

Вступ до Git

Що таке «система контролю версій»?

- спосіб керування файлами та каталогами
- відстежувати зміни в часі
- згадати попередні версії
- «керування джерелом» є підмножиною VCS.

Трохи історії контролю джерел

(1972) Source Code Control System (SCCS) - закритий код, частина UNIX

(1982) Revision Control System (RCS) - з відкритим кодом

(1986) Система паралельних версій (CVS) - з відкритим кодом

(2000) Apache Subversion (SVN) - відкритий код

(2000) BitKeeper SCM

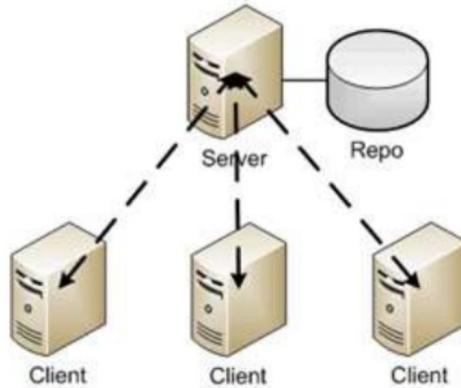
- із закритим кодом, пропрієтарний, використовується з керуванням вихідним кодом ядра Linux
- безкоштовно до 2005 року
- розподілений контроль версій

Розподілений контроль версій

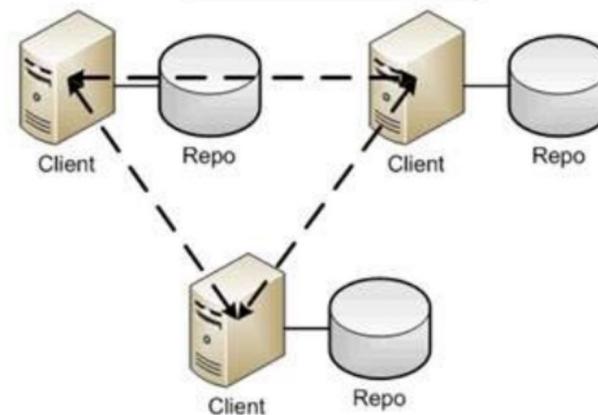
Немає центрального сервера

Кожен розробник є клієнтом, сервером і репозиторієм

Traditional



Distributed





Що таке git?

Що таке git?

- створено Лінусом Торвальдсом, квітень 2005 року
- заміна BitKeeper для керування змінами ядра Linux
- програма керування версіями командного рядка
- використовує контрольні суми для забезпечення цілісності даних
- розподілений контроль версій (наприклад, BitKeeper)
- кросплатформенність (включаючи Windows!)
- з відкритим кодом, безкоштовно



Розподілений контроль версій Git

- «Якщо вас не розповсюджують, ви не варти використання». – Linus Torvalds
- немає необхідності підключатися до центрального сервера
- може працювати без підключення до Інтернету
- жодної точки відмови
- розробники можуть працювати незалежно та пізніше об'єднати свою роботу
- кожна копія репозиторію Git може служити або сервером, або клієнтом (і має повну історію!)
- Git відстежує зміни, а не версії
- Купа дрібних наборів плаває навколо

Git для мене?

- Люди, які переважно працюють із вихідним кодом
- Будь-хто, хто хоче відстежувати зміни (особливо зміни в текстових файлах) - переглянути історію змін - будь-хто, хто хоче поділитися, об'єднати зміни
- Усі, хто не боїться інструментів командного рядка

Найбільш популярні мови, які використовуються з Git

- HTML
- CSS
- Javascript
- Python
- ASP
- Scala
- Shell scripts
- PHP
- Ruby
- Ruby on Rails
- Perl
- Java
- C
- C++
- C#
- Objective C
- Haskell
- CoffeeScript
- ActionScript

Не настільки корисний для зображень, фільмів, музики... та файлів, які потрібно інтерпретувати (.pdf, .psd тощо)

Як мені це отримати?

<http://git-scm.com>



Підказка щодо встановлення Git

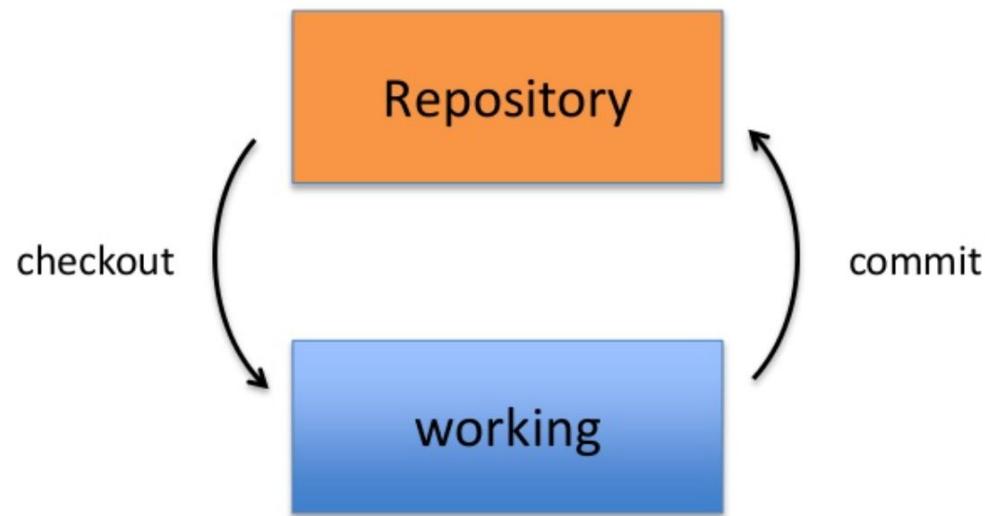
- Набагато краще налаштовувати для кожного користувача (замість глобальної, загальносистемної інсталяції)

Що таке репозиторій?

