STEP-BY-STEP GUIDE

Build a Fast Website with Claude Code

No code. No monthly hosting bill.

Ask an AI to build you a website, then host it free on Cloudflare. Scores 100 out of 100 on Google's speed test, and you never open a code editor.

Beginner · 60 to 90 min · Claude Code · Free hosting, paid AI plan

@learnwithseb

One system at a time. Built, not theorised.

Contents

01 The Game Plan
02 What You Need
03 The Jargon, Translated
04 Create Your Repo
05 Give Claude Access to GitHub
06 Connect Claude Code to GitHub
07 Open a Free Cloudflare Account
08 Ask Claude to Build the Site
09 Deploy It on Cloudflare
10 Check It Was Built Properly
11 Teach Claude Your Voice
12 Publish Your First Post by Asking
13 Putting Your Own Domain On It
14 When Something Goes Wrong
15 Your Prompt Library
16 Validation
17 Keep Building
18 Quick Reference

The Game Plan

You are going to end up with a live website that costs nothing to host and loads almost instantly. It scores 100 out of 100 on Google's own speed test, on mobile and on desktop.

You will not write a line of code. You describe what you want to Claude Code in plain English, and it builds the files for you.

How the three pieces fit together

Three services do one job each. Once you can see the shape of it, every step below makes sense rather than feeling like a list of buttons to press.

Piece	Its one job	Think of it as
GitHub	Stores your website's files	A folder in the cloud
Claude Code	Writes and edits those files when you ask	The person who builds the site
Cloudflare	Serves the files to visitors, fast and free	The shop window on the street

The site itself is built with Astro, a website framework known for being very fast. You do not need to learn Astro. Claude handles it, and you will likely never open one of its files.

Who this is for

- Non-technical people who want a proper website without a monthly bill
- Anyone paying for hosting they barely use
- Creators who want a blog they update by asking rather than logging in
- People who tried WordPress, found it heavy, and gave up

What you will not be doing

- Opening a code editor
- Learning HTML, CSS or JavaScript
- Paying a monthly hosting fee
- Installing anything beyond the Claude desktop app

> **NOTE**
>
> Based on the walkthrough in Learn With Seb's video "How to build a fast website with Claude Code (No hosting fees)". Watch the original at youtu.be/5zqEiteDrrc. This guide rebuilds it as a step-by-step reference you can follow without pausing the video, and adds the jargon, troubleshooting and prompt sections that a video cannot pause for.

What You Need

Hosting genuinely costs nothing at this size. The one real cost is a paid Claude plan, because Claude Code is not available on the free tier.

What	Why you need it	Cost
GitHub account	Stores the website files	Free

What	Why you need it	Cost
Claude GitHub app	Lets Claude read and write that repo	Free
Claude paid plan	Claude Code needs one	From about £18/month
Claude desktop app	Where you run Claude Code	Included in the plan
Cloudflare account	Free hosting on Workers	Free
A domain name	Optional, only for a custom address	About £10/year

BE CLEAR ON THE COST

"No hosting fees" is accurate. GitHub and Cloudflare are both free at this size and will stay free for a normal blog. But Claude Code sits behind a paid plan, so the setup is not free overall. If you already pay for Claude, you are adding nothing.

Before you start

Set aside an unbroken hour for the first run. Most of the time goes on account setup and authorising things, which is fiddly rather than hard. Every session after this one is just typing a request and watching it publish.

TIP

Turn on two-factor authentication on both GitHub and Cloudflare while you are creating the accounts. These two now hold your website between them. It takes a minute each and you will never get round to it later.

The Jargon, Translated

You will meet these words in the next few steps. None of them are as complicated as they sound. Skim this now and come back when one trips you up.

Word	What it actually means
Repository, or repo	A named folder that stores your files online
Commit	Saving a change, with a note about what changed

Word	What it actually means
Push	Sending your saved changes up to GitHub
Branch	A version of your files. Yours is called main, and one is plenty
Framework	A starter kit that saves building a site from nothing
Astro	The framework used here, chosen because it is very fast
Workers	Cloudflare's free hosting product
Deploy	Putting your site online, or updating what is online
Build	Turning your files into the finished pages visitors see
Subdomain	The first part of a web address, the bit before the main domain name
DNS	The address book of the internet. It points a domain at a host
CMS	A content management system, meaning a visual editor for writing posts

TIP

You do not need to memorise any of this. If Claude uses a word you do not know, ask it: "explain that in plain English, I am not technical". It will.

STEP 01

Create Your Repo

A repository, or repo, is just a named place where files are stored. In this case it will hold your website.

Sign up at **github.com** if you do not have an account, and turn on two-factor authentication. Then click **Create repository**.

