev and GitHub Codespaces allow you to edit your code straight from your repository. However, both have slightly different benefits, depending on your use case.
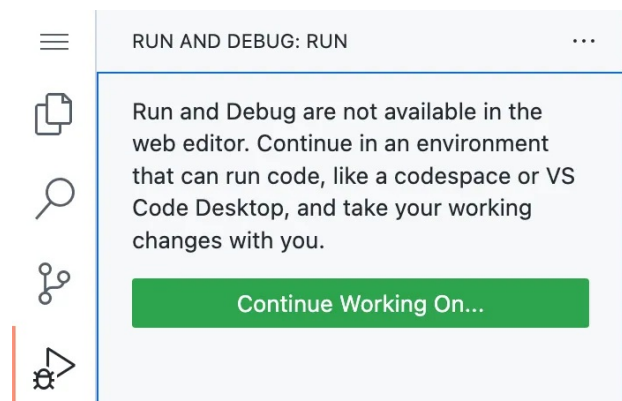
	github.dev	GitHub Codespaces
Cost	Free.	Free monthly quota of usage for personal accounts. For information on pricing, see "[About billing for GitHub Codespaces](#)."
Availability	Available to everyone on GitHub.com.	Available to everyone on GitHub.com.
Start up	github.dev opens instantly with a key-press and you can start using it right away, without having to wait for additional configuration or installation.	When you create or resume a codespace, the codespace is assigned a VM and the container is configured based on the contents of a `devcontainer.json` file. This set up may take a few minutes to create the environment. For more information, see "[Creating a codespace for a repository](#)."
Compute	There is no associated compute, so you won't be able to build and run your code or use the integrated terminal.	With GitHub Codespaces, you get the power of a dedicated VM on which you can run and debug your application.
Terminal access	None.	GitHub Codespaces provides a common set of tools by default, meaning that you can use the Terminal exactly as you would in your local environment.
Extensions	Only a subset of extensions that can run in the web will appear in the Extensions View and can be installed. For more information, see "[Using extensions](#)."	With GitHub Codespaces, you can use most extensions from the Visual Studio Code Marketplace.

Continue working on Codespaces 🔗

You can start your workflow in github.dev and continue working on a codespace. If you try to access the Run and Debug View or the Terminal, you'll be notified that they are not

available in github.dev.

To continue your work in a codespace, click **Continue Working on...** and select **Create New Codespace** to create a codespace on your current branch. Before you choose this option, you must commit any changes.
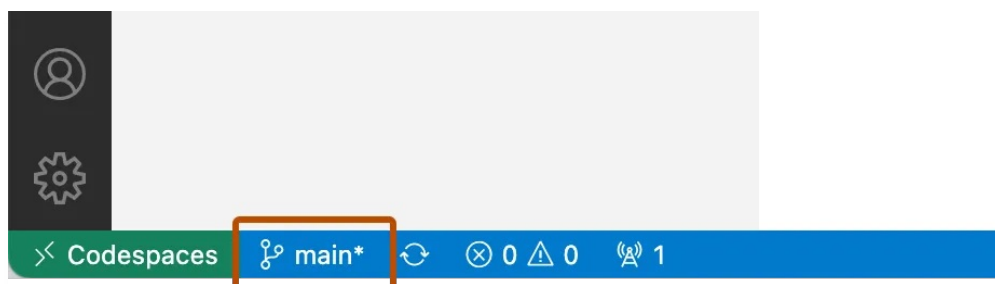

Using source control 🔗

When you use github.dev, all actions are managed through the "Source Control" view, which is located in the Activity Bar on the left hand side. For more information on the "Source Control" view, see "Version Control" in the VS Code documentation.
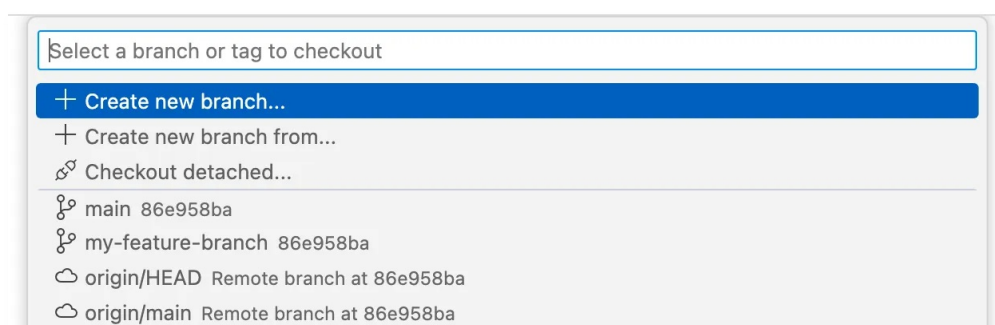
Because github.dev uses the GitHub Repositories extension to power its functionality, you can switch branches without needing to stash changes. For more information, see "GitHub Repositories" in the VS Code documentation.

Create a new branch 🔗

1. If the current branch is not shown in the status bar, at the bottom of your codespace, right-click the status bar and select **Source control**.

