

*Service web d'hébergement et de gestion social de développement de logiciels.*

**SOMMAIRE**

[1. Introduction. GitHub :Utile ou pas… ? 2](#_Toc11321200)

[2. Logiciels nécessaires 2](#_Toc11321201)

[2.1. Logiciel de traitement du projet (Windows) 2](#_Toc11321202)

[2.2. Logiciels de travail (local) 2](#_Toc11321203)

[3. GItHub 3](#_Toc11321204)

[3.1. A faire par tous. 3](#_Toc11321205)

[3.2. Pour le professeur. 3](#_Toc11321206)

[3.2.1. Par utilisation de l'IHM. 4](#_Toc11321207)

[3.2.2. En lignes de commande 4](#_Toc11321208)

[3.3. Pour les élèves 8](#_Toc11321209)

[4. Résumé : exemples pour ce fichier. 11](#_Toc11321210)

[5. Mise à jour suite à un Fork 12](#_Toc11321211)

[5.1. Si aucune modification n'a été faites après le "Fork" 12](#_Toc11321212)

[5.2. Si il y a eu des modifications après le "Fork" 14](#_Toc11321213)

[5.2.1. Méthode à l'arrache : 14](#_Toc11321214)

[5.2.2. Méthode 2 14](#_Toc11321215)

[6. Petites erreurs. 15](#_Toc11321216)

1. Introduction. GitHub : Utile ou pas… ?

Force de GitHub :

* Collaboratif,
* Horodatage des modifications,
* Noms des contributeurs (et donc des élèves passifs),
* Hébergement distant (pas de clé USB ou stockage des projets aléatoire, …),
* Récupération aisées des projets,
* Visualisation simplifiée des modifications des camarades,
* …

Inconvénients de Git Hub :

* Difficulté d'utilisation (pour les élèves),
* Clonage, push, pull en ligne de commande,
* …

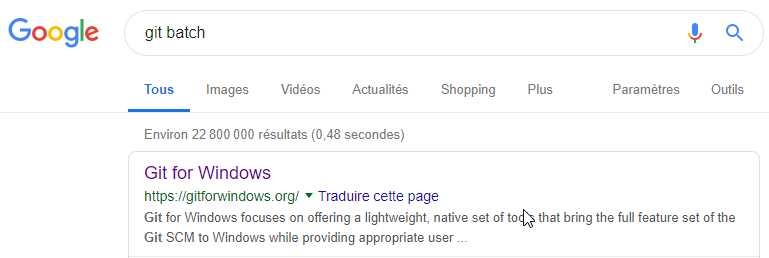
Il faut voir GitHub comme Dropbox qui fait des mises à jour permanentes en vérifiant les dates des fichiers.

Ici les mises à jour sur le serveur se font manuellement (**commit**) par l'utilisateur. GitHub permet de conserver les différentes version du projet.

1. Logiciels nécessaires
   1. Logiciel de traitement du projet (Windows)

Pour télécharger (**cloner**), déposer ses modifications (**push**), télécharger les modifications des copains (**pull**).

Après un **clonage** (ou un **pull**) on travail en local.



* 1. Logiciels de travail (local)

**IDE** en fonction du langage de programmation choisi pour le projet.

Pour moi utilisation d'Anaconda qui inclus, entre autre, l'IDLE **Spyder** pour programmer en **Python** ainsi que **Jupiter** pour créer des **NoteBook**.

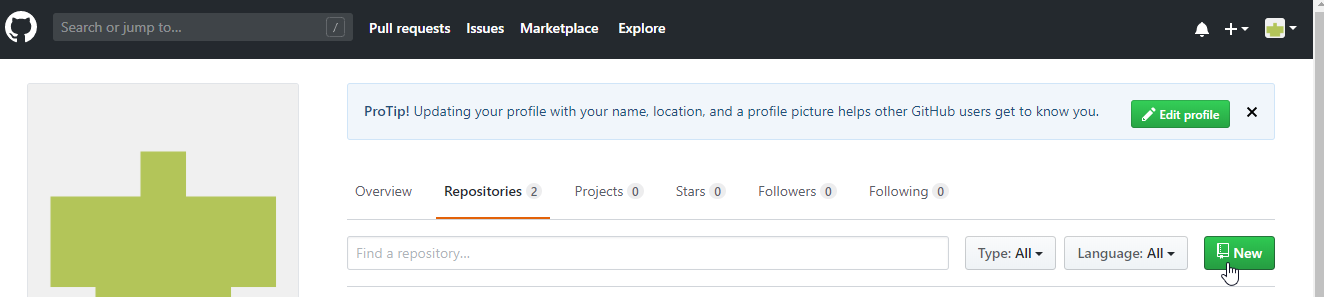
1. GItHub
   1. A faire par tous.

