

New to GitHub? Quick Start Guide

For someone who has never used GitHub. It takes you from no account to a first CodeDelta report on your own code in about twenty minutes. Nothing is downloaded or installed on your computer.

The idea in one minute. GitHub is a website that stores your code and remembers every version of it, like a filing cabinet that never throws anything away. CodeDelta is a tool that compares two versions of your code and tells you how much really changed, and whether any of it quietly calls AI services. On GitHub, CodeDelta runs on GitHub's computers, not yours. So the plan is: **put your code on GitHub, make one small change, and ask GitHub to show the before and after.** CodeDelta then writes its report on that before-and-after page.

What you need

An email address, a web browser, and your code in a folder on your computer. No licence file is needed: the GitHub version of CodeDelta is free to use until **31 December 2026**.

What it costs

Nothing. A free GitHub account lets you keep your code private and gives you 2,000 minutes a month of GitHub's computer time. The CodeDelta run on our own demo took 23 seconds.

GitHub has its own vocabulary, and most of it sounds stranger than it is. Each new word is explained where you first meet it, and the **Glossary** at the back explains every one again in everyday language.

1 Create your GitHub account

Go to github.com/signup and follow the prompts: email address, password, and a username of your choice. GitHub then asks you to **verify your email address**, to prove the address is yours. Do not skip this: GitHub will not let you store anything until it is done. If it offers paid plans, the free one is all you need.

2 Create a private home for your project

GitHub calls a project's home a **repository**: a folder on GitHub that holds your files and remembers every saved version. **Private** means only you can see it.

1. At the top right of any GitHub page, press the + icon and choose **New repository**.
2. Type a **Repository name**, for example `my-project`. Use letters, digits and hyphens, no spaces.
3. Where it asks who can see it, choose **Private**.
4. Turn on the option to **add a README**. A README is just a front-page note for the project. Adding one now means the repository is not empty, which the next steps need.
5. Press **Create repository**.

3 Put your code in it

1. On your repository's page press **Add file**, then **Upload files**.
2. Drag your project folder from your computer onto the page, or press **choose your files**.
3. At the bottom is a box labelled **Commit message**. A **commit** is GitHub's word for a save, and the message is a short note to yourself about what you saved. Type `First upload`, leave the other settings alone, and press the green button.

The browser takes up to **100 files at a time**, each up to **25 MB**. For a bigger project, repeat the upload one folder at a time.

Before uploading, take anything secret out of the folder: passwords, keys to online services, customer data. Your repository is private, but secrets should not be stored in any repository.

4 Switch CodeDelta on

You switch CodeDelta on by adding one small text file to your repository. It is an instruction sheet for GitHub, which GitHub calls a **workflow**. It says: “whenever a before-and-after page is opened in this project, run CodeDelta and write its report there.” You do not need to understand the text, only to copy it exactly.

1. On your repository's page press **Add file**, then **Create new file**.
2. In the file name box type exactly `.github/workflows/codedelta.yml` and note that it begins with a dot. Each time you type `/` GitHub turns the word before it into a folder. That is meant to happen.
3. Paste the text below into the large box. **The spaces at the start of each line matter**, and copying from a PDF can lose them. The safest way is to open codedelta.app/github-quickstart.html and copy the same text from there, under “Create the workflow file”.
4. Press **Commit changes...**, then **Commit changes** again in the small window that opens. That saves the file.

```
name: CodeDelta
on: [pull_request]

permissions:
  contents: read
  pull-requests: write      # post the summary comment
  security-events: write    # SARIF to the Security tab (public repo or Advanced Security)

jobs:
  codedelta:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with: { fetch-depth: 0 }
      - uses: code-delta-app/action@v1
```

Nothing happens yet. CodeDelta is waiting for a before and an after, which is the next step.

5 Make a change and ask for the before-and-after

Any small edit will do for a first try. The important part is *how* you save it. Until now each save went straight into the official copy of your project, which GitHub calls `main`. This time you save the change as a **draft** instead, so that GitHub has two versions to compare: the official copy (before) and your draft (after). GitHub calls a draft a **branch**, and it calls the before-and-after page a **pull request**. The name is odd; it simply means “please pull my draft into the official copy”. On your own project nobody else is involved: you ask, and you decide.

1. Click one of your code files to open it, then press the pencil icon, **Edit file**.
2. Change something: add a line, alter a line, delete a few lines.
3. Press **Commit changes....** In the small window, this time choose the option to **create a new branch** and start a pull request. Then press **Propose changes**.
4. On the next page press **Create pull request**. You are now looking at the before-and-after page.

