



Qu'est-ce que Git?

Git est un logiciel de gestion de versions open source

Débute en 2005 et est maintenant implémenté sur des plateformes en ligne comme **Github**

Garde un historique de toutes les versions du code, des changements et de ceux les ayant effectué

Tout le monde y a accès et peut contribuer au développement



20 février 2020

Formation n°1 : Github

**Github : Site d'hébergement
de répertoires**

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

[Edit](#)[Manage topics](#)

Pour télécharger une copie du répertoire

492 commits

10 branches

0 releases

8 contributors

Branch: develop ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

PaulGOUX27 Merge pull request #134 from ETICINSA Technologies/feature/accessRights ...

Latest commit 677e4e6 2 days ago

.deploy

Fix bugs for PhpStorm documentation

25 days ago

documents

test ok on linux/Travis

2 months ago

logs

Renamed "dev" to "staging", changed incorrect imports, changed code to

last year

src

Change function name

2 days ago

tests

Add tests for route access verifications

3 days ago

.gitignore

this is the end

2 months ago

.travis.yml

Feature/publipostage member inscription (#108)

3 months ago

Dockerfile

addinf ZipArchive class for publipostage

2 months ago

README.md

initial

last year

composer.json

Add mail sending feature (Untested)

29 days ago

composer.lock

Add mail sending feature (Untested)

29 days ago

Fichiers
dans le
répertoire



Commits on Oct 20, 2019

Add tests for route access verifications

 teckwan committed 3 days ago ✓

 7c80a30 

Add verification on routes other than core and adapted inscription tests

 teckwan committed 3 days ago ✓

 fdf62e4 

Fix thrown exception message

 teckwan committed 3 days ago ✗

 f477110 

Add access verification to inscription update and validation

 teckwan committed 3 days ago ✗

 64b6dc0 

Historique des
versions !

Commits on Oct 17, 2019

Merge pull request #133 from ETICINSATechnologies/fix/ValidateConsult...

 PaulGOUX27 committed 6 days ago ✓

Verified  f1b7c28 

Commits on Oct 5, 2019

modify status code validating consultant inscription

 PaulGOUX27 committed 18 days ago ✓

 0959e2d 

Commits on Oct 3, 2019

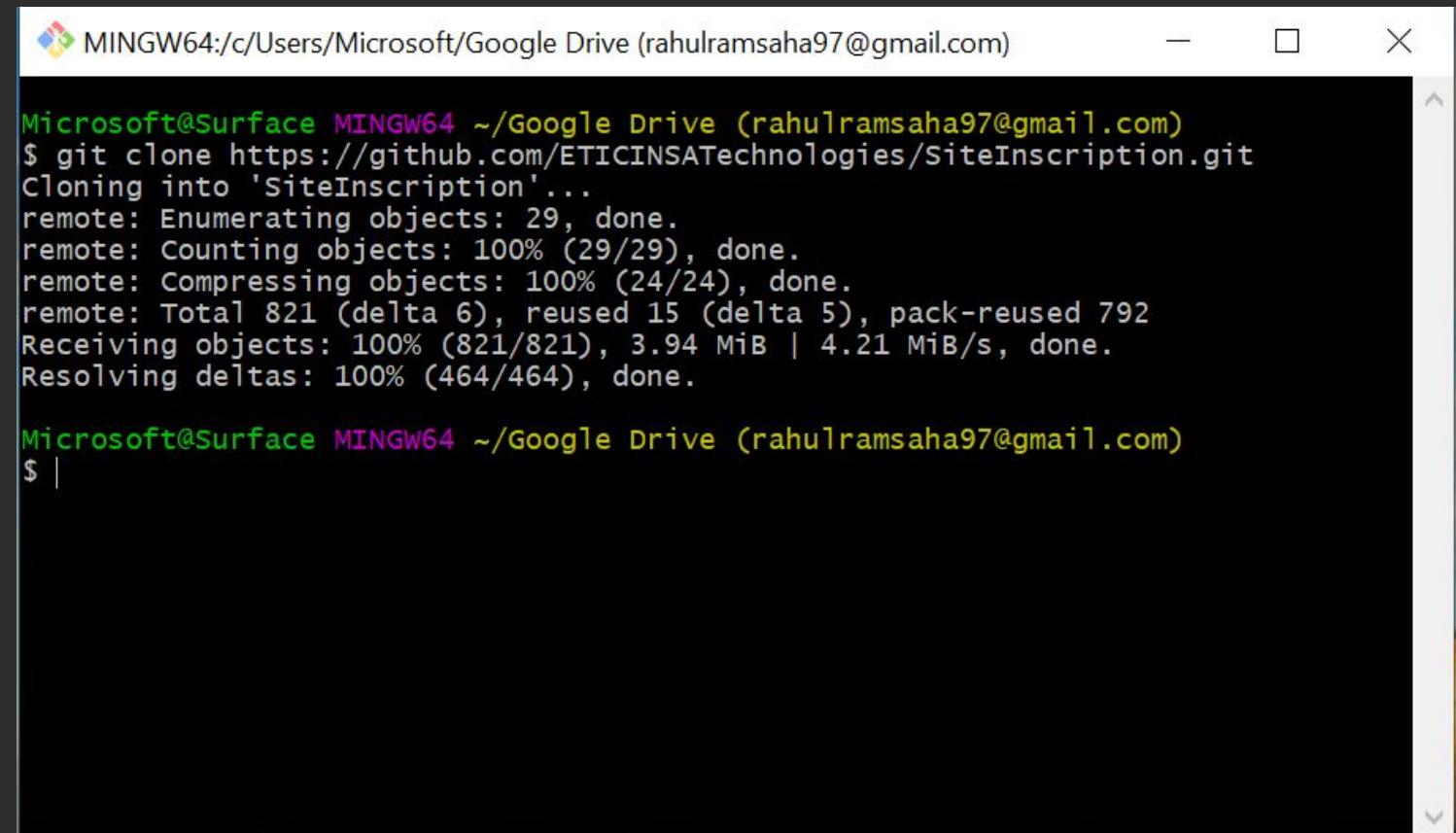
Merge pull request #131 from ETICINSATechnologies/feature/mailSender ...