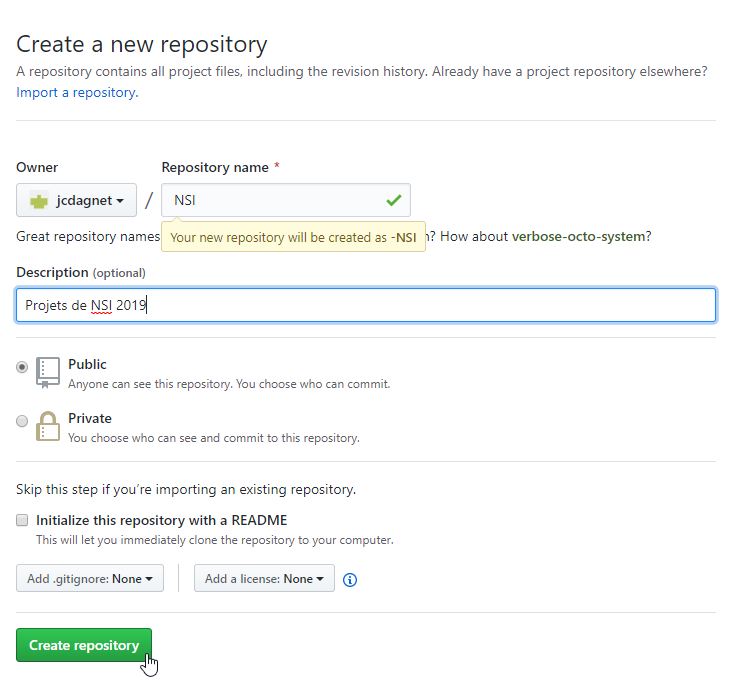
Création d'un compte **GitHub**. Super simple pas besoin de didacticiel ☺

Attention, choisir un nom d'utilisateur explicite car c'est avec ce dernier qu'on peut vous retrouver.

* 1. Pour le professeur.

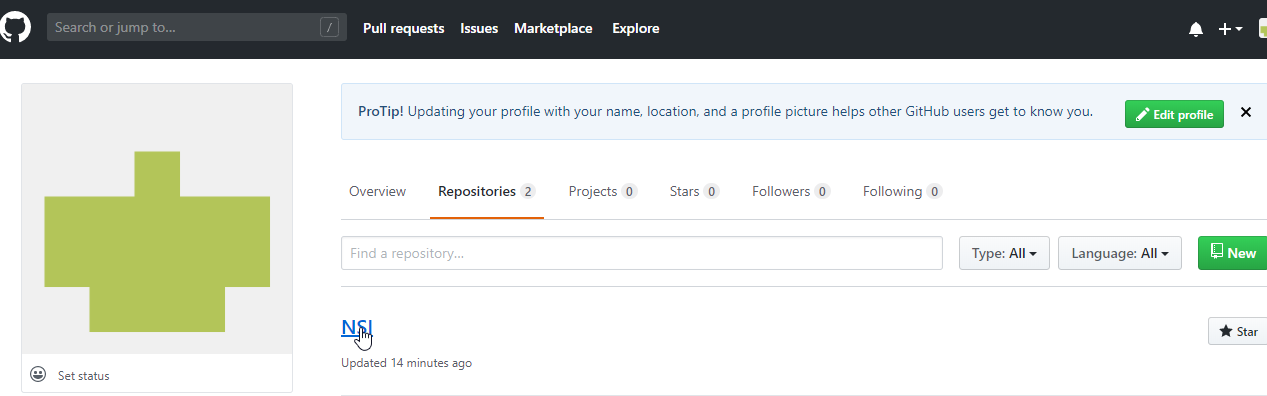
Création d'un "**repositories**" (attention pas un "**projet"** qui, lui, inclus plusieurs **repositories**)

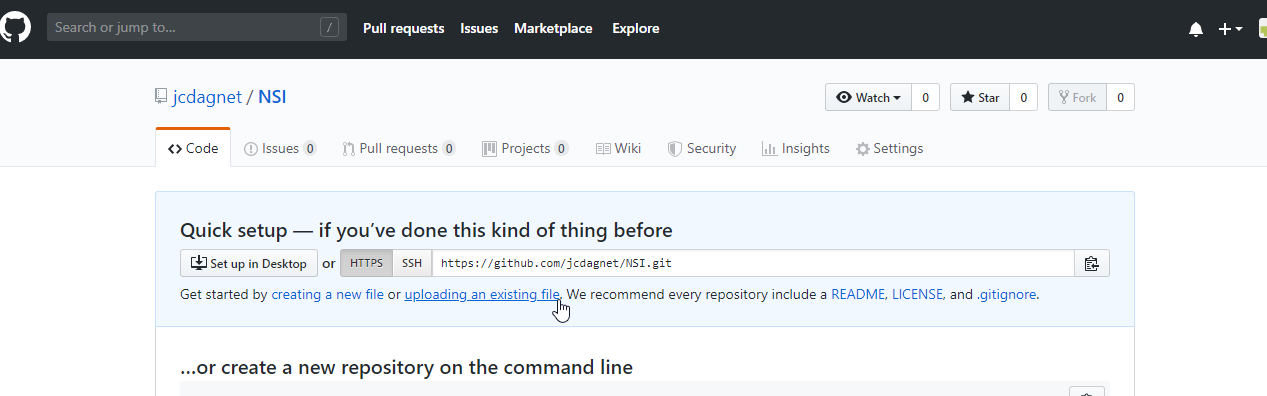


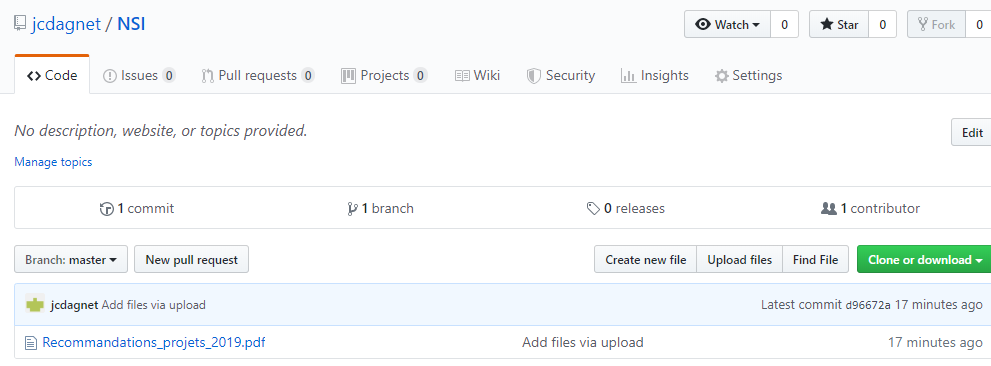


* + 1. Par utilisation de l'IHM.

