Introduction to Data Analytics

T5 Bootcamp by SDAIA

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

GitHub

Let's start together…

Agenda

What Is GitHub?

GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects.

- GitHub facilitates easy sharing and collaboration of code files among developers, particularly for open-source projects.
- Since its establishment in 2008, GitHub has amassed millions of users and become a prominent platform for collaborative software projects.

What Is GitHub?

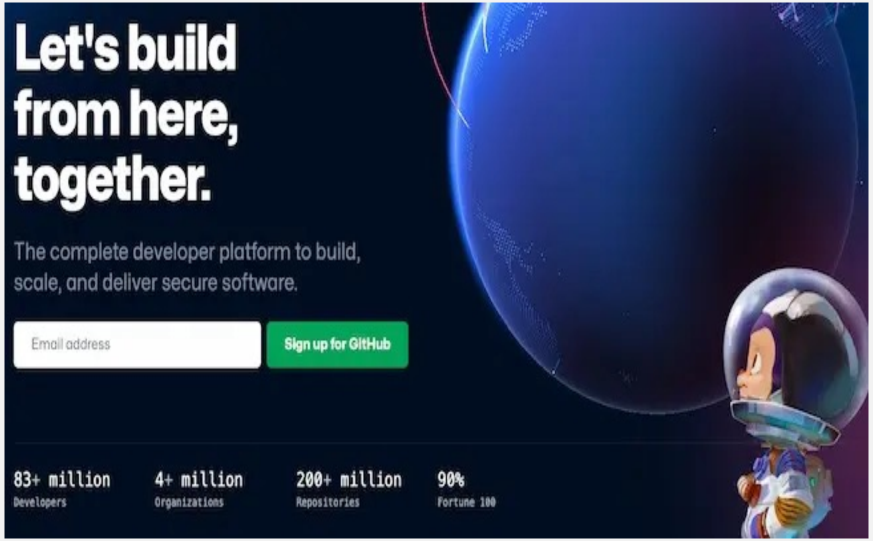
It functions as a social networking platform where developers can connect, collaborate, and showcase their work.

- In addition to its code-centric functionalities, GitHub encourages users to create personal profiles and establish their brand.
- Users can explore profiles to discover projects owned and contributed to by others, fostering a collaborative community for software and website development.
- To understand exactly what GitHub is, you need to know two connected principles:
  1. ***Version control***
  2. ***Git***

How does GitHub work?

GitHub users create accounts, upload files, and create coding projects. But the real work of GitHub happens when users begin to collaborate.

- Software projects often involve teams working together, presenting challenges in collaboration for distributed teams.
- GitHub simplifies this process by centralizing code and documentation within repositories, facilitating access for contributors and providing project guidelines.



Let's build from here, together.

The complete developer platform to build, scale, and deliver secure software.

83+ million Developers

4+ million Organizations

200+ million Repositories

90% Fortune 100

How does GitHub work?

- Coding involves creative processes, where compatibility between different pieces of code is crucial but prone to **errors**.
- GitHub addresses these issues by visualizing changes to both files and the **main branch**, enabling error detection before changes are pushed, thus enhancing coding efficiency.
- GitHub facilitates **version control**, allowing users to track changes and revert to previous versions of a project, leveraging the underlying technology, Git.

Why GitHub?

GitHub enables software developers and engineers to establish remote repositories on the cloud at no cost.

Users can duplicate repos to their local device, make modifications to files locally, and subsequently "push" these changes back to the repository, where they become visible to the public.

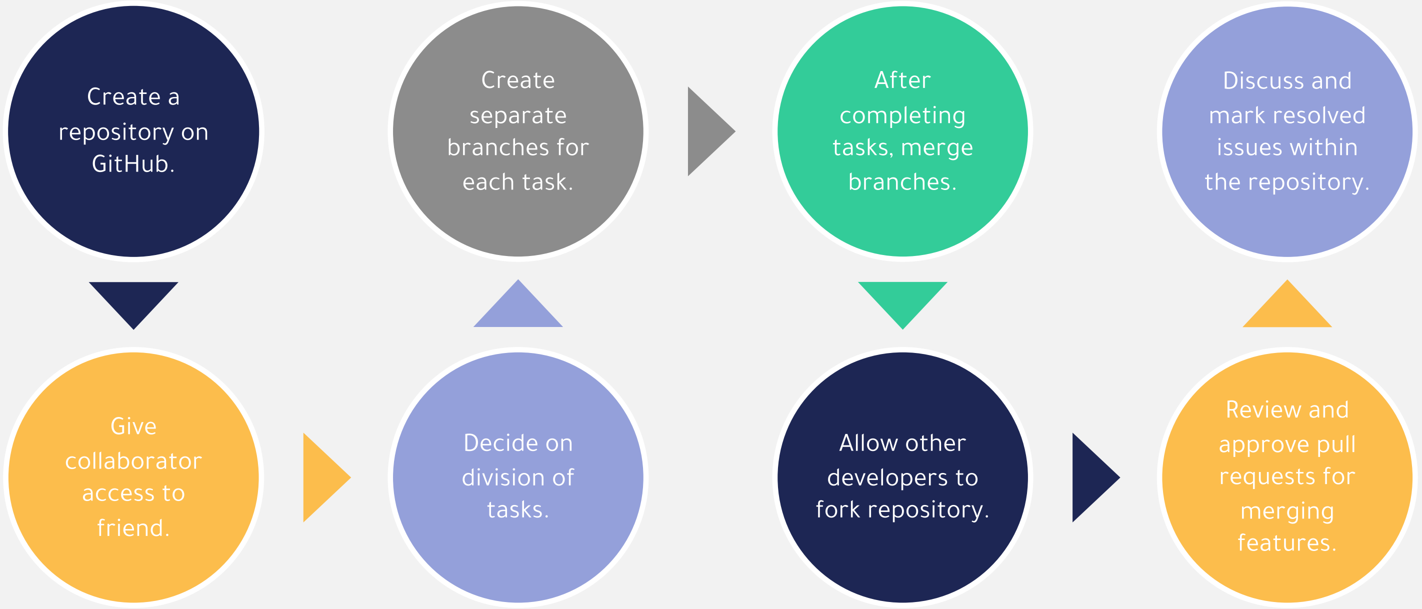
Now, why might one opt for GitHub instead of developing with a private repository?

- *Enhanced Collaboration*
- *Easy File Management*
- *Social Networking*
- *Open-Source Projects*
- *Private Repositories*

Why GitHub?

ENHANCED COLLABORATION

Imagine, you want to code an online game, and your friend will help you.



Create a repository on GitHub.

Give collaborator access to friend.

Decide on division of tasks.

Create separate branches for each task.

