GitHub Basics

You will never type a GitHub command: Claude does that for you. This course explains what is actually happening when Claude “saves to GitHub”, and why it quietly protects all your projects. The exercises are pretend, so click anything you like.

Beginner · 10 lessons · 3 hands-on exercises · 7 practice goals

Start lesson 1 → Open the word list



my-project Private

- Add a contact form — today
- Make the buttons green — yesterday

- Add opening hours — Monday
- First version — last week

This course includes

- Plain-language explanations
- A quiz on what to ask Claude
- Answers to common questions
- 3 click-through exercises
- A word list for GitHub terms
- Progress saved in your browser

Course content

10 lessons · 4 with hands-on practice

0 of 7 practice goals done

Reset my progress

What you'll learn

- ✓ What GitHub is, and how it differs from the files on your computer
- ✓ Why work only counts as backed up once it's uploaded
- ✓ Exactly what to say to Claude to save, compare or undo
- ✓ What a “save point” is, and how it lets you undo anything
- ✓ How to try a risky idea without putting your project at risk
- ✓ How to keep your projects private and safe

How each lesson works

1. Read

Each lesson starts with a short, plain explanation. No commands to learn.

2. Try

Some lessons have a small pretend exercise. Click the buttons and see what happens.

3. Use it

Lesson 07 turns it all into simple sentences you can say to Claude.

Lesson 01 Reading

GitHub in one minute

You may already know the “version history” in Google Docs or Word, where you can see older versions of a document and bring one back. GitHub is the same idea, but for **a whole project folder**: every file inside it, all at once.

Two names come up, and they are two halves of one system:



Git takes snapshots on your computer; GitHub keeps a safe copy of them online.

Git: the tool on your computer

A free program that takes **snapshots** of your project folder whenever you ask, and remembers every one of them. It works without the internet.

GitHub: the website

A website that keeps **a copy** of your project and all its snapshots online, safe from anything that happens to your computer. Only you can see your private ones.

> **The whole course in three sentences**
>
> Claude takes a snapshot of your project (a **save point**) whenever you're happy with it. Claude then **uploads** the save points to GitHub. From then on, nothing in that version can be lost, and you can always go back to it.

Lesson 02 Hands-on practice

Save points: snapshots you can go back to

Think of a video game. You save before a hard level, so if things go wrong you can restart from there instead of from the beginning. A **save point** works the same way for your project. The official name is a **commit**, and you'll see that word a lot.

- Each save point is a snapshot of **every file** in the project at that moment.
- Each one gets a short note saying what changed, like “Add opening hours”. Claude writes these notes for you.
- Save points pile up into a **history**: a timeline of your project, newest at the top.
- Changes you haven't saved yet are not protected. If something goes wrong, they can't be brought back.

Goal not done yet

Make the three changes one by one, and make a save point after each one. The third change goes wrong. When it does, **go back** to a save point that still has the menu.

The bakery website

All saved

Bakery name and logo

Menu: bread, cakes, coffee

Save points (history)

First version
save point 1 · you are here

Go back to this

This is a tiny website with one save point so far: “First version”. Press **Make the next change** to let “Claude” edit it.

Make the next change | Make a save point | Start over

Going back is never risky

Going back to an older save point doesn't erase the newer ones. They stay in the history, so if you change your mind you can jump forward again.

Lesson 03 Hands-on practice

Two copies: yours and GitHub's

There are always **two copies** of each project: one in a folder on your computer, and one on GitHub. They don't sync by themselves the way Dropbox or OneDrive do. They only match when Claude copies save points from one to the other:

