

SOARES
ALEXIS
2SIO



**Documentation Professionnelle :
GIT - Logiciel de gestion de versions**

Professeur - M. De Almeida Silva

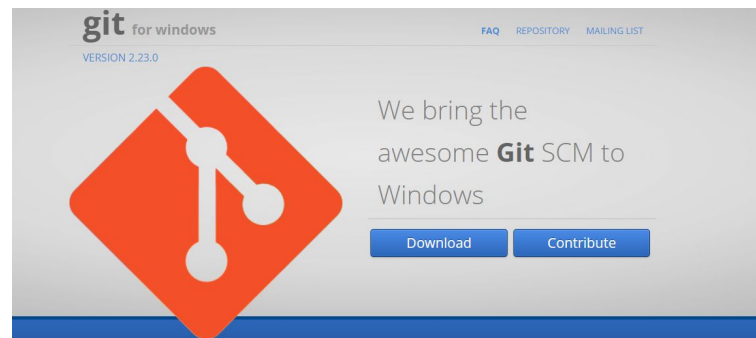
INTRODUCTION

Vous avez toujours rêvé de faire un travail de groupe en informatique, mais vous ne saviez comment faire ? Pouvoir ajouter vos lignes de codes avec d'autres personnes. C'est donc pour cela que l'on utilise un logiciel de gestion de versions, il permet donc de partager vos codes sources et que chacun puisse y ajouter sa pierre à l'édifice.

INSTALLATION

ATTENTION : Lors de l'installation cocher les cases : "Git BASH HERE" et "GIT GUI HERE"

- Sous windows, il vous suffit simplement d'utiliser le lien suivant : <https://msysgit.github.io/>
- Sous linux, il vous suffit d'utiliser un gestionnaire de paquets :
`sudo apt-get install git-core gitk`



INSCRIPTION GITHUB

Maintenant passons à l'inscription sur github, pour cela c'est tout aussi simple, il vous suffit d'aller sur le site suivant et de vous créer un compte si ce n'est pas déjà fait. Une fois votre compte créé, il vous suffira de vous connecter et de passer à la suite.

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

AJOUTER UNE CLEF SSH POUR UTILISER GITHUB

Même si les termes paraissent compliquer ne t'en fais pas, c'est tout aussi facile. tu as tout simplement à suivre mes indications, et tu seras tout faire comme un grand.

Ouvrez le logiciel suivant : Git BASH

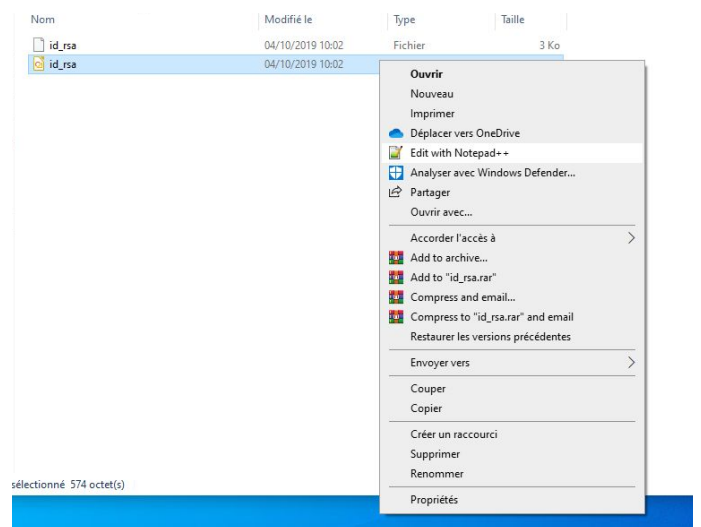
Pour le moment vous aurez simplement une console qui s'ouvre et vous pourriez ensuite y transmettre les commandes de git.

- **ssh-keygen -t rsa -C votreadressemail@gmail.com**

Puis ensuite vous devez appuyez trois fois sur ENTRER. C'est pour pouvoir rentrer un mot de passe vide, c'est à dire de ne rentrer aucun mot de passe pour rentrer dans git. Si vous souhaitez être plus sécurisant vous pouvez en mettre un, mais attention il faut vous en souvenir seulement pour la CLEF SSH !

