

[Accueil](#) > [Cours](#) > [Gérez du code avec Git et GitHub](#) > Travaillez avec un dépôt distant

## Gérez du code avec Git et GitHub

 6 heures  Facile

Mis à jour le 19/05/2022



## Travaillez avec un dépôt distant

### Accédez à un dépôt distant



Maintenant que vous savez travailler à partir d'un projet local et que vous maîtrisez le système de branches, voyons comment **accéder à un dépôt distant** et le copier en local.

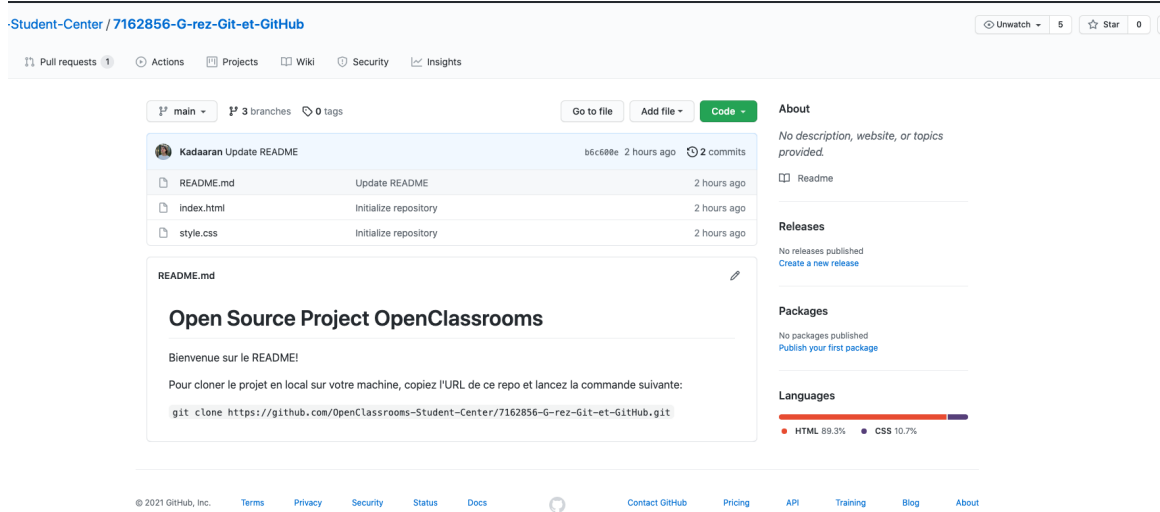
Je vous explique comment faire dans la vidéo ci-dessous et vous invite également à suivre les différentes étapes sous la vidéo :

01:26

Imaginons que vous deviez travailler sur un projet avec des amis. Ces derniers ont créé le repository sur GitHub. Il est temps pour vous de récupérer le code pour apporter vos modifications :

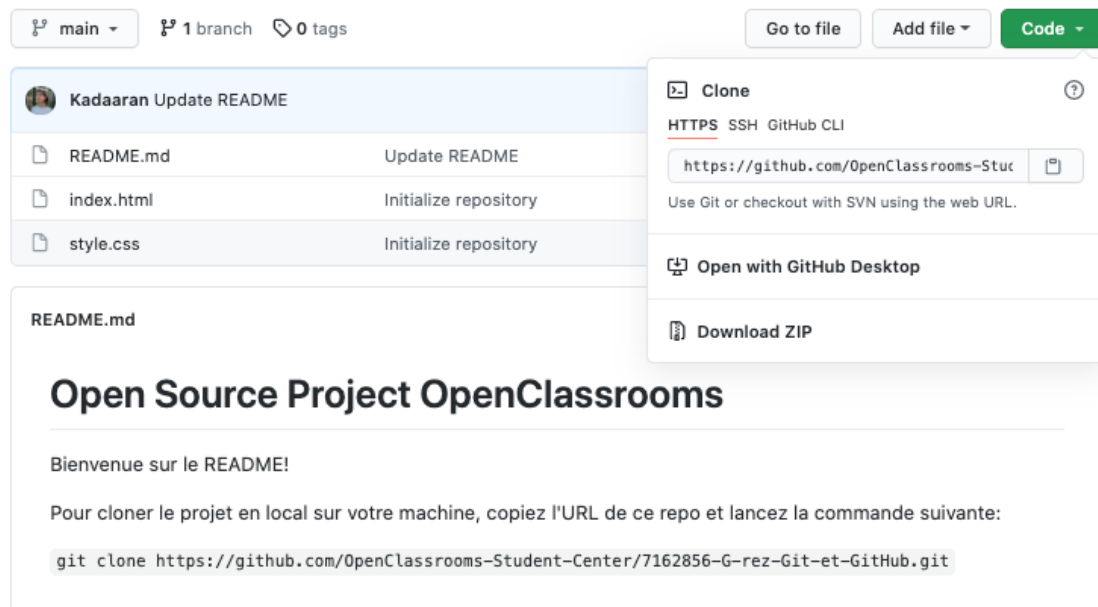
Tout d'abord, vous allez récupérer l'URL du dépôt distant : cela se passe sur GitHub !