2. Click the branch name in the status bar.

3. In the dropdown, either click the branch you want to switch to, or enter the name for a new branch and click **Create new branch**.
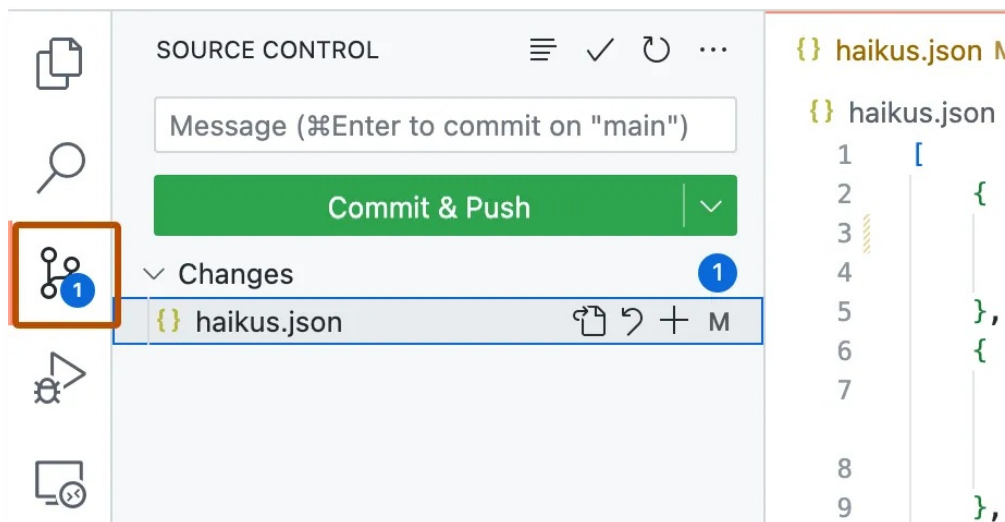

Any uncommitted changes you have made in your old branch will be available on
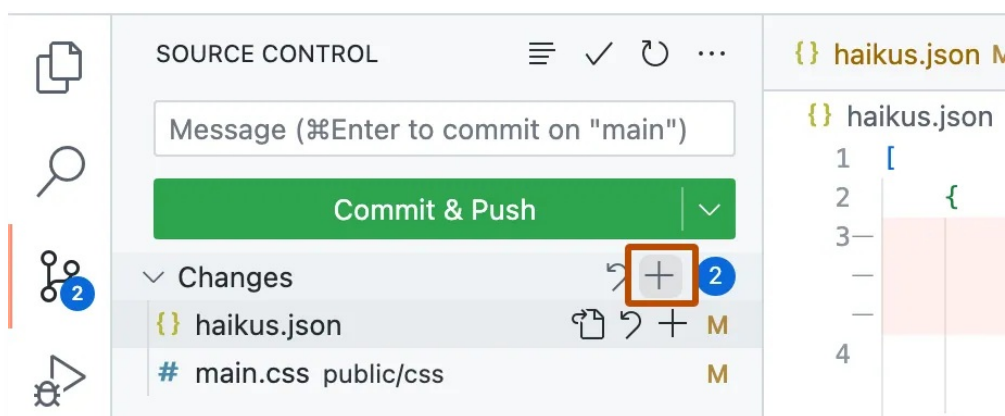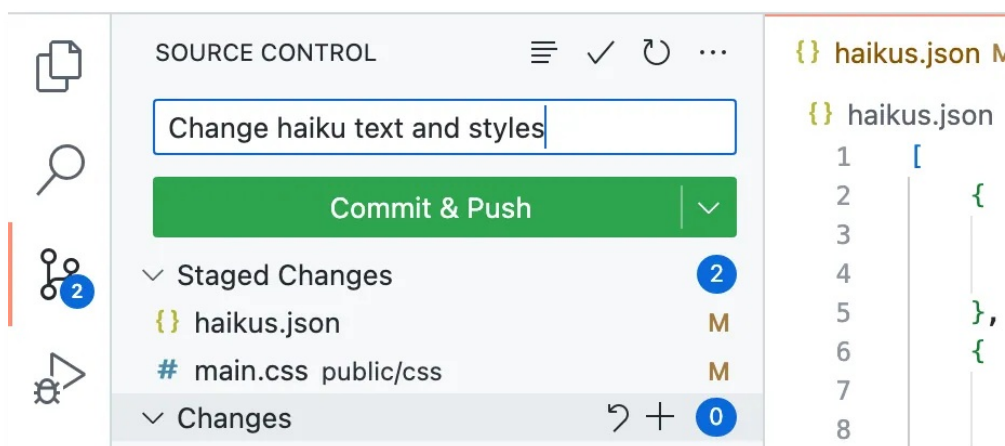
your new branch.

Commit your changes 🔗

1 In the Activity Bar, click the **Source Control** view.

2 To stage your changes, click **+** next to the file you've changed, or next to **Changes** if you've changed multiple files and you want to stage them all.