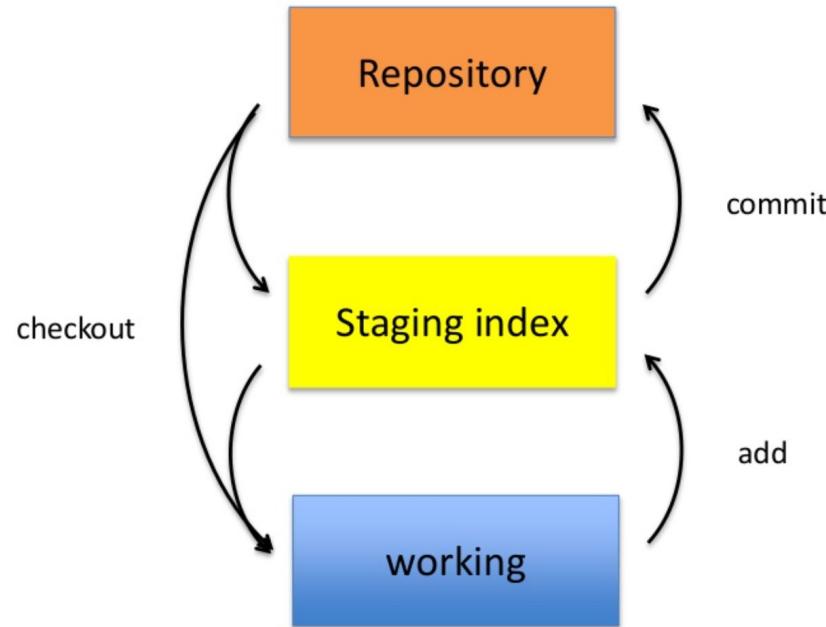
- “repo” = репозиторій
- зазвичай використовується для організації одного проекту
- сховища можуть містити папки та файли, зображення, відео, електронні таблиці та набори даних – усе, що потрібно вашому проекту

Двохдерева архітектура

інші VCS



Git використовує тридеревоподібну архітектуру



Простий робочий процес Git

1. Ініціалізація нового проекту в каталозі:

```
git init
```

```
[ dolanmi L02029756 ~/Desktop ]$ mkdir new_project
[ dolanmi L02029756 ~/Desktop ]$ cd new_project/
[ dolanmi L02029756 ~/Desktop/new_project ]$ git init
Initialized empty Git repository in /Users/dolanmi/Desktop/new_project/.git/
[ dolanmi L02029756 ~/Desktop/new_project ]$ █
```

2. Додайте файл за допомогою текстового редактора в каталог

3. Додайте всі зміни, внесені до каталогу:

```
git add
```

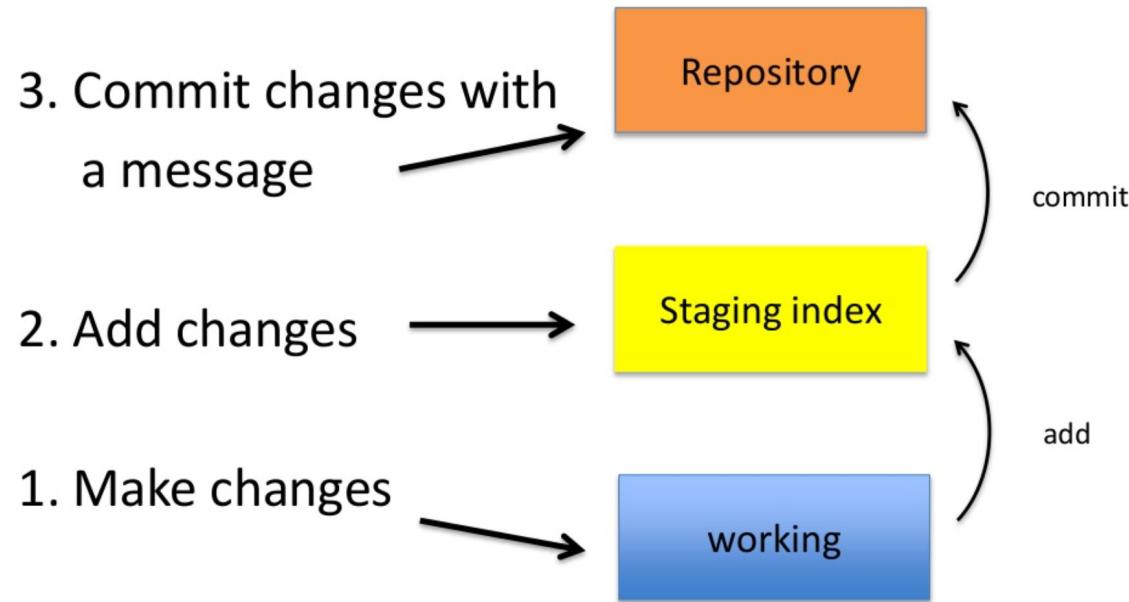
Простий робочий процес Git

4. Зафіксуйте зміни в репо:

`git commit -m «важливе повідомлення тут»`

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git add .
[ dolanmi L02029756 ~/Desktop/new_project ]$ git commit -m "Add message to file.txt"
[master (root-commit) 1a7e4a5] Add message to file.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
[ dolanmi L02029756 ~/Desktop/new_project ]$ █
```

Після ініціалізації нового сховища git...



Примітка щодо повідомлень фіксації

- Скажи, що він робить (теперішній час)
- Короткий опис в одному рядку, після якого йде пробіл, а потім більш повний опис
- Тримайте рядки до <= 72 символів
- Допоможе номер квитка або помилки

Хороші і погані приклади

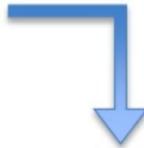
Поганий: “Виправлення друкарських помилок”

Хороший: “Додайте відсутні / в розділі CSS”

Поганий: “Оновлює таблицю. Наступного понеділка ми обговоримо це з Дарреллом.”

Bad: `git commit -m "Fix login bug"`

Good: `git commit -m`



Redirect user to the requested page after login

`https://trello.com/path/to/relevant/card`

Users were being redirected to the home page after login, which is less useful than redirecting to the page they had originally requested before being redirected to the login form.

- * Store requested path in a session variable
- * Redirect to the stored location after successfully logging in the user

Як побачити, що зроблено?

```
git log
```

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git log
commit 6c40ffd9ba4ba1567eb6fcd3715f12a15b0a678d
Author: mchldln <dolanmi@niaid.nih.gov>
Date:   Mon May 2 18:11:23 2016 -0400
```

Add message to text file

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ █
```

Контрольні суми, згенеровані алгоритмом шифрування SHA1

```
[ dolanmi L02029756 ~/Desktop/bcbb/portal_project/git/BCBBportalXI ]$ git log
commit f8c00639a649a122446040b15185cc09c4c5c71c
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Fri Apr 29 15:02:56 2016 -0400

    update headers

commit eb0cf49cc05786cbc7314982f06af5a9ad93149e
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Tue Apr 26 12:07:32 2016 -0400

    update name link and about page

commit 44c433a1794cefef211d5116568dcfbe67d518b2f
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Mon Apr 25 15:45:27 2016 -0400

    remove about, change font family in the name

commit 898be0093a995c08a7a4f99219abee255b94a874
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Fri Apr 22 09:30:49 2016 -0400

    updating header and sidenav bar

commit c5f689ed0b8c71582b3d301e2282f9e6472962c6
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Thu Apr 21 14:29:20 2016 -0400

    change the name to code

commit 4463ea2d1c75b80af9d2894feb2eb3ded7fe40c9
:
commit f8c00639a649a122446040b15185cc09c4c5c71c
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Fri Apr 29 15:02:56 2016 -0400

