

# Git & GitHub

---

*I won't be impressed with technology,  
until I can download food.*

# What is Git ?





**WIKIPEDIA**  
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Current events](#)

[Random article](#)

[About Wikipedia](#)

[Contact us](#)

[Donate](#)

[Contribute](#)

[Help](#)

[Community portal](#)

[Recent changes](#)

[Upload file](#)

[Tools](#)

[What links here](#)

[Related changes](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Cite this page](#)

[Wikidata item](#)

[Print/export](#)

[Download as PDF](#)

[Article](#) [Talk](#)

[Read](#) [Edit](#) [View history](#)



Wiki Loves Earth:

An international photographic contest where you can showcase Malaysia's unique natural environment and potentially win a prizes.

# Git

From Wikipedia, the free encyclopedia

*For other uses, see [Git \(disambiguation\)](#).*

**Git** (/ɡɪt/<sup>[7]</sup>) is a [distributed version-control](#) system for tracking changes in [source code](#) during [software development](#).<sup>[8]</sup> It is designed for coordinating work among [programmers](#), but it can be used to track changes in any set of [files](#). Its goals include speed,<sup>[9]</sup> [data integrity](#),<sup>[10]</sup> and support for distributed, non-linear workflows.<sup>[11]</sup>

Git was created by [Linus Torvalds](#) in 2005 for development of the [Linux kernel](#), with other kernel developers contributing to its initial development.<sup>[12]</sup> Its current maintainer since 2005 is [Junio Hamano](#). 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.<sup>[13]</sup> Git is [free and open-source](#) software distributed under the terms of the [GNU General Public License](#) version 2.

## Contents [hide]

- History
  - Naming
  - Releases
- Design
  - Characteristics
  - Data structures

