

# Version control, Git, GitHub and GitFlow

Criscely Luján<sup>1,2</sup>

*[criscely.lujan@ird.fr](mailto:criscely.lujan@ird.fr)*

Nicolas Barrier<sup>2</sup>

*[nicolas.barrier@ird.fr](mailto:nicolas.barrier@ird.fr)*

<sup>1</sup>Université Paris-Sud, UMR MARBEC

<sup>2</sup>IRD, UMR MARBEC

April 11, 2019

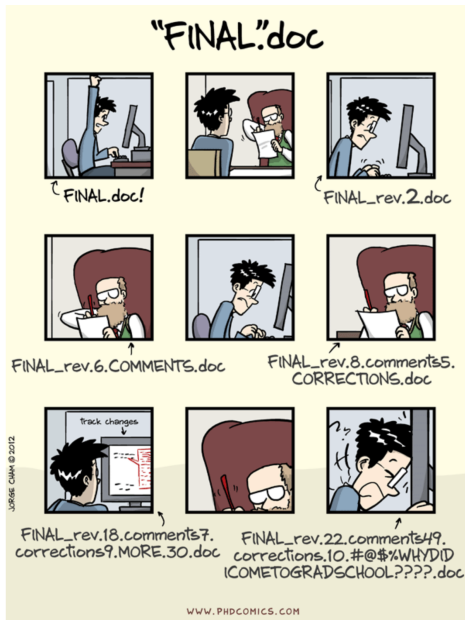


Also known as **revision control** or **source control**.

... *"is the management of changes:*

- *documents*
- *computer programs*
- *large web sites*
- *other collections of information ... "*

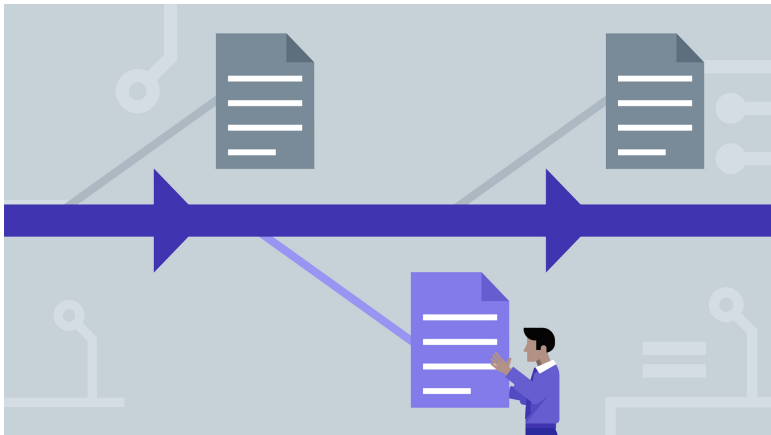
# Why version control is important?



# Why version control is important?

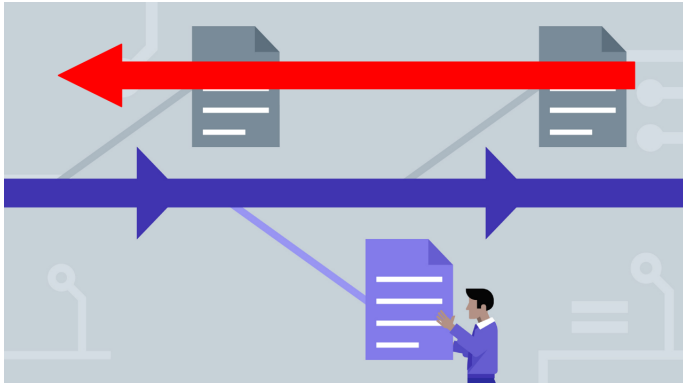
Storing **version** (properly).