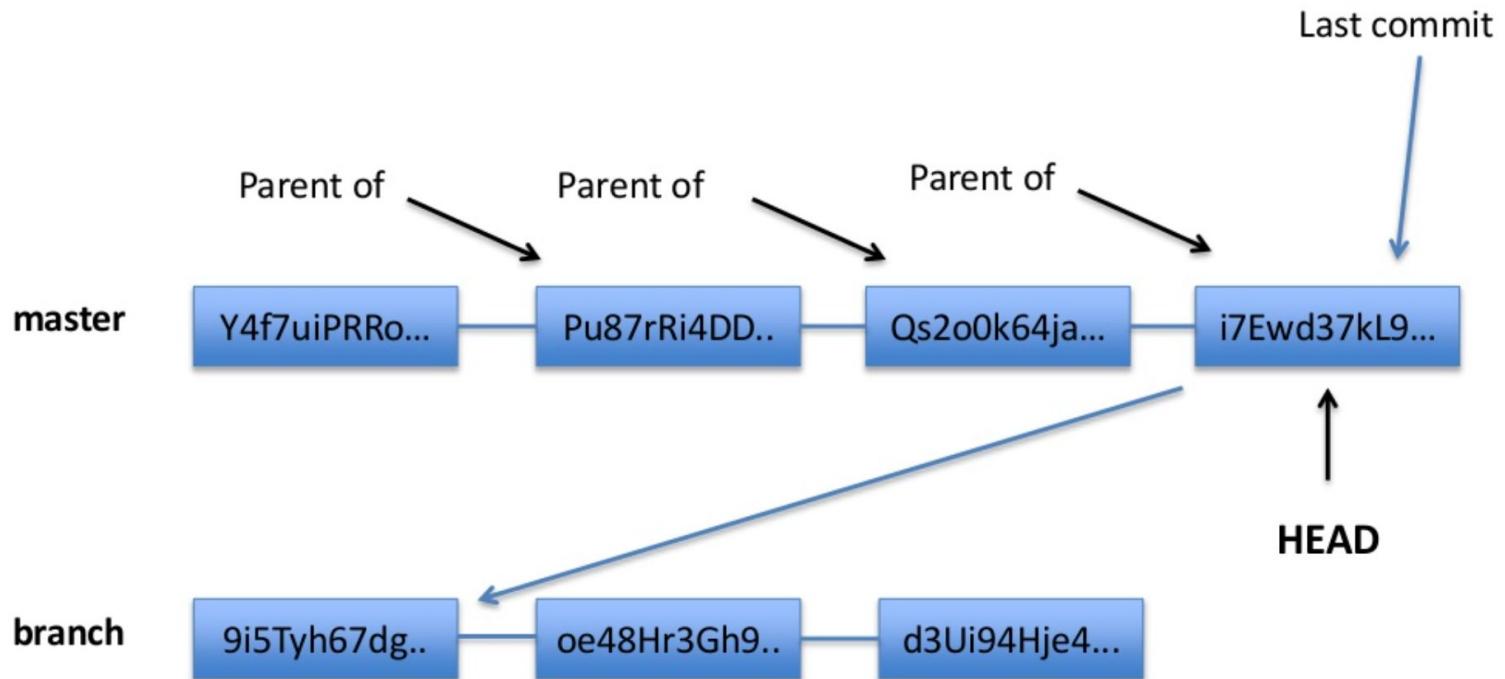
    update headers

commit eb0cf49cc05786cbc7314982f06af5a9ad93149e
Author: Yamil Boo <yamil.booirizarry@nih.gov>
Date:   Tue Apr 26 12:07:32 2016 -0400

    update name link and about page
```

Показник HEAD

- вказує на конкретну фіксацію в репо
- у міру створення нових комітів вказівник змінюється
- **HEAD завжди вказує на «підказку» поточної вилученої гілки в репозиторії**
- (не робочий каталог або проміжний індекс)
- останній стан репо (те, що було перевірено спочатку)
- HEAD вказує на батьківський елемент наступного коміту (де відбувається запис наступного коміту)



Які файли було змінено та де вони знаходяться в дереві трьох?

git status – дозволяє побачити, де знаходяться файли в схемі трьох дерев

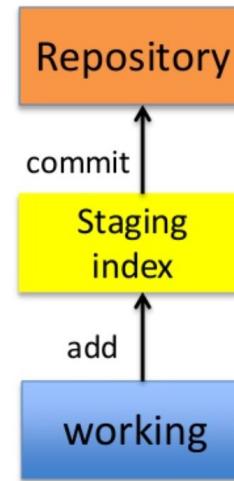
```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file.txt
```



Що змінилося в робочому каталогі?

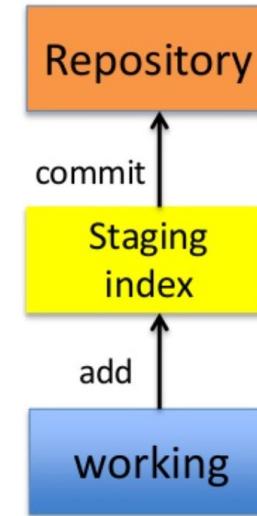
`git diff` – порівнює зміни файлів між сховищем і робочим каталогом

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git diff
diff --git a/file.txt b/file.txt
index 4e1c952..bd5fd23 100644
--- a/file.txt
+++ b/file.txt
@@ -1 +1 @@
-NIEHS is not great!
+NIEHS is great!
```

Line numbers in file
Removed
Added

Note: `git diff --staged` - compares staging index to repo

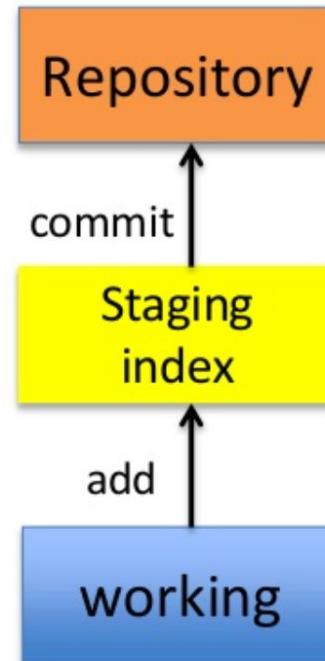
Note: `git diff filename` can be used as well



Видалення файлів із сховища

`git rm filename.txt`

- переміщує видалену зміну файлу до робочої області
- Недостатньо видалити файл у вашому робочому каталозі. Ви повинні зафіксувати зміни