 PaulGOUX27 committed 20 days ago ✓

Verified  0a2c982 

Petit mot sur Git bash : Installez le dès que possible

- Ligne de commande de type UNIX avec les commandes git pré-installées
- Nécessaire car à l'origine Git ne marchait que dans les environnements UNIX et non sur Windows



```
Microsoft@Surface MINGW64 ~/Google Drive (rahulramsaha97@gmail.com)
$ git clone https://github.com/ETICINSA Technologies/SiteInscription.git
Cloning into 'SiteInscription'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 821 (delta 6), reused 15 (delta 5), pack-reused 792
Receiving objects: 100% (821/821), 3.94 MiB | 4.21 MiB/s, done.
Resolving deltas: 100% (464/464), done.

Microsoft@Surface MINGW64 ~/Google Drive (rahulramsaha97@gmail.com)
$ |
```

Commandes UNIX utiles à savoir :

- **cd [directory_name]** (change directory)
- **ls** (list all files and folders in current directory)



Comment sauvegarder son code sur Git?

Aller sur le dossier du projet -> clic droit -> Git Bash here, puis :

git add .

Sélectionner les fichiers à mettre à jour.
"." signifie qu'on prend tout le dossier

git commit -m *"message expliquant les modifs"*

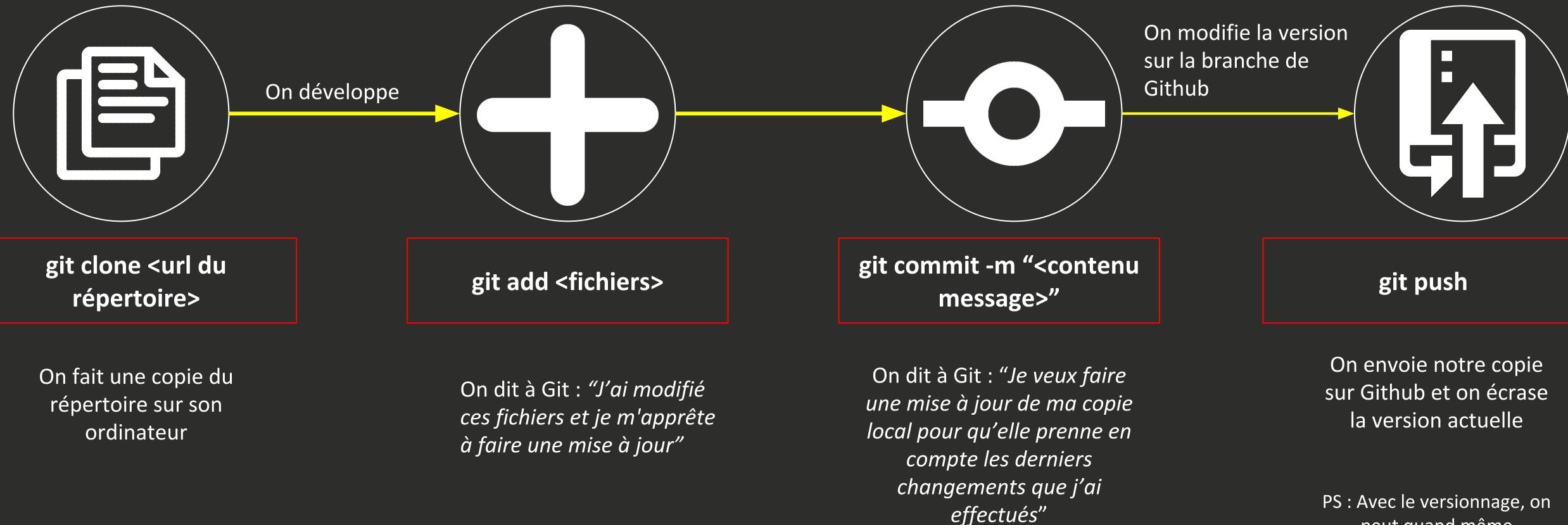
On crée une nouvelle version avec les fichiers modifiés accompagné d'un message explicatif

git push

On "pousse" la nouvelle version sur le répertoire Github en ligne



Les commandes Git : *Timeline*



Notion de localité des fichiers

- ❖ Adrien et Bastien récupèrent le répertoire en ligne avec **git clone** sur leur ordinateur
- ❖ Adrien fait des modifications
- ❖ Adrien add/commit/push sur le repo
- ❖ Le répertoire en ligne est à jour
- ❖ Le code sur l'ordinateur de Bastien s'est-il mis à jour ?

NON !

Il faut mettre à jour son code “en local” par rapport au répertoire en ligne.

Bastien doit lancer la commande : **git pull**

Notion de localité des fichiers

Adrien **git clone**
code version 1 en local

Adrien
code version 1 en local

Adrien
code version 1 en local

Adrien **git pull**
code version 2 en local

code version 1 sur Github

Bastien **git clone**
code version 1 en local

Bastien **git add/commit**
code version 2 en local

Bastien **git push**
code version 2 sur Github

Travailler en équipe avec Git

Que se passe-t-il si quelqu'un a ajouté une nouvelle version du code pendant que je travaillais sur l'ancienne version ?

2 cas :

- modifications sur des lignes de code différents -> pas de soucis
- modifications sur les mêmes lignes -> conflits à résoudre manuellement

Comment éviter cela le plus possible et bien travailler en équipe ?