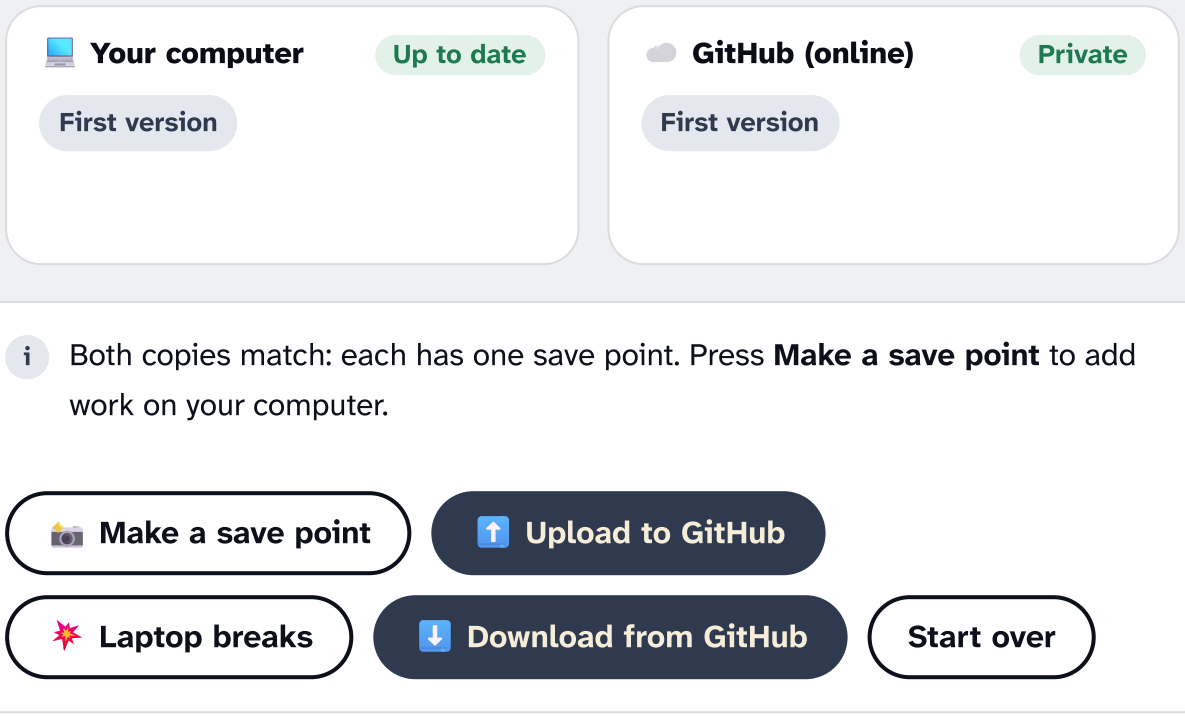
- **Upload** (official name: **push**): send your new save points from your computer to GitHub.
- **Download** (official name: **pull**, or **clone** the very first time): bring the save points from GitHub to a computer.

A save point that is only on your computer is still only as safe as your computer. It's properly backed up once it has been uploaded.

Goal not done yet

Make **two** new save points and upload them. Then press **Laptop breaks**, and get all your work back on a new laptop.

Curious? Start over and break the laptop *before* uploading, and see what happens.



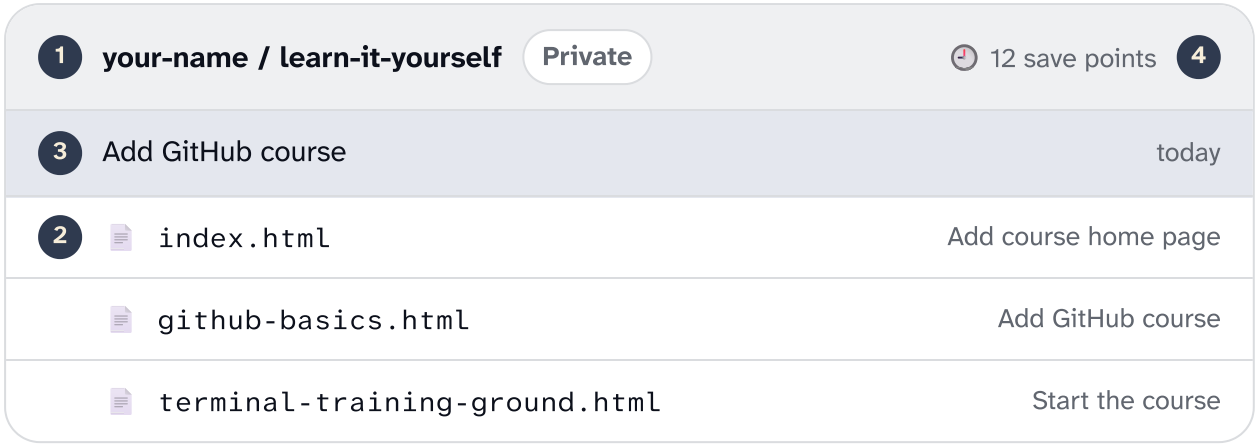
Lesson 04 Reading

One project, one repository

On GitHub, each project lives in its own **repository**, or **repo** for short. A repository is simply a project folder together with its full history of save points.

Your setup follows a simple rule: every folder in `piy-projects` is one project, and each project gets its own **private** repository with the same name. So this course lives in the `learn-it-yourself` folder on your computer, and in a private repository called `learn-it-yourself` on GitHub.