- Saving successive changes (“commit”)
- Versioning (v0.1)



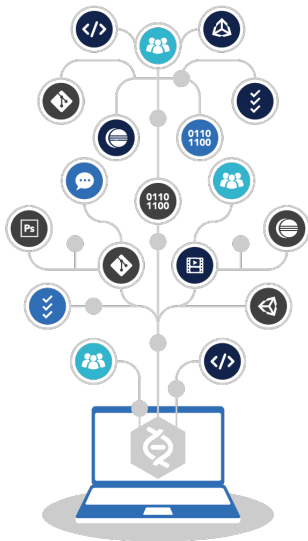
# Why version control is important?

**Restoring** previous versions.



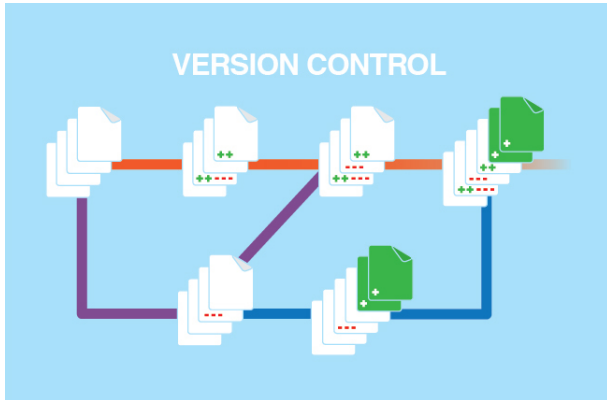
# Why version control is important?

**Collaborations** (networking).



# Why version control is important?

Save **time**.



# Version control software

Version control software			[hide]
Years, where available, indicate the date of first stable release. Systems with names <i>in italics</i> are no longer maintained or have planned end-of-life dates.			
Local only	Free/open-source	RCS (1982) · SCCS (1972)	





# What is Git?

**Git** is a distributed version control system for tracking changes in source code during the development of software.



# Why use Git?

- **Popular and successful**
  - Active development
  - Fast
- **Distributed**
  - Work online and offline
  - Collaborate with large groups
- **Tracks any type of file**
  - Works best with text
- **Branching**
  - Smarter merges

# What is GitHub Inc.?

**GitHub** is a web-based hosting service for version control using **Git**.

The GitHub logo, which consists of the word "GitHub" in a bold, black, sans-serif font.

[📄 Download logo](#)



[📄 Download mark](#)



[📄 Download Octocat](#)

- Access to the control and collaboration features for every **project**.

The screenshot shows the GitHub repository settings page for a repository named 'osmose\_configurations'. At the top, there is a navigation bar with links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings (which is highlighted with an orange bar). On the left side, there is a sidebar with a list of settings categories: Options (highlighted with an orange bar), Collaborators, Branches, Webhooks, Notifications, Integrations & services, and Deploy keys. The main content area is titled 'Settings' and contains several sections. The 'Repository name' section shows the current name 'osmose\_configurations' and a 'Rename' button. The 'Features' section lists several features that are enabled with checkboxes: Wikis, Restrict editing to collaborators only, and Issues. Each feature has a brief description. There is also a promotional box for 'Get organized with issue templates' with a 'Set up templates' button. At the bottom, the 'Projects' feature is also listed as enabled.

<> Code ① Issues 0 🔄 Pull requests 0 📁 Projects 0 📖 Wiki 📊 Insights ⚙️ Settings

**Options**

- Collaborators
- Branches
- Webhooks
- Notifications
- Integrations & services
- Deploy keys

## Settings

**Repository name**

osmose\_configurations Rename




## Features

- ☒ **Wikis**  
GitHub Wikis is a simple way to let others contribute content. Any GitHub user can create and edit pages to use for documentation, examples, support, or anything you wish.
- ☒ **Restrict editing to collaborators only**
- ☒ **Issues**  
Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and milestones, and reference them in commit messages.  

**Get organized with issue templates**  
Give contributors issue templates that help you cut through the noise and help them push your project forward. Set up templates
- ☒ **Projects**  
Project boards on GitHub help you organize and prioritize your work. You can create project boards for specific feature work, comprehensive roadmaps, or even release checklists.

- Work with public and private **repositories**.