→ Notion de branche sur Git

Les conflits intra-branches

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package myapplication;
7
8 <<<<< HEAD
9 import java.util.Scanner;
10 =====
11 import java.io.Scanner;
12 >>>>> origin/master
13
14 /**
15 *
16 * @author CCannon
17 */
18 public class MyApplication {
19
20     /**
21      * @param args the command line arguments
22     */
23     public static void main(String[] args) {
24         // TODO code application logic here
25         Scanner keyboard = new Scanner(System.in);
26 <<<<< HEAD
27         System.out.print("Enter an input: ");
28         int input = keyboard.nextInt();
29
30         System.out.println(input * 5);
31 =====
32         System.out.println("Enter user input: ");
33         int operand = keyboard.nextInt();
34 >>>>> origin/master
35     }
36
37 }
38 }
```

modifs en local

code de la branche
origin/master modifié par les
autres



Les branches sur Git

Lister les branches : `git branch`

Créer une branche : `git branch <nom_branche>`

Branch '**cool-feature**'

Changer de branche : `git checkout <nom_branche>`

Branch '**master**'

Branch
'coolest-feature'

Attention : Créer une
branche depuis la **branche
principale** !

Branch
'most-coolest-feature'

Le code présent sur la
branche principale doit
toujours marcher !

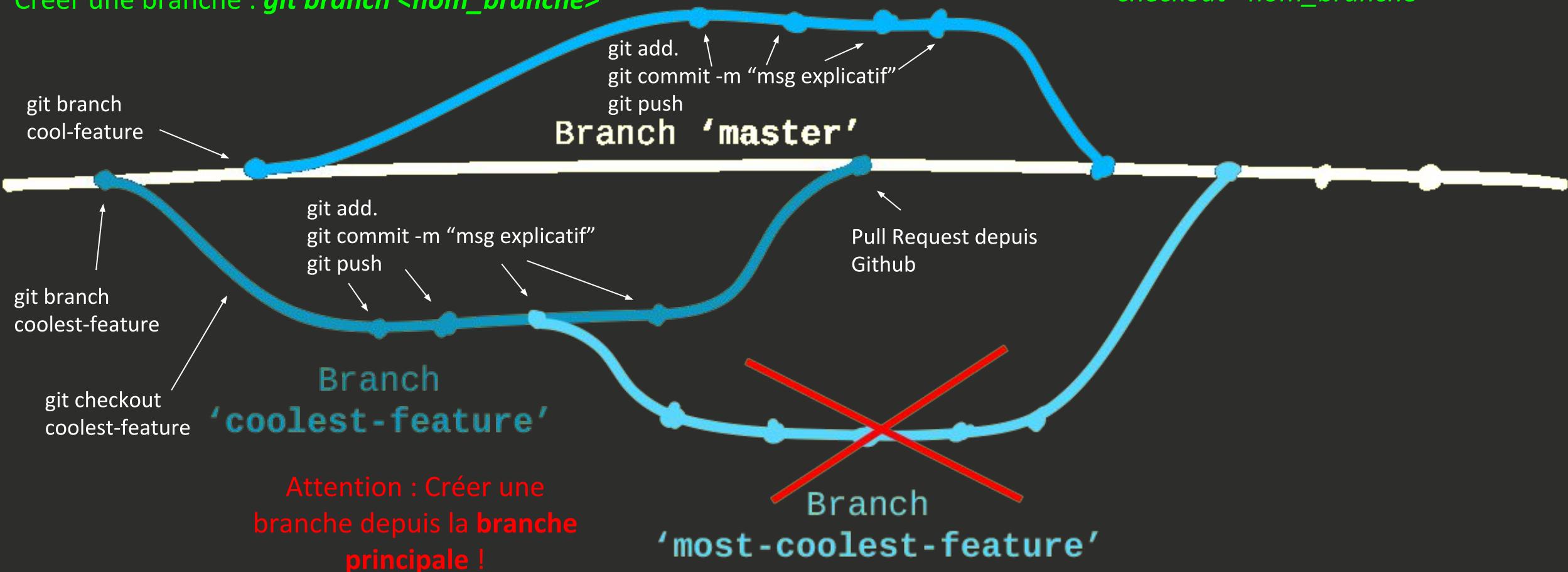


Les branches sur Git

Lister les branches : `git branch`

Créer une branche : `git branch <nom_branche>`

Changer de branche : `git checkout <nom_branche>`



Que se passe-t-il en local lorsqu'on change de branche ?

Adrien travaille sur la branche feature/emoji et Bastien sur la branche feature/gif.

Bastien lance la commande `git checkout feature/emoji`.

Que se passe-t-il au niveau des fichiers de code dans l'ordinateur de Bastien ?

Que se passe-t-il en local lorsqu'on change de branche ?

2 cas possibles :

- première fois qu'il change de branche alors **rien à faire de plus**
Tous ses fichiers en local sont remplacés par les fichiers mis en ligne de la branche feature/emoji
- Pas la première fois, il faut faire **git pull en plus.**
Sans git pull, ses fichiers en local seront remplacés par les fichiers d'une ancienne version de la branche feature/emoji.

Les Pull Request aka PR

- Faire une demande pour fusionner une branche avec la branche principale
- Possibilité de faire une review du code : pour l'Agilité
- Possibilité de lancer des tests auto

Manage topics

246 commits 8 branches 0 releases 7 contributors View license

A We found potential security vulnerabilities in your dependencies. You can see this message because you have been granted [access to security alerts for this repository](#). [View security alerts](#)

Branch: Sad/fileUpload ▾ New pull request Create new file Upload files Find file Clone or download ▾

This branch is 5 commits ahead, 28 commits behind develop. [Pull request](#) [Compare](#)



Les Pull Request aka PR

[torvalds / linux](#)

[Watch](#) 6.9 · [Code](#) · [Pull requests 322](#) · [Actions](#) · [Projects 0](#) · [Security](#) · [Insights](#)

Linux kernel source tree

900,358 commits · 1 branch · 0 packages · 638 releases · [contributors](#) · [View license](#)

Branch: master · [New pull request](#) · [Create new file](#) · [Upload files](#) · [Find file](#) · [Clone or download](#)

