

RE-COOP #8:

A round-up around GIT ...
... informal of course !!!

16th September 2020

Cedric Lemaître
Guillaume Lemaître
Joan Massich

Universitat de Girona, Université de Bourgogne



① History

② What is GIT?

Definitions

1st study case

2nd study case

3rd study case

4th study case

③ GIT more in details

The big picture

Branch principle

GitHub

Usual commands

④ GIT in even more details

git init

git add

git status

git commit

git log

git status

git diff

gitg

git branch

git checkout

Push forward

git merge

GitHub

git remote add

git push

History

What is GIT?

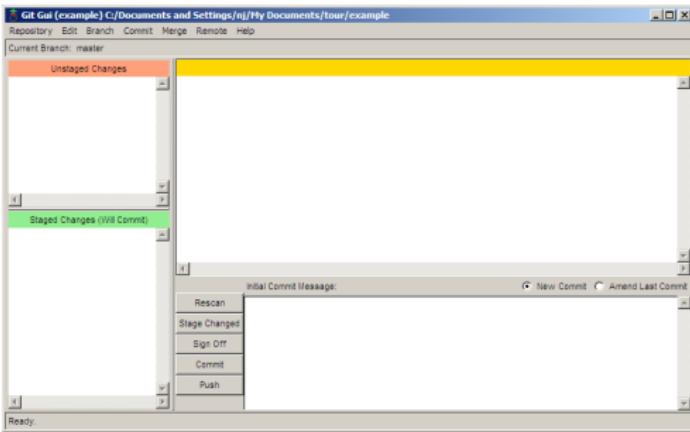
GIT more in details

GIT in even more details



A bit of History ...

For Microsoft Windows users





A bit of History ...

For Linux users

History

What is GIT?

GIT more in details

GIT in even more details



A bit of History ... Original author



A bit of History ... Original author

Linus Torvalds





What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.



What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- THIS IS COOL AND FANCY!!!!



What is GIT?

Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,



What is GIT?

Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,
 - ✓ Allow to have multiple versions,



What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,
 - ✓ Allow to have multiple versions,
 - ✓ Possibility to swap versions,



What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,
 - ✓ Allow to have multiple versions,
 - ✓ Possibility to swap versions,
 - ✓ Perform collaborative work.



What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,
 - ✓ Allow to have multiple versions,
 - ✓ Possibility to swap versions,
 - ✓ Perform collaborative work,
 - ✓ Easy way to merge work,



What is GIT? Definitions

Formal definition

- ▶ Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
 - ▶ Since 2005, it has become the most widely adopted version control system for software development.

In an everyday language

- ▶ THIS IS COOL AND FANCY!!!!
 - ✓ Keep track of all changes performed on a repository,
 - ✓ Allow to have multiple versions,
 - ✓ Possibility to swap versions,
 - ✓ Perform collaborative work,
 - ✓ Easy way to merge work,
 - ✗ Need to learn the tool.



What is GIT?

Study cases

Software versioning

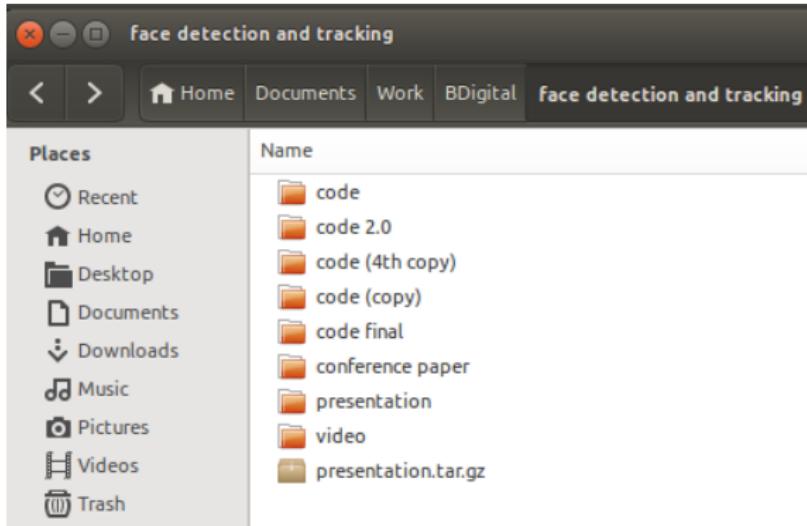




What is GIT?