If you open one of your repositories on the GitHub website, you'll see something roughly like this:



1. **The name and the Private label.** “Private” means only you can see it. If it ever says “Public”, anyone on the internet can.
2. **The files.** The same files as in your folder, as they were at the latest upload. Next to each file is the note from the last save point that changed it.
3. **The latest save point.** Its note and when it was made. If this looks older than you expect, your newest work hasn't been uploaded yet.
4. **The history.** Click here to see the whole timeline of save points and exactly what changed in each one.

> **You don't need to visit the website**
>
> Claude does everything from your computer. The website is just a nice window onto your projects if you're ever curious.

Lesson 05 Reading

Why this helps you

You build small tools for yourself with Claude. That's exactly the kind of work GitHub protects best:



A backup you never think about

Laptop stolen, broken or replaced? Every uploaded project comes back in a minute, history included.



An undo button for anything

If Claude's latest change broke your tool, you can go back to the last version that worked, even days later.



See what changed

Every save point records exactly which lines changed. Ask Claude “what did we change this week?” and get a clear answer.



Experiment without fear

Because you can always go back, you can let Claude try bolder ideas. The worst case is “go back to before”.



Work from any computer

On a new or second computer, Claude can download any of your projects and carry on.



Claude works more safely

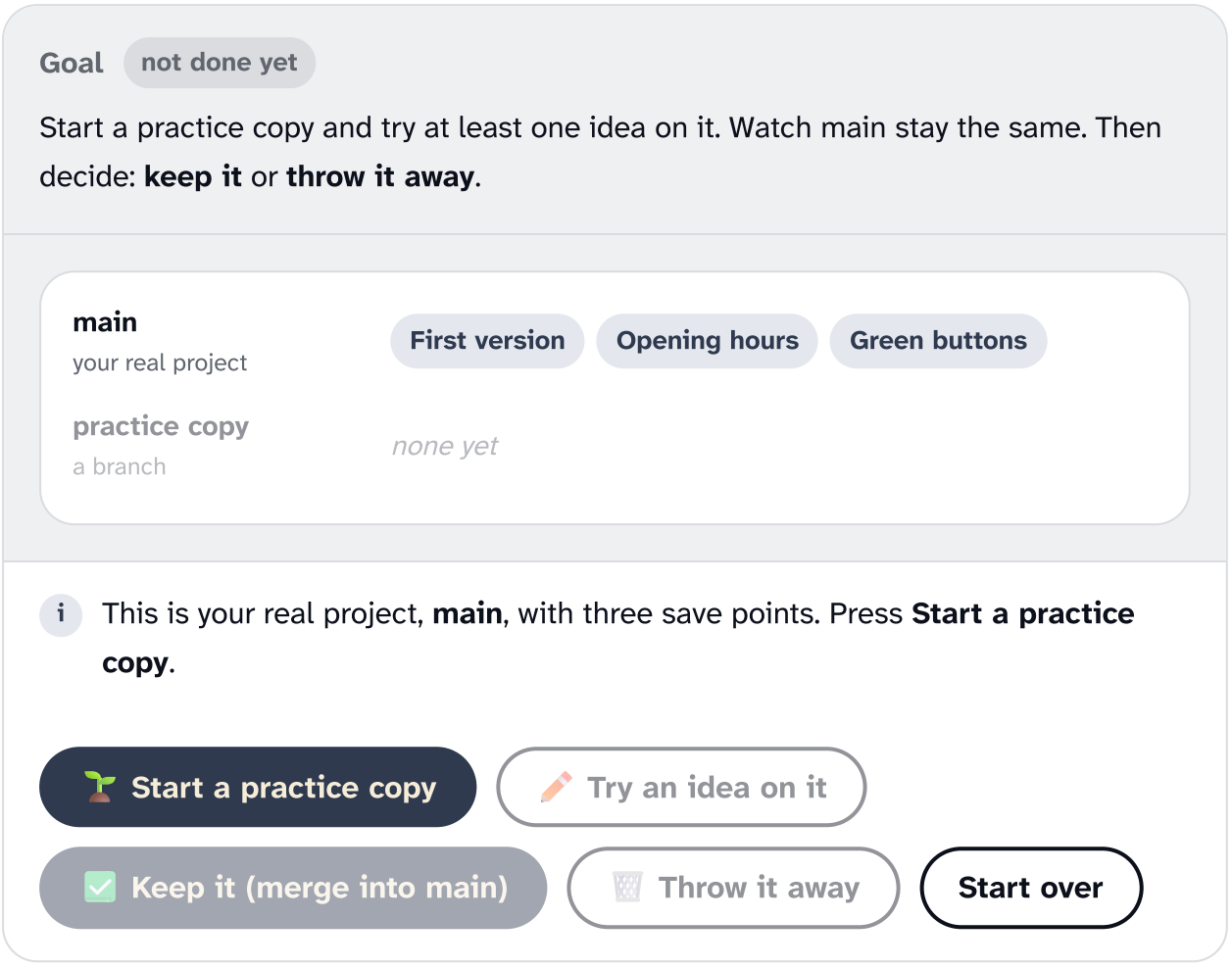
Save points give Claude a safe spot to return to, and the notes help it remember why things were changed.