Git



```
$ git init
Initialized empty Git repository in /tmp/tmp.IHBYSY7RBY/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4d6c69] 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:cdown/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](#)<sup>[1]</sup>

**Developer(s)** [Junio Hamano](#) and others<sup>[2]</sup>

**Initial release** 7 April 2005; 15 years ago

Let's put things in a more simpler way

Git is a technology/application that allow different ppl to work together just like Google Docs / Spreadsheet.

**And it does more than that**

So... ok... Why I need Git?  
Why can't we just use Google  
Dog?



Because Git is particularly designated for programmer  
**Tracking / Collaboration / Popular platform** (esp open source)

Google docs will make you cry...



---

# What about GitHub?

---

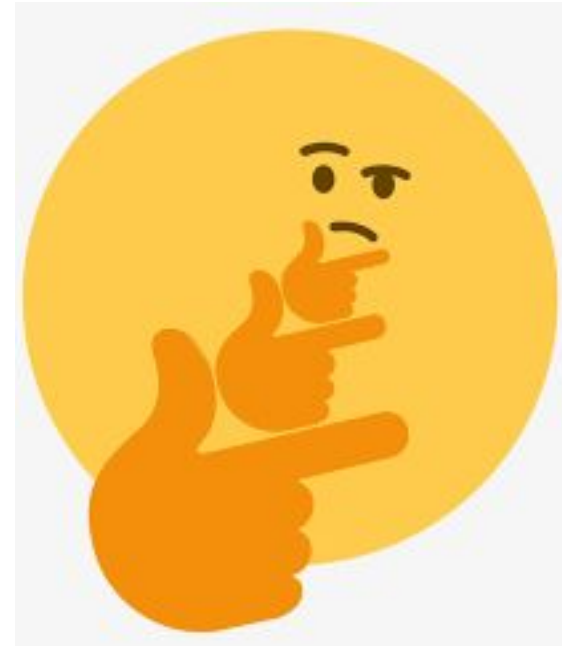
GitHub is a Git **repository hosting service**, but it adds many of its own features.

Git is a command line tool, GitHub provides a **Web-based graphical interface**.

GitHub also provides access control and several collaboration **features**:

- Wikis