After completing tasks, merge branches.

Allow other developers to fork repository.

Review and approve pull requests for merging features.

Discuss and mark resolved issues within the repository.

Why GitHub?

EASY FILE MANAGEMENT

- Using GitHub means you're not limited to one device or environment.
- A GitHub user may access their repository from any location and any device, download the repository, and push their changes.

SOCIAL NETWORKING

- All GitHub users have profiles to display their projects activity on the site and can see anyone's public-facing profile and repositories.
- For example, recruiters often use GitHub to scout talent, since prospects' code is available for anyone to review.

OPEN-SOURCE PROJECTS

- GitHub has fueled a surge of open-source collaboration.
- GitHub has also opened up software development to anyone who wants to learn programming, fostering an engaged, innovative, and productive community.

PRIVATE REPOSITORIES

- GitHub provides paid services as well, including private repositories.
- On a paid plan, teams can collaborate on GitHub while keeping their code behind closed virtual doors.

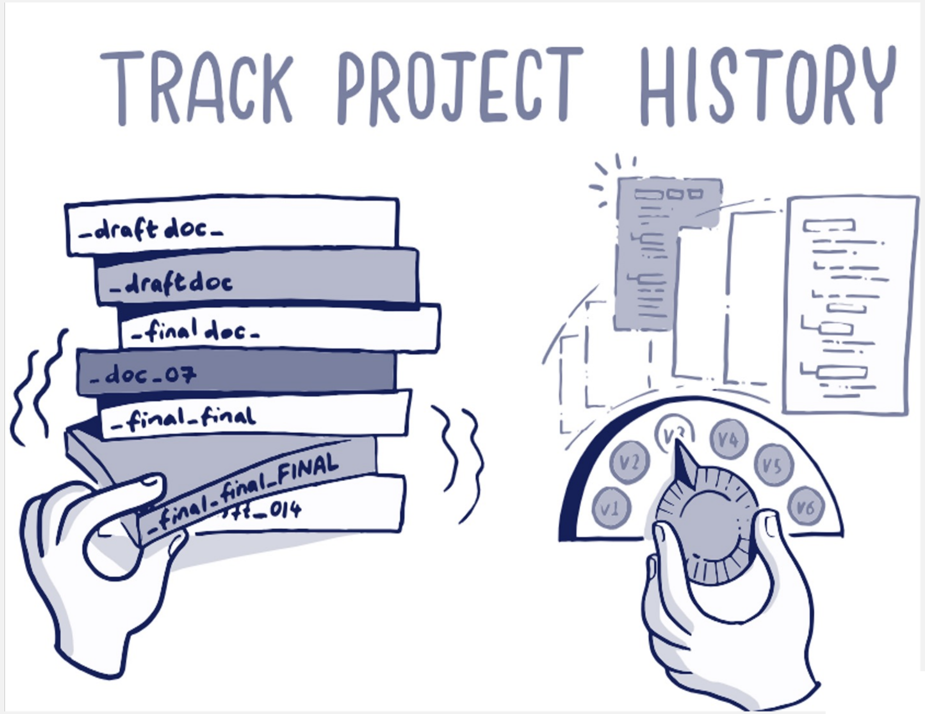
Version Control

WHAT IS VERSION CONTROL?

Version control, also referred to as source control, manages and tracks changes made to software code.

- Version control systems are software tools aiding software teams in managing source code alterations over time.
- Beneficial for DevOps teams, facilitating reduced development time and increased deployment success rates.
- Developers can revert to earlier versions of the code to rectify mistakes, minimizing disruption to team members.

Example *Version control in everyday use include Google Docs' "Version History" and Microsoft Office's "Track Changes" features. You might prefer saving multiple copies of a file and labeling them "v1", "v2", etc*



Version Control

WHY IS VERSION CONTROL HELPFUL FOR CODING?

When building software, developers frequently and simultaneously update the code to add features and fix bugs. It wouldn't make sense to make these changes to the source code directly, since any issues would affect users.

- Instead, developers work with their own copies of the code, then – after the code has been thoroughly tested – add this code to the main codebase.
- Collaborative coding can become chaotic without a system to merge contributions, track changes, and store versions, essential for troubleshooting and restoring previous iterations.
- That's especially true if there's no way to combine everyone's contributions into one unified codebase or see who contributed what changes.
- This is helpful when something breaks and the developers need to backtrack and restore a previous version.

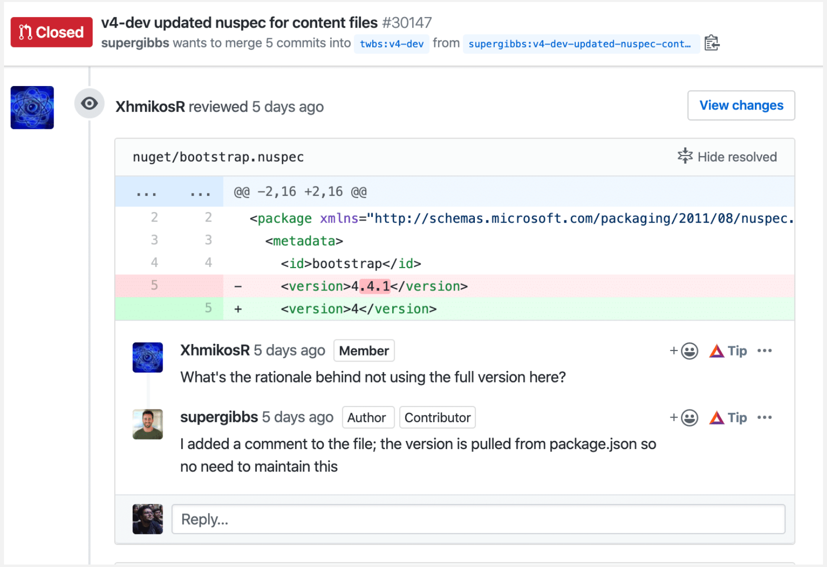
That's what Git is for ...!

Version Control

WHY IS VERSION CONTROL HELPFUL FOR CODING?

When a developer wants to make a change to a piece of software, they:

1. Download their copy of the source code from its central storage location (called a repository) to their local system
2. Make modifications safely to their copy
3. Merge their revised copy back with the source files in the repository
4. Add comments explaining the changes



Closed v4-dev updated nuspec for content files #30147
supergibbs wants to merge 5 commits into twbs:v4-dev from supergibbs:v4-dev-updated-nuspec-cont...