Et pour récupérer sa clef, il suffit d'aller dans le dossier : **C:/USER/utilisateur/.SSH**

Dans ce dossier vous avez id_rsa.pub que vous devez ouvrir avec Note Pad comme sur l'image suivante :



Lorsque vous avez ouvert le fichier vous devez tout sélectionner puis copier la clef ssh.

MAINTENANT SUR GITHUB

Il faut accéder au paramètre de votre compte avec le lien suivant :

<https://github.com/settings/profile>

Puis dans l'onglet : **SSH and GPG keys** vous devez appuyez sur **NEW SSH key** et puis rentrer votre clef SSH en faisant un **COLLER**, pour le titre vous pouvez mettre ce que vous souhaitez.

SSH keys / Add new

Title

PC- Sainte Marguerite

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC8x2zKqY592qKmmMgXRvGeJyE9V3MPm9YjRF42smdoARCAsfJPeU6kpFf
p1/HRAFXTPiUa8lb4un8zlMyl55sMVCaSKWJjs7K2Ac00Kg++bWy3X+y3SCTCt1PHjC9RYsOISk0HXbMCYxlpG6AEpSP7
b8scBxamp8Lv3EUUnRQwZMBsTBR9UYhE+NDPi7mETCmG+T2ywRV9t8aWrttq440kibm8yL3pc0KbbXIngeopPCdxC7X
d6V4gQMtOr7CoS77WgTBIN3e/mu8wmoFOseZP1xGEZscUa9BfyWGuUCtAFSWX7JXLHTAanNmTTII0G0tMyz29U8w
0z0bJYVYawC5M2UhOMeYXfp+ZnPz5/GLXuvyubiak3logkGD7OqRsbiGMq/Wats5a81jdvqAko7jz8P3Opl9tVcOvwDZ
0mzaKFx/ICiYklqvq2dm0+AjqPjailRg/LKLH3BmKZG4wF0FFLuxa+kKdG4eSbnlz3PKHh/huXXucZ+aTPqUi4LLk=
asoaespro@gmail.com
```

Add SSH key

Par la suite vous devez créer un nouveau REPOSITORY, c'est-à-dire un nouveau répertoire pour pouvoir y mettre votre projet de groupe.

- **Sur github :**
New > Repository
<https://github.com/new>

Entrer les informations que vous souhaitez, simplement ici c'est important de sélectionner Public si vous souhaitez partager avec tout les utilisateurs de GIT ou privée pour seulement les utilisateurs que vous souhaitez.

Create a new repository

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

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner

Repository name *