Télécharger les fichiers / dossiers (j'ai pas réussi à insérer un dossier) nécessaires.

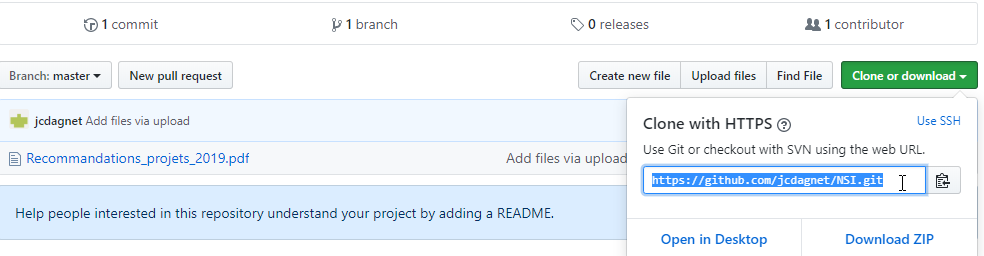






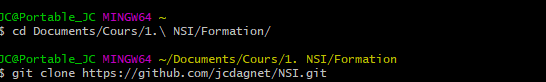
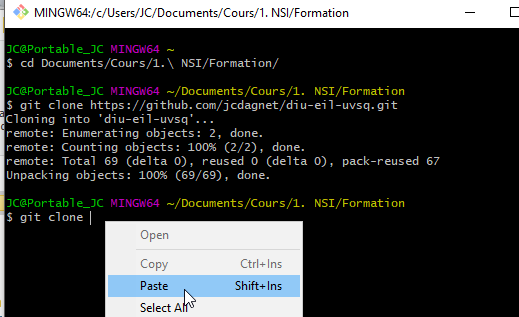
* + 1. En lignes de commande

Il faut "**cloner**" le "**repositories**" par l'URL que l'on trouve sur le site web :



Ouvrir le logiciel Git Bach :

Descendre dans le dossier où voulez créer notre dossier local (commande **cd**), puis **cloner** le "**repositories**"



Création du clone sur mon disque

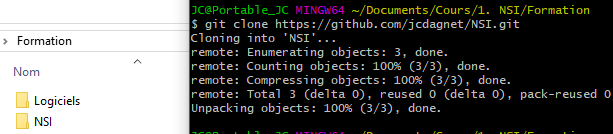
(en local)

Suppression de l'espace

***Aide pour descendre dans son dossier :***

|  |  |
| --- | --- |
| **Commandes** | **Explications** |
| **$ pwd** | Me permet de savoir où je suis ! |
| **$ cd ..** | Remonte d'un niveau dans mon arborescence |
| **$ cd Doc + Tab** | Il complète automatiquement ($ cd Documents/) |
| **$ cd Doc + Tab Tab** | Il me dit ce qui est dans le dossier |

Mon **clone** est maintenant créé en local :



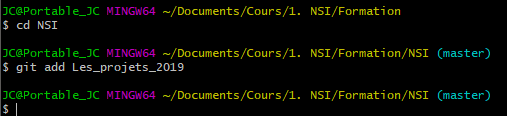
On modifie le contenu de notre projet… Modification du projet, ajout de fichier(s) …

Pour ajouter les modifications au projet global, il faut :

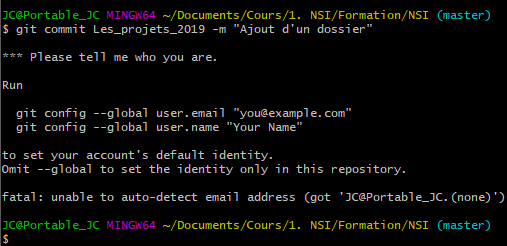
* Ajoutez les modifications à l'index (le .git qui s'est créé dans votre dossier ) avec la commande ***$git add***
* Créez un **commit** pour identifier les modifications créées avec la commande ***$git commit***
* Modifiez le contenu du projet commun pour en faire profiter tout le monde avec la commande ***$git push***

Pour ce faire :

On entre dans le dossier **cloné** et on ajoute (**git add**) nos fichiers et/ou dossiers à l'index (mais attention toujours pas au en ligne à nos ami(e)s).



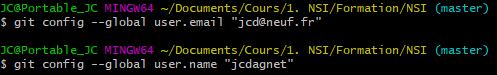
Puis on fait un **commit** (Waouh on y est presque) :



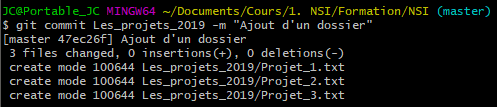
-m = ????????????????????

"Ajout…" = texte qui apparaitra sur le site GitHub comme notification de la MAJ

Correction de l'erreur éventuelle, il faut donner son adresse et son nom :



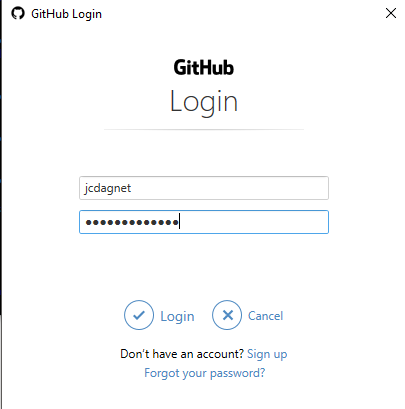
On recommence le commit (obligatoire car le "push" ne travail que sur les commits, je l'ai déjà dit, il faut suivre !!!!)