XhmikosR reviewed 5 days ago — View changes

nuget/bootstrap.nuspec — Hide resolved

```
@@ -2,16 +2,16 @@
  <package xmlns="http://schemas.microsoft.com/packaging/2011/08/nuspec.
    <metadata>
      <id>bootstrap</id>
-     <version>4.4.1</version>
+     <version>4</version>
```

XhmikosR 5 days ago Member
What's the rationale behind not using the full version here?

supergibbs 5 days ago Author Contributor
I added a comment to the file; the version is pulled from package.json so no need to maintain this

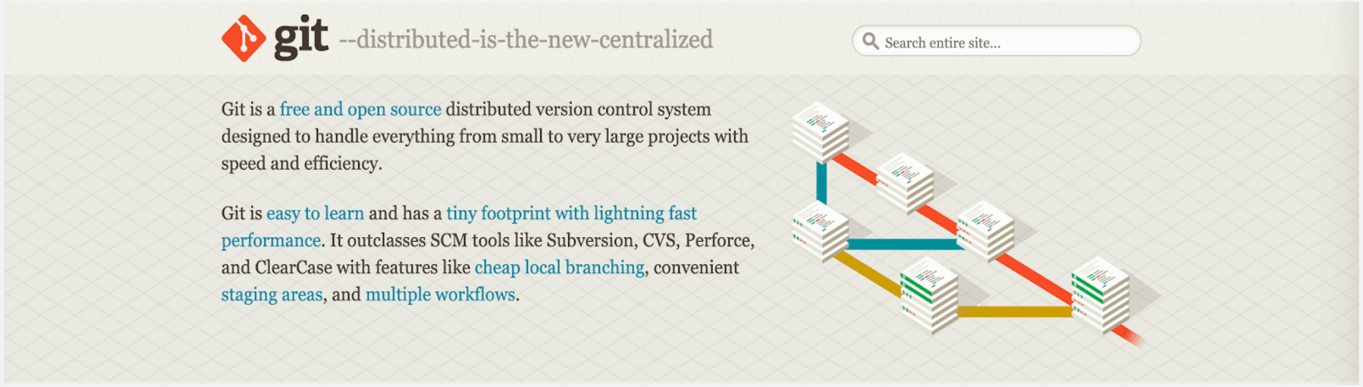
Reply...

What Is Git?

Git is open-source version control software created by Linus Torvalds in 2005, used for managing and tracking file revisions. You can use Git with any file type, but it's most often used for tracking code files.

Specifically, Git is a distributed version control system, which means that the entire codebase and history is available on every developer's computer, which allows for easy branching and merging.

According to a Stack Overflow developer survey, over 87% of developers use Git.



git --distributed-is-the-new-centralized

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

What is Git?

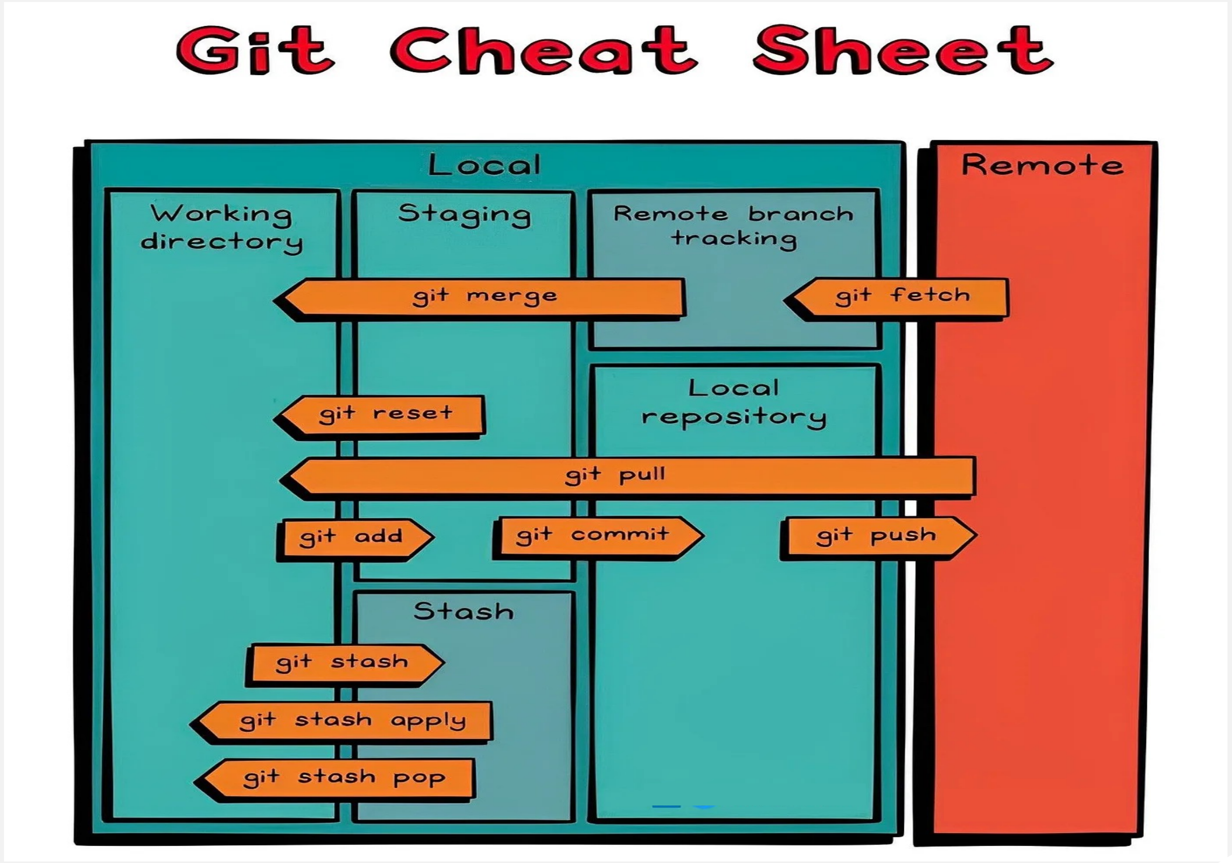
HOW TO INSTALL GIT?

- **Installing on Windows**

  Just go to https://git-scm.com/download/win and the download will start automatically.

- **Installing on macOS**

  A macOS Git installer is maintained and available for download at the Git website, at https://git-scm.com/download/mac.



What is Git?