 soaresalexis ▾

/ gitutilisation ✓

Great repository names are short and memorable. Need inspiration? How about musical-broccoli?

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.

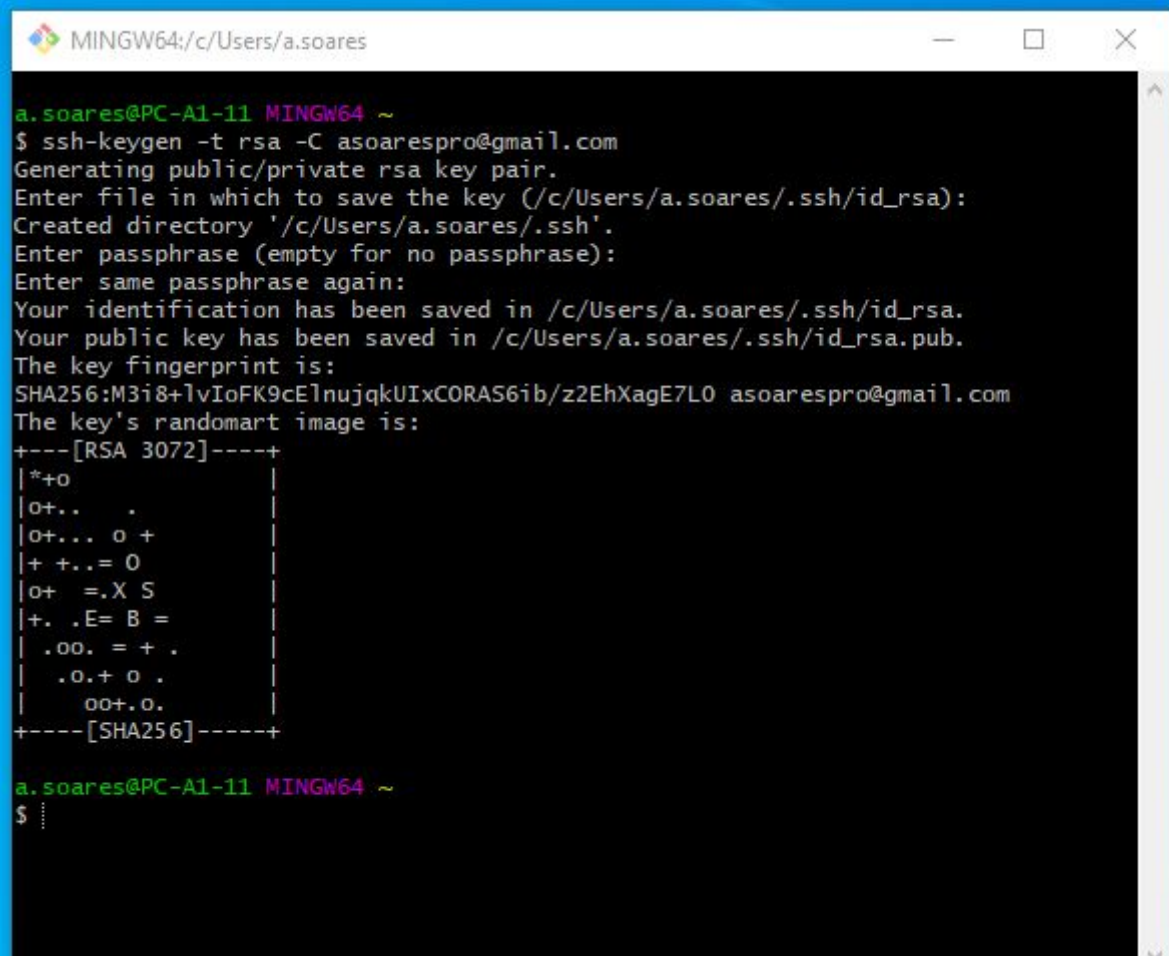
Add .gitignore: None ▾

Add a license: None ▾ ⓘ

Create repository

Maintenant que votre répertoire est créé il va falloir envoyer vos codes sources sur **GITHUB**, ce qui est le plus important pour partager votre travail avec les autres membres de votre groupe. Pour le moment il faut prendre le **SSH** juste en dessous du titre : "Quick setup - if you've done this kind of thing before"

Enfin, vous pourrez pouvoir déplacer vos dossiers, pour cela aller dans le dossier où se trouvent les fichiers de vos sources. Faites un clic droit dans le dossier et cliquez sur **GIT BASH HERE**, cela ouvrira la console de **GIT BASH**.



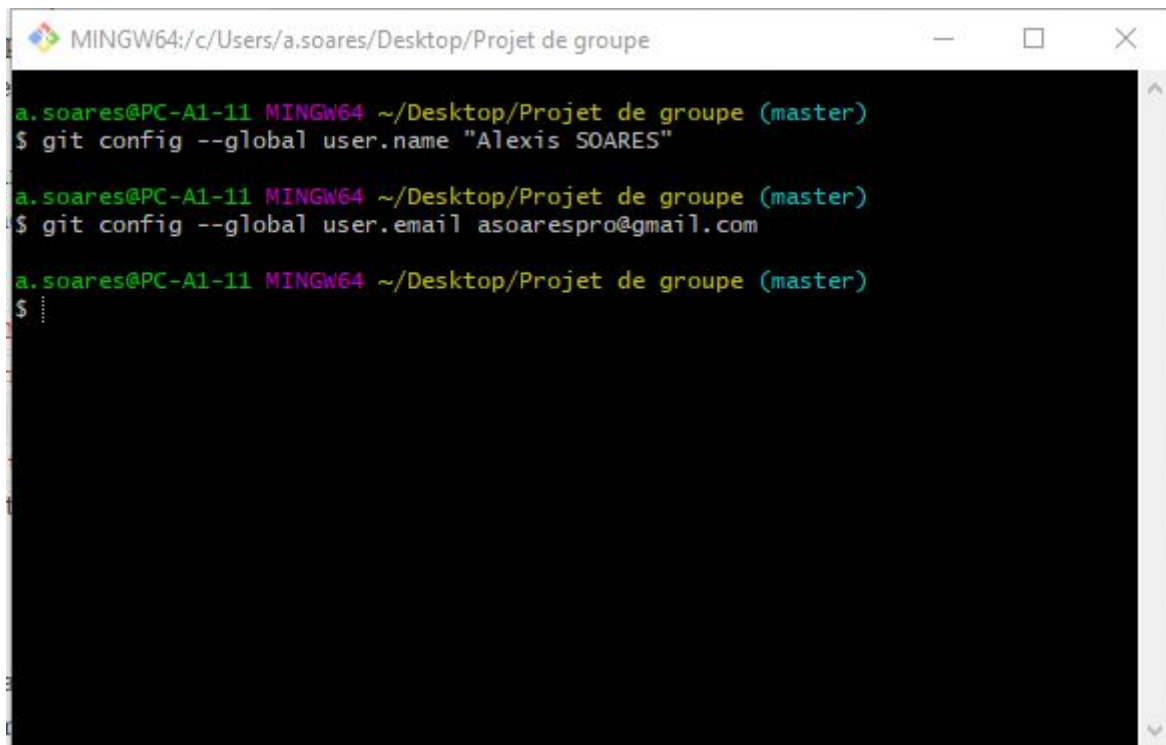
```
MINGW64:/c/Users/a.soares
a.soares@PC-A1-11 MINGW64 ~
$ ssh-keygen -t rsa -C asoarespro@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/a.soares/.ssh/id_rsa):
Created directory '/c/Users/a.soares/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/a.soares/.ssh/id_rsa.
Your public key has been saved in /c/Users/a.soares/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:M3i8+lvIoFK9cElnujqkUIxCORAS6ib/z2EhXagE7L0 asoarespro@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|*+0
|O+.. .
|O+... o +
|+ +..= O
|O+ =,X S
|+. .E= B =
|.oo. = + .
|.o.+ o .
|oo+.o.
+-----[SHA256]-----+
a.soares@PC-A1-11 MINGW64 ~
$ .....
```

Maintenant, il va falloir taper les commandes de git suivantes :

- git config --global user.name "Votre nom"

- git config --global user.mail votreadressemail@gmail.com

Cela initialisera la toute première utilisation de git bash, ensuite passons à initialiser la première fois votre projet, dossier.