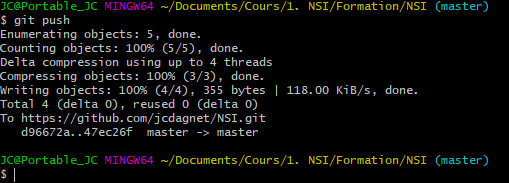
Enfin on envoi le tout vers le serveur :

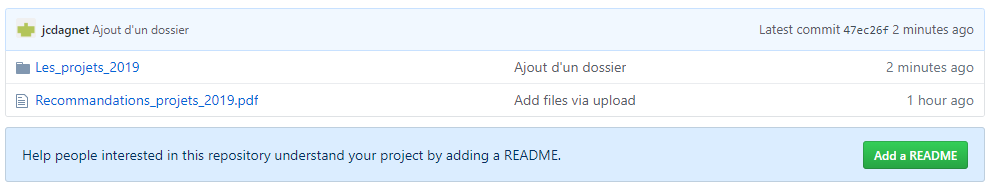


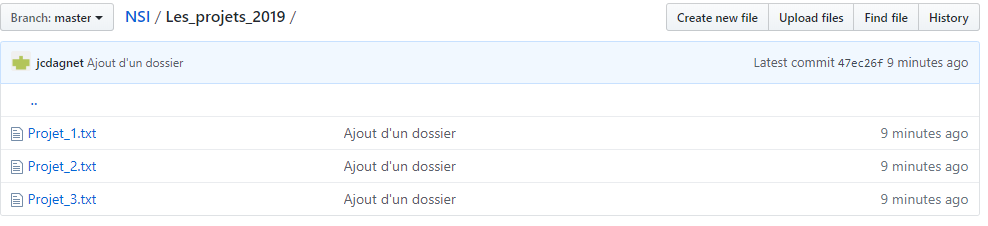
Eventuellement on vous invite de nouveau à vous identifier ! Faites-le ☺



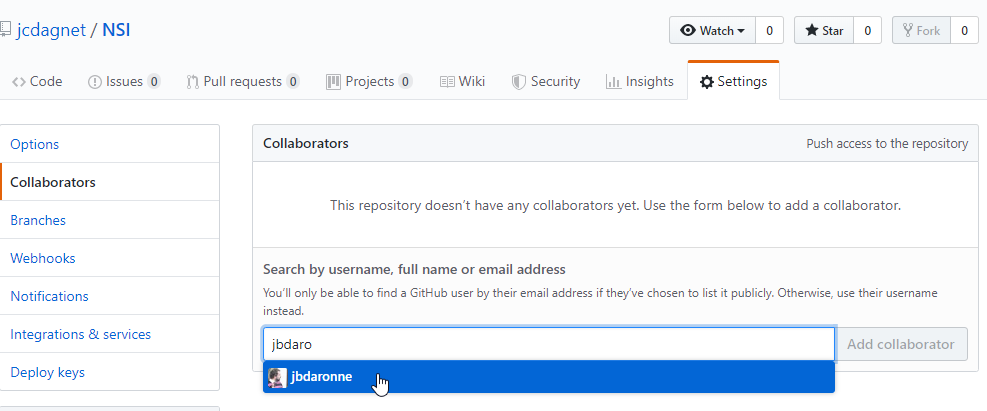
C'est fait :







Il faut ensuite ajouter des collaborateurs :



* 1. Pour les élèves

Faire un "**Fork"** pour copier le projet sur votre espace :

Elève 1

jcdagnet/ISN

GitHub



**fork**

Recommandations

Les\_projets\_2019

Recommandations

Les\_projets\_2019

Local

Chaque élève fait ensuite un **clone** en ligne de commande :

Recommandations

Les\_projets\_2019

Elève 3

Elève 1

Recommandations

Les\_projets\_2019

Local

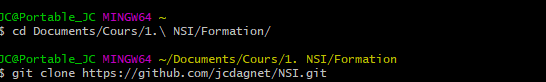
Elève 1

jcdagnet/ISN

GitHub

Recommandations

Les\_projets\_2019



**Clone**

Recommandations

Les\_projets\_2019

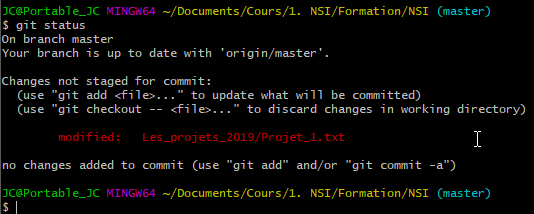
Elève 2

Recommandations

Les\_projets\_2019

Ils possèdent maintenant chacun le projet en local et peuvent le modifier à souhait.

Une fois modifié, ils peuvent voir leurs propres modifications avec un " ***$ git status***"



Il faut maintenant remonter le projet sur le site **GitHub**.