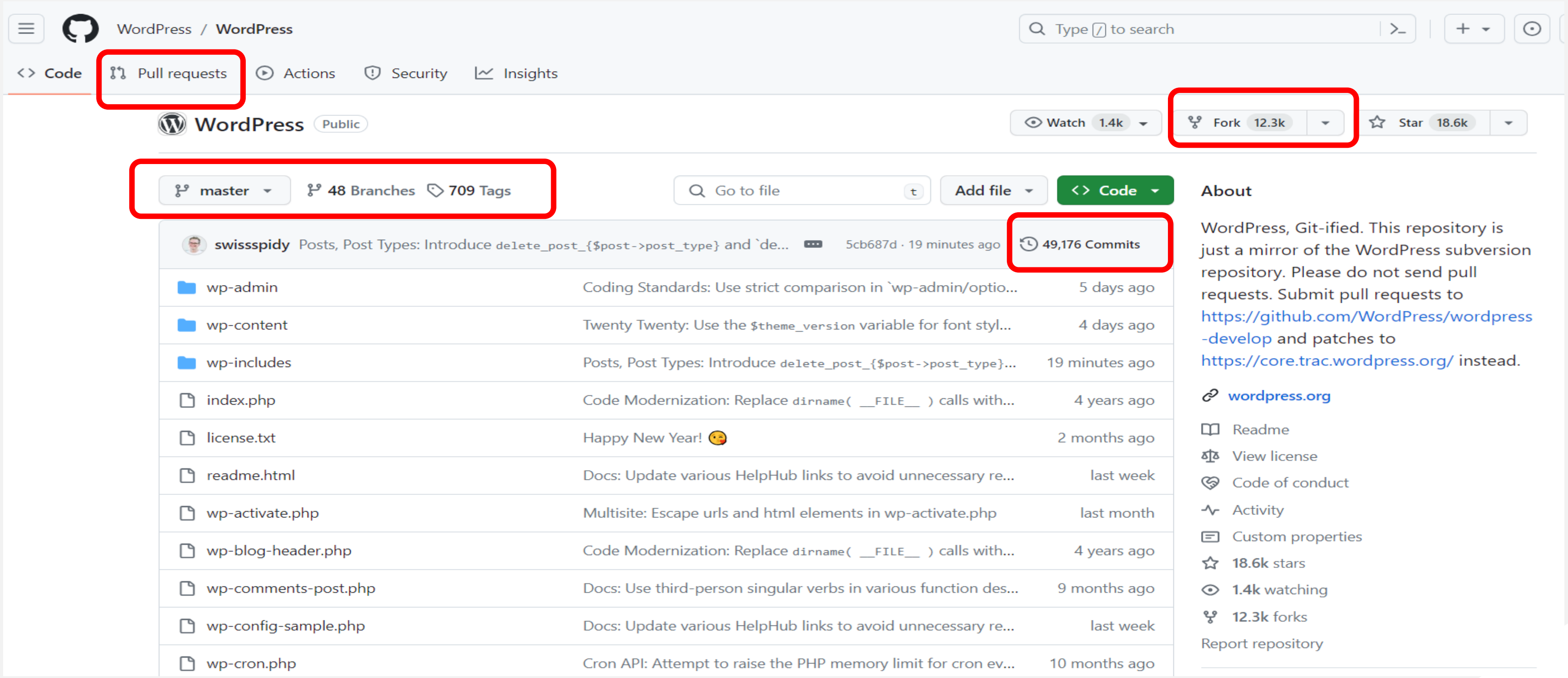
HOW TO INSTALL GIT?

There are a few terms you'll want to familiarize yourself with as you start using the software:

- **Repository:** The file location where your project is stored.
- **Commit:** The command used to save new changes to your project in the repository.
- **Stage:** Before you can commit changes in Git, you need to stage them - this gives you the chance to prepare your code before formally adding it to your project.
- **Branch:** The part of your project you're actively developing

Exploring the GitHub Interface

To give you a basic understanding of what the GitHub interface looks like, here's the WordPress source code [hosted at a GitHub](#) [repository](#):



swissspidy Posts, Post Types: Introduce `delete_post_{$post->post_type}` and `de... 5cb687d · 19 minutes ago 49,176 Commits

wp-admin	Coding Standards: Use strict comparison in `wp-admin/optio...	5 days ago
wp-content	Twenty Twenty: Use the `$theme_version` variable for font styl...	4 days ago
wp-includes	Posts, Post Types: Introduce `delete_post_{$post->post_type}`...	19 minutes ago
index.php	Code Modernization: Replace `dirname( __FILE__ )` calls with...	4 years ago
license.txt	Happy New Year! 😘	2 months ago
readme.html	Docs: Update various HelpHub links to avoid unnecessary re...	last week
wp-activate.php	Multisite: Escape urls and html elements in wp-activate.php	last month
wp-blog-header.php	Code Modernization: Replace `dirname( __FILE__ )` calls with...	4 years ago
wp-comments-post.php	Docs: Use third-person singular verbs in various function des...	9 months ago
wp-config-sample.php	Docs: Update various HelpHub links to avoid unnecessary re...	last week
wp-cron.php	Cron API: Attempt to raise the PHP memory limit for cron ev...	10 months ago

About

WordPress, Git-ified. This repository is just a mirror of the WordPress subversion repository. Please do not send pull requests. Submit pull requests to https://github.com/WordPress/wordpress-develop and patches to https://core.trac.wordpress.org/ instead.

wordpress.org

- Readme
- View license
- Code of conduct
- Activity
- Custom properties
- 18.6k stars
- 1.4k watching
- 12.3k forks

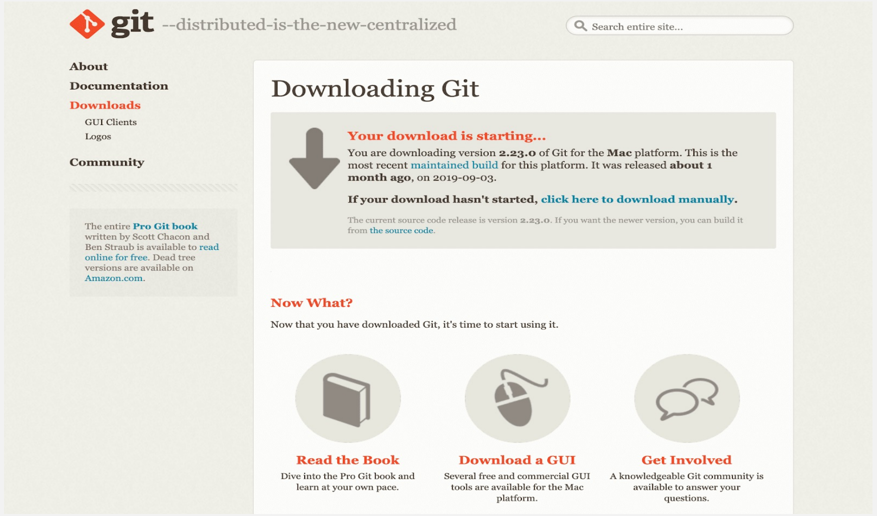
Report repository

Getting Started With GitHub

In order to use git and GitHub together for version control and collaboration, there are a few steps you'll need to take.