**Owner**      **Repository name**


PUBLIC   **hubot** ▾ /  

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

**Description** (optional)

---

☒  **Public**  
Anyone can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---

☒ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

▾ |  ▾ ⓘ

---

**Create repository**

- Develop a **networking**.

The screenshot displays a GitHub activity feed with three entries:

- jennybc** forked **jennybc/testthat-1** from **cran/testthat** 5 days ago.
  - cran/testthat**: A read-only mirror of the CRAN R package repository. **testthat** — Unit Testing for R. Homepage: <http://testthat.r-lib.org>, <https://github.com>...
  - Language: R, Updated Oct 13.
  - Star button.
- jeroen** created 2 repositories 5 days ago.
  - ropensci-docs/magick**: Updated Apr 5.
  - Star button.
- mkearney** starred 4 repositories 5 days ago.
  - bentrevett/pytorch-seq2seq**: Tutorials on implementing a few sequence-to-sequence (seq2seq) models with PyTorch and TorchText.
  - Language: Jupyter Notebook, 245 stars, Updated Apr 10.
  - Star button.
- barriern** created 2 repositories 6 days ago.

- Source of information.


The screenshot shows the GitHub search interface for the query 'ecosystem models'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. On the left sidebar, there are filters for Repositories (217), Code (289K), Commits (831), Issues (2K), Marketplace (0), Topics (1), Wikis (1K), and Users (13). Below these are filters for Languages, listing R (30), Java (29), Python (18), C++ (17), Fortran (11), JavaScript (11), Jupyter Notebook (8), HTML (7), MATLAB (5), and Ruby (3). The main content area displays '217 repository results' with a 'Sort: Best match' dropdown. Three repository results are visible: 1. 'EDmodel/ED2' (Fortran, 35 stars) with description 'Ecosystem Demography Model' and a note 'Updated on Dec 3, 2018 5 issues need help'. 2. 'kokkos/kokkos' (C++, 351 stars) with description 'Kokkos C++ Performance Portability Programming EcoSystem: The Programming Model - Parallel Execution and Memory Abstr...' and tags 'c-plus-plus', 'parallel-computing', and 'abstraction'. 3. 'ESCOMP/ctsm' (Fortran, 83 stars) with description 'Community Terrestrial Systems Model (includes the Community Land Model of CESM)' and tags 'ecosystem', 'climate', 'land', 'hydrology', and 'ncar'. A fourth result, 'sizespectrum/mizer' (R, 6 stars), is partially visible at the bottom with the description 'Multi-species size-based ecological modelling in R'.

Repository	Language	Stars	Description
EDmodel/ED2	Fortran	35	Ecosystem Demography Model
kokkos/kokkos	C++	351	Kokkos C++ Performance Portability Programming EcoSystem: The Programming Model - Parallel Execution and Memory Abstr...
ESCOMP/ctsm	Fortran	83	Community Terrestrial Systems Model (includes the Community Land Model of CESM)
sizespectrum/mizer	R	6	Multi-species size-based ecological modelling in R

- **Plans** for enterprise, teams, pro and free accounts.

## Plans for every developer

Whether you're starting an open source project or choosing new tools for your team, we've got you covered.



### Individuals

Free	Pro
<b>\$0</b>	<b>\$7</b>
Per month The basics of GitHub for every developer	Per month Pro tools for developers with advanced requirements
<ul style="list-style-type: none"><li>Unlimited public repositories</li><li>Unlimited private repositories</li><li>3 collaborators for private repositories</li><li>Issues and bug tracking</li><li>Project management</li></ul>	<ul style="list-style-type: none"><li>Unlimited public repositories</li><li>Unlimited private repositories</li><li>Unlimited collaborators</li><li>Issues and bug tracking</li><li>Project management</li><li>Advanced tools and insights</li></ul>

Included free alongside other real-world development tools in the [GitHub Student Developer Pack](#)