Exemple, suite à des modifications de l'élève 3, il faut :

* Faire un "***add***" (pour ajouter à l'index),
* créer un "***commit***" (sur lequel le "**push**" travaillera),
* puis le "***push"***

Elève 2

Elève 1

Recommandations

Les\_projets\_2019

Recommandations

Les\_projets\_2019

Elève 3

Local

**add**

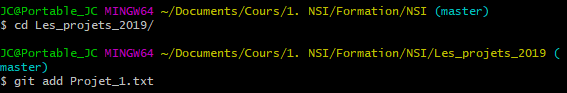
Recommandations\_v2

Les\_projets\_2019

*Index du fichier "****.git****" créé dans votre dossier*

*Identification des modifications*

**commit**





Recommandations

Les\_projets\_2019

Recommandations

Les\_projets\_2019

Elève 1

jcdagnet/ISN

GitHub

Elève 1

Recommandations

Les\_projets\_2019

Elève 3

*Index du fichier "****.git****" créé dans votre dossier*

Recommandations

Les\_projets\_2019

Recommandations

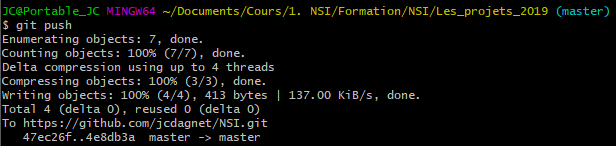
Les\_projets\_2019

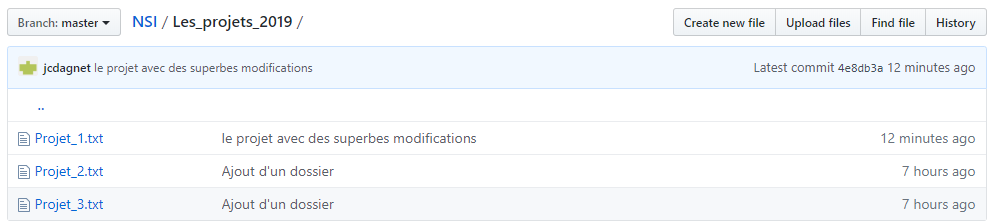
Elève 2

Local

**Push**

*Identification des modifications*





Reste aux autres élèves à faire un "**pull**" pour récupérer les modifications de l'élève 3



