



GitHub Pages: Free Hosting Straight From a Repo

Turn a repo of plain files into a live website at no cost

Here is something most people do not realize GitHub gives away: free web hosting. If your site is made of plain files (HTML, CSS, images, a static site generator's output), GitHub will serve it to the public at a real URL, at no cost, straight from a repo. It is called GitHub Pages.

What it is good for

Pages serves *static* sites: files that are the same for every visitor. That covers more than it sounds like:

- A firm one-pager or landing page
- A personal or professional bio site
- Documentation for a tool you built
- A simple blog from a static generator (Jekyll, Hugo, Eleventy)

What it does **not** do is run server-side code. No PHP, no database, no login system, no form that emails you by itself. If you need those, Pages is the wrong tool; that is what a VPS or a proper app host is for. For a brochure-style site, though, free is hard to beat.

Turning it on

The fast path, from a repo that has an `index.html` at its root:

1. Push your site files to the repo (see Clone, Pull, Push).
2. Go to **Settings**, then **Pages** in the sidebar.
3. Under "Build and deployment," set Source to "Deploy from a branch," pick `main` and the `/root` folder, and Save.
4. Wait a minute. GitHub shows you the live URL: `https://yourname.github.io/your-repo/`.

That is it. Your files are on the public web. Every time you push a change, the live site updates on its own within a minute or two.

Your own domain

The default address is `yourname.github.io/your-repo`. You can point a real domain (like `guide.yourfirm.com`) at it for free: add the domain under Settings, then Pages, then create a

CNAME record at your registrar pointing to GitHub. The site keeps updating from the repo; visitors just see your name on it.

The catch a lawyer should hear

A GitHub Pages site is public, and its source repo is usually public too. That is fine for a brochure site, where public is the point. But it means the same rule from the `.gitignore` tip applies with force: nothing client-related, no draft you would not want indexed by Google, no internal note. Everything in that repo is readable, and the site itself is crawlable by the entire internet by design.

Also know that Pages is meant for reasonable personal and project use, not high-traffic commercial hosting; GitHub publishes soft bandwidth and build limits. For a firm landing page or a tool's docs, you will never come close.

The playbook

Static files only, no server code. Push to a repo, flip on Pages in Settings, get a free public URL that redeploys on every push. Point a custom domain at it if you want. Keep the repo clean, because public means public. For a simple site you would otherwise pay to host, it is the easiest free win in this whole toolbox.