- Open Issue
- Comment
- basic task management tools for every project

Wait, what about  
GitLab and  
BitBucket?





# They are all repository management platform

Though there are some differences amongst their management style

<https://stackshare.io/stackups/bitbucket-vs-github-vs-gitlab>



BitBucket



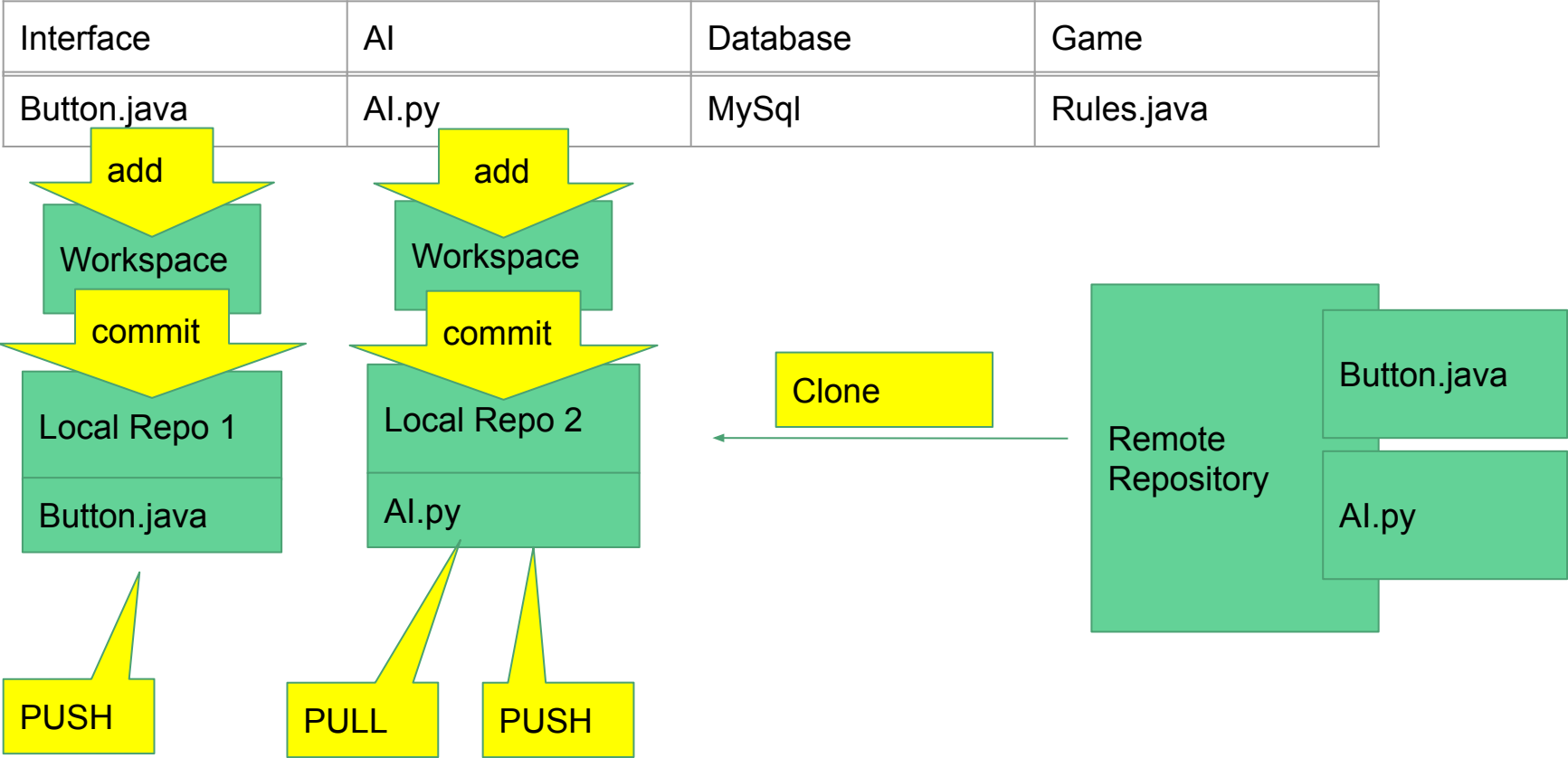
GitHub



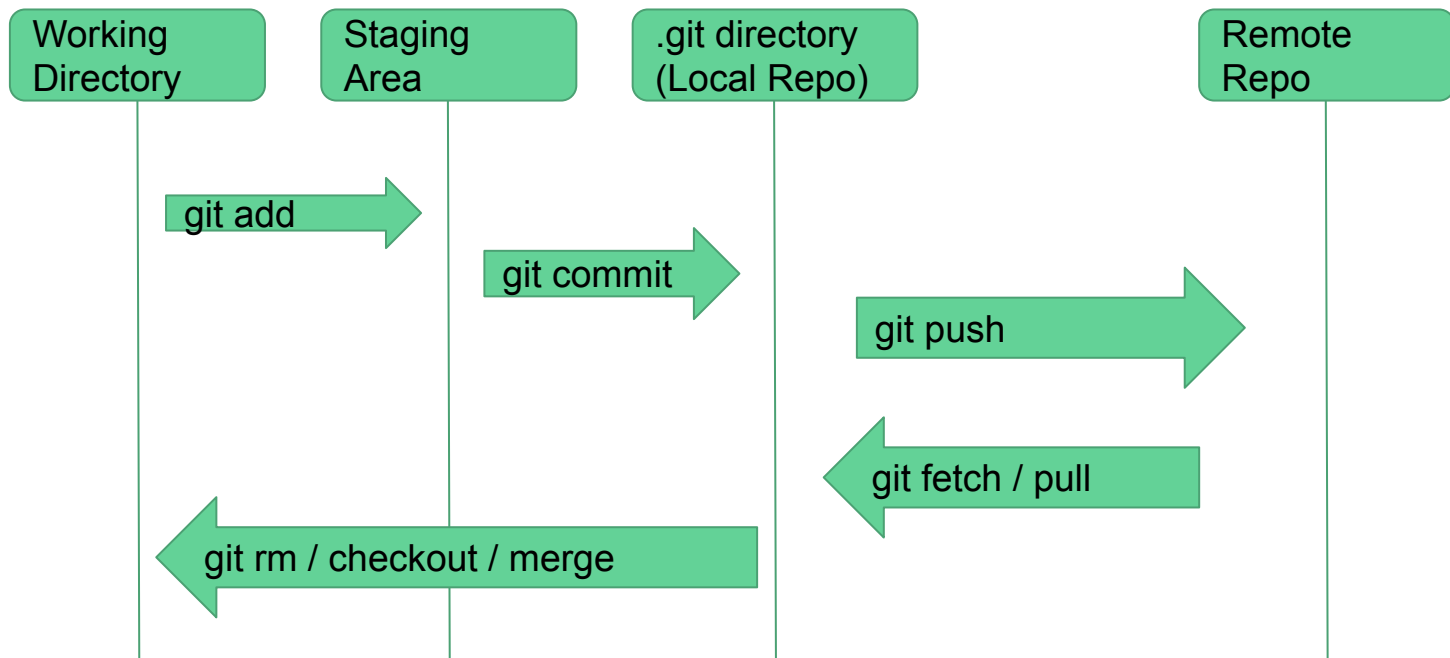
GitLab

# How Git Works

## Chess Game



# Helicopter view

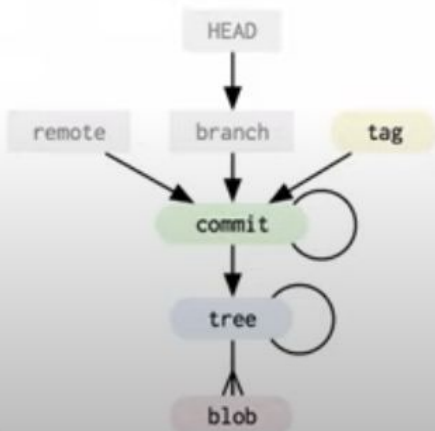




# Git Internals

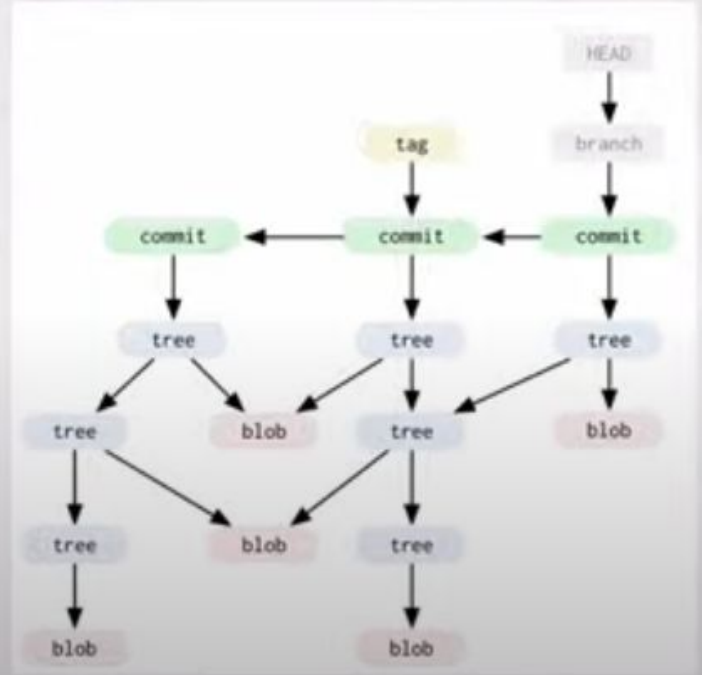
- Object Types
  - Commit - Author, message, pointer to a tree of changes
  - Tree - Pointer(s) to file names, content, other trees
  - Blob - Data (source code, pictures, videos, etc.)
- These are stored as objects on the filesystem
- Tags and Branches
  - Pointers to commits (lightweight)
  - Not full copies
  - Allows a name for a commit
    - Contains additional meta information

## Model



## Git Models across 3 commits

```
.  
|-- init.rb  
-- lib  
    |-- base  
    |-- base_include.rb  
    -- my_plugin.rb
```





***Git Bash!!***