A quick intro to Git and GitHub

Miguel Lázaro-Gredilla
miguel@tsc.uc3m.es

June 2014
Machine Learning Group
http://www.tsc.uc3m.es/~miguel/MLG/

Contents

Git

- Don't confuse Git with GitHub
  - Git is a version control tool
  - GitHub provides cloud services using Git (remote repositories, bug tracking, wiki page...)

Git

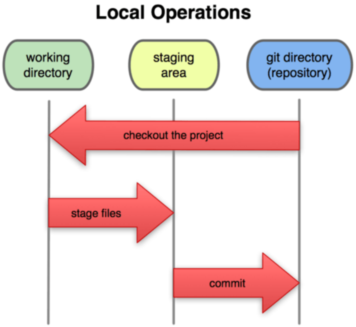
- Don't confuse Git with GitHub
  - Git is a version control tool
  - GitHub provides cloud services using Git
    (remote repositories, bug tracking, wiki page...)
- Git is not like Dropbox or Google Drive
  - True version control, not just file history
  - Need to resort to console sooner or later

Git

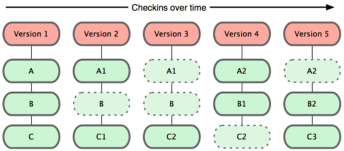
- Don’t confuse Git with GitHub
    - Git is a version control tool
    - GitHub provides cloud services using Git (remote repositories, bug tracking, wiki page...)
- Git is not like Dropbox or Google Drive
    - True version control, not just file history
    - Need to resort to console sooner or later
- Git is not like CVS, Subversion or Perforce
    - There is no need for a central (such as cloud) repository
    - You can work offline most of the time
    - Each local copy contains the full repository’s history
- Devised by Linus, motivated by Linux Kernel development

Git: General concepts (I/II)

- Local operations (staging area == index)



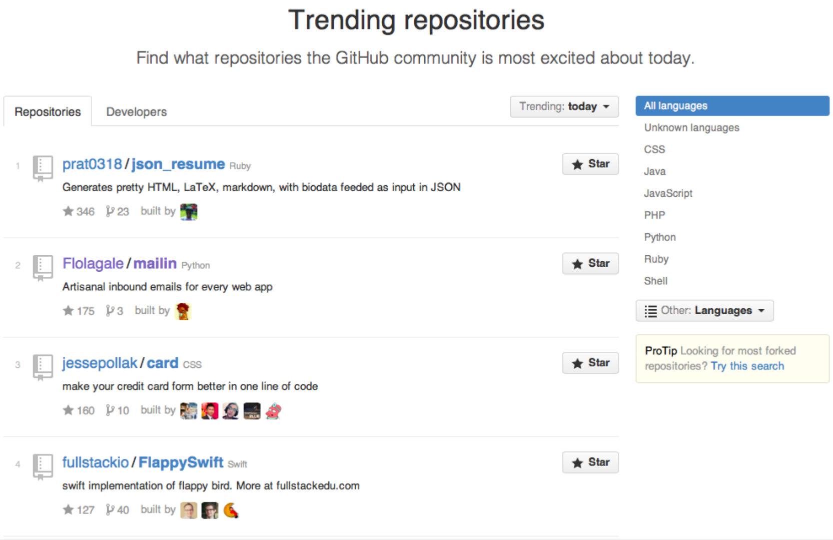
- Evolution over time



Git: General concepts (II/II)

- clone: Clone remote repository (and its full history) to your computer
- stage: Place a file in the staging area
- commit: Place a file in the git directory (repository)
- push: Update remote repository using local repository
- pull: Update local repository using remote repository
- add: Start tracking a new file, or stage a modified file
- branch: An end point in the commit tree
- fork: A copy of another repository for reuse
- merge: Combine two commits

Github: trending repositories


Trending repositories

Find what repositories the GitHub community is most excited about today.