Included in Pro

Already signed up

### Teams

Team	Enterprise
<b>\$9</b> Per user / month Advanced collaboration and management tools for teams	Contact Sales for pricing Security, compliance, and deployment controls for organizations
<ul style="list-style-type: none"><li>Unlimited public repositories</li><li>Unlimited private repositories</li><li>Team access controls</li><li>User management and billing</li><li>Issues and bug tracking</li><li>Project management</li><li>Advanced tools and insights</li></ul>	<ul style="list-style-type: none"><li>Everything included in Team</li><li>Self-hosted or cloud-hosted</li><li>SAML single sign-on</li><li>Access provisioning</li><li>Simplified account administration</li><li>Unified search and contributions</li><li>Priority support</li><li>99.95% uptime SLA for Enterprise Cloud</li><li>Invoice billing</li><li>Advanced auditing</li></ul>

Starts at \$15 / month and includes your first 5 users

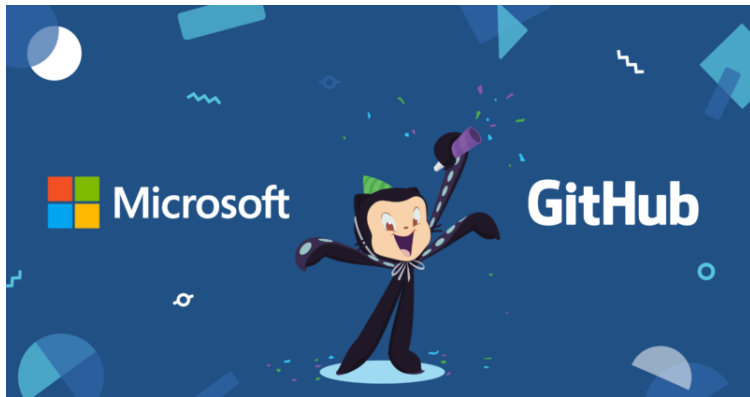
Free for academic faculty for teaching or non-profit research

Questions? — [Learn more about Enterprise](#)

Free for educational institutions participating in the [GitHub Education program](#)



- Is the **largest** host of source code in the world! (*28 million users, 57 million repositories (28 million public) - June 2018*).

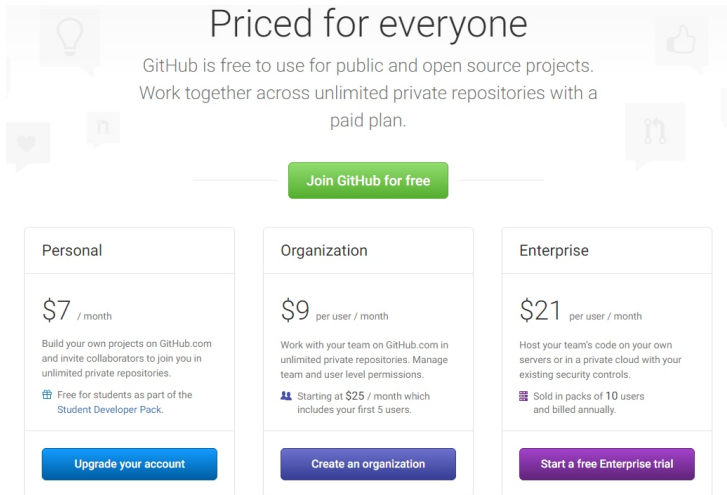


# Register a GitHub account

- Create an account in [GitHub](#) is free!
- Free private repositories
  - Students, faculty, and educational / research staff: [GitHub Education](#).
  - Official nonprofit organizations and charities: [GitHub for Good](#).

# Register a GitHub account




- Pay for private repositories
  - Individual cost is 7 dollars per month: [GitHub Pricing](#).

A screenshot of the GitHub Pricing page. At the top, the heading "Priced for everyone" is centered. Below it, a paragraph states: "GitHub is free to use for public and open source projects. Work together across unlimited private repositories with a paid plan." A green button labeled "Join GitHub for free" is centered below the text. The page is divided into three columns representing different pricing tiers: Personal, Organization, and Enterprise. Each column contains a price per month, a description of the plan's features, and a button to upgrade or create an account.

