

Tutoriel Git/GitHub

INTRODUCTION

Il faut garder à l'esprit que Git/GitHub est prévu pour permettre le versionning lors de la création de projets. Il facilite le travail à plusieurs, mais sert également pour un utilisateur unique afin de partager avec tout le monde (ou un nombre limité de personnes) son travail.

C'est de cela que je vais parler ici, avec une rapide description des étapes principales pour commencer avec Git et GitHub. Chaque étape présentera les différentes commandes à exécuter. Et je reviendrais en détails sur quelques-unes d'entre-elles.

Voici donc un plan très sommaire :

Je précise que ce tutoriel n'engage que moi, il est basé sur ce que j'ai moi-même pu expérimenter et il n'a pour seul but que d'aider ceux qui débutent et veulent en savoir plus.

Plan

- I. INITIALISATION ET INSTALLATION
- II. CREATION DE REPOSITORY (REPertoire).
- III. CLONAGE EN LOCAL.
- IV. LES COMMANDES GIT PRINCIPALES.
 - a. git add
 - b. git commit
 - c. git push
- V. GESTION DES ERREURS.

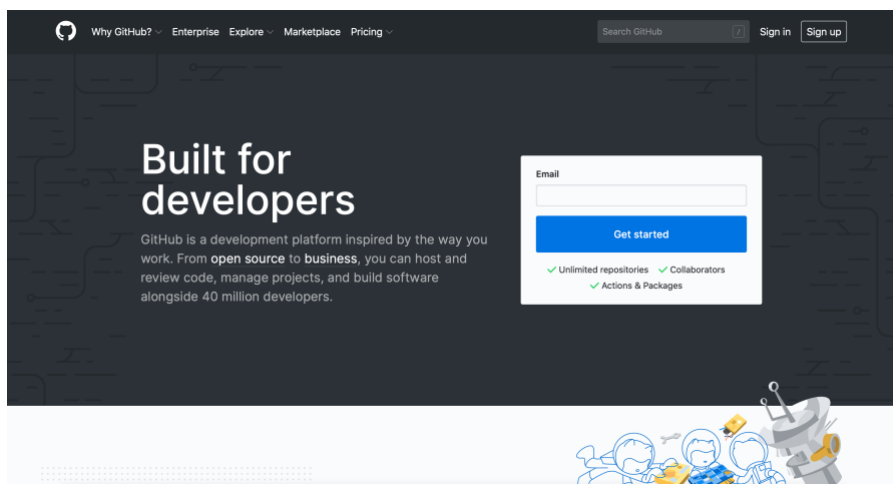
I. INSTALLATION

L'installation se passe en 2 étapes :

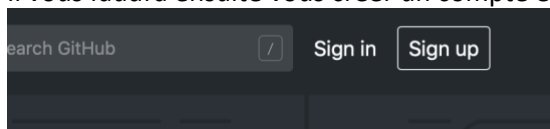
- Création de compte sur Github
- Installation de Git

- Création de compte

Si vous lisez ce que j'écris c'est que vous êtes bien déjà sur le bon site. Mais au cas ou voici le site sur lequel il vous faudra créer un compte : <https://github.com>
Cela ressemblera à cela :



Il vous faudra ensuite vous créer un compte en cliquant en haut à droite.



Un email valide suffira pour vous créer un compte.

Passons maintenant à la 2^e étape : l'installation de Git.

- Installation de git

Git s'installe en local sur votre ordinateur, pour faire court Git permet de connecter ce qui est en ligne sur Github et ce qui est sur votre ordinateur en local. Ainsi vous allez pouvoir effectuer vos modifications en local et ensuite poster le tout en ligne sur Github avec de simple commande git.

Pour installer Git, cela dépend de votre environnement :

- Pour Windows allez sur ce site : <https://git-scm.com/download/win> et téléchargez la version qui convient à votre système.
Il se peut que lors de l'installation on vous demande de choisir sur quel terminal utiliser Git.
Le mieux est de mettre « use default console window »
- Pour MacOS de même mais sur ce site : <https://git-scm.com/download/mac>

Le plus simple reste tout simplement d'écrire sur un moteur de recherche « installer git sur ... » et vous écrivez le nom de votre environnement.