- Accédez au [dépôt distant](#) sur lequel vous souhaitez travailler.



Accédez au dépôt distant sur lequel vous souhaitez travailler

- Cliquez sur le bouton "Code" pour copier le dépôt, puis sur la section HTTPS.



Cliquez sur "Code" pour copier le dépôt

Et tadam, vous avez la fameuse URL dont nous allons avoir besoin. Copiez-la dans le presse-papier.

- Retournez sur Git Bash, et tapez la commande suivante :

```
git clone https://github.com/OpenClassrooms-Student-Center/7162856-G-rez-Git-et-GitHub.git
```

Le nom de ce repository est beaucoup trop long ! Je vous conseille de créer un raccourci pour gagner du temps.

- Lancez la commande suivante :

```
git remote add OC https://github.com/OpenClassrooms-Student-Center/7162856-G-rez-Git-et-GitHub.git
```

OC représente le nom court que vous utiliserez ensuite pour appeler votre dépôt. Appelez-le comme bon vous semble, mais un nom court et simple est toujours plus facile.

Cette ligne ne permet pas de copier le dépôt, mais permet de dire au dépôt que l'on pointe vers le dépôt distant.

Et voilà! Vous pouvez maintenant vous diriger dans le dossier sur votre ordinateur et accéder au code.

## Mettez à jour le dépôt en local

Imaginons que durant la semaine, un de vos amis ait ajouté des modifications sur la branche main et que vous souhaitiez les récupérer. Comment faire ? Utilisez la commande `git pull` :

```
git pull origin main
```

Cela devrait afficher ces petites lignes de commande !

```
→ git pull OC main
From https://github.com/OpenClassrooms-Student-Center/7162856-G-rez-Git-et-GitHub
 * branch          main          -> FETCH_HEAD
 * [new branch]    main          -> OC/main
Already up to date.
```

Résultat de la commande git pull

C'est bon signe, la magie a opéré ! 🎉

Vous devriez maintenant avoir l'ensemble des fichiers et dossiers du repository à jour dans votre répertoire courant. La duplication s'effectue très rapidement, puisque les fichiers vont être compressés avant le transfert. Vous allez donc les recevoir à la vitesse de l'éclair ! Enfin... rapidement, quoi.

Vous pouvez maintenant vous mettre à vos développements, et cela sans faire de bêtises ! Mais avant, vérifions quelques détails et parcourons ce nouveau repository.

## Collaborez sur GitHub

Je vous disais dans les chapitres précédents que GitHub est un excellent outil de collaboration. Mais, en quoi GitHub aide les développeurs à collaborer ?

GitHub est avant tout une interface, son rôle est de faciliter la collaboration en ajoutant un aspect design, facile, ludique et simple d'utilisation pour les développeurs. Jusqu'ici nous avons observé les changements effectués sur des branches via des lignes de commande. Si nous travaillons à plusieurs sur un projet, il serait bien plus pratique d'accéder aux modifications sans ligne de commande, n'est-ce pas ?

## Vérifiez les branches de votre projet

Commençons par vérifier les différentes branches du projet : vos collègues travaillent peut-être sur de nouvelles fonctionnalités. Pour vous en assurer, lancez la commande `git branch` .

Vous devriez voir :

```
* main
new-version-css
update-readme
```

! Nous avons en plus de la branche principale main, deux autres branches : `new-version-css` et `update-readme` : vos amis ont déjà envoyé (ou push en anglais) des branches, certainement pour créer de nouvelles fonctionnalités !

Dans les chapitres précédents, git merge nous permettait de fusionner les modifications de notre branche avec la branche principale.

Mais dans un contexte professionnel, c'est un peu plus compliqué ! 😊 Lorsque vous travaillez en équipe sur un repository, la branche principale est souvent bloquée. Vous ne pouvez pas pusher directement votre code sans qu'il soit vérifié. Vous ne pouvez donc pas fusionner vos modifications vous-même !

? Hein !? Mais comment on fait alors ? On ne va pas demander l'autorisation quand même ?

Eh bien presque... On fait ce qu'on appelle une **pull request** !