**Priced for everyone**

GitHub is free to use for public and open source projects. Work together across unlimited private repositories with a paid plan.

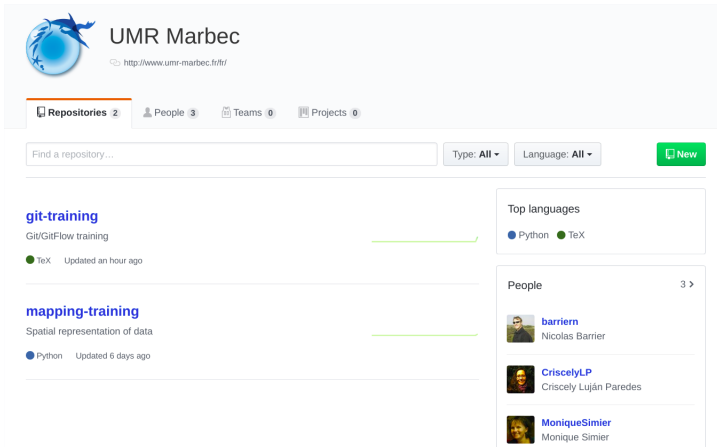
[Join GitHub for free](#)

Personal	Organization	Enterprise
<b>\$7</b> / month	<b>\$9</b> per user / month	<b>\$21</b> per user / month
Build your own projects on GitHub.com and invite collaborators to join you in unlimited private repositories.	Work with your team on GitHub.com in unlimited private repositories. Manage team and user level permissions.	Host your team's code on your own servers or in a private cloud with your existing security controls.
 Free for students as part of the Student Developer Pack.	 Starting at \$25 / month which includes your first 5 users.	 Sold in packs of 10 users and billed annually.
<a href="#">Upgrade your account</a>	<a href="#">Create an organization</a>	<a href="#">Start a free Enterprise trial</a>

# Marbec in GitHub

All the materials of Pole Modelisation's technical "workshop" are now stored in an institutional GitHub account:

<https://github.com/umr-marbec>.



The screenshot shows the GitHub profile page for UMR Marbec. At the top, there is a profile picture of a blue globe with a fish-like shape, the name "UMR Marbec", and a website link "http://www.umar-marbec.fr/fr/". Below this, navigation tabs show "Repositories 2", "People 3", "Teams 0", and "Projects 0". A search bar "Find a repository..." is present, along with filters for "Type: All" and "Language: All", and a green "New" button. The main content area lists two repositories: "git-training" (Git/GitFlow training, TeX, updated an hour ago) and "mapping-training" (Spatial representation of data, Python, updated 6 days ago). On the right, a "Top languages" section shows Python and TeX, and a "People" section lists three contributors: barriern (Nicolas Barrier), CriscelyLP (Criscely Luján Paredes), and MoniqueSimier (Monique Simier).

GitHub is a private US company. There are also *institutional* repositories on which Git can be used:

- [Sourcesup](#): this is a Renater platform (login possible from any French research institute or through CRU accounts)
- [Forge Ifremer](#): very close to SourceSup (Ifremer extranet account required)
- [IRD GitLab](#): GitLab IRD platform (IRD account required).

However, the projects hosted on these repositories may have less visibility...

# Git clients

Git and Git client **are not** the same! Like R and RStudio is not the same thing!

Git client:

- IDE (Integrated development environment)!
- Make the experience more pleasant providing a richer visual representation.

Some example of Git clients:

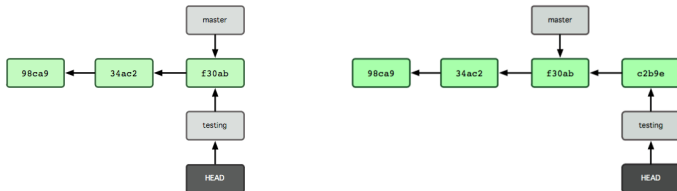
- [SourceTree](#)
- [GitKraken](#)
- [GitUp](#)
- [SmartGit](#)
- [git-cola](#)
- [RStudio](#)

# Git branches

One main advantage of Git is the use of *branches*, which allow multiple developments of the same code at the same time.

