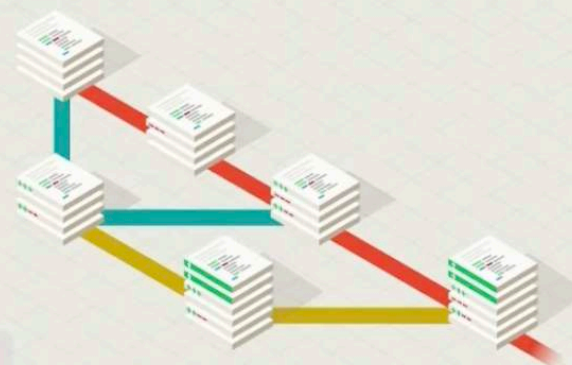


git --distributed-even-if-your-workflow-isnt

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**.

Just follow the onscreen instructions.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.

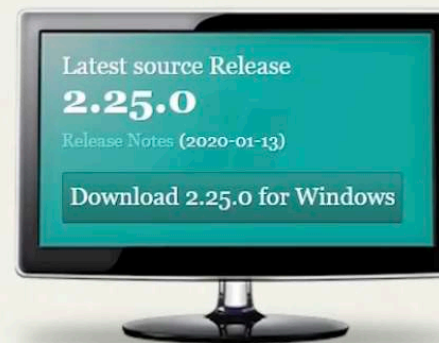


Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Windows GUIs



Tarballs



Mac Build



Source Code

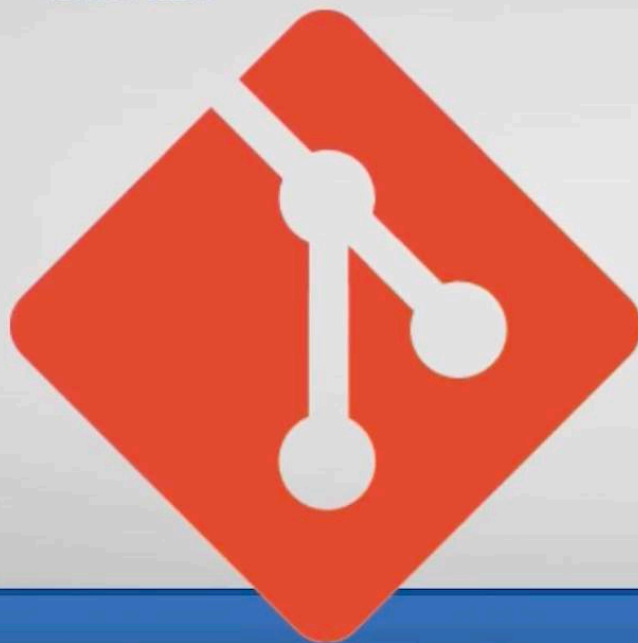
git for windows

VERSION 2.25.0

[FAQ](#)

[REPOSITORY](#)

[MAILING LIST](#)



We bring the awesome
Git SCM to Windows

[Download](#)

[Contribute](#)

Tools & Features

Git for Windows focuses on offering a lightweight, native set of tools

What is the difference between `git-scm` (downloaded from git-scm.com) and `msysGit` (hosted on msysgit.github.io)? They seem pretty similar, and even though I have used both, they seem to work fine.



... called `Git Bash`, or do both have the

asked Mar 10 '14 at 19:56

388 4

flow.com

io and v

and git-s

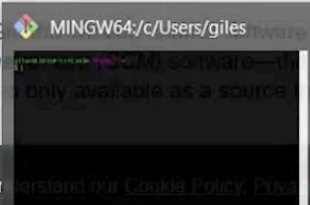
same downloadable binaries, and github.com/git-for-windows/git is the official repository in either case. The gitforwindows.org site seems like the official "face" of Git for Windows; rest assured it's not some "separate but deceptively similar" project. It's literally the same software. — John Y Jan 12 '18 at 16:39

add a comment

1 Answer

active oldest votes

108 The website git-scm.com is the official website for `Git SCM`, that's just the URL of the *source control management* software. Originally written for Linux, the original Git software is only available as a source that doesn't compile easily on Windows.



There is no "Git-SCM" software—the name is just `Git`. It doesn't compile

Blog

Micro-interactions with react-spring: Part 3

This week, #StackOverflowKnows the coronavirus's long tail, sulfuric acid...

Featured on Meta

Linked

- 105 MSysGit vs. Git for Windows
- 4 Which is official GUI of Git for windows?