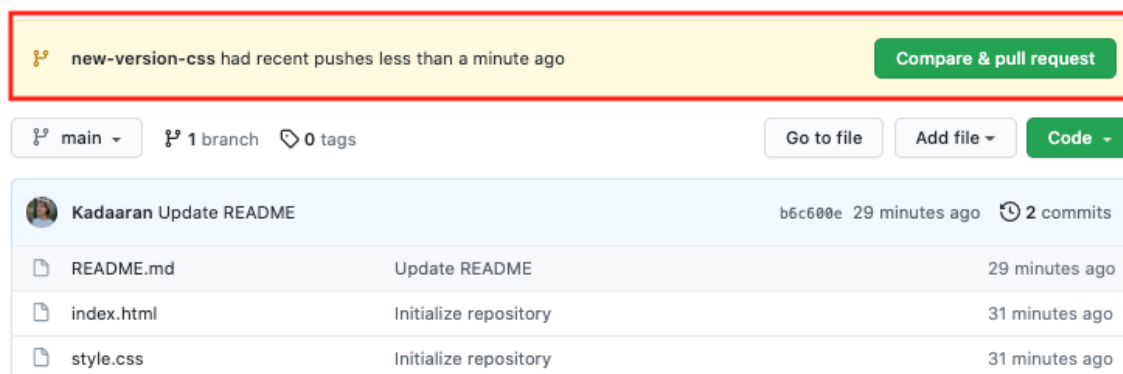
## Réalisez une pull request

Une **pull request**, ou *demande de pull*, en français, est une fonctionnalité de GitHub qui permet de demander aux propriétaires d'un repository l'autorisation de fusionner nos changements sur la branche principale ou toute autre branche sur laquelle on souhaite apporter nos modifications.

Donc si vous créez une pull request, vous avez au préalable :

1. Créé une nouvelle branche.
2. Envoyé votre code sur cette même branche.

D'ailleurs, lorsque ces deux conditions sont remplies, un bandeau apparaît à l'écran pour vous suggérer de créer une pull request :



Le bouton "Compare & pull request" vous suggère de créer une pull request

Parfait !

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: new-version-css ✓ **Able to merge.** These branches can be automatically merged.

Update style, turn all pages into blue

Write

Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

[GitHub Community Guidelines](#)

1 commit

1 file changed

0 comments

1 contributor

Commits on Aug 22, 2021

Update style, turn all pages into blue 1bcf85d

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

style.css

@@ -1,3 +1,3 @@

1 1 h1 {

2 - color: red;

2 + color: blue;

3 3 }

La fenêtre Open a pull request s'ouvre

- Ajoutez un commentaire pour expliquer les raisons de vos modifications.

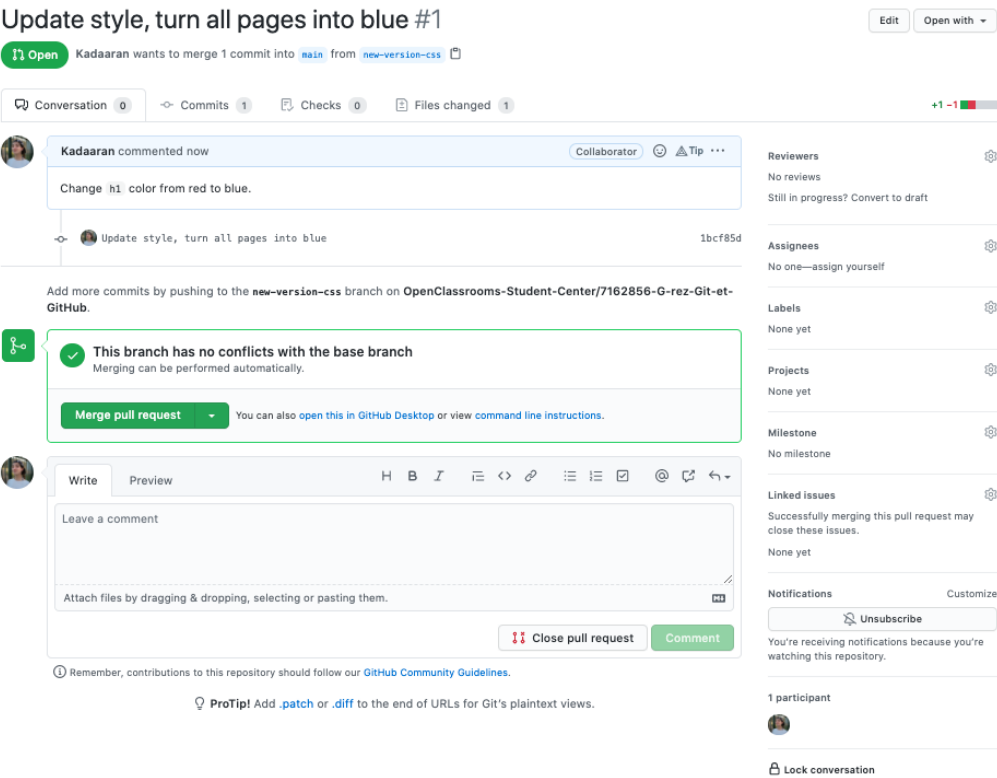
Ici, votre modification consiste à changer la couleur d'une balise. Je vous conseille donc de commenter : "Change h1 color from red to blue".