Видалення файлів із сховища

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git add file.txt
[ dolanmi L02029756 ~/Desktop/new_project ]$ git commit -m "message"
[master (root-commit) 1edeeae] message
 1 file changed, 1 insertion(+)
  create mode 100644 file.txt
[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
nothing to commit, working directory clean
[ dolanmi L02029756 ~/Desktop/new_project ]$ git rm file.txt
rm 'file.txt'
[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    deleted:    file.txt

[ dolanmi L02029756 ~/Desktop/new_project ]$ git commit -m "delete file.txt"
[master c4f8073] delete file.txt
 1 file changed, 1 deletion(-)
  delete mode 100644 file.txt

[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
nothing to commit, working directory clean
```

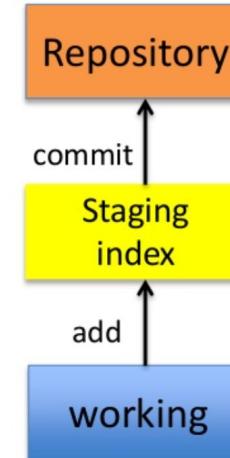
Переміщення (перейменування) файлів

git mv filename1.txt filename2.txt

```
[ dolanmi L02029756 ~/Desktop/new_project ]$ git mv file2.txt file1.txt
[ dolanmi L02029756 ~/Desktop/new_project ]$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

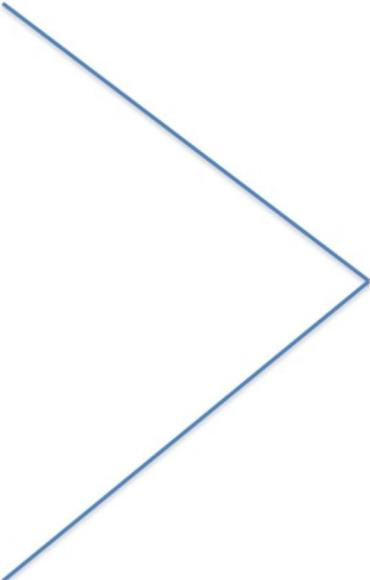
    renamed:   file2.txt -> file1.txt
```

Note: File file1.txt was committed to repo earlier.



Хороші новини!

- git init
- git status
- git log
- git add
- git commit
- git diff
- git rm
- git mv



75% of the time you'll be using
only these commands

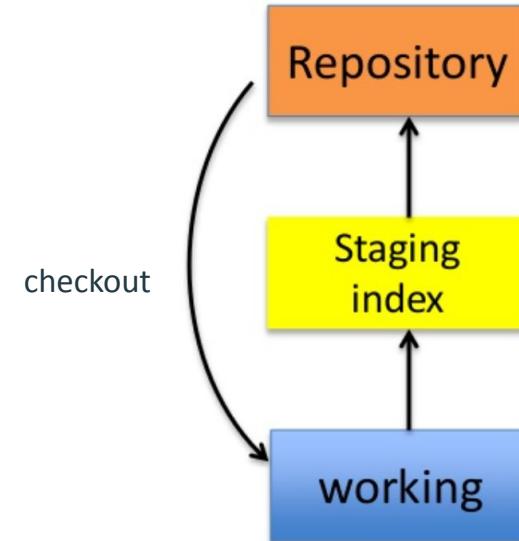
Що робити, якщо я хочу скасувати зміни, зроблені в робочому каталозі?

git checkout something

(де «щось» — це файл або ціла гілка)

- git checkout захопить файл із сховища
- Приклад: **git checkout -- file1.txt**

(«витягти файл «file1.txt» із поточної гілки»)



Що робити, якщо я хочу скасувати зміни, додані до робочої області?

`git reset HEAD filename.txt`

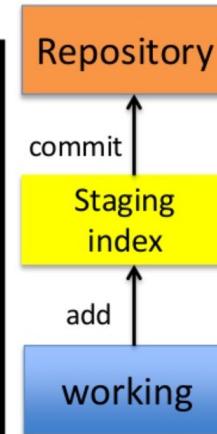
```
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ vi file4.txt
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git add .
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file4.txt

[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git reset HEAD file4.txt
Unstaged changes after reset:
M      file4.txt
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   file4.txt

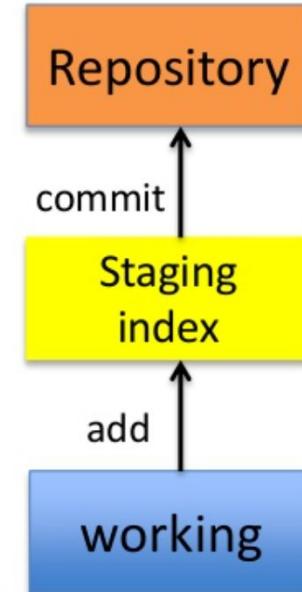
no changes added to commit (use "git add" and/or "git commit -a")
```



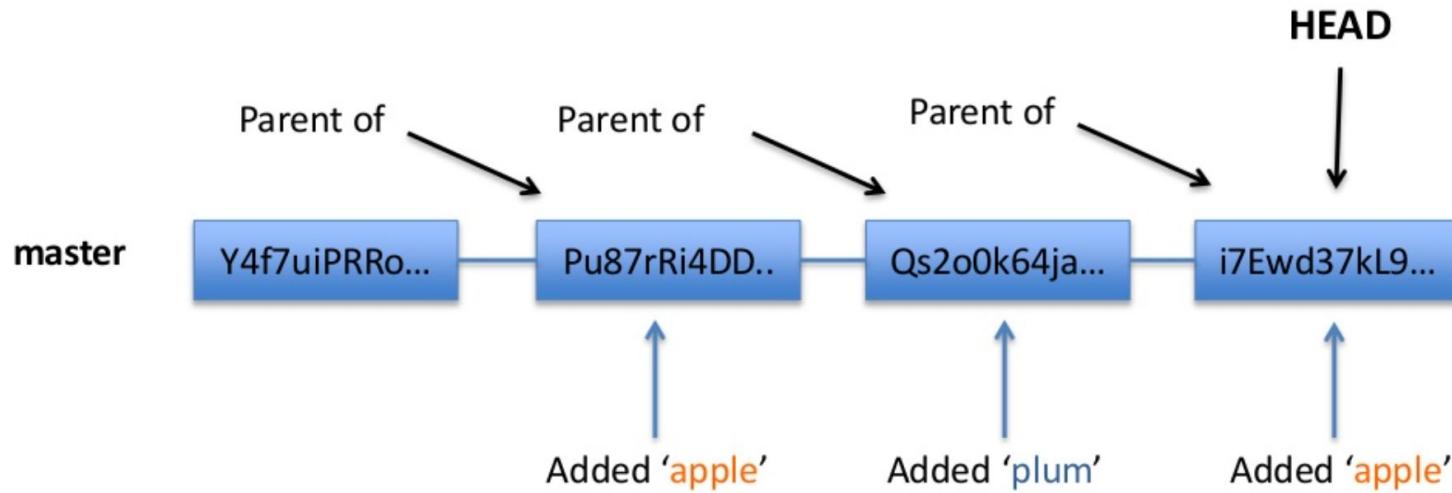
Що робити, якщо я хочу скасувати зміни, внесені до репо?

`git commit --amend -m "message"`

- дозволяє вносити зміни до останнього коміту
- будь-що в проміжній області буде змінено до останнього коміту



Примітка. Щоб скасувати зміни в старих комітах, зробіть новий коміт



Отримати старіші версії