Related

- 6651 How to remove local (untracked) files from the current Git working tree
- 11630 What is the difference between 'git pull' and 'git fetch'?

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ git config --global user.name "Your Name"  
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ git config --global user.email "Yourname@domain.com"
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ git config --global core.editor "notepad"
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ touch test.txt
```

Create and open file test.txt

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ notepad test.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ git config --global core.editor "Atom"
```

Changing text editor configuration

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ Atom test.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ git config --list
```

```
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true
```

See all the configuration

4

Configure Git

☆

1. Username
2. E-mail
3. Favourite text editor


```
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ mkdir shop  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~  
$ cd shop  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop  
$ git init  
Initialized empty Git repository in C:/Users/giles/shop/.git/  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ ls  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ ls -A  
.git/  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ cd .git  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop/.git (GIT_DIR!)  
$ ls -A  
config description HEAD hooks/ info/ objects/ refs/  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop/.git (GIT_DIR!)  
$ cd ..  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ touch lists.txt  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ ls  
list.txt ✓  
  
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)  
$ rm lists.txt
```

1

Make a directory

2

Change the directory

3

Track this directory

4

Create a file

No files yet.

Create a file

Remove a file

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ clear
```

Clear screen.

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
```

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

list.txt

nothing added to commit but untracked files present (use "git add" to track)

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git add list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
```

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: list.txt

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
```

```
$ git commit -m "Create shopping list"
```

```
[master (root-commit) 027e390] Create shopping list
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
```

```
$ git status
```

On branch master

nothing to commit, working tree clean

5

To track the file, we need to use "git add file name"

The "add" command tells Git to start tracking this particular file

👉 This step is called "Staging"

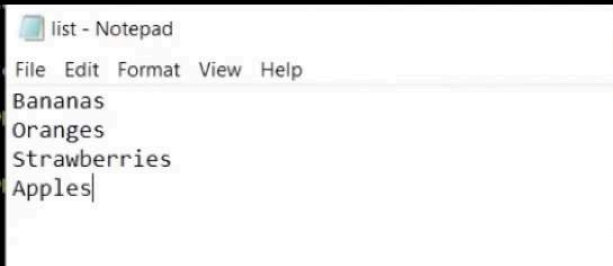
Now we need to commit this file

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 027e3900241aa87f02dffc613aaa9881e3584099 (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:46:15 2020 +0000
```

Create shopping list

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt
```

Open a file



list - Notepad

File Edit Format View Help

Bananas
Oranges
Strawberries
Apples

1

Add some items to the list

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   list.txt
```

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

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
```

```
$ git diff
diff --git a/list.txt b/list.txt
index e69de29..e8e0963 100644
```

```
--- a/list.txt
+++ b/list.txt
```

b version has some items added.

```
@@ -0,0 +1,4 @@
+Bananas
+Oranges
+Strawberries
+Apples
```

\ No newline at end of file

log- tells us the history of commits for this file

!

Git knows that this file has been changed but it hasn't been told to save a snapshot of this yet

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git add list.txt

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   list.txt

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git commit -m "Add fruit and vegetables"
[master 8bdeb9a] Add fruit and vegetables
1 file changed, 4 insertions(+)

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 8bdeb9a6fdc9e5f024d448095d27674ba6ee783d (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date:   Sat Feb 1 14:51:39 2020 +0000

    Add fruit and vegetables

commit 027e3900241aa87f02dffc613aaa9881e3584099
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date:   Sat Feb 1 14:46:15 2020 +0000

    Create shopping list

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
```



Let's revamp this
file


```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt
```

list - Notepad

File Edit Format View Help

FRUIT AND VEGETABLES
Bananas
Oranges
Strawberries
Apples

MEAT

DAIRY

HOUSEHOLD

We are not happy with the structure

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   list.txt
```

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

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git add list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git commit -m "Restructure list: new headings for different categories of purchases"
[master 3872c32] Restructure list: new headings for different categories of purchases
1 file changed, 15 insertions(+), 1 deletion(-)
```

We have these different versions of the file in the repository

We use "add" to stage the file

The descriptions need to be detailed

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log -1
commit 3872c32502a348efdc84df64ff451e69507cc896 (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:56:06 2020 +0000
```

Return most recent commit.

Restructure list: new headings for different categories of purchases

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git add list.txt
```

Applying changes to file and then
add to staging area.