What to put in each field

Field	What to do
Repository name	Something plain and lowercase, such as my-blog
Description	One line for your own benefit later
Public or private	Either works. You can change it afterwards
Everything else	Leave exactly as it is

Click **Create repository**. It will look empty, which is correct. Claude fills it in step 5.

PUBLIC OR PRIVATE?

Public means anyone can read your site's files, not that your site is more visible in search. Private keeps them to you. For a normal blog neither choice causes a problem, so do not lose time on it.

STEP 02

Give Claude Access to GitHub

Claude needs permission to put files into that repo.

Go to **github.com/apps/claude** and install the official Claude app. You can grant it access to all your repos or just one. Picking only the repo you made in step 1 is the tidier choice, and you can widen it later.

Click **Install**, then authorise. A passkey works here if you have one set up. You should see

"GitHub app installed".

The app will not appear inside the Claude desktop app yet. That is expected and nothing has gone wrong. You can confirm it installed by visiting **github.com/settings/installations**, which is also where you go to change its permissions or remove it later.

DO THIS BEFORE YOU START BUILDING

If you skip the app install and go straight to asking Claude to build the site, Claude gets partway through and stops to ask you to install it. Doing it now keeps the build in one clean run instead of two interrupted ones.

STEP 03

Connect Claude Code to GitHub

Installing the app gave Claude permission. This step tells the Claude desktop app which repo to work in.

Open the desktop app, click the **Claude Code** tab, then **New**.

Switch from your computer to the cloud

Above the chat box there is a folder selector, and by default it points at your own computer. Click **Local** and switch it to the cloud, then choose **Add cloud environment**. Name it something you will recognise later, such as "GitHub", and click **Create environment**.

Working in the cloud rather than on your own machine means the files live in one place, and nothing depends on your laptop being switched on.

Connect the account

A **Select repo** button appears next to your new environment. Click it and Claude asks you to connect to GitHub. Find the GitHub integration in the connectors list, click **Connect**, and authorise it. Claude reopens itself when that finishes.

THE BIT THAT LOOKS BROKEN

Straight after authorising, click **Select repo** and your repository will not be in the list. Nothing has gone wrong. Quit the desktop app completely and reopen it. The repo now appears, with its main branch, and Claude has access. Almost everyone hits this and assumes the connection failed.

STEP 04

Open a Free Cloudflare Account

Cloudflare is the free host that will put your site on the web.

Go to **cloudflare.com**, click **Products**, then **Workers**, then **Start building for free**. Create an account if you do not have one, and turn on two-factor authentication once you are in.

That is all you need here for now. You will come back to Cloudflare in step 6 to connect it to your repo.

WHY WORKERS AND NOT THE FREE PLAN ELSEWHERE

Cloudflare Workers serves your site from locations all over the world, so it loads quickly wherever the visitor is. That is most of where the speed score comes from. The free tier is generous enough that a normal blog will not approach its limits.

STEP 05

Ask Claude to Build the Site

All three accounts are now connected, so you can describe the site and let Claude build it.

Go back to Claude Code with your repo selected and paste this in:

PROMPT: COPY & PASTE INTO CLAUDE CODE

Create an Astro site set up to deploy on Cloudflare Workers, and commit it to this repository. Walk me through it step by step, and explain what each part does in plain English before you create it. I am not technical. Keep it simple to start: a homepage, a blog section and an about page.

Claude creates the files and pushes them to your repo. Asking it to walk you through is worth doing even if you do not follow every detail. You will understand your own site far better the first time you want to change something.

If you already like the look of another site

You do not have to accept whatever design Claude invents. Point it at a reference:

PROMPT: COPY & PASTE INTO CLAUDE CODE

Use my existing site at [paste the address] as a reference for the layout, typography and colours. Match the feel of it rather than copying it exactly.

TIP

Resist adding features on this first pass. Get a plain three-page site live, see it working, then ask for changes. A build that does one thing is far easier to fix than one that tried to do nine.

STEP 06

Deploy It on Cloudflare

The files exist in GitHub. This step points Cloudflare at them and puts the site online.

In Cloudflare, click **Build**, then **Workers and Pages**, then **Create application**, then **Connect GitHub**. Pick your repo from the drop-down, then click install and authorise. Click **Continue with GitHub**, check the right repo is selected, and click **Next**.

Leave every build setting exactly as it is. Click **Deploy**.

It takes a few seconds. When it finishes, Cloudflare shows you the URL. Paste it into your browser. Your site is live on the internet, with no hosting fee.

What you should see

A basic blog: a homepage, a blog section and an about page. It is deliberately plain at this point. Everything about how it looks is changeable later by asking.

ABOUT THAT ADDRESS

Cloudflare gives you a free address ending in .workers.dev. You can rename that subdomain at any time to something more readable, and point your own domain at it later. See "Putting Your Own Domain On It" below.