Study cases

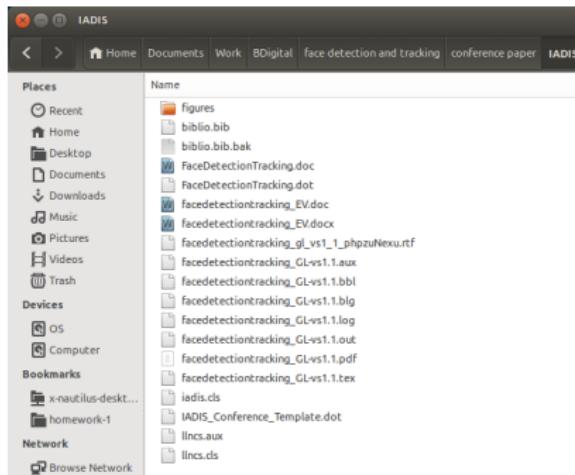
Student case





What is GIT? Study cases

Writing case



History
ooo

What is GIT?
oooo●

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



What is GIT?

Study cases

Education case

The screenshot shows a Git commit history interface. At the top, there's a header bar with a user icon, the name "Guillaume Lemaître / ViBOT Course - Medical Image Ana...", and a search bar. Below the header, a navigation bar includes tabs for "Project", "Files" (which is selected), "Commits", "Network", "Graphs", "Issues 0", and "Merge Requests".

The main area displays a list of commits. On the left, a sidebar shows a dropdown menu set to "master" and a list of branches: "course_2012_2013_dev_handout", "course_2013_2014_dev", "course_2013_2014_dev_handout", and "master". The "master" branch is highlighted with a blue background.

The commit list on the right shows the following entries:

- Last Commit > [b5abfec194a](#) – Add the grate and report for the segmentation
- Guillaume Lemaître update the master branch
- Guillaume Lemaître update the master branch
- Guillaume Lemaître Add anonymized version
- Guillaume Lemaître Add the grate and report for the segmentation
- Guillaume Lemaître Update the README file

Each commit entry includes a small user icon, the author's name, and a timestamp.