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git diff
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git diff --staged
diff --git a/list.txt b/list.txt
index 70de999..6103d4d 100644
--- a/list.txt
+++ b/list.txt
@@ -6,7 +6,7 @@ Strawberries
 Apples

 MEAT
-
+Sausages
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 940e8c355f146a2fcd88ef96439177de0b58815b (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 15:17:32 2020 +0000
```

Sausages added to meat section

```
commit 3872c32502a348efdc84df64ff451e69507cc896
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:56:06 2020 +0000
```

Restructure list: new headings for different categories of purchases

```
commit 8bdeb9a6fdc9e5f024d448095d27674ba6ee783d
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:51:39 2020 +0000
```

Add fruit and vegetables

```
commit 027e3900241aa87f02dffc613aaa9881e3584099
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:46:15 2020 +0000
```

Create shopping list

```
$ git diff 8bdeb9a list.txt
diff --git a/list.txt b/list.txt
index e8e0963..6103d4d 100644
--- a/list.txt
+++ b/list.txt
@@ -1,4 +1,18 @@
+
+FRUIT AND VEGETABLES
+Bananas
+Oranges
+Strawberries
-Apples
\ No newline at end of file
```

Differences between versions

! This is the head of all the changes that we've made

We just need to type the first digits of the commit identifier.


```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 940e8c355f146a2fcd88ef96439177de0b58815b (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 15:17:32 2020 +0000
```

Sausages added to meat section

```
commit 3872c32502a348efdc84df64ff451e69507cc896
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:56:06 2020 +0000
```

Restructure list: new headings for different categories of purchases

```
commit 8bdeb9a6fdc9e5f024d448095d27674ba6ee783d
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:51:39 2020 +0000
```

Add fruit and vegetables

```
commit 027e3900241aa87f02dffc613aaa9881e3584099
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 14:46:15 2020 +0000
```

Create shopping list

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git checkout HEAD~3 list.txt
Updated 1 path from b50a881
```

Original version

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
list.txt
```

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
$ git checkout HEAD~2 list.txt
path from d65b70c
```

Version 2

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
list.txt
```

Let's go back to
the file as it was

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
$ git checkout HEAD~1 list.txt
path from 0a68a0d
```

Version 3

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
list.txt
```

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
$ git checkout HEAD list.txt
path from 7b6e982
```

Current version

```
TOP-CLFCJ43R MINGW64 ~/shop (master)
```



```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git add list.txt

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   list.txt
```

?

What happens
when we make a
mistake?

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git reset list.txt
Unstaged changes after reset:
M       list.txt
```

→

We have
unstaged those
changes

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git diff --staged
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git commit -m "Add content to Dairy"
[master f175af1] Add content to Dairy
1 file changed, 6 insertions(+)
```

?

What if we
committed the
wrong changes?

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit f175af1a5201e551e246616d96e19996a66671d3 (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date:   Sat Feb 1 15:40:20 2020 +0000
```

Add content to Dairy

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git reset HEAD~1 --soft
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 940e8c355f146a2fcd88ef96439177de0b58815b (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date:   Sat Feb 1 15:17:32 2020 +0000
```

Sausages added to meat section

1

Use reset

- soft - removes
the commit

- soft - doesn't change
what happened in the
staging area

1

Reset options:

- soft
- mixed
- hard

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git reset HEAD~1 --mixed
Unstaged changes after reset:
M   list.txt
```

- mixed - removes the commit and also makes the changes unstaged

- mixed - doesn't change the file

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git commit -m "Dairy section update"
[master bb0a0d9] Dairy section update
1 file changed, 6 insertions(+)
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git reset HEAD~1 --hard
HEAD is now at 940e8c3 Sausages added to meat section
```

- hard - removes the changes we've added to the file

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git log
commit 940e8c355f146a2fcd88ef96439177de0b58815b (HEAD -> master)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date:   Sat Feb 1 15:17:32 2020 +0000
```

Sausages added to meat section

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git branch dairy
```

Create branch

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git branch
dairy
* master
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git branch meat
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git branch
dairy
* master
meat
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git checkout dairy
Switched to branch 'dairy'
```

Work on branch

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/s
$ notepad list.txt
```

Open file and modify it

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (dairy)
$ git add list.txt
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (dairy)
$ git commit -m "update dairy"
[dairy b58e937] update dairy
1 file changed, 5 insertions(+), 1 deletion(-)
```

Add & commit file

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (dairy)
$ git log
commit b58e937f248406a0f10136aeef22fe244b30db6 (HEAD -> dairy)
Author: Giles <54141847+PythonJournalist@users.noreply.github.com>
Date: Sat Feb 1 15:57:53 2020 +0000
```

update dairy



Different branches



The different versions of the file can be developed in parallel

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (meat)
$ git checkout master
Switched to branch 'master'