Step 1: Install git and Add a Repository

First, download the git software for your Operating System (OS)



The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloading Git

Your download is starting...

You are downloading version **2.23.0** of Git for the **Mac** platform. This is the most recent maintained build for this platform. It was released **about 1 month ago**, on 2019-09-03.

If your download hasn't started, click here to download manually.

The current source code release is version **2.23.0**. If you want the newer version, you can build it from the source code.

Now What?

Now that you have downloaded Git, it's time to start using it.

Read the Book
Dive into the Pro Git book and learn at your own pace.

Download a GUI
Several free and commercial GUI tools are available for the Mac platform.

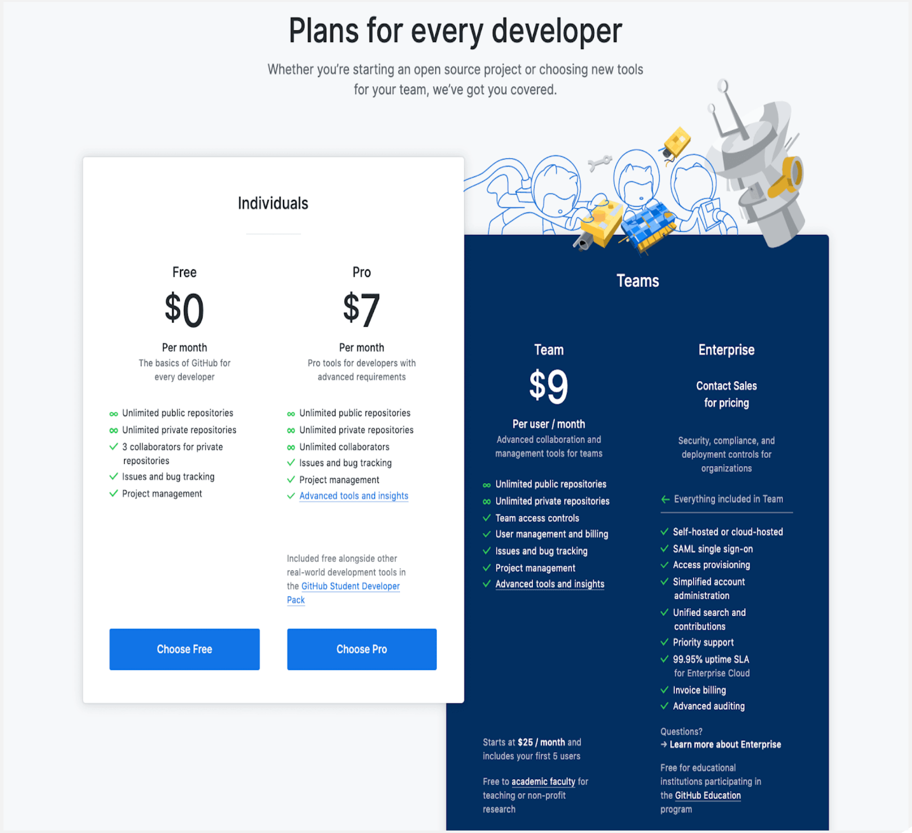
Get Involved
A knowledgeable Git community is available to answer your questions.

Getting Started With GitHub

Step 2: Create a GitHub Account

Next, you'll need a GitHub account. You can sign up for one for free or invest in a paid plan:

- **Free** account works well for new developers looking to hone their skills.
- **Pro** plan is better suited to freelancers and advanced coders, while agencies will want to invest in a team plan in order to access more project management and communication tools.



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

$0

Per month

The basics of GitHub for every developer

- ∞ Unlimited public repositories
- ∞ Unlimited private repositories
- ✓ 3 collaborators for private repositories
- ✓ Issues and bug tracking
- ✓ Project management

Choose Free

Pro

$7

Per month

Pro tools for developers with advanced requirements

- ∞ Unlimited public repositories
- ∞ Unlimited private repositories
- ∞ Unlimited collaborators
- ✓ Issues and bug tracking
- ✓ Project management
- ✓ Advanced tools and insights

Included free alongside other real-world development tools in the GitHub Student Developer Pack

Choose Pro

Teams

Team

$9

Per user / month

Advanced collaboration and management tools for teams

- ∞ Unlimited public repositories
- ∞ Unlimited private repositories
- ✓ Team access controls
- ✓ User management and billing
- ✓ Issues and bug tracking
- ✓ Project management
- ✓ Advanced tools and insights

Starts at **$25 / month** and includes your first 5 users

Free to academic faculty for teaching or non-profit research

Enterprise

Contact Sales for pricing

Security, compliance, and deployment controls for organizations

← Everything included in Team

- ✓ Self-hosted or cloud-hosted
- ✓ SAML single sign-on
- ✓ Access provisioning
- ✓ Simplified account administration
- ✓ Unified search and contributions
- ✓ Priority support
- ✓ 99.95% uptime SLA for Enterprise Cloud
- ✓ Invoice billing
- ✓ Advanced auditing

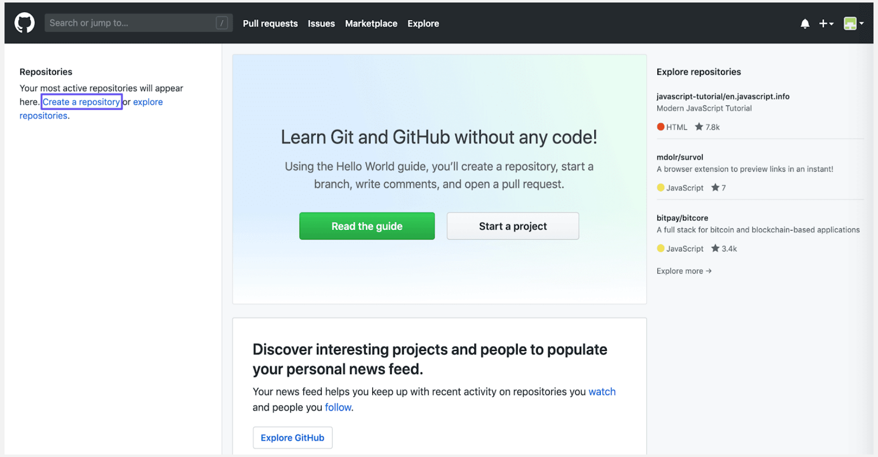
Questions?
→ Learn more about Enterprise

Free for educational institutions participating in the GitHub Education program

Getting Started With GitHub

Step 3: Add a GitHub Repository to Your Account

After you've created and set up your account, you'll need to create a repository in GitHub where you can store your project when you move it over from git