```
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git log
commit 60f1c1a034fdcab4e8127d36556a881e7778c2ec
Author: mchldln <dolanmi@niaid.nih.gov>
Date:   Tue May 3 17:00:36 2016 -0400

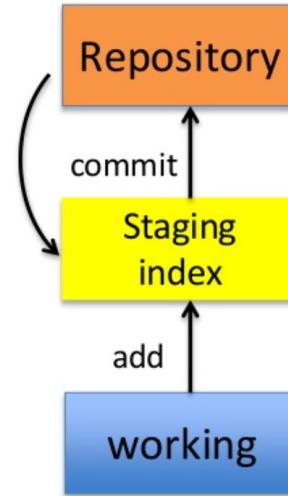
    another message yet

commit d685ff9a41a9eec62e6010827513e33ba1abc0d6
Author: mchldln <dolanmi@niaid.nih.gov>
Date:   Tue May 3 17:00:09 2016 -0400

    another message

commit 6e073c640928b1470f8443e594fb63063c87bcf7
Author: mchldln <dolanmi@niaid.nih.gov>
Date:   Tue May 3 14:25:38 2016 -0400

    message
```



git checkout 6e073c640928b -- filename.txt

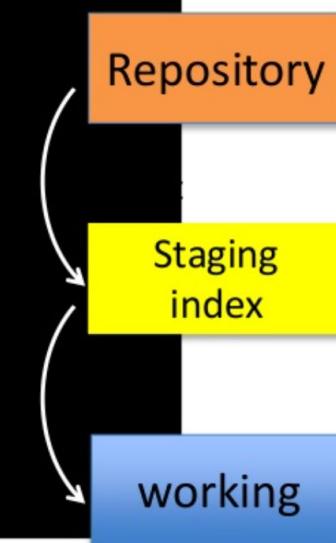
Примітка. Перевірка старіших комітів переміщує їх у проміжну область Робочий проміжний індекс Репозиторій додавання комітів

```
git checkout 6e073c640928b -- filename.txt
```

```
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git checkout 6e073c640928b -- file4.txt
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file4.txt