## Definition

A branch in Git is simply a lightweight movable pointer to one of the commits.



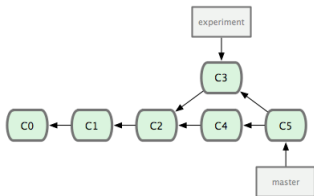
In this example, the `master` branch points to the `f30ab` commit, while the `testing` branch points to the `c2b9e` one. `HEAD` points to the active branch (here, `testing`).

Source: <https://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

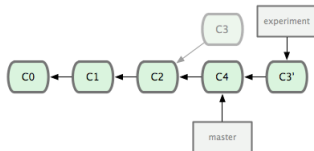
# Merging branches

To merge a branch (for instance a feature branch) to another branch (for instance the main one), several options are offered.

- merge: Three-points branch (common ancestor + tips of the two branches)
- rebase: Compresses all the changes into a single “patch.”



(a) Merge



(b) Rebase

Figure: Merging versus rebasing

Source: <https://git-scm.com/book/fr/v1/Les-branches-avec-Git-Rebaser>



There are several ways to use Git branches (we talk about **workflows**).

- *Centralized workflow*: one main branch, everyone commit in the same place.
- *Feature Branch Workflow*: developments are made in dedicated branches (feature branches), which are regularly merged into the master one.
- **Gitflow Workflow**: Strict branching model designed around the project release.

Source: <https://www.atlassian.com/git/tutorials/comparing-workflows>

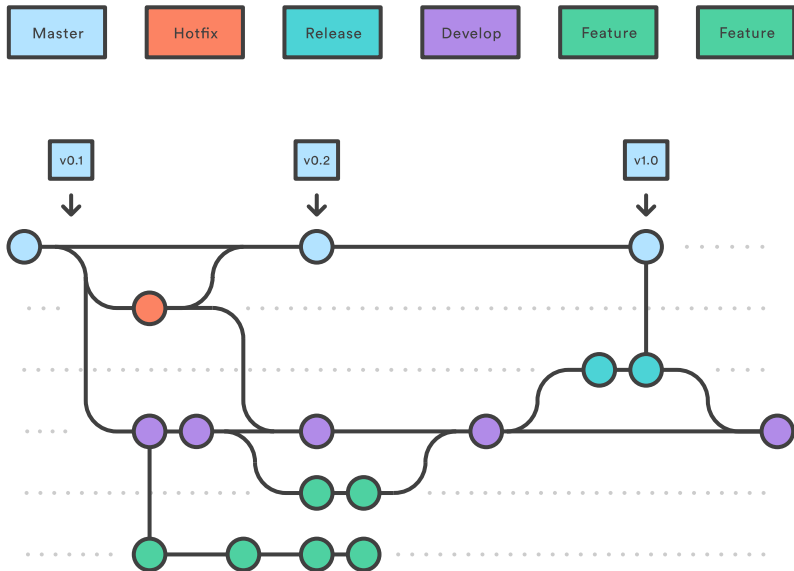
GitFlow workflow contains two main branches:

- **master**: official release history. Branch which is shared to the world!
- **develop**: integration branch for features

It also contains additional temporal branches:

- **feature**: feature branches (one for each new feature to add to the code)
- **release**: branch created when enough features have been added (new version of the code) to develop
- **hotfix**: branch for maintenance and bug correction of the production release

## In summary...



Source: <https://www.atlassian.com/git/tutorials/comparing-workflows>

# Thanks for your attention

**Now, let's crack on it!**



# This is the end



Source: [poshpete117.deviantart.com/journal/Thats-All-Folks-427323458/](https://poshpete117.deviantart.com/journal/Thats-All-Folks-427323458/)