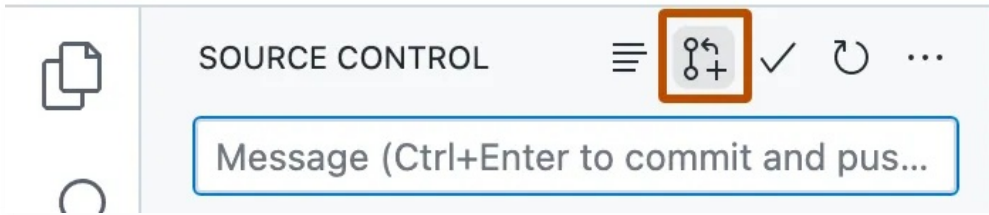**3** In the text box, type a commit message describing the change you've made.
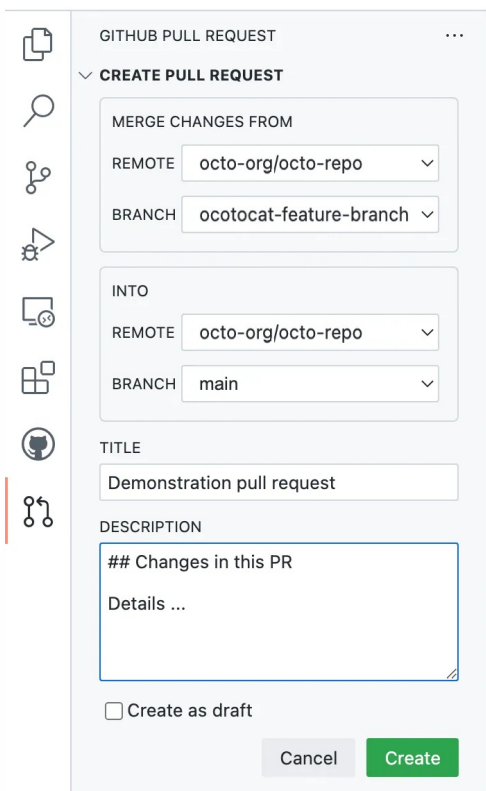

4 Click **Commit & Push**.

Your changes are automatically be pushed to your branch on GitHub.

Create a pull request 🔗

① After you've committed changes to your local copy of the repository, click the pull request icon at the top of the "Source Control" side bar.

② Check that the local branch and repository you're merging from, and the remote branch and repository you're merging into, are correct. Then give the pull request a title and a description.

③ Click **Create**.

Working with an existing pull request 🔗

You can use github.dev to work with an existing pull request.

① Browse to the pull request you'd like to open in github.dev.

② Press . to open the pull request in github.dev.

③ Once you have made any changes, commit them using the steps in Commit your changes. Your changes will be committed directly to the branch, it's not necessary to push the changes.

Using extensions 🔗

The github.dev editor supports VS Code extensions that have been specifically created or updated to run in the web. These extensions are known as "web extensions". To learn how you can create a web extension or update your existing extension to work for the web, see "[Web extensions](#)" in the VS Code documentation.

Extensions that can run in github.dev will appear in the Extensions View and can be installed. If you use Settings Sync, any compatible extensions are also installed automatically. For information, see "[Settings Sync](#)" in the VS Code documentation.

Using github.dev behind a firewall 🔗

If you are working behind a firewall you will need to add the following URLs to your firewall's allow list.

URL	Use
`https://*.vscode-cdn.net`	github.dev runs entirely in the browser. To do so it needs to download VS Code assets from endpoints at this URL.
`https://update.code.visualstudio.com`	github.dev runs entirely in the browser. To do so it needs to download VS Code assets from endpoints at this URL.
`https://api.github.com`	Used to retrieve source files from GitHub
`https://vscode-sync-insiders.trafficmanager.net`	*Optional.* To allow settings to be synchronized via Settings Sync.

Every extension installed in github.dev is run under an independent web worker. This adds a layer of security between multiple extensions running in the same browser. As a result, request URLs coming from extensions are similar to this: `https://v--151hfiju3s93ktt2rqh65902gukb27osot905m4g52k40kaea3h6.vscode-cdn.net`.

Data is retrieved from the repository at runtime using the [GitHub Repositories](#) extension. This data is not stored on the local computer between github.dev sessions, with the exception of the browser storage of unsaved files and currently displayed files (to allow for page reloads). The only non-repository data that's stored locally between sessions are some user settings and the credentials sent by GitHub's authentication flow.

Troubleshooting 🔗

If you have issues opening github.dev, try the following:

- Make sure you are signed in to GitHub.
- Disable any ad blockers.
- Use a non-incognito window in your browser to open github.dev.

Known limitations 🔗

- The github.dev editor is currently supported in Chrome (and various other Chromium-based browsers), Edge, Firefox, and Safari. We recommend that you use the latest versions of these browsers.
- Some keybindings may not work, depending on the browser you are using. These keybinding limitations are documented in the "[Known limitations and adaptations](#)"

section of the VS Code documentation.

- `.` may not work to open github.dev according to your local keyboard layout. In that case, you can open any GitHub repository in github.dev by changing the URL from `github.com` to `github.dev` .

Legal