Your most active repositories will appear here. Create a repository or explore repositories.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

javascript-tutorial/en.javascript.info
Modern JavaScript Tutorial
HTML ★ 7.8k

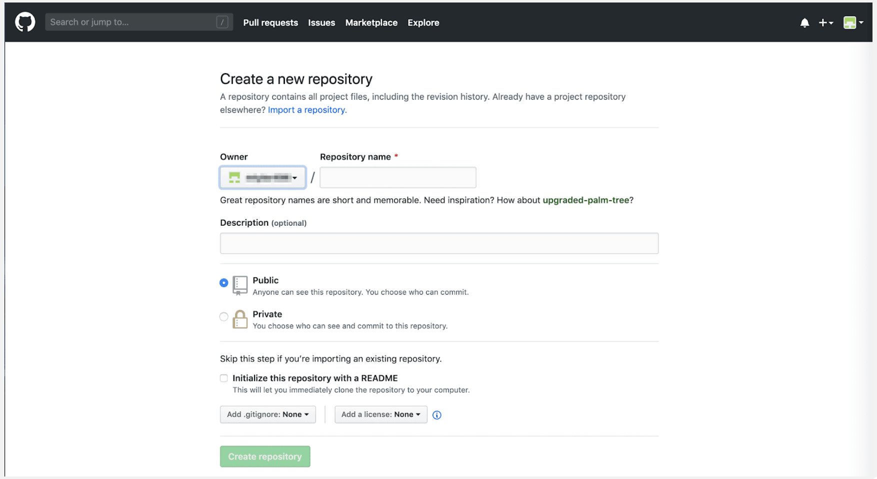
mdolr/survol
A browser extension to preview links in an instant!
JavaScript ★ 7

bitpay/bitcore
A full stack for bitcoin and blockchain-based applications
JavaScript ★ 3.4k

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch and people you follow.

Then, you'll need to choose a name for your repository:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner / Repository name *

Great repository names are short and memorable. Need inspiration? How about upgraded-palm-tree?

Description (optional)

[✓ Public] Anyone can see this repository. You choose who can commit.

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

Skip this step if you're importing an existing repository.

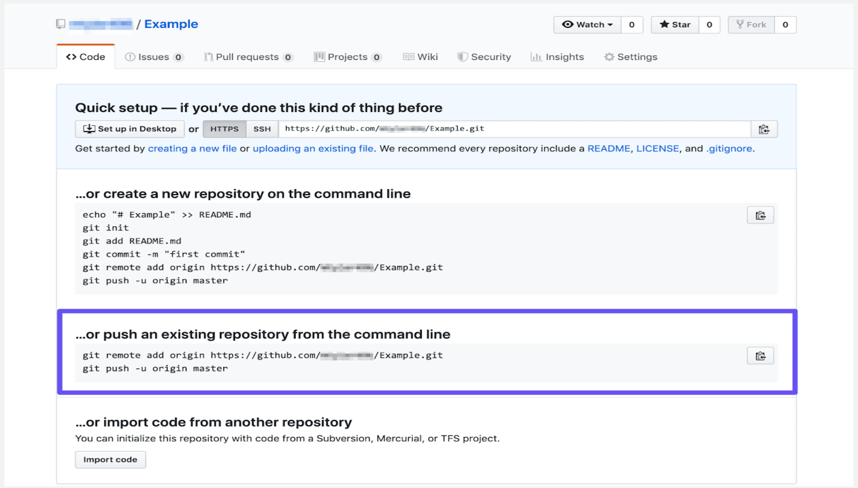
☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None

Getting Started With GitHub

Step 4: Push a Repository to GitHub

Next, you'll have the option to add code to your repository in a few different ways.



Quick setup — if you've done this kind of thing before

https://github.com/[redacted]/Example.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Example" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/[redacted]/Example.git
git push -u origin master
```

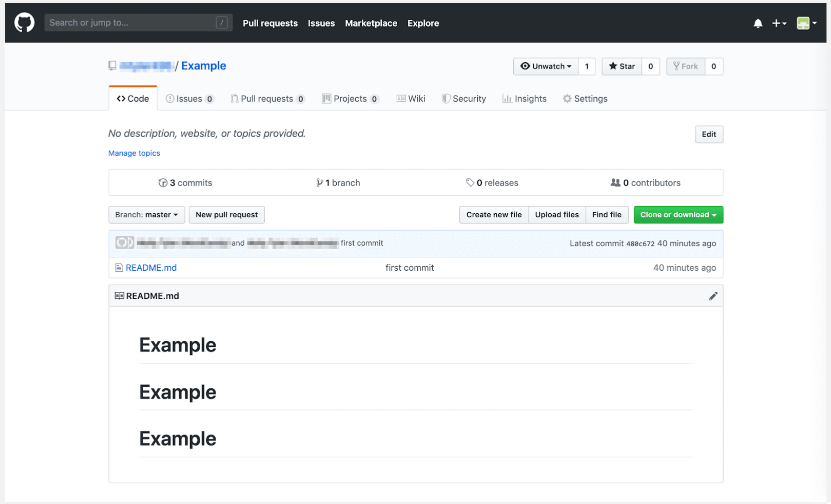
...or push an existing repository from the command line

```
git remote add origin https://github.com/[redacted]/Example.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Once that's done, refresh your GitHub page:



No description, website, or topics provided.

[redacted] and [redacted] first commit — Latest commit 480c672 40 minutes ago

README.md — first commit — 40 minutes ago

README.md

Example

Example

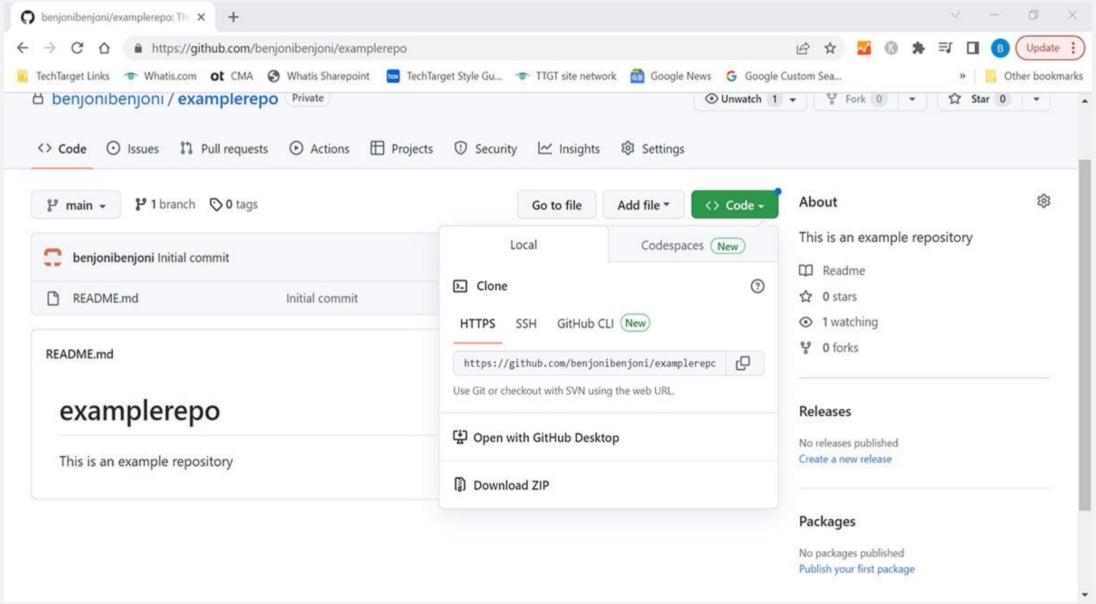
Example

Getting Started With GitHub

Step 5: Create a local copy of your repository.