Lesson 06 Hands-on practice

Practice copies: trying ideas safely

Sometimes you want to try something big, like a complete redesign, without touching the version that works. For that, Claude can make a **branch**: a separate line of save points that splits off from your project, like a practice copy.

- Your real project lives on the main line, called **main**.
- On a branch, Claude can make as many changes and save points as it likes. Main stays exactly as it was.
- If you like the result, Claude **merges** it: the branch's changes are added to main.
- If you don't, the branch is simply thrown away, and main never knew it existed.



Do I need branches?

> Not often. For everyday changes, save points on main are perfect. Branches are handy for big “I'm not sure about this” experiments, so it's enough to know that you can ask for one.

Lesson 07 Quiz

What to say to Claude

You never need the official words. Say what you want in normal language and Claude will run the right commands. Here are some sentences that work well:

You want to...	Say something like...	Official name
Back up your work	“Save this to GitHub.”	commit + push
See what changed	“What did we change since yesterday?”	diff, log
Undo a change	“Go back to the version from before the redesign.”	revert, restore
Try something risky	“Try this on a separate branch first.”	branch
Keep the experiment	“I like it. Merge it into main.”	merge
Drop the experiment	“Throw that branch away.”	delete branch
Start a new project online	“Put this project on GitHub as a private repository.”	create repo
Get a project on another computer	“Download my learn-it-yourself project from GitHub.”	clone

You want to...	Say something like...	Official name

Now try it. Pick the best answer for each situation.

Question 1 of 4 not done yet

Your laptop is stolen. Which of your projects can you get back?

- All of them. GitHub saves everything automatically.
- The ones that were saved and uploaded to GitHub, up to their last upload.
- None of them. They were on the laptop.

Question 2 of 4 not done yet

Claude changed your tool this morning and now it's broken. Yesterday it worked. What do you say?

- “Delete the project and let's start again.”
- “Upload it to GitHub, that will fix it.”
- “Go back to yesterday's version, from before this morning's change.”

Question 3 of 4 not done yet

You want to try a complete redesign, but you're not sure you'll like it.

- “Try the redesign on a separate branch first.”
- “Turn off save points while you try it.”
- “Make me a second GitHub account for it.”