GitHub indique les modifications effectuées par un code couleur. Les lignes en rouge indiquent une suppression, et les lignes vertes une addition. Ici on voit bien qu'il y eu un changement sur une ligne, l'attribut red a été supprimé, et a été remplacé par blue.

- Cliquez sur **Create pull request** pour valider la pull request.
- Et voilà une belle pull request prête à l'emploi :

<https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github/7165692-travaillez-avec-un-depot-distant>

5/9

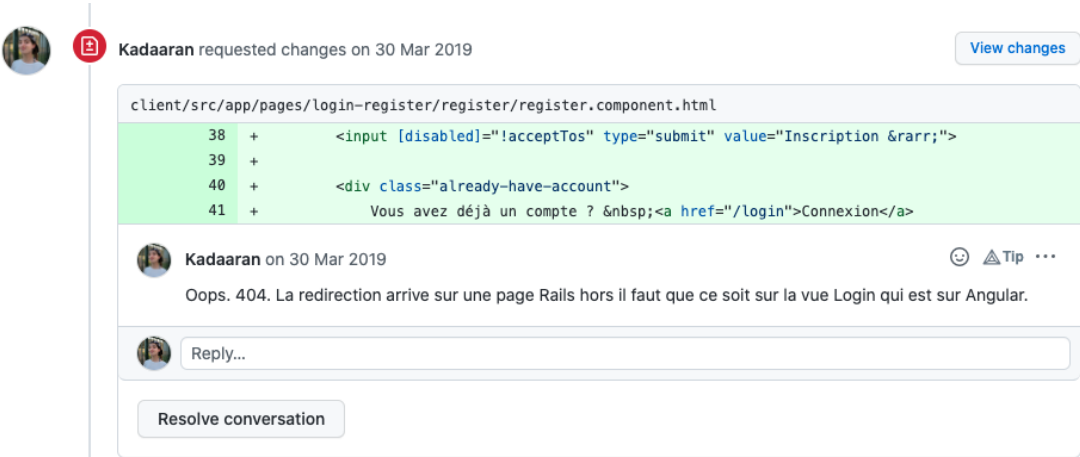


Vous avez créé votre pull request

Votre pull request est créée et peut être fusionnée avec la branche main. Vous pouvez y accéder grâce à [ce lien](#).

Demandez une relecture de code

Sur des projets d’envergure, il peut arriver que votre code ne puisse être fusionné sur la branche principale sans être relu et validé par d’autres membres du projet. C’est ce qu’on appelle une *Code Review*, ou *revue de code*, en français. Cela permet de prévenir les erreurs éventuelles, de discuter sur un choix, une prise de position ou même de poser des questions.



La revue de code permet d’échanger sur une Pull Request

Et voilà, vous savez maintenant collaborer en équipe grâce à GitHub !

N’hésitez pas à revoir ces différentes étapes dans la vidéo ci-dessous :

02-26

## En résumé



- Sur GitHub, nous pouvons récupérer l'URL d'un dépôt distant.
- `git clone` permet de copier en local un dépôt distant.
- `git remote add` permet de lier un dépôt à un "nom court", pour une plus grande facilité d'utilisation.
- `git pull` permet de dupliquer un dépôt GitHub en local.
- Une Pull Request permet de demander à fusionner votre code sur la branche principale.

*Cette partie vous a permis de découvrir la gestion de versions pour le développement, et de prendre en main Git. Bien sûr, il vous arrivera de faire des erreurs et nous allons voir dans la partie suivante comment les réparer!*

*Mais avant cela, passons au quiz !*

J'ai terminé ce chapitre et je passe au suivant



Appréhendez le système de branches

Quiz : Utiliser les commandes de base de Git

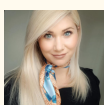


### Les professeurs



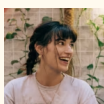
**Luc Bourrat**

Mentor et développeur web



**Tiffany Lestroubac**

Développeur Fullstack / Mentor / Evaluatrice / Rédactrice



**Kassandre Pedro**

Backend Engineer at Pexels | Ruby on Rails Developer



**Mila Paul**

I am a researcher and teacher with a PhD in Cyber Operations. I also love trekking, scuba diving and traveling!

  
OPENCCLASSROOMS

OPPORTUNITÉS



AIDE



POUR LES ENTREPRISES



EN PLUS



Français

Télécharger dans  
l'App Store