Elève 1

Recommandations

Les\_projets\_2019

Elève 3

Recommandations

Les\_projets\_2019

Recommandations

Les\_projets\_2019

Elève 2

Recommandations

Les\_projets\_2019

Recommandations

Les\_projets\_2019

Elève 1

jcdagnet/ISN

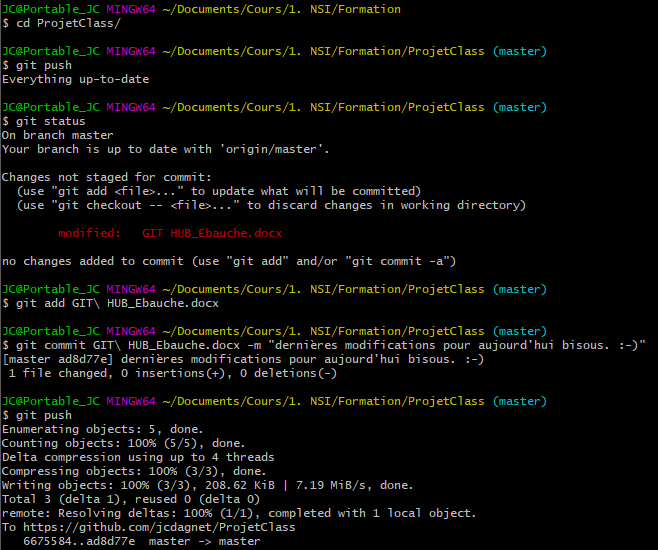
**Pull**

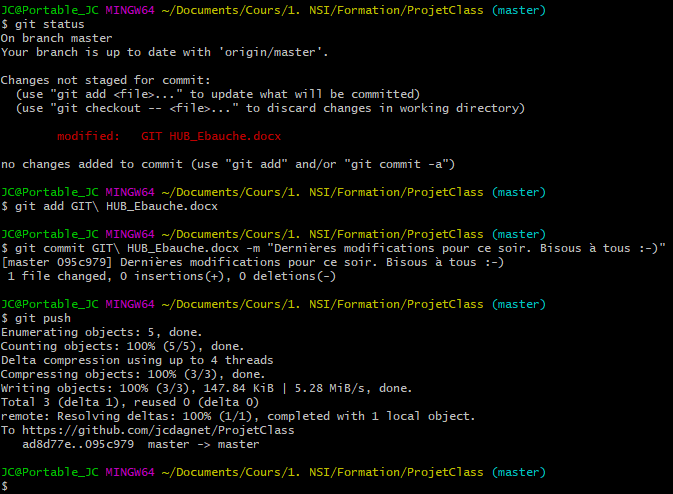


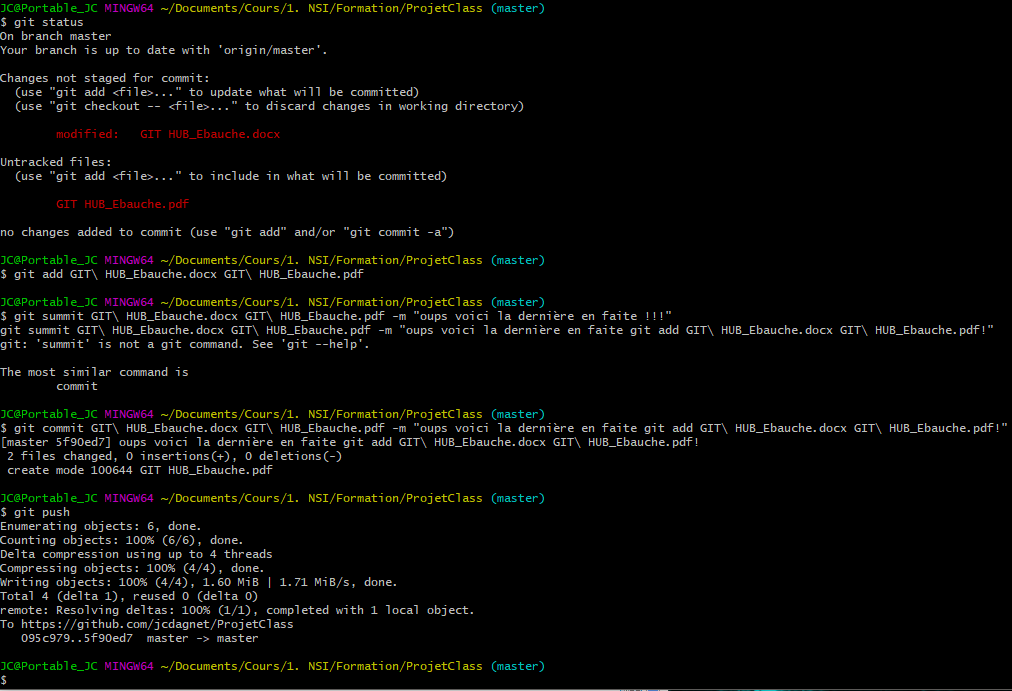
Local



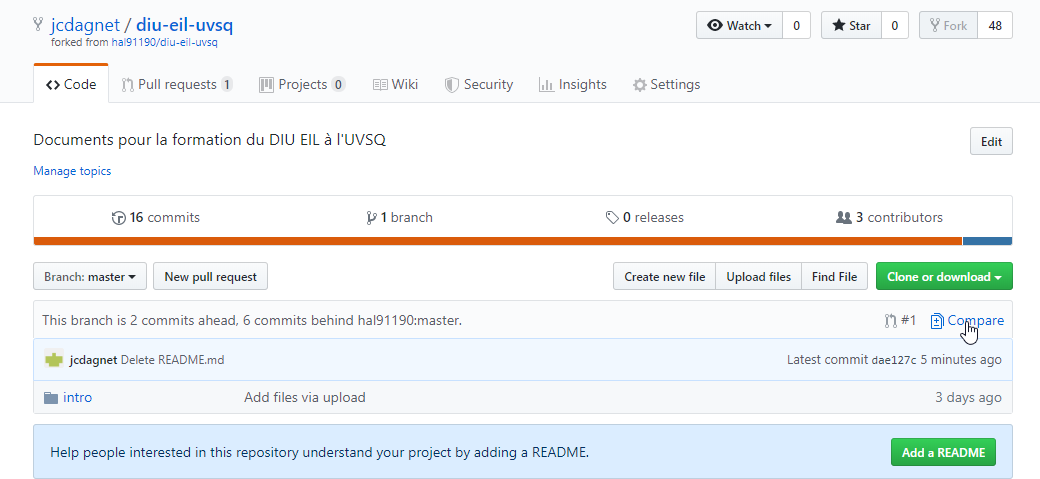
1. Résumé : exemples pour ce fichier.

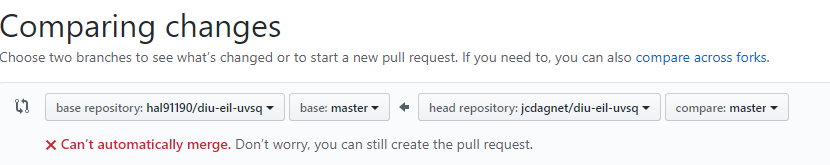




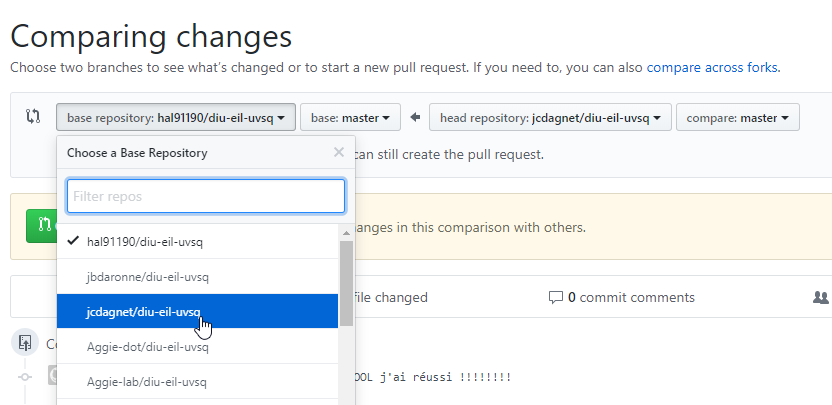
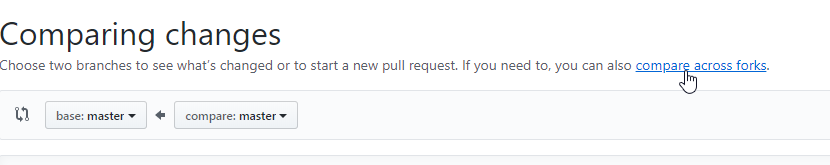


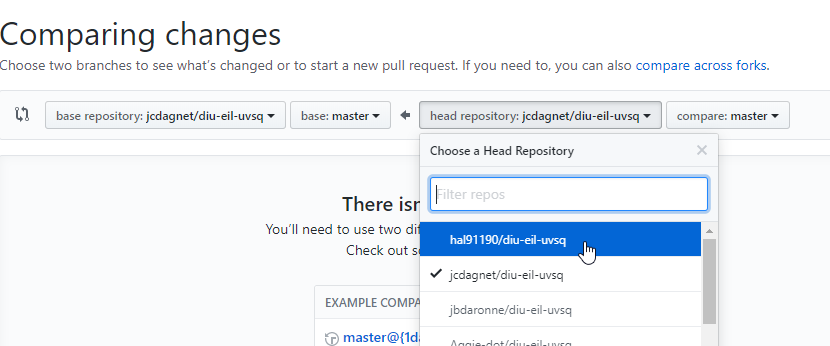
1. Mise à jour suite à un Fork
   1. Si aucune modification n'a été faites après le "Fork"