You'll now create a local copy of your GitHub repository (or "clone" your repository) where you'll edit your files and push your changes.

*On your main repository page, click the green **Code** button, then copy the HTTPS URL of your repository.*

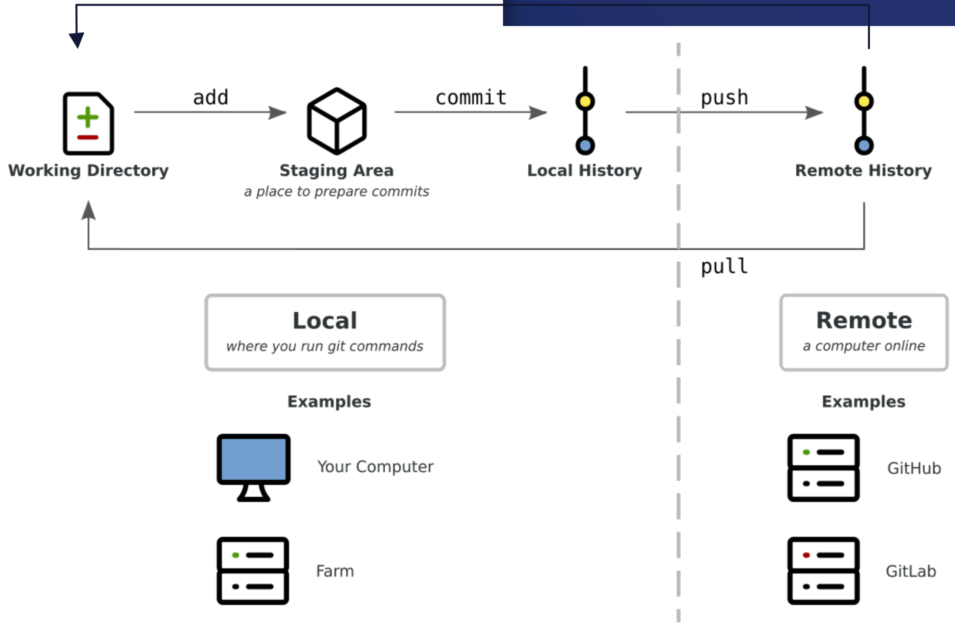


Getting Started With GitHub

Step 6: Local Updates

Open your terminal and navigate to the directory you want to place your repository copy.

```
# Clone the remote repository
git clone <remote_repository_URL>

# Change directory to the cloned repository
cd <repository_name>

# Make adjustments to the README file (you can
use any text editor you prefer)
echo "Your adjustments here" >> README.md

# Add the README file to the staging area
git add README.md

# Commit the changes
git commit -m "Adjusted README file"

# Push the changes to the remote repository
git push origin master
```

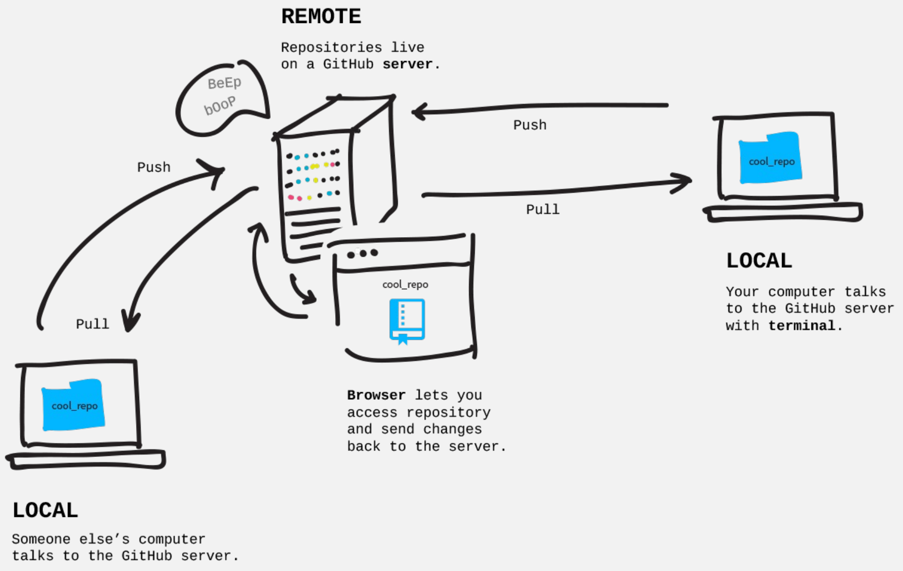


Getting Started With GitHub

Step 7: Pull Your Changes Back to git

While you can see all the changes you and others have made to your project on GitHub, the platform doesn't have direct access to your computer's files.

- In order to keep your project up-to-date on your computer, you'll need to pull your edits via git.
- To do so, simply enter git pull origin master into your command-line interface. This should update your files so that everything is in sync across all iterations of your project.



REMOTE
Repositories live on a GitHub **server**.
BeEp bOoP
Push
Pull
Push
Pull
cool_repo
cool_repo
cool_repo
LOCAL
Your computer talks to the GitHub server with **terminal**.
Browser lets you access repository and send changes back to the server.
LOCAL
Someone else's computer talks to the GitHub server.

Git cheat sheet

Most common Git commands.