IT UPDATES ITSELF FROM NOW ON

Cloudflare watches your repo. Every time Claude pushes a change, the site rebuilds and republishes on its own. You will not repeat this step.

STEP 07

Check It Was Built Properly

Before you start adding content, confirm the foundations are sound. Two free tools tell you.

Paste your URL into Google PageSpeed Insights and then into GTmetrix. A text-only Astro site on Cloudflare should score at or very near 100 on both, on mobile and desktop.

This is worth doing now precisely because the site is empty. It gives you a clean baseline. When the score drops later, you will know it was something you added rather than something built wrong at the start.

WHAT DRAGS THE SCORE DOWN

Almost always images. A photo straight off a phone can be several megabytes. Ask Claude to resize and compress images before adding them, and re-run the speed test after any post with pictures or video in it.

STEP 08

Teach Claude Your Voice

Do this before you publish anything, because it changes everything that follows.

Without guidance, an AI writes in generic blog English: tidy, forgettable, slightly American. A short tone of voice guide kept in the project fixes that. It makes more practical difference than any clever prompt wording.

What to put in it

- Who you are writing for, in one sentence
- How long your sentences and paragraphs run
- Words and phrases you never use
- Whether you write in British or American English
- Two or three paragraphs of your own writing as a sample
- What you want the reader to feel or do at the end of a post

Ask Claude to set it up and keep it where it will be read:

PROMPT: COPY & PASTE INTO CLAUDE CODE

Create a tone of voice guide in this project and refer to it every time you write a post for my site. Here is how I write: [paste two or three paragraphs of your own writing]. Read it back to me as a set of rules, so I can check you understood before we use it.

TEST IT PROPERLY

Ask Claude to write the same post twice, once ignoring the guide and once using it, then read both. If you cannot tell them apart, your guide is too vague. Add specifics until you can.

STEP 09

Publish Your First Post by Asking

This is the part that makes the whole setup worth having. You do not log into anything to publish. You ask.

You can paste in text you have already written, or have Claude draft it from a brief:

PROMPT: COPY & PASTE INTO CLAUDE CODE

Add a new blog post to my site announcing that new videos are coming soon on my YouTube channel. Follow my tone of voice guide. Give it a clear title, keep it to about 200 words, and embed my latest YouTube video at the bottom. Then commit it and push it live.

Claude writes the post, adds it to the blog section, and pushes it. Cloudflare rebuilds automatically. Refresh your site and it is there, with the title, the text and the embedded video.

Changes work the same way. "Change the title to X." "Add a contact link to the header." "Make the About page shorter." You describe it, Claude does it.

READ BEFORE YOU PUBLISH

The AI writes the draft, not the final word. Read every post before it goes live, especially anything containing a claim, a statistic or a promise. It is your name on the site, and a confident sentence is not the same as a true one.

Putting Your Own Domain On It

The free .workers.dev address is fine while you are building. At some point you will want your own.

There are two stages, and they are worth separating in your head.

Stage one: rename the free subdomain

Cloudflare lets you change the first part of the free address at any time, so you can make it readable rather than whatever was auto-generated. This takes seconds and costs nothing. Do it now if the current address is ugly.

Stage two: point a domain you own at Cloudflare

Buy a domain from any registrar, then change its DNS settings to point at Cloudflare. DNS is the internet's address book: it is what tells a browser where your site actually lives. Once it points at Cloudflare, your site answers on your own address.

Changes can take a few hours to spread across the internet, so do not panic if it does not work instantly.

BEYOND THE VIDEO

The original video stops before the custom domain and points at the Cloudflare documentation, which is the honest answer: the exact screens depend on where you bought the domain. Cloudflare's own guide at developers.cloudflare.com is current and clear. Ask Claude to walk you through it with your registrar named, and it will follow along.

When Something Goes Wrong

Almost every problem on a first run is one of these five. None of them mean you have broken anything.

What you see	What it usually is	What to do
Your repo is not in the Select repo list	The desktop app has not picked up the new connection	Quit the app completely and reopen it

What you see	What it usually is	What to do
Claude stops and asks you to install the GitHub app	Step 2 was skipped or the app has no access to this repo	Install at github.com/apps/claude and grant it this repo
Cloudflare build fails	A setting was changed on the deploy screen	Start the deploy again and change nothing at all
Site is live but a change is missing	The change was not committed and pushed	Ask Claude: "did you commit and push that? Please do it now"
Speed score has dropped	An image or video you added	Ask Claude to compress the images, then test again

When you are genuinely stuck

Describe the problem to Claude exactly as you see it, including the error text, and say you are not technical. It has the context of your whole project, which no generic search result does.