1 prat0318 / json_resume Ruby
Generates pretty HTML, LaTeX, markdown, with biodata feeded as input in JSON
346 23 built by

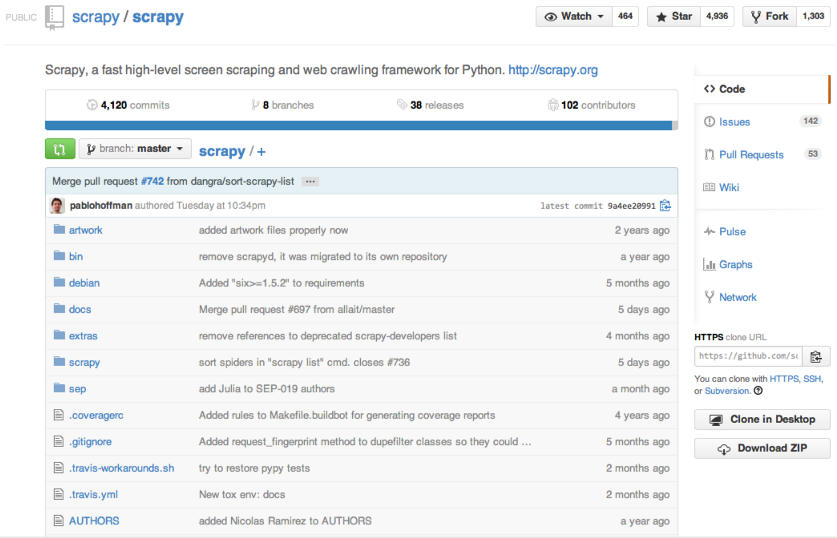
2 Flolagale / mailin Python
Artisanal inbound emails for every web app
175 3 built by

3 jessepollak / card CSS
make your credit card form better in one line of code
160 10 built by

4 fullstackio / FlappySwift Swift
swift implementation of flappy bird. More at fullstackedu.com
127 40 built by

ProTip Looking for most forked repositories? Try this search

Github: repository view



Scrapy, a fast high-level screen scraping and web crawling framework for Python. http://scrapy.org

Merge pull request #742 from dangra/sort-scrapy-list ...

pablohoffman authored Tuesday at 10:34pm latest commit 9a4ee20991

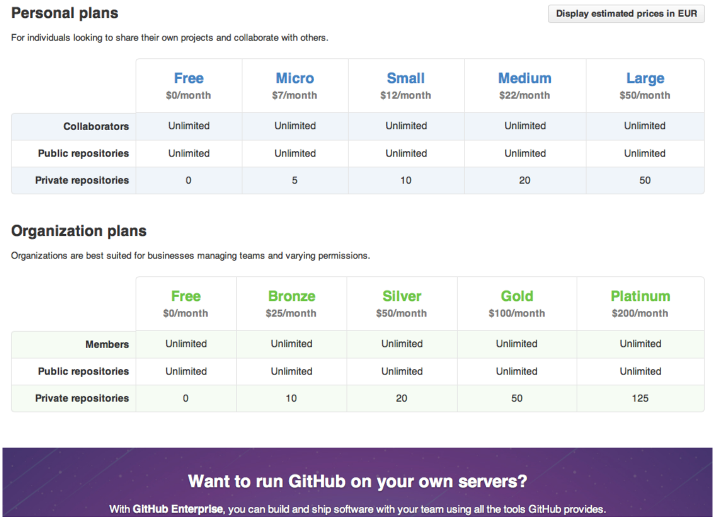
artwork	added artwork files properly now	2 years ago
bin	remove scrapyd, it was migrated to its own repository	a year ago
debian	Added "six>=1.5.2" to requirements	5 months ago
docs	Merge pull request #697 from allait/master	5 days ago
extras	remove references to deprecated scrapy-developers list	4 months ago
scrapy	sort spiders in "scrapy list" cmd. closes #736	5 days ago
sep	add Julia to SEP-019 authors	a month ago
.coveragerc	Added rules to Makefile.buildbot for generating coverage reports	4 years ago
.gitignore	Added request_fingerprint method to dupefilter classes so they could ...	5 months ago
.travis-workarounds.sh	try to restore pypy tests	2 months ago
.travis.yml	New tox env: docs	2 months ago
AUTHORS	added Nicolas Ramirez to AUTHORS	a year ago

HTTPS clone URL

https://github.com/s

You can clone with HTTPS, SSH, or Subversion.

Github pricing



Personal plans

Display estimated prices in EUR

For individuals looking to share their own projects and collaborate with others.

