

What is Git? <https://en.wikipedia.org/wiki/Git>

Git


From Wikipedia, the free encyclopedia

*For other uses, see [Git \(disambiguation\)](#).
Not to be confused with [GitHub](#).*

Git (/ɡɪt/^[7]) is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).^{[8][9][10]}

Git was created by [Linus Torvalds](#) in 2005 for development of the [Linux kernel](#), with other kernel developers contributing to its initial development.^[11] Since 2005, Junio Hamano has been the core maintainer. As with most other [distributed version control](#) systems, and unlike most [client–server](#) systems, every Git [directory](#) on every [computer](#) is a full-fledged [repository](#) with complete history and full version-tracking abilities, independent of network access or a central server.^[12] Git is [free and open-source software](#) distributed under the [GPL-2.0-only](#) license.

Contents [hide]
1 History
1.1 Naming
1.2 Releases
2 Design
2.1 Characteristics
2.2 Data structures
2.3 References
3 Implementations
4 Git GUIs
5 Git server
5.1 Open source
5.2 Git server as a service
6 Adoption
6.1 Extensions
7 Conventions
8 Security
9 Trademark



git

```
$ git init
Initialized empty Git repository in /tmp/tmp.I8bY5V788Y/.git/
$ cat > README << 'EOF'
> EOF
$ git add README
$ git commit
[master (root-commit) e4dccc89] You can edit locally and push
to any remote.
1 file changed, 1 insertion(+)
 create mode 100644 README
$ git remote add origin git@github.com:down/thats.git
$ git push -u origin master
```

A command-line session showing repository creation, addition of a file, and remote synchronization

Original author(s) [Linus Torvalds](#)^[1]

Developer(s) [Junio Hamano](#) and others^[2]

Initial release 7 April 2005; 16 years ago

Stable release 2.35.0^[3] / 24 January 2022

Repository git.kernel.org/pub/scm/git/git.git^[4]

Written in C, Shell, Perl, Tcl^[4]

Operating system POSIX (Linux, macOS, Solaris, AIX), Windows

Available in English

Type Version control

License [GPL-2.0-only](#)^{[1][6]}

Website git-scm.com^[5]

Other places you might use Git:

- Bitbucket (Github competitor): <https://bitbucket.org/>
- Gitlab (DevOps): <https://about.gitlab.com/>
- Gitbook (Docs oriented): <https://www.gitbook.com/>

What's in a name (Git)?

Naming [edit]

Torvalds sarcastically quipped about the name *git* (which means "unpleasant person" in [British English](#) slang): "I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."^{[23][24]} The [man page](#) describes Git as "the stupid content tracker".^[25] The read-me file of the source code elaborates further:^[26]

"git" can mean anything, depending on your mood.

- Random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- Stupid. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.
- "Global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "Goddamn idiotic truckload of sh*t": when it breaks.

A much older version of the Matrix (for version control):

<https://subversion.apache.org/>

Do a pull request:

1) fork a repo — make a change and submit:

- Chris can make a fork of [rokwire/rokwire-community](https://github.com/rokwire/rokwire-community).
- make a change in the interface.

2) what is in a repository?

- creating a new repository (why, how). Clone a repository on your local machine.
- templates for README, License, Contributors, Code of Conduct:
<https://github.com/rokwire/illinois-app>,
<https://github.com/rokwire/rokwire-community>,
<https://github.com/rokwire/rokwire-docs/tree/main/Templates>

3) styles of maintenance:

- Trunk-based vs. fork-based development.
- Git-forking vs. git-branching model of contributions.
- Code review: asynchronous, synchronous, meeting-based.

4) create an issue:

- address an already created issue (#140):
<https://github.com/rokwire/rokwire-community/issues/140>
- create a new issue of your own.
- review project board: <https://github.com/rokwire/rokwire-community/projects/1>

5) take a file and look at its history:

- example: Community Wiki README:
<https://github.com/rokwire/rokwire-community/blob/master/Community%20Wiki/README.md>

* look at commit history.

* look at diffs (line changes).

6) Wikis

- new pages, how to manage topical pages.
- building resources in Markdown (.md).

7) other features

- social aspects (stars, pins, watches).
- insights for rokwire/ <https://github.com/orgs/rokwire/insights>

- code frequency, dependency graph, networks, forks. Use the Illinois App repo:
<https://github.com/rokwire/illinois-app/pulse>

Github has resources for students:

- How to run virtual events (concept of a tech stack):
<https://github.blog/2021-06-24-run-online-campus-events-github-virtual-event-kit/>
- Github community (good features on Github functionality):
<https://github.blog/category/community/>