[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git diff
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git diff --staged
diff --git a/file4.txt b/file4.txt
index 56392a0..9c595a6 100644
--- a/file4.txt
+++ b/file4.txt
@@ -1 +1 @@
-temp temp temp 2
+temp
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git reset HEAD file4.txt
Unstaged changes after reset:
M      file4.txt
```



Які файли знаходяться в репо?

`git ls-tree tree-ish`

tree-ish – спосіб посилання на репо повний SHA, частина SHA, HEAD тощо

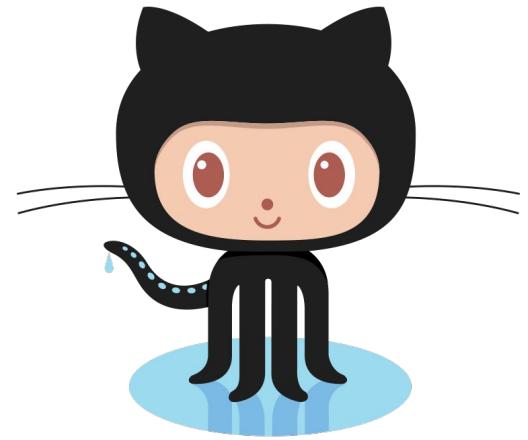
```
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git ls-tree HEAD
100644 blob 5626abf0f72e58d7a153368ba57db4c673c0e171      file1.txt
100644 blob f719efd430d52bcfc8566a43b2eb655688d38871      file2.txt
100644 blob a5648e79c58aab29ec5e45e99781edd7263e19e7      file3.txt
100644 blob 9c595a6fb7692405a5c4a10e1caf93d7a5bd9c37      file4.txt
040000 tree 6460ee80311f76a04b884e60f25400cf30b477b9      sub_dir
```

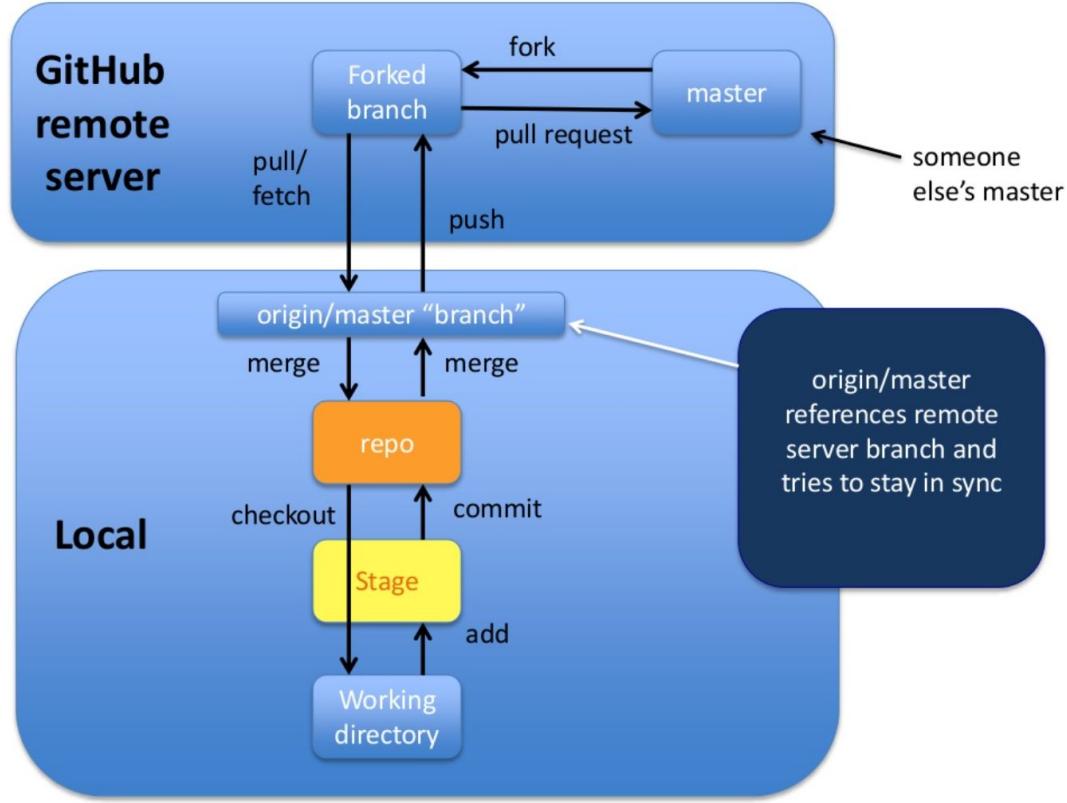
blob = file, tree = каталог

Що таке GitHub?

GitHub

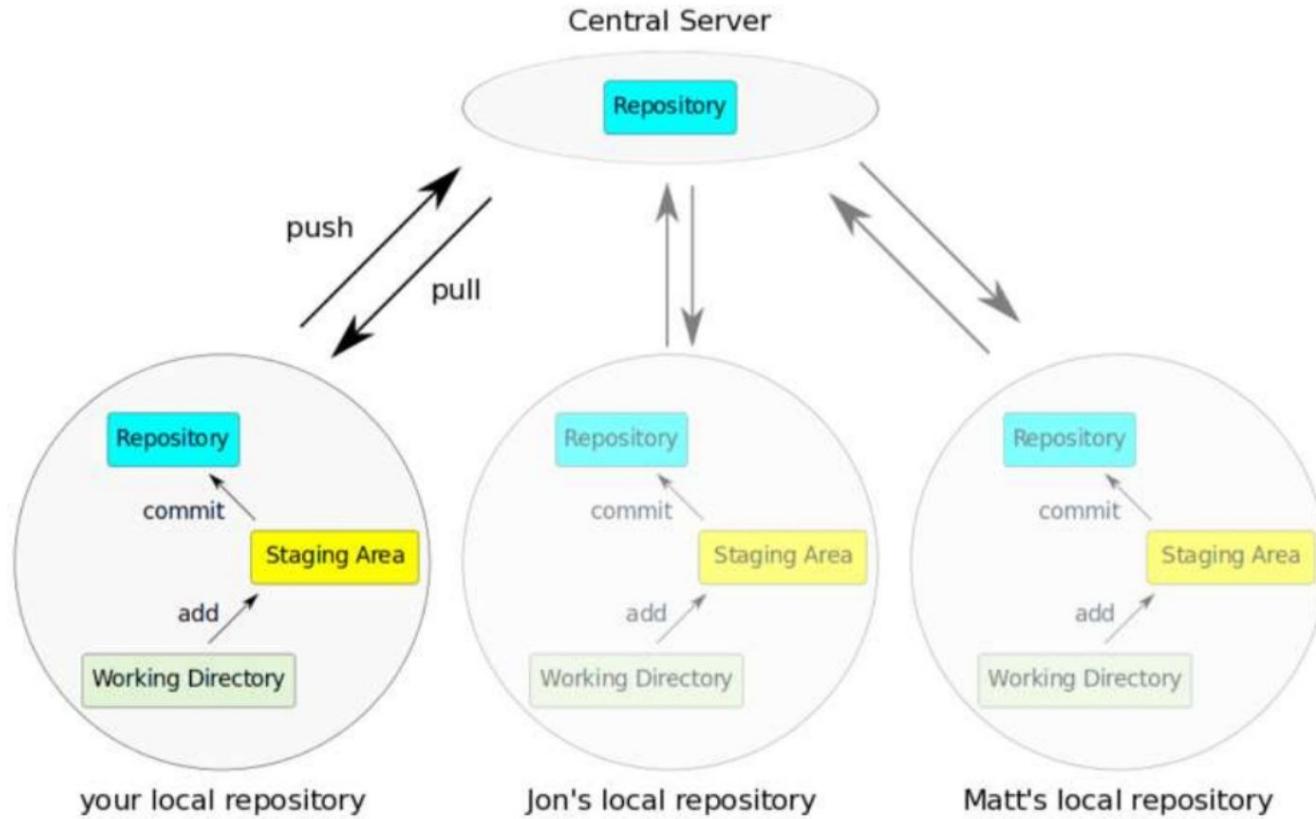
- платформа для розміщення сховищ коду git
- <http://github.com>
- запущений у 2008 році
- найпопулярніший хост Git
- дозволяє користувачам співпрацювати над проектами з будь-якого місця
- GitHub робить git соціальним!
- Безкоштовно





Важливо пам'ятати

Іноді розробники вирішують розмістити репо на GitHub як централізоване місце, де всі вносять зміни, але це не обов'язково має бути на GitHub



This repository Search

Pull requests Issues Gist

Unwatch 1 Star 0 Fork 113

Code Pull requests 0 Wiki Pulse Graphs Settings

DIY Skill – Edit

124 commits 1 branch 0 releases 36 contributors

Branch: master New pull request New file Upload files Find file HTTPS https://github.com/mchld1

This branch is 2 commits ahead of diy:master.

[Pull request](#) [Compare](#)

mchld1 Corrected typo Latest commit ecd0d3b 6 days ago

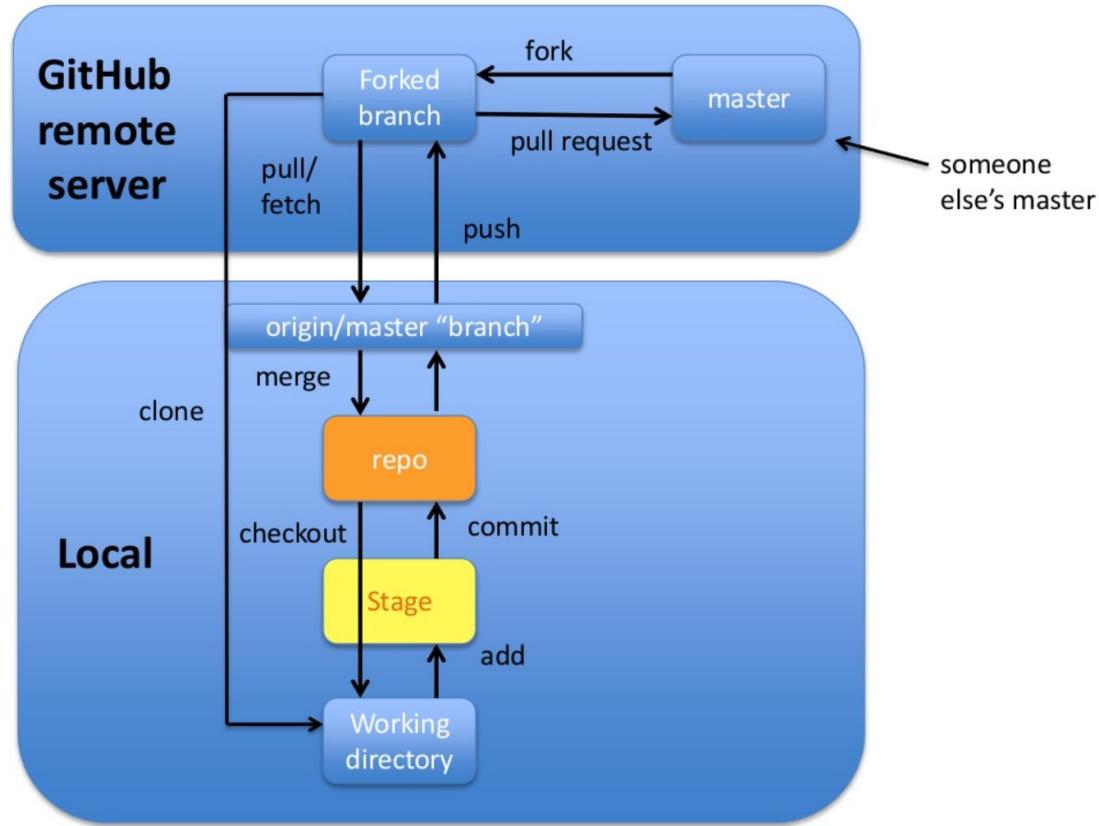
File	Commit Message	Time Ago
.gitignore	Add gitignore	3 years ago
README.md	added my name on README.md	2 years ago
collaborative-story.txt	Corrected typo	6 days ago
new-features.txt	Add lines to collaborative-story, new-features and ultimate-cookie vi...	2 years ago
script.md	Updated script.md	3 years ago
ultimate-cookie.txt	Add lines to collaborative-story, new-features and ultimate-cookie vi...	2 years ago

README.md

Копіювання (клонування) файлів із віддаленого репо на локальну машину

`git clone URL <new_dir_name>`