	Free $0/month	Micro $7/month	Small $12/month	Medium $22/month	Large $50/month
Collaborators	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Public repositories	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Private repositories	0	5	10	20	50

Organization plans

Organizations are best suited for businesses managing teams and varying permissions.

	Free $0/month	Bronze $25/month	Silver $50/month	Gold $100/month	Platinum $200/month
Members	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Public repositories	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Private repositories	0	10	20	50	125

Want to run GitHub on your own servers?

With **GitHub Enterprise**, you can build and ship software with your team using all the tools GitHub provides.

Set up Git and GitHub

Go to `http://www.github.com`
and create a user. Choose a nickname you like!

Then go to `http://help.github.com`, look for "Set up Git" and download and install the native app for your platform.

In the process you should also be installing the command-line tools. The apps aren't that good, but keep them just in case.

You should be all set!

Contents

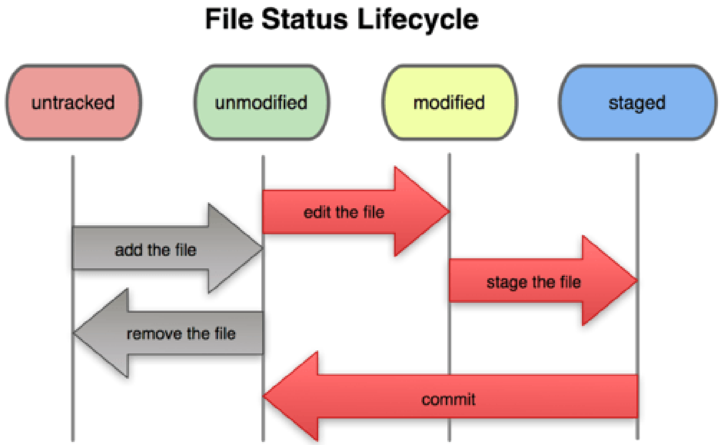
Cloning a repository

This clones a repository (and its full history) to your computer

```
$ git clone https://github.com/lazarox/charla.git
```

- It creates folder `charla` (your working directory)
- It creates folder `charla/.git` (the local repository)
- Might include text files
  - `charla/.gitignore`
  - `charla/.gitattributes`
- Remote server will be referred to as `origin`.

Possible file statuses



Checking the status of things

Tells you about modified files, untracked files, staged files ready for commit, etc. Even provides suggestions about what to do next. Very useful!

Also informs you about which branch you are at (`master`)

```
$ cd charla
$ git status
```

ACTION: Put a file with your name (e.g., miguel.txt) in directory "charla" and then check the status

Start tracking a file

Untracked files in the working directory can't be staged or committed. You can track a file using

```
$ git add miguel.txt
```

Multiple files can be added at a time (don't forget the quotes)

```
$ git add '*.txt'
$ git add folder
```

ACTION: Track the file you just added and check the status
QUESTION: was the previous diagram accurate?
ACTION: Modify the file that you just added and check the status
QUESTION: What if I commit now? How can we avoid that issue?

Committing changes

Commit *everything* from the staging area to the repository (locally)

```
$ git commit -m 'I improved all files'
```

The message is compulsory! Otherwise, you'd avoid it and soon forget that commit's purpose

Each commit has an identifying SHA-1 hash and comment. You can roll back to past commits.

You can stage everything that is tracked and commit in a single step using

```
$ git commit -a -m 'I improved all files'
```

ACTION: Commit everything.

Removing and moving files

Untrack a file and delete a file, or rename/move a file are modifications

```
$ git rm miguel.txt
$ git mv miguel.txt newmiguel.txt
```

You will need to commit this changes! As any commit, they can be undone.

If you delete or rename files on your own, Git will notice and you'll still have to stage and commit

ACTION: Delete the file you created, check state, commit

Browsing existing commits

You can see the list of previous commits with

```
$ git log
```

There are many options to filter commits, compare them, see what was added or removed, etc.