PROMPT: COPY & PASTE INTO CLAUDE CODE

Something is not working and I am not technical. Here is exactly what I did, and here is exactly what I see: [describe it, and paste any error text]. Explain what has gone wrong in plain English, then tell me the smallest thing I can do to fix it.

DO NOT DELETE AND START AGAIN

The instinct when something looks broken is to delete the repo and redo everything. Resist it. Your files are safe in GitHub, nothing you have done is unrecoverable, and starting over usually reproduces the same problem twenty minutes later.

Your Prompt Library

These cover most of what you will ask for in the first few months. Adapt the wording and keep the ones that work.

Publishing from something you already have

PROMPT: COPY & PASTE INTO CLAUDE CODE

Here is the transcript of my latest video. Turn it into a blog post that follows my tone of voice guide. Keep the useful detail and cut the repetition and filler. Add a clear title, subheadings, and a short introduction that says what the reader will get. Then commit it and push it live.

Changing how the site looks

PROMPT: COPY & PASTE INTO CLAUDE CODE

Redesign the homepage to be cleaner: more white space, larger body text, and the three most recent posts listed with their dates. Show me what you plan to change before you change it, and keep the speed score as close to 100 as possible.

Editing something already published

PROMPT: COPY & PASTE INTO CLAUDE CODE

In my post titled [title], change the headline to [new headline], tighten the opening paragraph, and fix any American spellings. Show me the changes before you push them.

Keeping the site fast

PROMPT: COPY & PASTE INTO CLAUDE CODE

Check every image on the site, compress any that are larger than they need to be, and make sure each one has descriptive alt text for screen readers and search engines. Tell me what you changed.

Understanding your own site

PROMPT: COPY & PASTE INTO CLAUDE CODE

Explain how my site is put together, in plain English, as if I have never seen a website's files before. Tell me which file I would change to edit the About page, and which one controls the site's colours.

Validation

Run through this once the site is live. If everything here passes, your setup is sound and you can start writing.

- ✓ The Cloudflare URL opens and shows your homepage
- ✓ The blog and about pages both load
- ✓ PageSpeed Insights scores near 100 on mobile and desktop
- ✓ GTmetrix agrees
- ✓ A post you asked Claude to add appears on the live site
- ✓ Your repo shows the files Claude created
- ✓ Two-factor authentication is on for GitHub and Cloudflare
- ✓ A tone of voice guide exists in the project
- × A published post you never actually read through
- × Large unoptimised images pushed straight to the live site
- × Build settings you changed on the Cloudflare deploy screen

Keep Building

The site works. Here is where it goes next, roughly in the order most people want it.

- Point your own domain at Cloudflare, so the address is yours rather than a .workers.dev one
- Rename the free subdomain in the meantime so it reads properly
- Write the tone of voice guide properly if you rushed it, then have Claude rewrite an old post with it and compare the two
- Look at a CMS such as Sanity if you want a WordPress-style visual editor for writing posts by hand
- Add an email signup so visitors can subscribe, rather than relying on people remembering to come back
- Turn each video or long post into two or three shorter pieces, since the publishing cost is now almost nothing
- Ask Claude to redesign a page whenever you tire of it, because changes are cheap now and that is the whole point

PUBLISH BEFORE YOU POLISH

Keep the site text-only until you have published a handful of posts. It is the fastest it will ever be, and you will discover what you actually want from it before spending an afternoon on design.

Quick Reference

The whole setup on one page. Use this on your second site, when you no longer need the explanations.

Stage	Where	What you do
Create repo	github.com	Create repository, name it, turn on 2FA
Give Claude access	github.com/apps/claude	Install the app, grant it the repo, authorise
Check the install	github.com/settings/installations	Confirm it is listed
Start Claude Code	Claude desktop app	Claude Code tab, then New
Switch to cloud	Claude desktop app	Local, then Add cloud environment, then Create
Connect GitHub	Claude desktop app	Select repo, Connect, authorise
Fix the empty list	Claude desktop app	Quit and reopen the app, then Select repo
Open host account	cloudflare.com	Products, Workers, Start building for free, 2FA
Build the site	Claude Code	Prompt for an Astro site on Cloudflare Workers
Deploy	Cloudflare	Build, Workers and Pages, Create application, Connect GitHub, Deploy
Test speed	PageSpeed Insights and GTmetrix	Paste the URL, expect near 100
Set your voice	Claude Code	Create a tone of voice guide in the project
Publish	Claude Code	Describe the post, Claude commits and pushes
Custom domain	Cloudflare	Point the domain's DNS at Cloudflare

BuildWithSeb | A Fast Website with Claude Code

Built by @learnwithseb · One system at a time. Built, not theorised.

© 2026 Learnwithseb. All rights reserved.