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git merge meat
Updating 940e8c3..327fc07
Fast-forward
 list.txt | 5 +++++
 1 file changed, 5 insertions(+)

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git merge dairy

giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ notepad list.txt
```

Merging files from
different branches to the
master branch

FRUIT AND VEGETABLES
Bananas
Oranges
Strawberries
Apples

MEAT
Sausages
Bacon
Lamb
Beef
Pork
Rabbit

DAIRY
Milk
Eggs
Yoghurt
Butter
Cheese

HOUSEHOLD

PETS



2.00



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

New

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.

[Read the guide](#)

[Start a project](#)



Together we will create
a new repository
on Github

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.

[Explore GitHub](#)



New repository

Import repository

New gist

New organization

New project

See what launched
Say hello to our new
more from GitHub A
how we're shaping t



GitHub Sponsors is out of beta

We're now generally available in 30 countries, with more coming soon.



Securing software together

Introducing new ways to identify and prevent security vulnerabilities across your code base.

Explore repositories

[jonhoo/flurry](#)

A port of Java's ConcurrentHashMap to Rust

Java ★ 105

[FluxML/Zygote.jl](#)

Intimate Affection Auditor


Julia ★ 622

[haikuports/haikuports](#)

Software ports for the Haiku operating system.

Shell ★ 174

Owner

 PythonJournalist ▾

Repository name *

/ git_practice ✓

Great repository names are short and memorable. Need inspiration? How about **refactored-meme**?

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 ▾



Creating repository...

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



or

HTTPS

SSH

https://github.com/PythonJournalist/git_practice.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 "# git_practice" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/PythonJournalist/git_practice.git
git push -u origin master
```



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

```
git remote add origin https://github.com/PythonJournalist/git_practice.git
git push -u origin master
```



```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git remote add origin https://github.com/PythonJournalist/git_practice.git
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git remote -v
origin https://github.com/PythonJournalist/git_practice.git (fetch)
origin https://github.com/PythonJournalist/git_practice.git (push)
```

```
giles@LAPTOP-CLFCJ43R MINGW64 ~/shop (master)
$ git push origin master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (21/21), 1.87 KiB | 480.00 KiB/s, done.
Total 21 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/PythonJournalist/git_practice.git
 * [new branch]      master -> master
```

We want to synchronize our local repository with the new repository on GitHub

You will be asked for your username and password

We can push our repository to the remote repository on GitHub

Read the guide

PythonJournalist / git_practice Private

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Security

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

7 commits

1 branch

0 packages

0 releases

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

PythonJournalist Merge branch 'dairy'

Latest commit e3ad90f 16 minutes ago

list.txt

Merge branch 'dairy'

16 minutes ago

Add a README with an overview of your project.

Add a README

A list of Medical imaging datasets.

medical-imaging image-dataset

The screenshot shows the GitHub interface for the repository 'medical-imaging-datasets'. At the top, it displays '20 commits', '1 branch', '0 packages', '0 releases', and '5 contributors'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A yellow callout box with the number '1' points to the 'Clone or download' button. A red box highlights the 'Clone with HTTPS' section, which includes the text 'Use Git or checkout with SVN using the web URL.' and the URL 'https://github.com/sfikas/medical-imaging-datasets'. Below this, there are buttons for 'Open in Desktop' and 'Download ZIP'.

Git Clone - a way of downloading a repository that exists on GitHub

Clone with HTTPS ⓘ Use SSH

Use Git or checkout with SVN using the web URL.

<https://github.com/sfikas/medical-imaging-datasets>

Open in Desktop Download ZIP

medical-imaging-datasets

- A list of Medical imaging datasets. Source : <https://sites.google.com/site/aacruzr/image-datasets>
- An additional, possibly overlapping list can be found at : <https://github.com/beamandrew/medical-data>

Multimodal databases

- Center for In vivo Microscopy (CIVM), Embryonic and Neonatal Mouse (H&E, MR)
<http://www.civm.duhs.duke.edu/devatlas/> user guide: <http://www.civm.duhs.duke.edu/devatlas/UserGuide.pdf>
- LONI image data archive <https://ida.loni.usc.edu/services/Menu/IdaData.jsp?project=>
- Radiology (Ultrasound, Mammographs, X-Ray, CT, MRI, fMRI, etc.)
- Collaborative Informatics and Neuroimaging Suite (COINS) <https://portal.mrn.org/micis/index.php?subsite=dx>