`http://git-scm.com/book/en/Git-Basics-Viewing-the-Commit-History`

ACTION: Browse your current commits

Rolling back a previous commit

You can destructively roll back to a previous commit (as if posterior commits never happened) with

`git reset --hard <HASH>` (to move to that commit)
`git reset --hard master^` (to move one step back in `master`)

Or you can move to a previous commit and play around without destroying anything

`git checkout <HASH>` (to move to that commit)
`git checkout master^^` (to move two steps back in `master`)

Working with remotes (such as GitHub)

You can download and merge from the remote repository you cloned from

```
$ git pull origin
```

Or upload your local repository to the remote repository you cloned from

```
$ git push origin master
```

This will fail if the repository is ahead of you. Pull, check everything is fine, then push. You can force it with `$ git push --force`, but that destroys the remote! Never do that!

To get info about the remote: `$ git remote show origin`

ACTION: Push your changes. Try an actual merge with collisions!

Browsing current changes

Differences between your working directory and staging area

```
$ git diff
```

Differences between your staging area and your repository

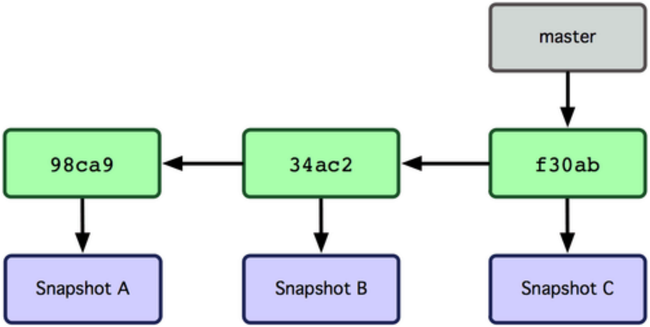
```
$ git diff --staged
```

Contents

Branching

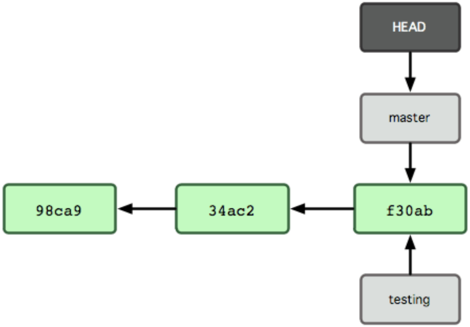
- Each commit has a pointer to its parent(s) commits.
- Each branch is a pointer to a concrete commit.
- The HEAD is a pointer to the current branch.
- When create a new commit, the current branch moves forward to point to it

Several commits



Creating a new branch

```
$ git branch testing
```

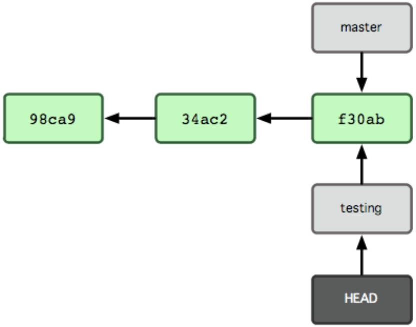


The current branch is still the old one

Use `$ git branch` to list existing branches

Moving to a different branch

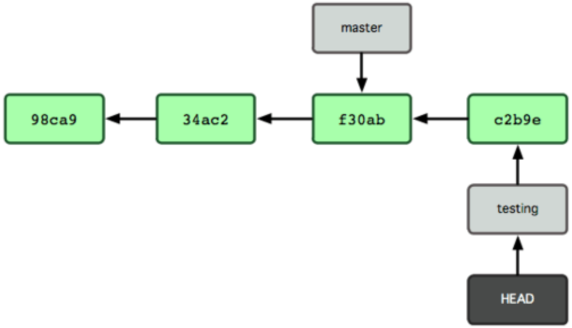
```
$ git checkout testing
```



Short-hand for the previous steps: `$ git checkout -b testing`
You could also checkout commits, and then provide a branch name
Always commit before leaving the current branch!

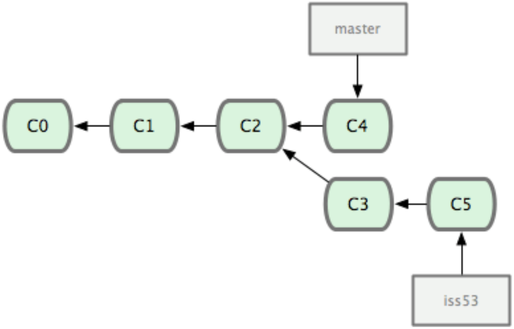
Bifurcations may appear (I/II)

If you now make a new commit while on branch `testing`...