BRAVO la 1ere étape est finie, on va pouvoir passer aux choses sérieuses !!!

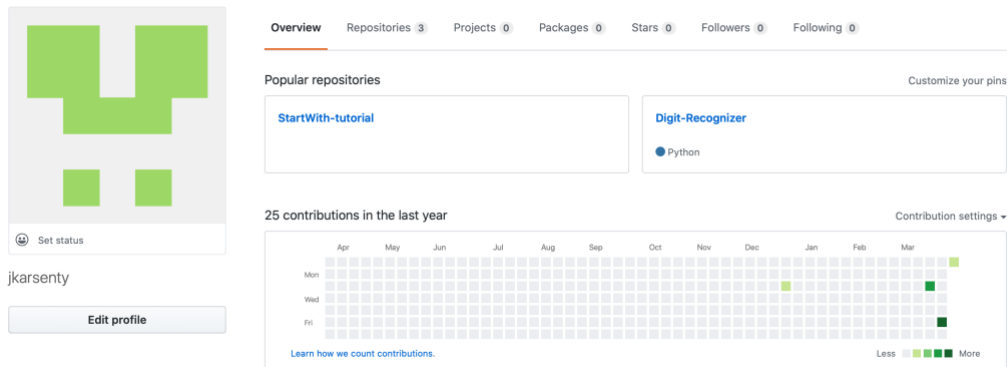
II. CREATION DE REPOSITORY (REPertoire).

Pour cette étape il va falloir aller sur le site de Github, se connecter avec vos identifiants. Et ensuite il faut appuyer sur le bouton vert « New ».

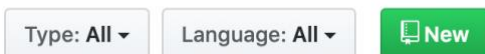
Repositories



Si vous ne le voyez pas apparaître, c'est que vous êtes sûrement dans cette page.



Dans ce cas il faut d'abord sélectionner l'onglet « Repositories » et vous verrez apparaître en haut à droite le bouton « New ».



Vous arriverez alors à cette page et il vous suffira de remplir les informations.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: jkarsenty / Repository name:

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

Description (optional):

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

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

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

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

Public = tout le monde peut voir votre répertoire.

Private = vous avez le choix de qui peut accéder à votre répertoire.

Initialize this repository with a README = crée le répertoire avec déjà un 1er fichier à l'intérieur.

Ce Fichier README est au format Markdown et c'est le texte de présentation de votre répertoire. Ce texte est modifiable et c'est la 1^{ère} chose que les visiteurs de votre repository vont regarder.

Je vous conseille vivement de toujours cocher cette case.

Vous pouvez maintenant cliquer sur le bouton « Create Repository »

BRAVO La 2^e étape est terminée !!!

III. CLONAGE EN LOCAL

Le clonage est le terme utilisé pour signifier que l'on crée un lien du répertoire situé sur Github mais en local sur notre ordi. On le copie, on le clone du serveur en ligne vers notre poste en local.

Pour cela créez un dossier sur votre poste de travail (*je l'ai appelé Git pour ma part*).

Puis une fois que c'est fait ouvrez un terminal dans ce dossier. Pour faire cela plusieurs méthodes selon vos systèmes d'exploitation :

- (Pour MacOS) Clic droit sur le dossier et « Nouveau Terminal au dossier ».
- (Pour Windows) Ouvrir le dossier en question et dans la barre de recherche (voir ci-dessous) écrire « cmd ».



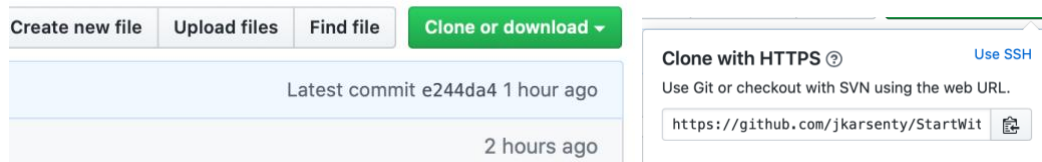
- La dernière méthode fonctionne pour tous les systèmes, vous ouvrez un terminal puis vous écrivez « cd » suivi du chemin de votre dossier. Ex : « cd /Users/jeremy/Desktop/Git » ou « cd C:\Users\ votreNom\Desktop\Git »

```
Git — -bash — 80x24
Last login: Sun Mar 29 17:36:35 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Macbook-Air-de-Jkarsenty:~ karsentyjk$ cd /Users/jeremy/Programmation/Git
Macbook-Air-de-Jkarsenty:Git karsentyjk$
```

Une fois que votre terminal est ouvert et bien situé dans votre dossier (vous verrez sur le terminal écrit le nom du dossier, ici on voit « :Git »).