Commit	Message	Date
	torvalds Merge branch 'for-5.6-fixes' of git://git.kernel.org/pub/scm/linux/ke...	Latest commit 0a679e1 15 hours ago
	Merge tag 'kbuild-v5.6-2' of git://git.kernel.org/pub/scm/linux/kerne...	2 days ago
	LICENSES: Rename other to deprecated	9 months ago
	Merge tag 'kbuild-v5.6-2' of git://git.kernel.org/pub/scm/linux/kerne...	2 days ago
	Merge tag 'block-5.6-2020-02-05' of git://git.kernel.dk/linux-block	5 days ago
	certs: Add wrapper function to check blacklisted binary hash	3 months ago
	treewide: remove redundant IS_ERR() before error code check	8 days ago
	Merge tag 'kbuild-v5.6-2' of git://git.kernel.org/pub/scm/linux/kerne...	2 days ago
	Merge tag 'kbuild-v5.6-2' of git://git.kernel.org/pub/scm/linux/kerne...	2 days ago
	Merge tag 'zonefs-5.6-rc1' of git://git.kernel.org/pub/scm/linux/kern...	2 days ago
	Merge tag 'trace-v5.6-2' of git://git.kernel.org/pub/scm/linux/kernel...	5 days ago
	proc: convert everything to "struct proc_ops"	8 days ago
	Merge branch 'for-5.6-fixes' of git://git.kernel.org/pub/scm/linux/ke...	15 hours ago

Les Pull Request aka PR

 **videolan / vlc** Watch ▾ 488 Star 5.7k Fork 2.1k

Code Pull requests 0 Actions Security Insights

VLC media player - All pull requests are ignored, please follow https://wiki.videolan.org/Sending_Patches_VLC/ <http://www.videolan.org/vlc>

c vlc multimedia framework open-source gplv2 lgpl

84,250 commits 1 branch 0 packages 50 releases 535 contributors View license

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

Commit	Message	Time
	robUx4 d3d11_device: cleanup exit on error	Latest commit 885e4eb 1 hour ago
	autotools Fix .gitignore	2 years ago
	bin macOS packaging: Use DESTDIR install for packaging	2 months ago
	compat Remove old mail address from sources	5 months ago
	contrib: add /usr/lib/TRIPLET/pkgconfig to PKG_CONFIG_LIBDIR	11 days ago
	doc: add the OpenGL callbacks libvlc player sources	3 hours ago
	extras macOS build.sh: Add check for python3	9 days ago
	include threads: forbid the use of vlc_cleanup_push/pop in portable C++ code	6 hours ago
	libvlc: merge libvlc_video_direct3d_set_callbacks into libvlc_video_s...	yesterday

Les conflits inter-branches

```
1 Octo-Repo
2 =====
3
4 Contact @octo-org/core for questions about this repository.
5
6 ### Installing
7
8 Instructions for Installing!
9
10 ### Contributing
11
12 See the [CONTRIBUTING file](../CONTRIBUTING) for guidance on contributing to this project.
13
14 ### Authors
15
16 <<<< add-emoji ←
17 We're all authors :star: :zap: :sparkles:
18 =====
19 We're all authors :star: :zap: :heart:
20 >>>> master ←
21
22 This commit will be verified!
23
```

modifs de la branche
add-emoji

code de la branche principale master



Conclusion

2 objectifs pour cette formation :

- S'initier aux commandes de bases pour survivre
- Pouvoir comprendre la documentation en ligne par vous-même

Git Cheatsheet

Résumé de toutes les commandes Git

<https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>

Les plus commandes les plus importantes selon nous

- git clone
- git add/commit/push
- git pull
- git status/diff
- git branch
- git checkout

IN CASE OF FIRE



git add

git commit

git push

exit building



Merci de votre
attention

Des questions ?