Basic merge

Given this structure

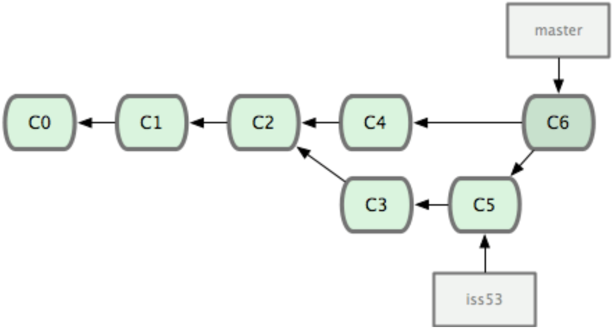


you can merge two commits using

```
$ git checkout master
$ git merge iss53
```

Deleting branches

After the previous merge, we get



since iss53 is no longer needed, we can delete it

```
$ git branch -d iss53
```

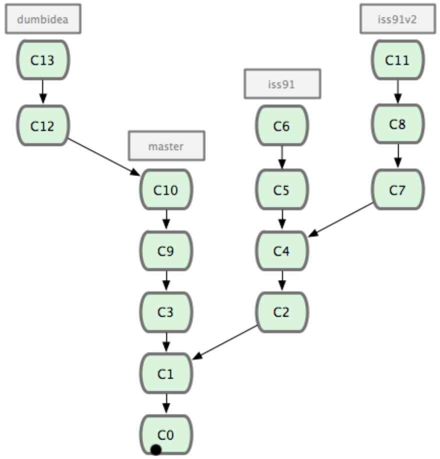
Solving merge conflicts

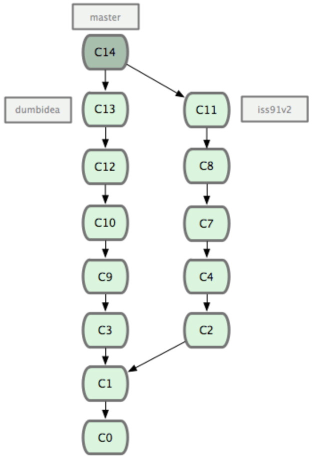
In case of conflict, no commit occurs. Instead, the working directory has conflicting files like this

```
<<<<<<< HEAD
<div id="footer">contact :
email.support@github.com</div>
=======
<div id="footer">
please contact us at support@github.com
</div>
>>>>>>> iss53
```

Conflicting files are shown on `$ git status`
To solve conflicts, fix each file and stage it