Vous allez sur Github, sur le répertoire que vous venez de créer vous sélectionnez le bouton « Clone or download » et vous copiez le lien (comme sur les photos).



Vous allez ensuite sur le terminal que l'on a ouvert précédemment. Et vous tapez la commande `git clone` suivi de ce que vous venez de copier. Cela doit donner :

« `git clone https://github.com/VotreNomUtilisateur/VotreNomDeRepository` »

ATTENTION !!! Souvent lorsque vous copier-coller il y a « .git » qui s'ajoute à la fin de votre lien vers le repository. Il faut enlever le « .git ».

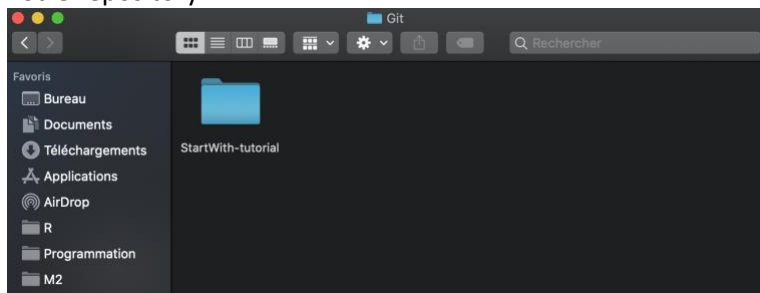
Cela doit donner cela : `git clone https://github.com/jkarsenty/StartWith-tutorial`

Et non pas : `git clone https://github.com/jkarsenty/StartWith-tutorial.git`

Et normalement si vous avez bien suivi les étapes vous aurez ce message qui apparaîtra :

```
For more details, please visit https://support.apple.com/kb/HT208050.
Macbook-Air-de-Jkarsenty:~ karsentyjk$ git clone https://github.com/jkarsenty/StartWith-tutorial
Cloning into 'StartWith-tutorial'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 17 (delta 1), reused 5 (delta 0), pack-reused 0
Unpacking objects: 100% (17/17), done.
```

Et vous verrez également dans votre dossier (que j'avais appelé Git) un nouveau dossier du nom de votre repository



Avant de finir cette étape il faut passer le terminal dans le dossier qui vient d'être créé. Pour cela on va utiliser la commande que l'on a déjà vu juste avant, le commade « cd » et il suffit d'écrire « cd » suivi du nom du dossier car nous sommes déjà dans le dossier Git (donc il comprend ce qu'il y a à l'intérieur). Et vous verrez comme avant le nom de votre dossier (ici appelé StartWith-tutorial) apparaitre dans le Terminal :

```
Macbook-Air-de-Jkarsenty:Git karsentyjk$ cd StartWith-tutorial
Macbook-Air-de-Jkarsenty:StartWith-tutorial karsentyjk$
```

BRAVO La 3^e étape est terminée !!! Vous avez fait le plus dur !! Le lien entre votre repository Github et votre dossier Local est maintenant parfaitement établi !

IV. LES COMMANDES GIT PRINCIPALES.

Maintenant que votre repository est bien cloné en local, vous avez pu commencer à travailler dessus. Votre projet est bien débuté vous avez rajouter des fichiers et vous avez même peut être modifié le fichier « README.md ».