```
MINGW64:/c/Users/a.soares/Desktop/Projet de groupe
a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git config --global user.name "Alexis SOARES"
a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git config --global user.email asoarespro@gmail.com
a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ .....
```

- git init

Pour pouvoir ajouter vos fichiers utiliser la commande suivante :

ATTENTION : Il televersera tout les fichiers, dossier présent dans votre dossier sélectionné.

- git add .

Permet d'accepter en paramètre un chemin qui correspond à un fichier ou un répertoire, dans ce cas, la commande ajoute tous les fichiers de ce répertoire

- git commit -m "commentaire"

Permet d'utiliser un message donné comme message de validation. Si il y a plusieurs options données, leurs valeurs sont concaténées avec des paragraphes séparés.

- git remote add origin (SSH)*
*([git@github.com](https://github.com):soaresalexis/gitutilisation.git)

Permet de créer une nouvelle connexion à un référentiel distant. Ici on ajoute une télécommande avec mon nom et mon url dans Git. Supprimez la connexion au référentiel distant appelé. Renommez une connexion distante avec l'ancien nom en nouveau nom

- git push origin master

Permet de mettre en référentiel l'origine. Il spécifie d'être poussé dans une branche appelé "master"

```
a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git init
Reinitialized existing Git repository in C:/Users/a.soares/Desktop/Projet de groupe/.git/

a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git add .

