GitHub Projects for Dummies

A beginner-friendly guide to GitHub Organizations, Repositories, and Projects.

By Georgios Giannakopoulos · January 2026

Table of Contents

Who This Tutorial Is For

This tutorial is written for **absolute beginners** who find GitHub Projects confusing or overwhelming.

It is especially useful if you:

- Have a **personal GitHub account** and are not sure when to use organizations
- Don’t understand the difference between **repositories, projects, and issues**
- Want to **organize research, learning, or side projects**
- Prefer **step-by-step explanations** instead of abstract documentation

No prior knowledge of GitHub Projects is assumed.

1. Home

This tutorial explains GitHub Projects in a way GitHub rarely does: **simple, step-by-step, no vague jargon**.

If you are confused by Projects, it’s usually because of these problems:

- You expect Projects to work like folders for repositories (they don’t)
- You created a Project in your personal account but expected it to live in the organization.
- You tried to “link a project to a repo directly” (GitHub doesn’t work like that).

The simplest summary

- **Organization** = container that can own repositories and projects
- **Repository** = where files/code/papers live
- **Project** = a task board that tracks work (cards), often using Issues

---

2. Core Concepts

This section fixes the #1 beginner confusion: these are different things.

1) Organization (org)

An **Organization** is like a workspace that can own:

- repositories
- projects
- people / teams

Organizations are the only real way to “group repositories”.

✅ Example: – `progenesis-research` (organization)

2) Repository (repo)

A **Repository** is where your content lives:

- code
- LaTeX papers
- figures
- datasets
- documentation

A repo belongs to either:

- your personal account, OR
- an organization

✅ Example: – `progenesis-research/mdpi-paper-mmwave-v2x`

3) Project (GitHub Projects)

A **Project** is a planning tool (like Trello/Jira-lite):

- board view (Todo → In Progress → Done)
- table view (spreadsheet)
- roadmap view (timeline)

A project does NOT contain repositories.

> 💡 **Key sentence:**
> **Projects organize work. Repositories store content.**

Correct hierarchy (memorize this)

```
Personal GitHub Account
└── Organization
├── Repositories
│   ├── repo-1
│   ├── repo-2
│   └── repo-3
└── Projects
├── Project A
└── Project B
```

This hierarchy means:

- Your **personal account** is just how you log in.
- An **organization** is a container that groups related work.
- **Repositories** store files, code, papers, figures, and data.
- **Projects** track work using tasks and issues.

Repositories and Projects live **side by side** inside an organization. Projects do **not** contain repositories.

Important rule

- ✅ Use **Organizations** to group repositories.
- ❌ Do **not** expect Projects to act like folders for repositories.

> 📌 **Think of it this way:**
> Repositories hold *things*.
> Projects track *work*.

3. How Projects Link to Repositories

This is the “aha moment”.

The rule

> Projects do NOT link to repositories directly.

Instead, GitHub links them through **Issues** (or Pull Requests).

The real chain

Project → Issue → Repository

Meaning:

- An **Issue** belongs to a **Repository**
- A **Project** contains a card that can reference that issue
- That card is what makes it *feel linked*

Step-by-step: link a repo to a project

Step 1 — Create an Issue in the repository

1. Open your repo
2. Click **Issues**
3. Click **New issue**
4. Give it a title (example):
    - `MDPI paper – mmWave V2X antennas`
5. Click **Submit**

Step 2 — Add that Issue to the Project

Method A (from the Issue page):

1. Open the issue
2. On the right sidebar find **Projects**
3. Select your project (e.g., `Research Roadmap`)

Method B (from the Project board):

1. Open the project
2. Click **+ Add item**
3. Choose **Add issue**
4. Select the issue from your repository

Result

Now your project shows a card that includes:

- the issue title
- the repo name

---

4. Organization vs Personal Projects

GitHub makes it easy to accidentally create Projects in the wrong place.

Important rule

A Project belongs to ONE owner:

- your personal account, OR
- an organization

It cannot belong to both.

How to tell (this is the only reliable check)

Personal project

You will see something like: – `@yourusername's Projects`

Organization project

You will see: – `org-name / Projects / project-name`

Example: – `progenesis-research / Projects / Research Roadmap`

Why this matters

If you create a project in your personal account but your repositories are in an organization:

- you'll feel like nothing connects
- issues may not show up the way you expect
- you'll keep fighting the interface

Correct approach for research

If the org is for research:

- Create your Projects inside the org
- Create repos inside the org
- Link via Issues

---

5. Archive, Delete, and Backup Safely

Beginners often say "`close the repo/project/org`", but GitHub uses different actions.

What "close" really means in GitHub

Thing	"Close" exists?	What you can do
Issue	✅ Yes	Close
Project	❌ No	Archive or Delete
Repository	❌ No	Archive or Delete
Organization	❌ No	Delete only

Safe option: Archive (recommended)

Archiving is a “soft close”:

- keeps everything
- makes it read-only
- can be restored later

Archive a repository

Repo → **Settings** → scroll to **Danger Zone** → **Archive this repository**

Archive a project

Project → **Settings** → **Archive project**

Dangerous option: Delete (irreversible)

Deleting removes it permanently. Only do it if:

- it’s a test
- it’s empty
- you are 100% sure you never need it again

Delete a repository

Repo → **Settings** → **Danger Zone** → **Delete this repository**

Delete a project

Project → **Settings** → **Delete project**

Delete an organization

Org → **Settings** → **Danger Zone** → **Delete organization** This deletes EVERYTHING inside it (repos + projects).

Backups (do this before deleting)

Quick backup

Repo → **Code** → **Download ZIP**

Proper backup (best)

Use git:

```
git clone https://github.com/ORG/REPO.git
```

Downloads

You can download this tutorial in multiple formats:

- 📄 **PDF version** — for printing or offline reading
- 📝 **Word (.docx) version** — for editing or annotations
- 📘 **Markdown (.md) version** — for reuse or contribution

All files are available in the repository's exports/ folder.

License

🏁 **End of tutorial**