Vous souhaitez désormais tout mettre sur Github, c'est là que les commandes interviennent. *Pour cela vous allez en Local sur le terminal, ouvert (comme vu plus tôt) toujours dans le répertoire de votre projet (ici StartWith-tutorial) et c'est ici aue vous écrivez les commandes.* Découvrons ensemble les principales commandes :

i. Git Status

Commençons par la commande « git status ». Git Status permet de vérifier les différences entre votre répertoire local et celui en ligne. Il vérifie comme son nom l'indique, le statut de votre répertoire local. Si il y a dans votre dossier de nouveaux fichiers ou des fichiers existants que vous avez modifiés. Alors Git status vous affichera les modifications effectuées ;

Si vous n'avez rien modifié il affiche cela :

```
Macbook-Air-de-Jkarsenty:StartWith-tutorial karsentyjk$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Et sinon il vous affichera la modification effectuée comme ci-joint :

```
Macbook-Air-de-Jkarsenty:StartWith-tutorial karsentyjk$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Tutoriel Git copie.docx

nothing added to commit but untracked files present (use "git add" to track)
```

ii. Git add

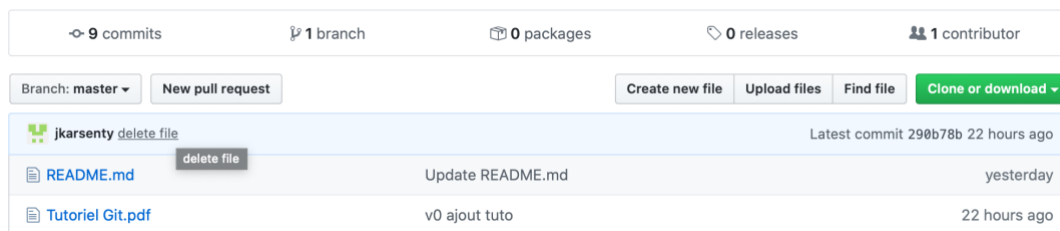
Git add permet de notifier à Github quels fichiers vous allez lui envoyer. C'est-à-dire quels fichiers vous allez passer du local au « en Ligne ». Il s'agit là de la 1^{ère} étape pour mettre des fichiers locaux sur Github. C'est une étape nécessaire. Git add agit un peu comme une notification que l'on enverrait à Github pour lui signifier que l'on veut ajouter des documents et lesquels.

Il faut donc écrire « git add » suivi du nom de fichier cela vous donnera quelque chose du style « git add nomDeFichier.extension ». Mais si vous souhaitez ajouter tous les fichiers modifiés, vous pouvez écrire « git add * ».

Lorsque vous écrivez git add sur le terminal il ne se passe rien en réponse c'est normal Il faut pour cela attendre les étapes suivantes.

iii. Git commit

Git commit est la 2^{nde} étape nécessaire pour transférer nos fichiers sur Github. Je le rappelle Git/Github servent principalement à faire du versionning lors de la réalisation de projet. Ces différentes versions sont identifiables grâce à des « commit ». Il s'agit là d'un commentaire qui apparait sur Github à coté de votre fichier et permet de montrer de quand date le fichier présent en ligne. Ici on peut voir 2 commit « Update README.md » et « v0 ajout tuto ».



Pour écrire un commit (ce qui est nécessaire avant la 3^e et dernière étape) il faut taper la commande « git commit -m "Texte de votre commit entre guillemet" ».

Il se peut que chez certains vous receviez une erreur qui vous demande de vous identifier.

Pour cela écrivez les 2lignes suivantes :

« git config --global user.email "votre adresse mail de github" »

« git config --global user.name "votre nom sur github" »

Et ensuite vous pouvez écrire votre commande de commit « git commit -m " " ».

Vous verrez s'afficher la modification effectuée sous le "commit" (ici appelé test).

```
Macbook-Air-de-Jkarsenty:StartWith-tutorial karsentyjk$ git commit -m "test"
[master 642c5d5] test
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Tutoriel Git copie.docx
```

iv. Git push

Vous pouvez enfin tout transférer, on appelle cela « pusher », vers Github. Rien de plus simple il suffit d'écrire la commande « git push ». Cela vous donnera ceci :

```
Macbook-Air-de-Jkarsenty:StartWith-tutorial karsentyjk$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.88 MiB | 1.62 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/jkarsenty/StartWith-tutorial
290b78b..642c5d5 master -> master
```

BRAVO La 4^e étape et dernière étape est terminée !!! Vérifiez par vous-même vos fichiers sont sur Github !!

V. GESTION DES ERREURS

Je vais vous montrer à présent juste quelque simple commande si vous avez par mégarde, ajouter un fichier en trop sur Github et que vous souhaitez le supprimer.