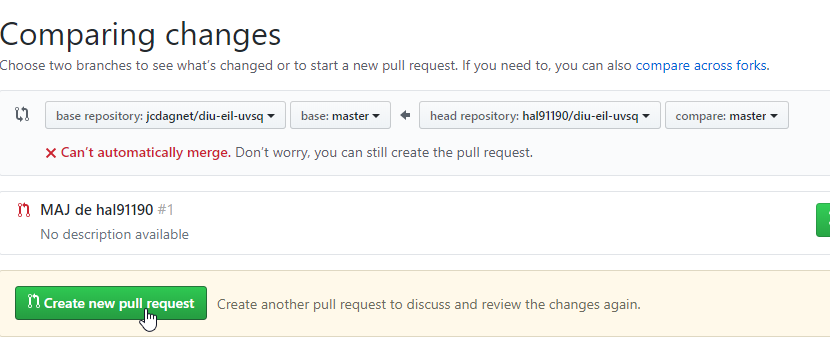
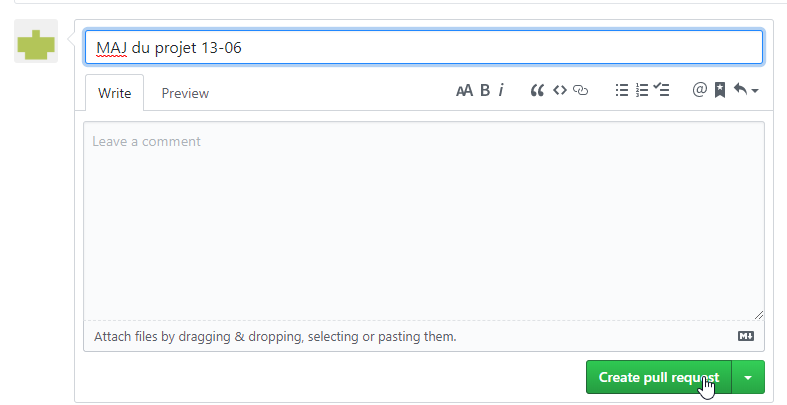