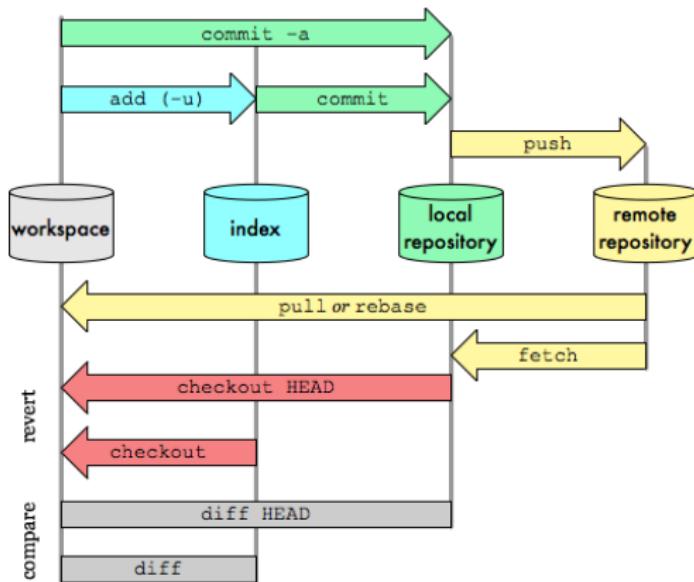


GIT more in details

The big picture

Git Data Transport Commands

<http://csteelle.com>





GIT more in details

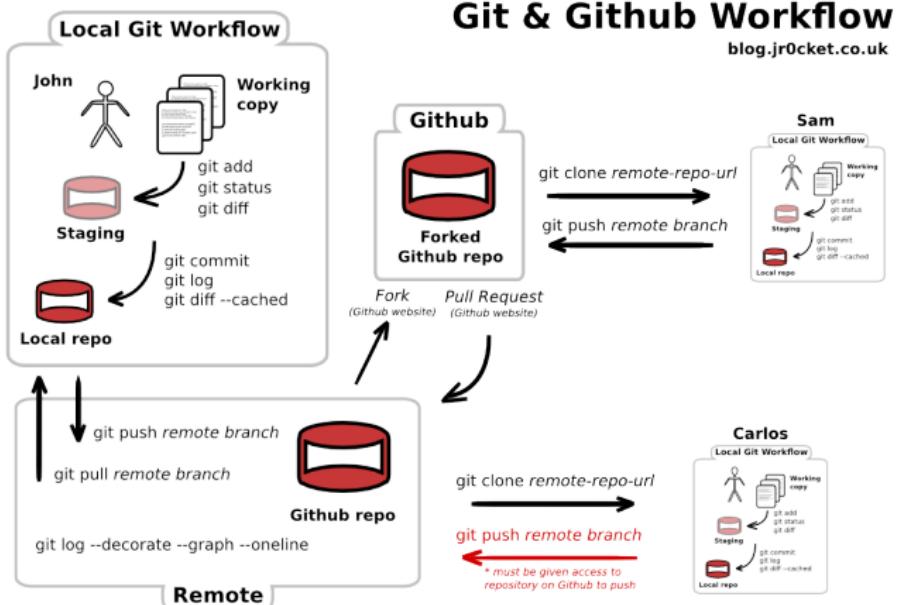
Branch principle





GIT more in details

And GitHub in all that ...





GIT more in details Usual commands

Local GIT - usual

- ▶ `git init`: initialize a git repository,
- ▶ `git add`: command to track change for a file,
- ▶ `git status`: get the status of the repository,
- ▶ `git diff`: see the difference inside the file,
- ▶ `git commit`: save the changes. It will be possible to come back on this node,
- ▶ `git checkout`: come back to a specific commit.

Local GIT - branching

- ▶ `git branch`: to create a new branch,
- ▶ `git checkout`: navigate to other branch,
- ▶ `git merge`: merge branches,
- ▶ `git cherry-pick`: select a specific commit for merging.

History
ooo

What is GIT?
ooooo

GIT more in details
oooo●

GIT in even more details
oooooooooooooooooooo



GIT more in details Usual commands

Remote GIT - synchronization

- ▶ `git remote add`: add the url of the server,
- ▶ `git push`: upload changes from local to remote,
- ▶ `git pull`: download changes from remote to local and apply changes,
- ▶ `git fetch`: download changes from remote,
- ▶ `git rebase`: apply changes downloaded from remote.

History
ooo

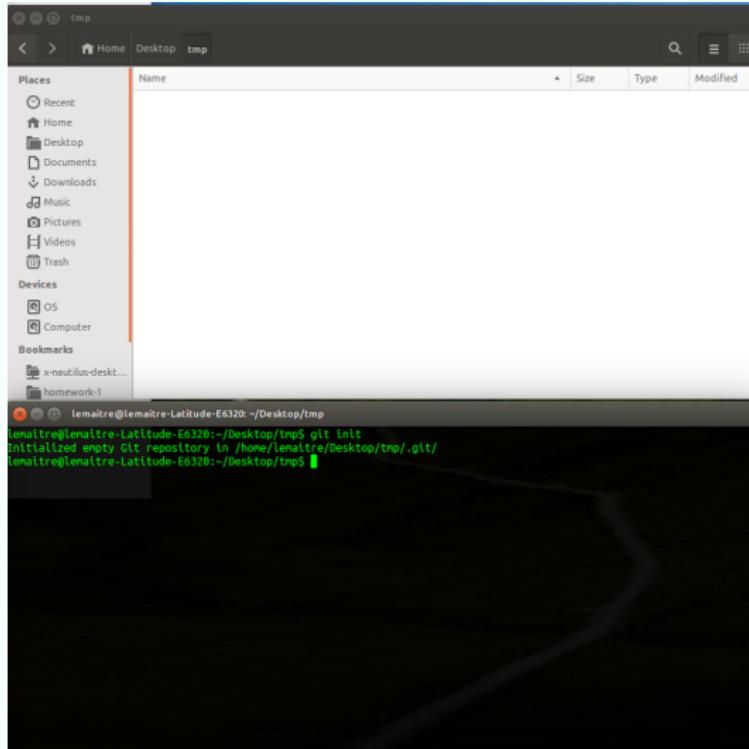
What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details git init



History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details git add

The screenshot shows a desktop environment with a file manager window open. The file manager lists a single file named "README.md" in the "tmp" directory. The terminal window in the bottom left shows the command "git add README.md" being entered.

```
lennaire@lennaire-Latitude-E6320: ~/Desktop/tmp
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git init
Initialized empty Git repository in /home/lennaire/Desktop/tmp/.git/
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ touch README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git add README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$
```