```
[dolanmil]$ git clone https://github.com/mchldln/open-sourcerer.git program_one
Cloning into 'program_one'...
remote: Counting objects: 294, done.
remote: Total 294 (delta 0), reused 0 (delta 0), pack-reused 294
Receiving objects: 100% (294/294), 45.83 KiB | 0 bytes/s, done.
Resolving deltas: 100% (149/149), done.
Checking connectivity... done.
[dolanmil]$ ls
program_one
[dolanmil]$ cd program_one/
[dolanmil]$ ls -aFlt
total 72
drwxrwxr-x  9 dolanmi  NIH\Domain Users   306 May  4 17:26 .
drwxrwxr-x 13 dolanmi  NIH\Domain Users  442 May  4 17:26 .git/
-rw-rw-r--  1 dolanmi  NIH\Domain Users    19 May  4 17:26 .gitignore
-rw-rw-r--  1 dolanmi  NIH\Domain Users   586 May  4 17:26 README.md
-rw-rw-r--  1 dolanmi  NIH\Domain Users 2938 May  4 17:26 collaborative-story.txt
-rw-rw-r--  1 dolanmi  NIH\Domain Users   138 May  4 17:26 new-features.txt
-rw-rw-r--  1 dolanmi  NIH\Domain Users 12984 May  4 17:26 script.md
-rw-rw-r--  1 dolanmi  NIH\Domain Users   192 May  4 17:26 ultimate-cookie.txt
drwxrwxr-x  3 dolanmi  NIH\Domain Users   102 May  4 17:26 ..
```



Як зв'язати мое локальне репо з віддаленим репо?

`git remote add <alias> <URL>`

Примітка. Це лише встановлює з'єднання... файли не копіюються/переміщуються

Примітка: так! Ви можете мати більше одного пульта, пов'язаного з вашим локальним каталогом!

До яких пультів я підключений?

`git remote`

Надсилання до віддаленого репо