6 Read the report

Stay on that page. Near the bottom, a box shows CodeDelta running and then finished. On our demo that took 23 seconds; a bigger project takes longer. Then a note headed **CodeDelta report** appears on the page, written by CodeDelta. GitHub calls such a note a **comment**. This is the start of the real one on our public demo:

CodeDelta report
Churn: 47 logical statements (+25 / -14 / Δ8); REP_CHURN 0.83
Agent scan: 64 CRITICAL · 7 HIGH · 81 ELEVATED · 17 endpoint-level · 57 text-level

Churn: 47 logical statements	Churn is how much code changed between the before and the after. It is counted in statements , the single instructions a program is made of, so reformatting or moving blank lines does not inflate it. Here 47 statements were touched.
+25 / -14 / Δ8	Of those 47: 25 were newly added , 14 were deleted , and 8 were existing statements edited in place .
REP_CHURN 0.83	The share of the change that was adding and deleting, as opposed to editing: $(25 + 14) \div 47 = 0.83$. Near 1 means code was thrown away and rewritten. Near 0 means existing code was carefully repaired.
Agent scan	A search of all your files for code that talks to AI services or starts AI “agents” (AI programs that act on their own), graded by how risky it looks. If your code uses no AI services, there may be nothing to report.

Our demo's figures are large on purpose: it is packed with harmless fake AI files so that the scan has something to find. You can see the whole page at github.com/code-delta-app/demo/pull/4.

Finishing off. If you want to keep your change, press **Merge pull request**. **Merge** means accept: the draft becomes part of the official copy. From now on, every time you save a change as a draft and open a before-and-after page, CodeDelta reports on it by itself. There is nothing more to set up.

If something does not look right

No CodeDelta report appears	Press the Actions tab at the top of your repository. It lists everything GitHub has run for you. Press the top entry: the same report is on that page, along with a step-by-step record of what happened.
The Actions tab is empty	CodeDelta never started. Check that the file from step 4 is named exactly <code>.github/workflows/codedelta.yml</code> , and that you did step 5 <i>after</i> step 4.
A red cross and the words “workflow is not valid”	Some spaces at the start of a line were lost when you pasted the text in step 4. Open the file, press the pencil icon, and paste the text again from codedelta.app/github-quickstart.html .
A warning that mentions “SARIF”	Safe to ignore. CodeDelta also tries to fill in GitHub's separate Security page, which GitHub only offers for public projects or paid plans. Your report is not affected.

Where to go next

More on the GitHub version	Once GitHub feels familiar, the companion guide Quick Start Guide for GitHub users covers the options: blocking a change, your own licence, running every night. Also codedelta.app/github-quickstart.html and the questions and answers at codedelta.app/faq.html#github .
The desktop app	The same CodeDelta as a program on your own computer, with full reports and a side-by-side view of what changed: codedelta.app/download.html . It has its own Quick Start Guide.
What the figures mean	codedelta.app/user-guide.html explains every figure with a worked example.
Questions	Use the chat on codedelta.app , or the contact button on the download page. A person replies.

Glossary: GitHub's words in everyday language

Account	Your personal login to GitHub. Free.
Repository ("repo")	Your project's home on GitHub: a folder that holds your files and remembers every version you have ever saved, so nothing is lost and you can always look back.
Private / Public	Who can see a repository. Private: only you, and anyone you choose to invite. Public: anyone on the internet.
README	A front-page note describing the project. Optional to write, but it stops a new repository being empty.
Commit	A save. Each commit is kept for ever as a version of your project, with the date and your note.
Commit message	The short note you attach to a save, such as "First upload" or "Fixed the date bug". It is for you, later.
main	GitHub's name for the official, up-to-date copy of your project.
Branch	A draft copy of your project. You change the draft; the official copy (<code>main</code>) is untouched until you accept the draft.
Pull request ("PR")	The before-and-after page. It puts your draft beside the official copy, shows every line that differs, and waits for you to accept or discard the change. The name means "please pull my draft into the official copy". CodeDelta writes its report on this page.
Propose changes	The button that saves your edit as a draft and takes you towards the before-and-after page.
Merge	Accept. The draft's changes become part of the official copy.
Comment	A note written on a pull request page. CodeDelta's report arrives as a comment.
Workflow	An instruction sheet you give GitHub: "when this happens, run that." Ours says: when a pull request is opened, run CodeDelta. It lives in the file you made in step 4.
.yml file ("YAML")	The kind of text file a workflow is written in. It is fussy about the spaces at the start of each line, which is why you copy it exactly.
GitHub Actions	GitHub's robot helper: computers at GitHub that carry out your workflows for you. The Actions tab in your repository lists everything it has run. The free account includes 2,000 minutes of it a month.
Run / check	One occasion of GitHub Actions carrying out a workflow. A tick means it finished; a cross means something stopped it.
Churn	CodeDelta's word for how much code changed between two versions.
Statement	One instruction in a program, such as <code>total = price * quantity;</code> . CodeDelta counts statements, not lines, so layout changes do not count as work.
REP_CHURN	"Replacement churn": the share of a change that was adding and deleting code, as opposed to editing what was there. From 0 (all careful edits) to 1 (all replaced).
Agent scan	CodeDelta's search for code that calls AI services or starts AI programs that act on their own. CRITICAL, HIGH and ELEVATED are its grades of concern, highest first.
SARIF	A file format GitHub uses for its Security page. You can ignore it; it only matters for public projects and paid plans.
Evaluation licence	CodeDelta's free trial permission, built into the GitHub version. Valid to 31 December 2026.