Question 4 of 4 not done yet

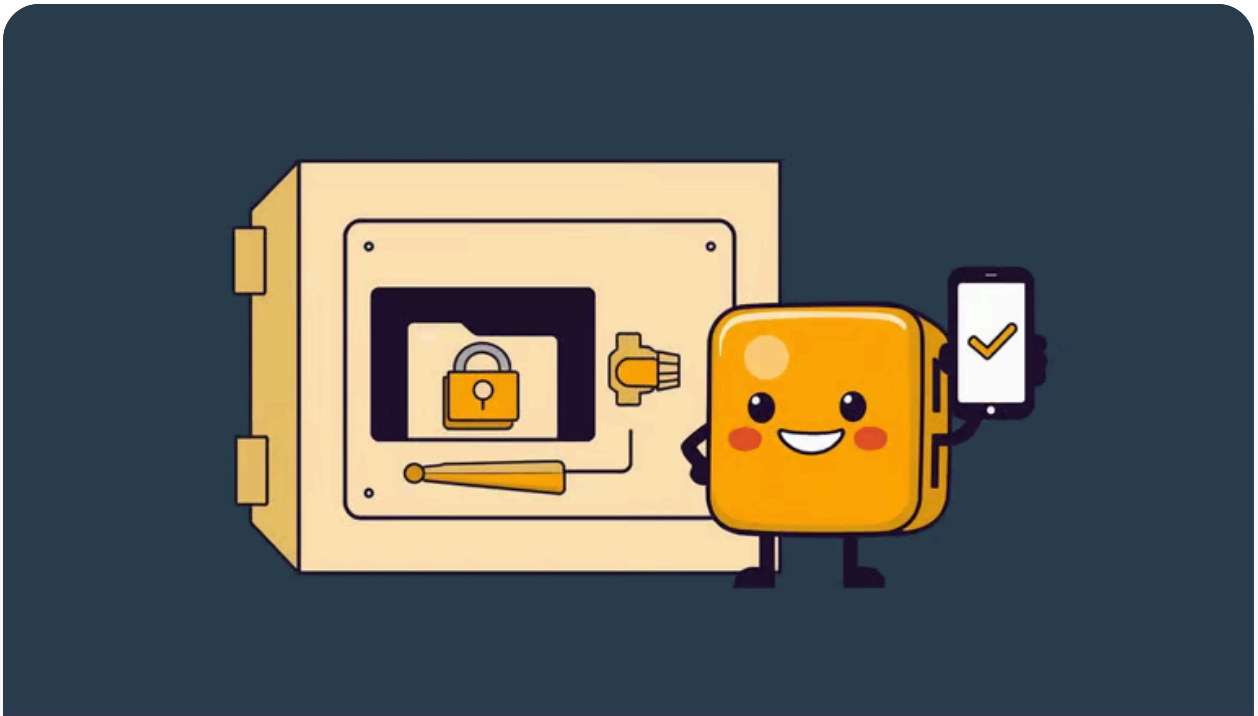
Who can see your private repositories?

- Anyone who searches for them on GitHub.
- Only you, plus anyone you personally invite.
- Nobody, not even you, without asking Claude.

Lesson 08 Reading

Staying safe

- **Private by default.** All your repositories are private. Claude won't make anything public unless you clearly ask for it.
- **Keep passwords out of projects.** Never put passwords, bank details or secret keys inside a project. A save point remembers everything, so even if you delete a password later, it is still sitting in the history.
- **Protect your GitHub account.** Use a strong password and turn on two-step login, where GitHub also asks for a code from your phone. Then a stolen password alone isn't enough to get in.
- **Deleting a repository is permanent.** Deleting files inside a project is fine, because the history keeps them. But deleting a whole repository on GitHub removes its history too.



Private repositories plus two-step login keep your projects locked away from everyone but you.

If something private slipped in

Tell Claude straight away: “I think a password ended up in this project.” Change that password on the service it belongs to first. That is the step that really protects you.

Lesson 09 Reference

Word list

Claude sometimes uses these words in its replies. Here is what each one means in plain language.

The basics

Git the save-point tool on your computer

Saving and history

commit a save point

commit message the save point's short

GitHub	the website that stores copies online
repository / repo	one project plus its history
private / public	only you / anyone can see it
local	on your computer
remote / origin	the copy on GitHub

	note
history / log	the list of all save points
diff	what changed between two versions
revert	undo a save point by adding a new one
uncommitted changes	edits with no save point yet

Moving copies around

push	upload save points to GitHub
pull	download newer save points
clone	download a whole project the first time
up to date	both copies match
ahead by 2	2 save points not uploaded yet

Practice copies

branch	a separate line of save points
main	the real, main line
merge	add a branch's changes to main
conflict	two changes to the same spot; Claude sorts it out
pull request	a “please merge this” proposal, used mostly in teams

Lesson 10 Common questions

Common questions

▶ **Does GitHub save my work automatically?**

▶ **I changed something, but GitHub still shows the old version.**

▶ **Does it cost anything?**

▶ **Can other people see my projects?**

▶ **If I go back to an old version, do I lose the newer work?**

▶ **I deleted a file by mistake. Is it gone?**

▶ **Is GitHub the same as OneDrive or Dropbox?**

▶ **Claude mentioned a “conflict”. Should I worry?**

The exercises on this page are simplified pretend versions. Real GitHub has more buttons, but the ideas are exactly the same. Your ticks are saved in this browser, and in your account when you're logged in.

Learn It Yourself is part of **Prompt It Yourself**.

Teaching humans to work with their AI agents.

Take the free Fundamentals course

Prompt It Yourself

Home

Learn: Fundamentals course

About us

Help & legal

contact@promptityourself.com

Privacy

Disclaimer

Follow

YouTube

LinkedIn

GitHub

© 2026 Prompt It Yourself · Operated by Mergodon Pty Ltd, Brisbane, Australia