a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git commit -m "Commentaire"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git remote add origin git@github.com:soaresalexis/gitutilisation.git
fatal: remote origin already exists.

a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git push origin master
Everything up-to-date

a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$
```

Maintenant si un membre de votre groupe a complété votre code sources, vous pourrez récupérer la version qu'il a complété en faisant les commandes suivantes :

- git fetch

Permet de récupérer des fichiers d'une branche dans un autre dossier souhaité


```
MINGW64/c:/Users/a.soares/Desktop/test

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git init
Initialized empty Git repository in C:/Users/a.soares/Desktop/test/.git/

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git origin master https://github.com/soaresalexis/gitutilisation.git
git: 'origin' is not a git command. See 'git --help'.

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git remote add origin https://github.com/soaresalexis/gitutilisation.git

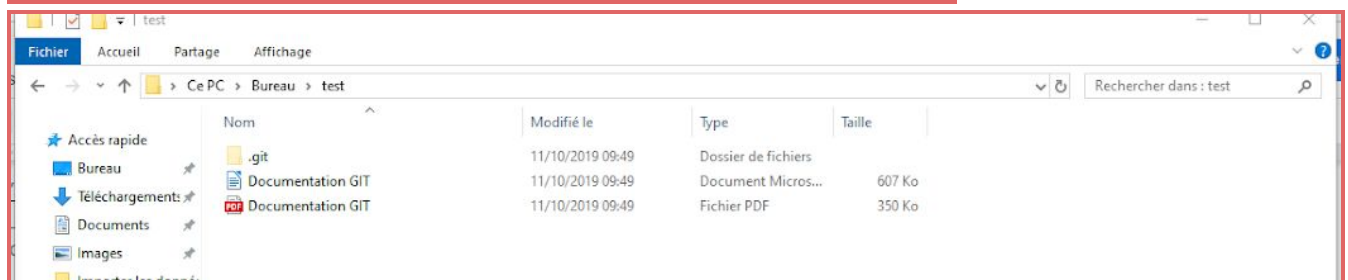
a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git fetch master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git fetch origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/soaresalexis/gitutilisation
 * branch      master      -> FETCH_HEAD
 * [new branch] master      -> origin/master

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$ git checkout master
Already on 'master'
Branch 'master' set up to track remote branch 'master' from 'origin'.

a.soares@PC-A1-11 MINGW64 ~/Desktop/test (master)
$
```



- git status

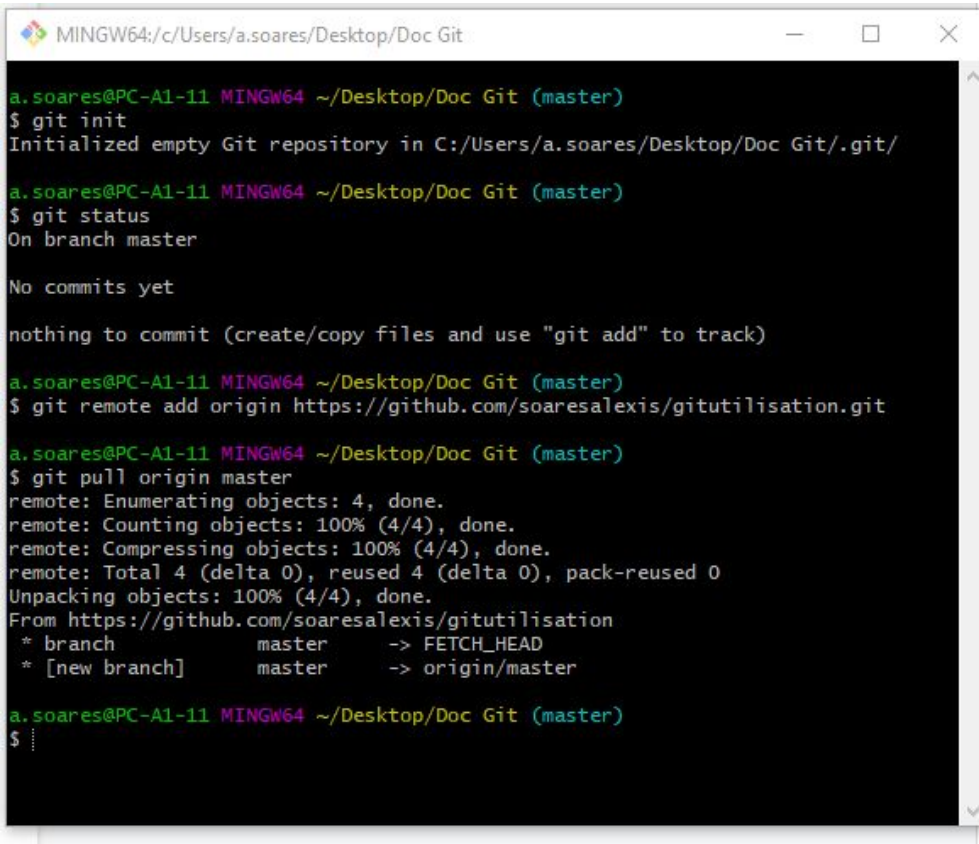
Permet de voir que la copie de travail est propre, aucun fichier suivi n'a été modifié, Git ne voit pas de non plus de fichiers non-suivis, sinon ils seraient listés. Et la commande indique aussi dans quelle branche nous sommes.