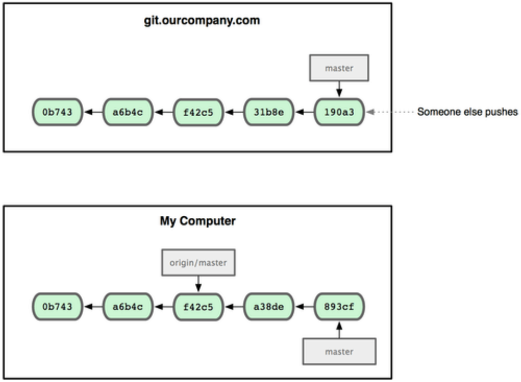
Sample workflow





Remote repositories

The remote repository and its local snapshot may diverge

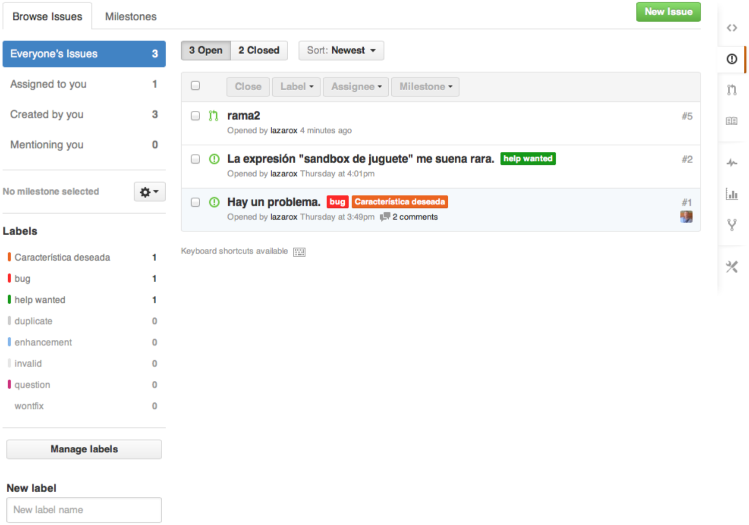


Option 1: Pull, solve any conflicts, and then push

Option 2: Use another branch, push it, and issue a pull request

Contents

Using "Issues"


Browse Issues Milestones New Issue

Everyone's Issues 3
Assigned to you 1
Created by you 3
Mentioning you 0

No milestone selected

Labels

Característica deseada	1
bug	1
help wanted	1
duplicate	0
enhancement	0
invalid	0
question	0
wontfix	0

Manage labels

New label

New label name

3 Open 2 Closed Sort: Newest

Close Label Assignee Milestone

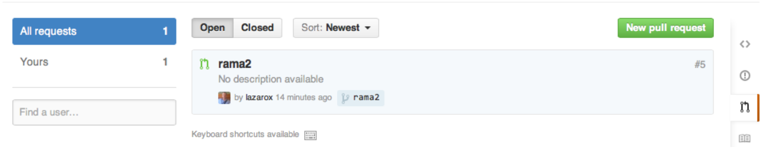
rama2 #5
Opened by lazarox 4 minutes ago

La expresión "sandbox de juguete" me suena rara. help wanted #2
Opened by lazarox Thursday at 4:01pm

Hay un problema. bug Característica deseada #1
Opened by lazarox Thursday at 3:49pm 2 comments

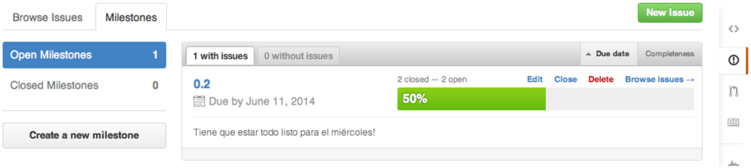
Keyboard shortcuts available

Pull requests



rama2 #5
No description available
by lazarox 14 minutes ago rama2

Progress tracking towards milestones



0.2
Due by June 11, 2014
2 closed — 2 open
50%
Tiene que estar todo listo para el miércoles!

Other Features: labels, cross-referencing, mark-down, per-line comments in commits, mentions, gists, etc.

Contents

Exercise

1. Make sure your local repo is up-to-date (i.e, pull)
2. Don't interact again with the server unless you are told to

Exercise

1. Make sure your local repo is up-to-date (i.e, pull)
2. Don't interact again with the server unless you are told to
3. Make a branch called `myfixName` (use your own name) at the current commit and switch to it

Exercise

1. Make sure your local repo is up-to-date (i.e, pull)
2. Don't interact again with the server unless you are told to
3. Make a branch called `myfixName` (use your own name) at the current commit and switch to it
4. Go to folder `fibos` and fix only the file with your name on it

Exercise

1. Make sure your local repo is up-to-date (i.e, pull)
2. Don't interact again with the server unless you are told to
3. Make a branch called `myfixName` (use your own name) at the current commit and switch to it
4. Go to folder `fibos` and fix only the file with your name on it
5. There is a bug that requires your immediate attention! Switch to branch `master` and pull new files from server
6. Go to folder `factorials` and fix only the file with your name
7. Push corrections in `master` to server

Exercise

1. Make sure your local repo is up-to-date (i.e, pull)
2. Don't interact again with the server unless you are told to
3. Make a branch called `myfixName` (use your own name) at the current commit and switch to it
4. Go to folder `fibos` and fix only the file with your name on it
5. There is a bug that requires your immediate attention! Switch to branch `master` and pull new files from server
6. Go to folder `factorials` and fix only the file with your name
7. Push corrections in `master` to server
8. Switch to `myfixName` and finish corrections in folder `fibos`
9. Push that branch to GitHub and issue a Pull Request to get your branch merged