GIT in even more details git status

The screenshot shows a desktop environment with a file manager window displaying a single file named "README.md" in the "/tmp" directory. The file is 0 bytes, a Text file, and was modified at 11:36. Below the file manager is a terminal window with the following content:

```
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$  
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git init  
Initialized empty Git repository in /home/lennaire/Desktop/tmp/.git/  
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ touch README.md  
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git add README.md  
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git status  
On branch master  
  
Initial commit  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  
    new file: README.md  
  
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$
```



GIT in even more details
git commit

tmp

Places

- Recent
- Home
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Trash

Devices

- OS
- Computer

Bookmarks

- x-nautilus-desk...
- home-work-1

"README.md" selected (0 bytes)

```
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git init
Initialized empty Git repository in /home/lemaitre/Desktop/tmp/.git/
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ touch README.md
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git add README.md
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file: README.md

lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git commit -a
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details git commit

The screenshot shows a desktop environment with a file manager window open in the foreground, displaying a single file named "README.md" in the "/tmp" directory. The file is 0 bytes long and is a text file. Below the file manager is a terminal window titled "lennaire@lennaire-Latitude-E6320: ~/Desktop/tmp". The terminal is running the "nano 2.2.6" editor and displays the standard commit message template:

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file: README.md
```

At the bottom of the terminal window, there is a menu bar with various options like Get Help, WriteOut, Read File, Where Is, Cut Text, Cur Pos, etc.



GIT in even more details git commit

The screenshot shows a desktop environment with a file manager window open in the foreground and a terminal window in the background.

File Manager Window:

- Title bar: tmp
- Toolbar: Back, Forward, Home, Desktop, tmp
- Places sidebar:
 - Recent
 - Home
 - Desktop
 - Documents
 - Downloads
 - Music
 - Pictures
 - Videos
 - Trash
- Devices sidebar:
 - OS
 - Computer
- Bookmarks sidebar:
 - x-nautilus-deskt...
 - homework-1
- Content area:
 - Name column
 - File: README.md
 - Size: 0 bytes
 - Type: Text
 - Modified: 11:36

Terminal Window:

```
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git init
Initialized empty Git repository in /home/lennaire/Desktop/tmp/.git/
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ touch README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git add README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file: README.md

lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git commit -a
[master (root-commit) e38038c] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$
```



GIT in even more details

git log

The screenshot shows a terminal window with the following content:

```
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ git log
Initialized empty Git repository in /home/lenaire/Desktop/tmp/.git/
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ touch README.md
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ git add README.md
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ git status
On branch master
Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file: README.md

tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ git commit -a
[master (root-commit) e39030c] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$ git log
commit e39030c8c54bd3e967fff67783fbf0053adfb0b56
Author: Guillaume Lemaire <glenalitregvisor.udg.edu>
Date:   Thu Mar 19 11:39:37 2015 +0100

  Initial commit
tmp@lenaire-Latitude-E6320:~/Desktop/tmp$
```



GIT in even more details

git status

The screenshot shows a desktop environment with a file manager window open, displaying a file named 'README.md'. The terminal window below it is running the command 'git status'.

```
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git commit -a
[master (root-commit) e3303bc] Initial commit
 1 file changed, 2 insertions(+), 0 deletions(-)
 create mode 100644 README.md
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git log
commit e3303bc8c54bd3ea67ff6f7783fbf0053adfb6b56
Author: Guillaume Lennaire <glenair@visor.udg.edu>
Date:   Thu Mar 19 11:39:37 2015 +0100

Initial commit
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ 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: README.md

no changes added to commit (use "git add" and/or "git commit -a")
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ █
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooo



GIT in even more details git diff

The screenshot shows a Linux desktop environment with a file manager window open, displaying a folder named 'tmp' containing a file named 'README.md'. The file content is shown in the main pane: 'I modify my readme file'. Below the file manager is a terminal window with the following session:

```
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git log
commit e3303c8c54bd3ea67ff67783fbf0653adfb0b56
Author: Guillaume Lennaire <glennaire@visor.udg.edu>
Date: Thu Mar 19 11:39:37 2015 +0100