```
a.soares@PC-A1-11 MINGW64 ~/Desktop/Projet de groupe (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

- git pull origin master

Prendre tout les fichiers présent dans la branche "MASTER".



```
MINGW64:/c/Users/a.soares/Desktop/Doc Git
a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)
$ git init
Initialized empty Git repository in C:/Users/a.soares/Desktop/Doc Git/.git/

a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)
$ git status
On branch master

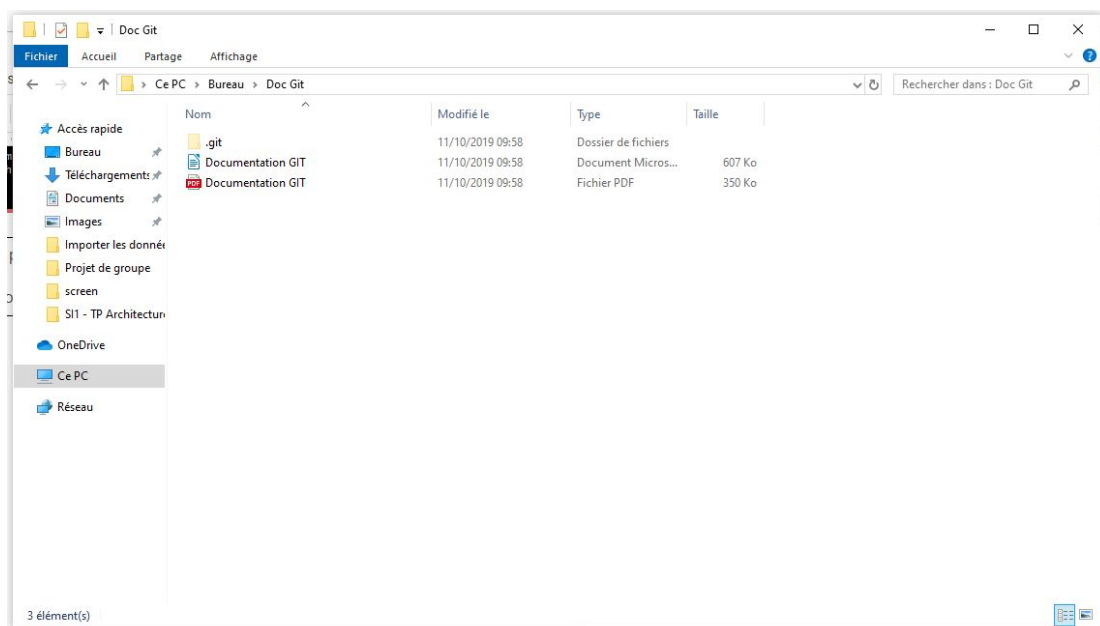
No commits yet

nothing to commit (create/copy files and use "git add" to track)

a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)
$ git remote add origin https://github.com/soaresalexis/gitutilisation.git

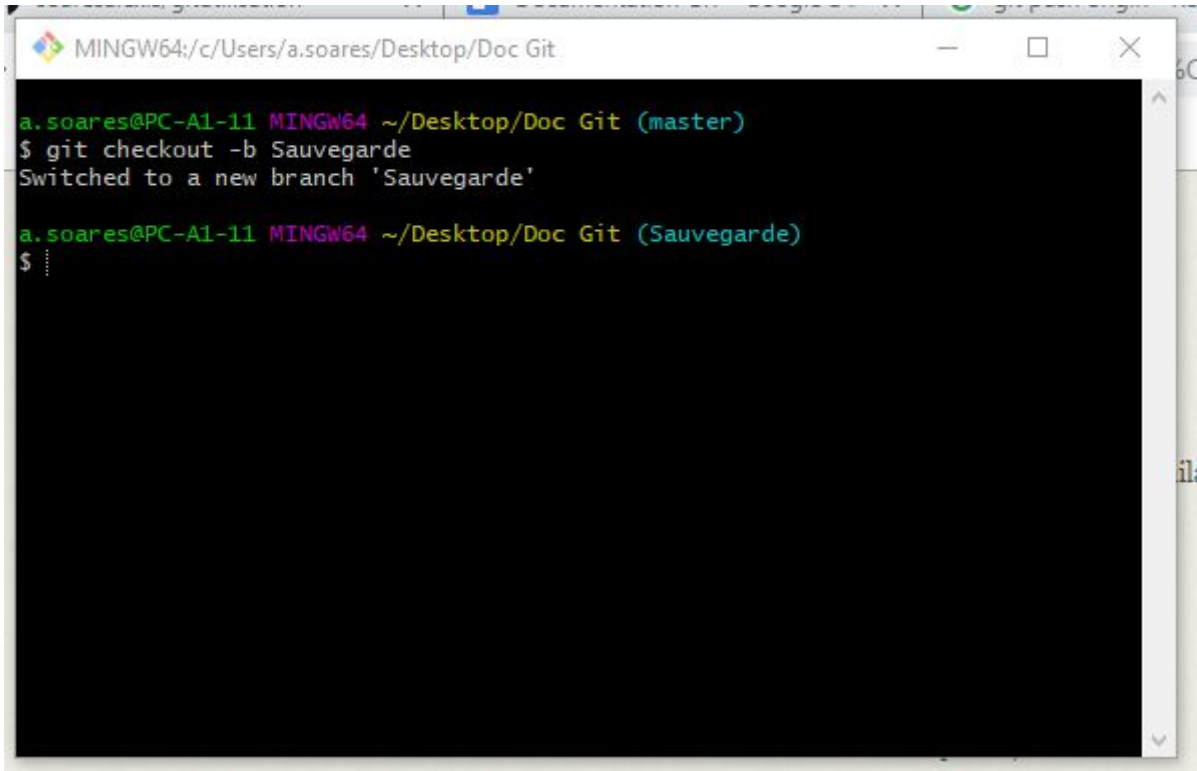
a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)
$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/soaresalexis/gitutilisation
 * branch      master      -> FETCH_HEAD
 * [new branch] master      -> origin/master

a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)
$
```



- `git checkout -b nomBranche`

Permet de créer une branche pour pouvoir gerer ces versions. Donc ici, création d'une branche de sauvegarde pour éviter tout problème avec le code principal

A screenshot of a terminal window titled 'MINGW64:/c/Users/a.soares/Desktop/Doc Git'. The prompt is 'a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (master)'. The user enters '\$ git checkout -b Sauvegarde', and the terminal responds 'Switched to a new branch 'Sauvegarde''. The prompt then changes to 'a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (Sauvegarde)'. The user enters '\$', and the terminal shows a blank line with a prompt '\$'.

Permet de changer de branche.

```
a.soares@PC-A1-11 MINGW64 ~/Desktop/Doc Git (Sauvegarde)
$ git checkout master
Switched to branch 'master'
```

- `git log`

Permet de montrer l'historique des modifications faites, et des connexions