Create a Repository

From scratch -- Create a new local repository
```
$ git init [project name]
```
Download from an existing repository
```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed
```
$ git status
```
Show the changes to files not yet staged
```
$ git diff
```
Show the changes to staged files
```
$ git diff --cached
```
Show all staged and unstaged file changes
```
$ git diff HEAD
```
Show the changes between two commit ids
```
$ git diff commit1 commit2
```
List the change dates and authors for a file
```
$ git blame [file]
```
Show the file changes for a commit id and/or file
```
$ git show [commit]:[file]
```
Show full change history
```
$ git log
```
Show change history for file/directory including diffs
```
$ git log -p [file/directory]
```

Working with Branches

List all local branches
```
$ git branch
```
List all branches, local and remote
```
$ git branch -av
```
Switch to a branch, my_branch, and update working directory
```
$ git checkout my_branch
```
Create a new branch called new_branch
```
$ git branch new_branch
```
Delete the branch called my_branch
```
$ git branch -d my_branch
```
Merge branch_a into branch_b
```
$ git checkout branch_b
$ git merge branch_a
```
Tag the current commit
```
$ git tag my_tag
```

Make a change

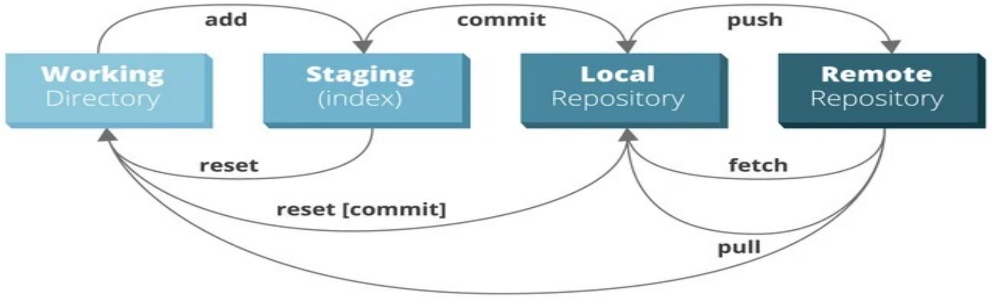
Stages the file, ready for commit
```
$ git add [file]
```
Stage all changed files, ready for commit
```
$ git add .
```
Commit all staged files to versioned history
```
$ git commit -m "commit message"
```
Commit all your tracked files to versioned history
```
$git commit -am "commit message"
```
Unstages file, keeping the file changes
```
$ git reset [file]
```
Revert everything to the last commit
```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)
```
$ git fetch
```
Fetch the latest changes from origin and merge
```
$ git pull
```
Fetch the latest changes from origin and rebase
```
$ git pull --rebase
```
Push local changes to the origin
```
$ git push
```

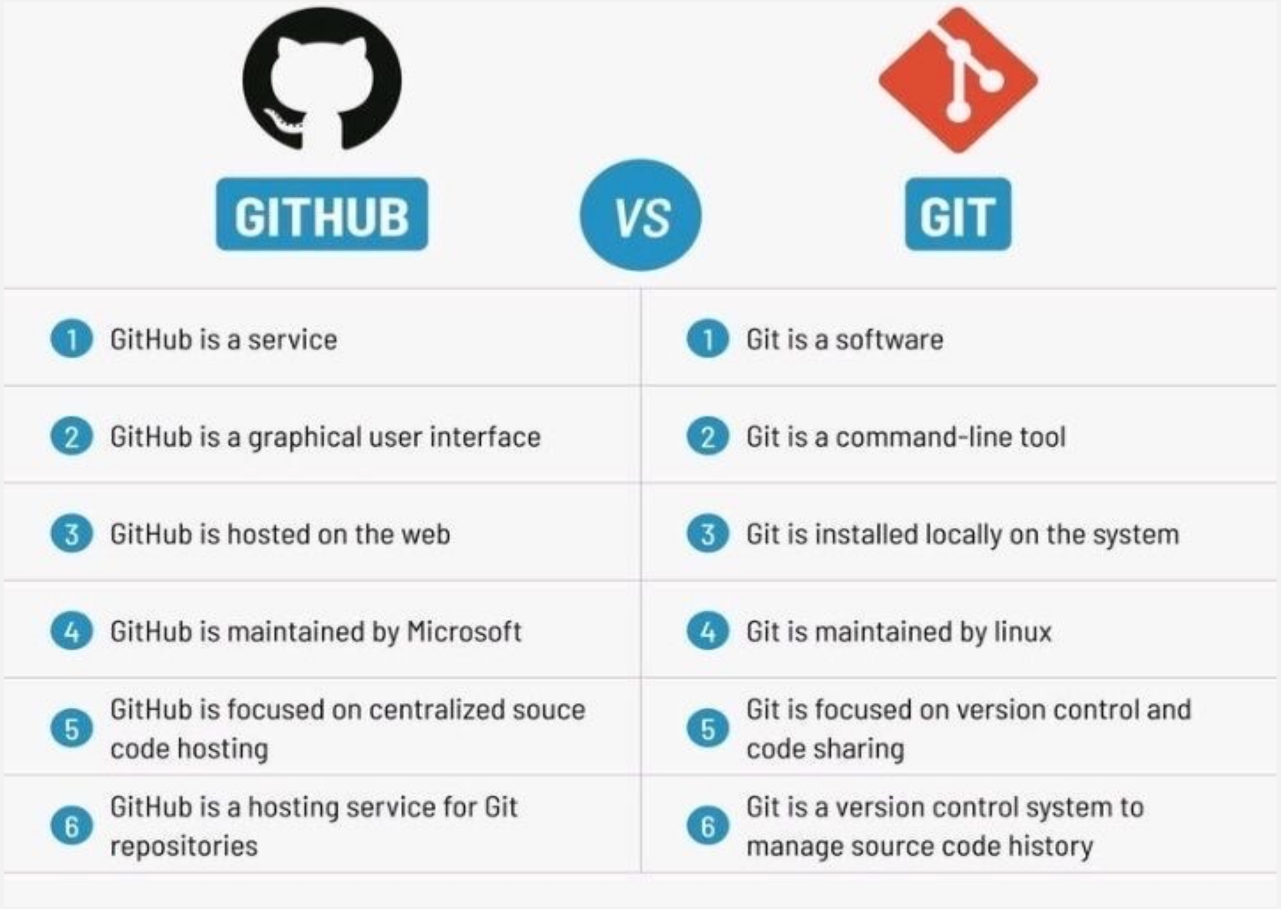
Finally!

When in doubt, use git help
```
$ git command --help
```
Or visit https://training.github.com/ for official GitHub training.



RECAP

Recap



GITHUB	GIT
1 GitHub is a service	1 Git is a software
2 GitHub is a graphical user interface	2 Git is a command-line tool
3 GitHub is hosted on the web	3 Git is installed locally on the system
4 GitHub is maintained by Microsoft	4 Git is maintained by linux
5 GitHub is focused on centralized souce code hosting	5 Git is focused on version control and code sharing
6 GitHub is a hosting service for Git repositories	6 Git is a version control system to manage source code history

Thank You

SDAIA
الهيئة السعودية للبيانات
والذكاء الاصطناعي
Saudi Data & AI Authority