Initial commit
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ 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: README.md

no changes added to commit (use "git add" and/or "git commit -a")
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$ git diff
diff --git a/README.md b/README.md
index 1000000..226a87a 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1 @@
@@ -0,0 +1 @@
I modify my readme file
lennaire@lennaire-Latitude-E6320:~/Desktop/tmp$
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooo●oooooooooooo



GIT in even more details

gitg

The screenshot shows the gitg application interface. At the top, there's a toolbar with icons for file operations. Below it, a navigation bar with tabs for 'History' (which is selected) and 'Commit'. A dropdown menu for 'Branch' is set to 'master'. On the right, there's a search bar with a magnifying glass icon. The main area displays a table of commits:

Subject	Author	Date
master Second changes	Guillaume Lemaitre	jue 19 mar 2015 11:43:59 CET
Initial commit	Guillaume Lemaitre	jue 19 mar 2015 11:39:37 CET



GIT in even more details

git branch

The screenshot shows a desktop environment with a file manager window open, displaying a file named 'README.md'. The terminal window below it is running on a system named 'lemaitre@lemaitre-Latitude-E6320'. The terminal output is as follows:

```
U:--- README.md All L3 Git:master (Markdown)
Wrote /home/lemaitre/Desktop/tmp/README.md

Date: Thu Mar 19 11:43:59 2015 +0100
Second changes

commit e33038c854bd3ea67fff677783fbf0053adfb0b56
Author: Guillaume Lemaitre <glemaitre@csor.ulg.edu>
Date: Thu Mar 19 11:39:37 2015 +0100

Initial commit
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git checkout 31009dbc36bc8290687f292c93e5e720b2651937
Note: checking out '31009dbc36bc8290687f292c93e5e720b2651937'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 31009db... Second changes
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git branch experimental
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$
```



GIT in even more details git checkout

The screenshot shows a desktop environment with a file manager window open in the foreground. The file manager displays a single file named "README.md" in the "/tmp" directory. The terminal window below it is running a command-line interface, specifically a shell session on a laptop named "lemaitre". The terminal output shows the execution of a "git checkout" command, which creates a new branch named "experimental" and switches to it. The user is prompted to create a new branch to retain commits, and the command "git checkout -b new_branch_name" is demonstrated.

```
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git checkout -b experimental
Switched to branch 'experimental'
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git branch
* experimental
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$ git checkout experimental
Switched to branch 'experimental'
lemaitre@lemaitre-Latitude-E6320:~/Desktop/tmp$
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooo●oooooooo



GIT in even more details

The screenshot shows a window titled 'gitg - tmp (experimental)' with two tabs: 'History' and 'Commit'. The 'History' tab is selected, showing a list of commits. The 'Branch' dropdown is set to 'All branches'. The commit list includes:

Subject	Author	Date
[master] Third changes	Guillaume Lemaitre	jue 19 mar 2015 11:45:59 CET
[experimental] Second changes	Guillaume Lemaitre	jue 19 mar 2015 11:43:59 CET
Initial commit	Guillaume Lemaitre	jue 19 mar 2015 11:39:37 CET



GIT in even more details

The screenshot displays the gitg graphical user interface for managing Git repositories. At the top, there's a terminal window titled "emacs24@lemaitre-Latitude-E6320" showing the command "I modify my readme file" and "my experimental changes". Below it is a file editor window titled "tmp" showing a single file named "README.md" with the same content. On the left, a file browser window titled "Places" shows the current directory is "tmp". The main area of the interface is a commit history viewer titled "gitg - tmp (experimental)". It lists four commits:

Branch	Subject	Author	Date
All branches	experimental Make changes in experimental	Guillaume Lemaitre	jue 19 mar 2015 11:49:37 CET
	master Third changes	Guillaume Lemaitre	jue 19 mar 2015 11:45:59 CET
	Second changes	Guillaume Lemaitre	jue 19 mar 2015 11:43:59 CET
	Initial commit	Guillaume Lemaitre	jue 19 mar 2015 11:39:37 CET

At the bottom of the commit history, there are tabs for "Details", "Changes", and "Files". The "Details" tab is selected, showing fields for "SHA:", "Author:", "Committer:", "Subject:", and "Parent:". A note at the bottom states "Loaded 4 revisions in 0,05s".



GIT in even more details git merge

```
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp
state without impacting any branches by performing another checkout.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:
  git checkout -b new_branch_name