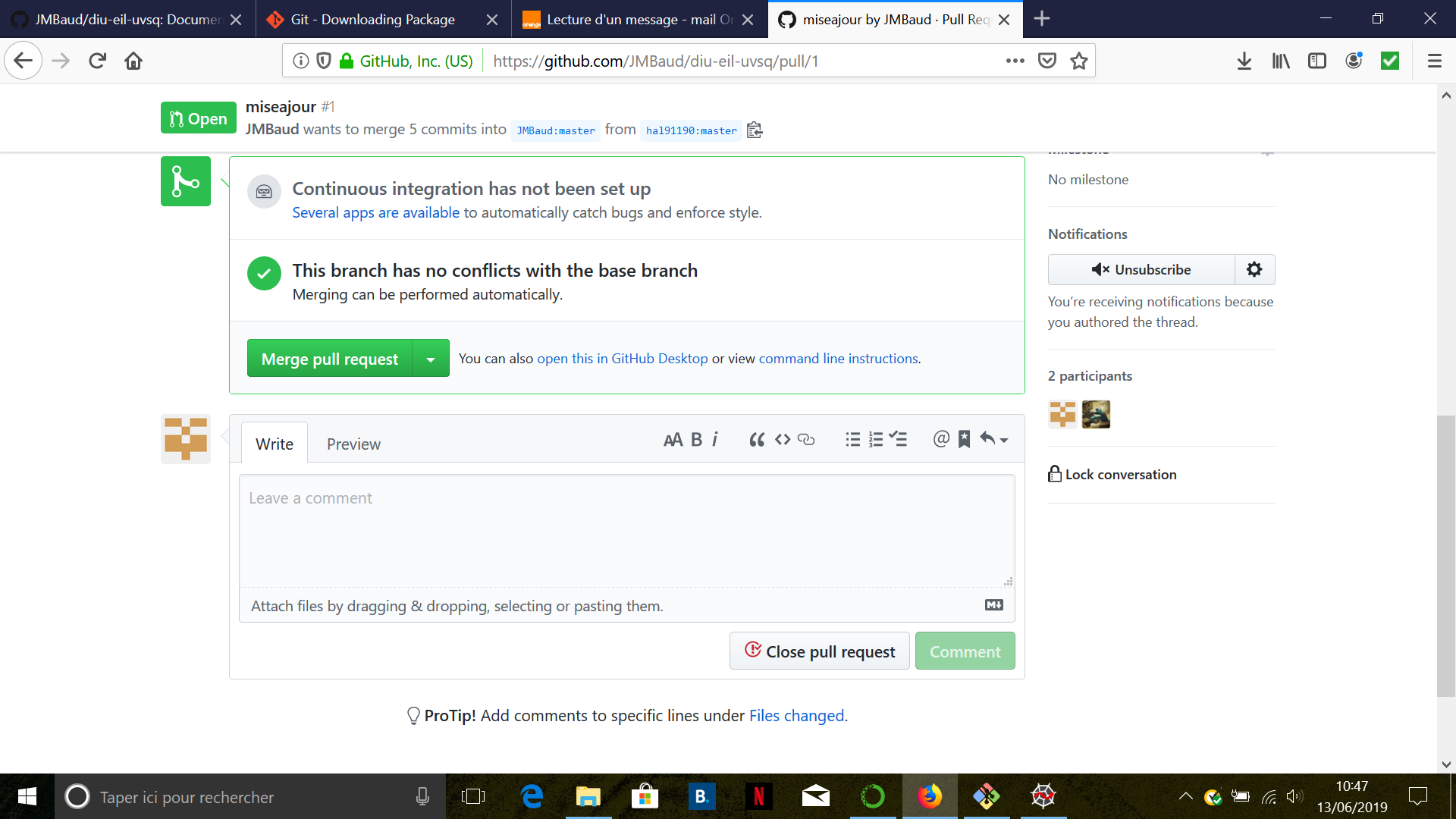


Attention il faut changer le sens de la comparaison

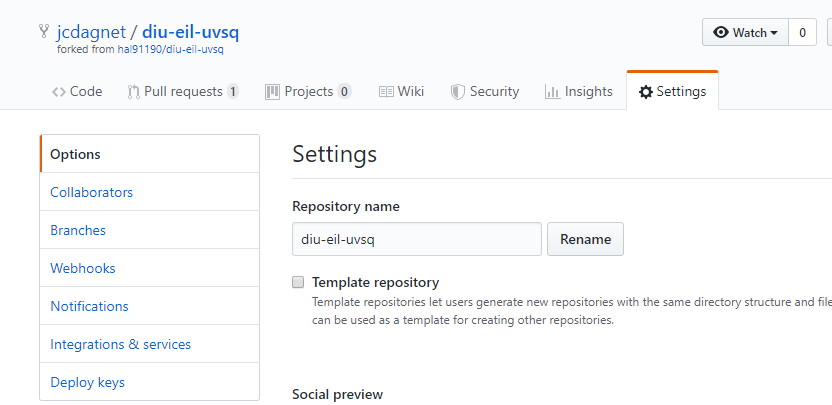


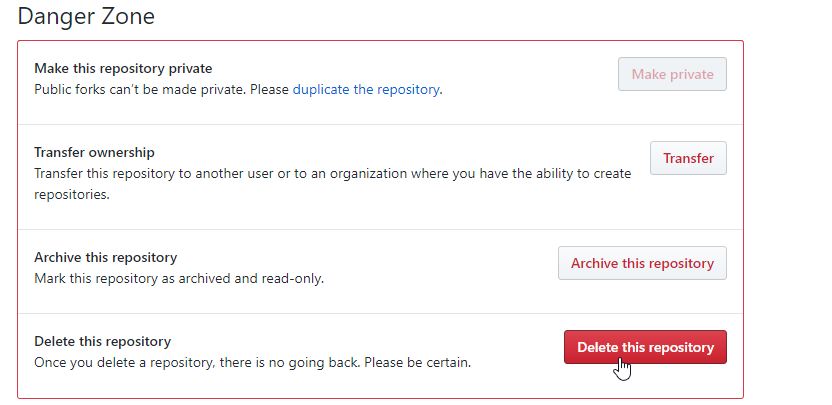




* 1. Si il y a eu des modifications après le "Fork"
     1. Méthode à l'arrache :





Et ensuite on refait le "**Fork**" puis on clone

* + 1. Méthode 2

GitHub

jcdagnet/ISN

Elève 1

Recommandations

Les\_projets\_2019

Nouveau fichiers

Recommandations

Les\_projets\_2019

Local

Recommandations

Les\_projets\_2019

ISN.zip

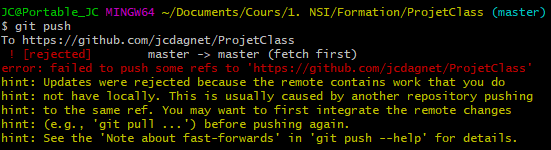
**1. Téléchargement du .zip**

**2. Insertion dans le dossier local**

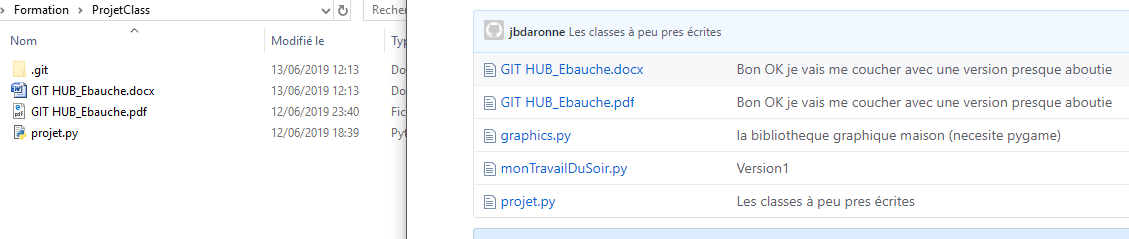
**(après avoir décompressé)**

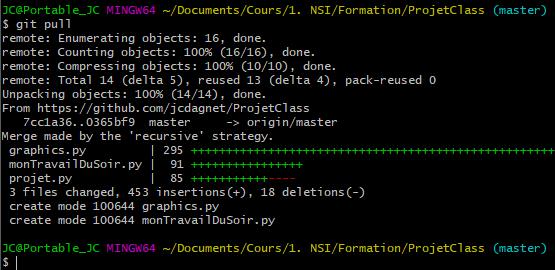
**3. Push**

1. Petites erreurs.



Les deux dossiers sont pas les mêmes donc il refuse de me faire un push





A noter que mes modification n'ont pas été supprimer

**Avant Après**