```
git push local_branch_alias branch_name
```

```
[dolanmi]$ git push origin master
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 280 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To https://github.com/mchldln/open-sourcerer.git
  ecd0d3b..212432e  master -> master
```

Отримання з віддаленого репо

`git fetch remote_repo_name`

Отримання жодним чином не змінює ваш робочий каталог або будь-які зроблені вами коміти.

- Отримайте перед роботою
- Отримайте, перш ніж натискати
- Отримуйте часто

git merge потрібно виконати, щоб об'єднати отримані зміни в локальну гілку

Співпраця з Git

This is a test account for NIEHS. — Edit

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request New file Upload files Find file HTTPS https://github.com/mchld1 Download ZIP

mchldn Update README.md Latest commit 6d509e4 on Mar 28

README.md Update README.md a month ago

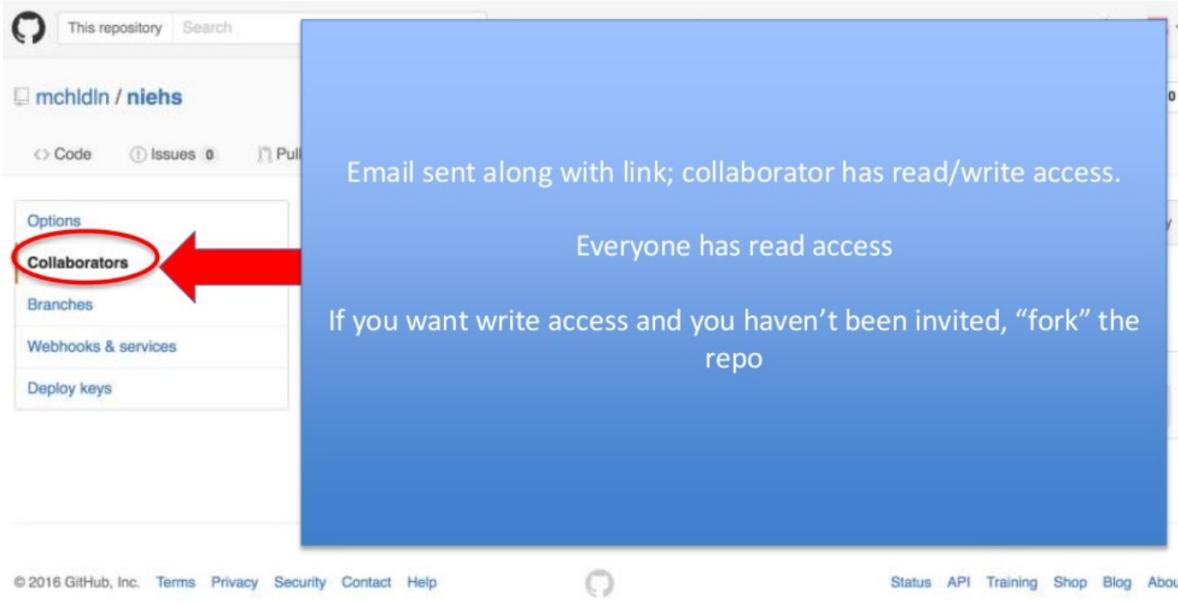
README.md

niehs

This is a test account for NIEHS. new text

A red arrow points to the 'Settings' button in the top navigation bar.

Співпраця з Git



This screenshot shows the GitHub repository settings for the user mchldn/niehs. A red arrow points from the text "Email sent along with link; collaborator has read/write access." to the "Collaborators" tab in the sidebar menu.

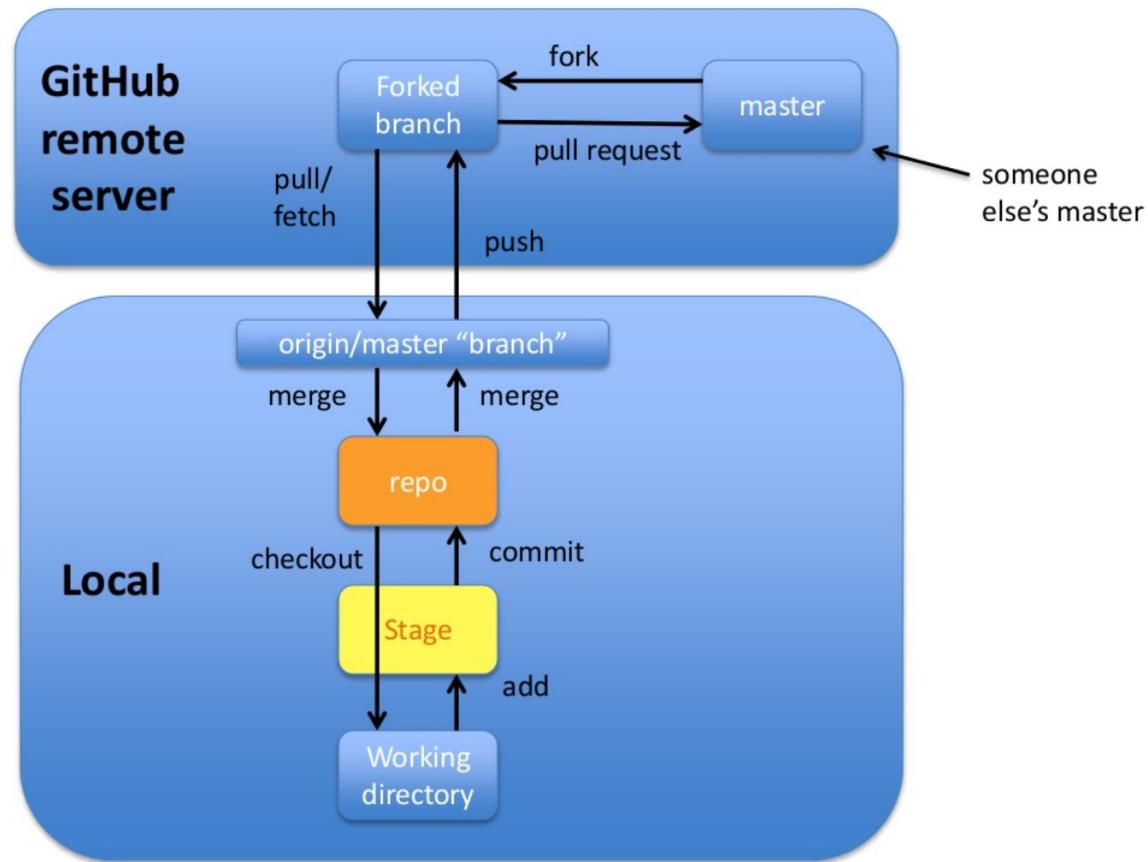
Email sent along with link; collaborator has read/write access.

Everyone has read access

If you want write access and you haven't been invited, "fork" the repo

© 2016 GitHub, Inc. Terms Privacy Security Contact Help

Status API Training Shop Blog About



GitHub Gist

<https://gist.github.com/>

The screenshot shows the GitHub Gist interface. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for "All gists" and "GitHub". There are also buttons for "Sign up for a GitHub account" and "Sign in". Below the navigation, a large blue header area contains the text "Instantly share code, notes, and snippets.". The main content area is a code editor. It has a "Gist description..." input field above a text area. The text area has a "Filename including extension..." input field at the top right, followed by three dropdown menus for "Spaces", "2", and "No wrap". The text area itself is empty, with only the number "1" visible in its top-left corner. At the bottom of the editor, there are buttons for "Add file", "Create secret gist", and "Create public gist".

git commit -a

- Дозволяє додавати до проміжного індексу та фіксувати *одночасно*
- Захоплює все в робочому каталозі
- Файли, які не відстежуються або видаляються, не включені

git log --oneline

- отримує перший рядок і контрольну суму всіх комітів у поточній гілці

```
[ dolanmi L02029756 ~/Desktop/new_project2 ]$ git log --oneline
3789cd3 file3.txt
6bfebcf new dir
730c6bd files
48f1ecf c
60f1c1a another message yet
d685ff9 another message
6e073c6 message
```

git diff g5iU0oPe7x

Якщо використовується контрольна сума старішого коміту, буде показано всі зміни порівняно з тими, які є у вашому робочому каталозі

Перейменування та видалення гілок

git branch -m/--move old_name new_name

git branch -d branch_name

Примітка: не має бути в branch_name

Примітка. Коміти в branch_name не повинні бути відключені в гілці, з якої ви видаляєте

git branch -D branch_name

Примітка. Якщо ви *справді* впевнені, що хочете видалити гілку з комітами

Тегування

- Git має можливість позначати певні моменти в історії як важливі, наприклад версії випусків (v.1.0, 2.0, ...)

git tag

```
$ git tag  
v0.1  
v1.3
```

Тегування

Два типи тегів:

- *lightweight* – вказівник на конкретний коментар – в основному SHA, що зберігається у файлі

git tag tag_name

- *annotated* – повний об'єкт, що зберігається в базі даних Git – SHA, ім'я тегера, адреса електронної пошти, дата, повідомлення, який можна підписати та перевірити за допомогою GNU Privacy Guard (GPG)

git tag -a tag_name -m "message"

Як я бачу теги?

git show tag_name

```
$ git show v1.4-lw
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700
```

changed the version number

Lightweight tag

```
$ git show v1.4
tag v1.4
Tagger: Ben Straub <ben@straub.cc>
Date:   Sat May 3 20:19:12 2014 -0700
```

my version 1.4

```
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700
```

changed the version number

Annotated tag

Питання та відповіді