HEAD is now at 31009db... Second changes
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git branch experimental
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git checkout experimental
Switched to branch 'experimental'
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ gitg
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git commit -a
[experimental f54935d] Make changes to experimental
 1 file changed, 2 insertions(+)
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ gitg
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git merge --help
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git checkout master
Switched to branch 'master'
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$ git merge experimental
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
lémaitre@lémaitre-Latitude-E6320:~/Desktop/tmp$
```

History
ooo

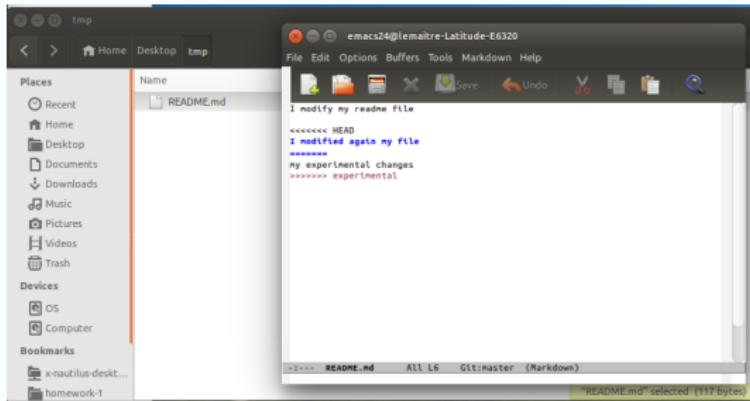
What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details git merge



History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details

GitHub

Owner  glemaitre / Repository name 

Great repository names are short and memorable. Need inspiration? How about [animated-dubstep](#).

Description (optional)

 **Public**
Anyone can see this repository. You choose who can commit.

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

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** | 

Create repository



GIT in even more details
git remote add

```
git checkout -b new_branch_name

HEAD is now at 310ff9b... Second changes
(glenmaitre@Latitude-E6320:~/Desktop/tmp)
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git branch experimental
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git checkout experimental
Switched to branch 'experimental'
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git g
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git commit -a
experimental F54035d) Make changes in experimental
1 file changed, 2 insertions(+)
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git merge --help
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git checkout master
Switched to branch 'master'
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git merge experimental
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git commit -a
[master 5d1fb40] Merge branch 'experimental'
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ y
y: command not found
(glenmaitre@Latitude-E6320:~/Desktop/tmp$ git remote add origin https://github.com/glenmaitre/example-git.git
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo



GIT in even more details git push

```
lemaître@lemaître-Latitude-E6320:~/Desktop/tmp

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced to Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': C
lemaître@lemaître-Latitude-E6320:~/Desktop/tmp$ git push origin master
Username for 'https://github.com': glenmaître
Password for 'https://glenmaître@github.com':
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (13/13), 1.00 KiB | 0 bytes/s, done.
Total 13 (delta 2), reused 0 (delta 0)
To https://github.com/glenmaître/example-git.git
 * [new branch]    master -> master
lemaître@lemaître-Latitude-E6320:~/Desktop/tmp$
```

History
ooo

What is GIT?
ooooo

GIT more in details
ooooo

GIT in even more details
oooooooooooooooooooo●



GIT in even more details git push

glemaître / example-git

Unwatch 1

Description Website

Short description of this repository Website for this repository (optional) Save or Cancel

5 commits 1 branch 0 releases 0 contributors

branch: master example-git / +

Merge branch 'experimental' ...

Guillaume Lemaitre authored 4 minutes ago latest commit 5d1fb40d5c

README.md Make changes in experimental 9 minutes ago

README.md

I modify my readme file

my experimental changes

The screenshot shows a GitHub repository page for 'glemaître / example-git'. At the top, there's a navigation bar with 'Unwatch' and a counter '1'. Below it, there are fields for 'Description' and 'Website'. Underneath, there are summary statistics: 5 commits, 1 branch, 0 releases, and 0 contributors. A dropdown menu shows 'branch: master' and a link to 'example-git / +'.

The main area displays a commit history. The most recent commit was made by Guillaume Lemaitre 4 minutes ago, with a link to the commit hash '5d1fb40d5c'. Below the commit, there's a link to 'README.md' and a note 'Make changes in experimental' with a timestamp of '9 minutes ago'. A preview of the 'README.md' file content is shown, containing the text 'I modify my readme file' and 'my experimental changes'.