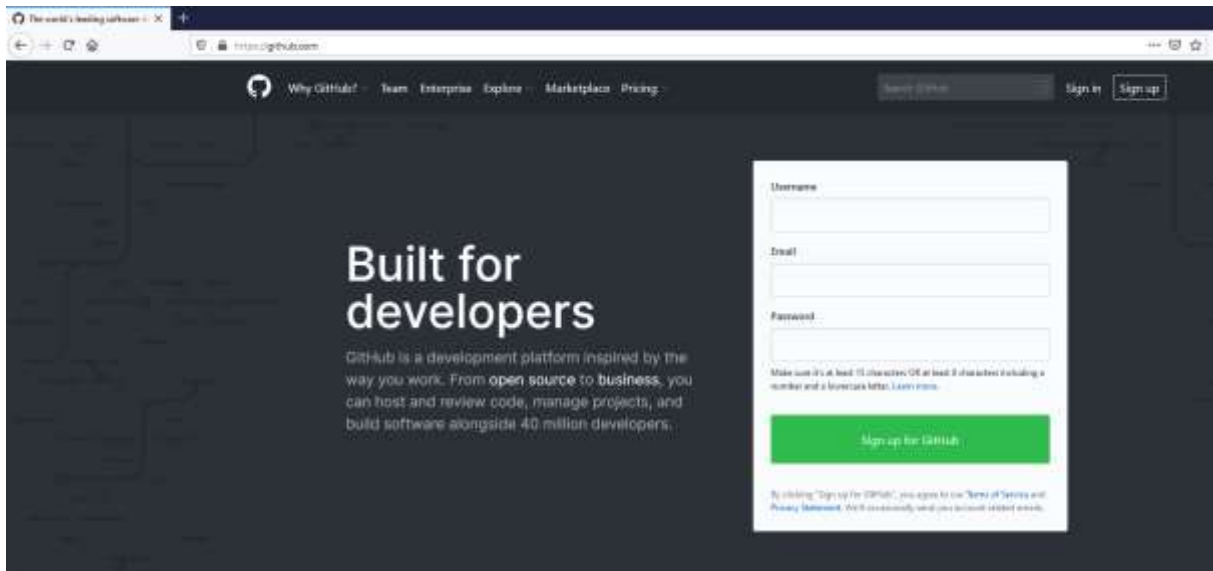
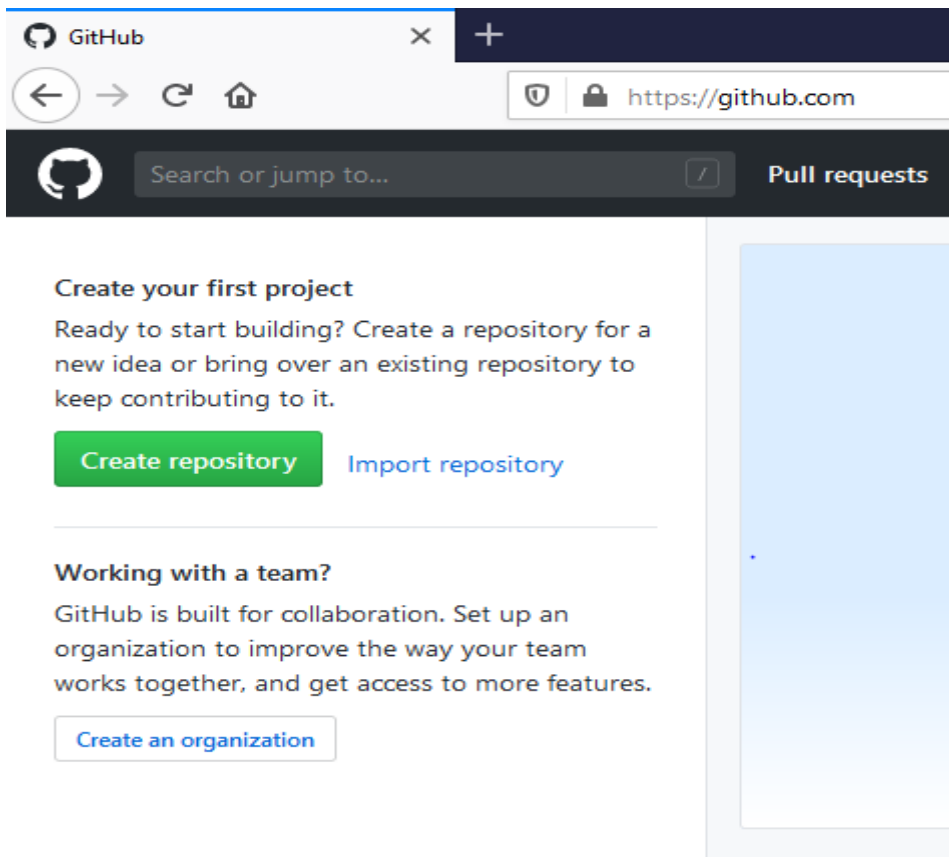


GITHUB

Etape 1 : créer un compte sur le site <https://github.com/>



Etape 2 : créer un repository




- Cliquer sur « Create repository »
- Remplir les informations suivantes :
 - Le nom du repository
 - Sa description(facultatif)
 - Cocher la case Public ou Private
 - Cocher la case Initialize this repository with a README
- Cliquer sur « Create repository »


Create a new repository

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

Owner

 DamsDams68 ▾

Repository name *

/ MonSite 

Great repository names are short and memorable. Need inspiration? How about **fuzzy-system**?

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.

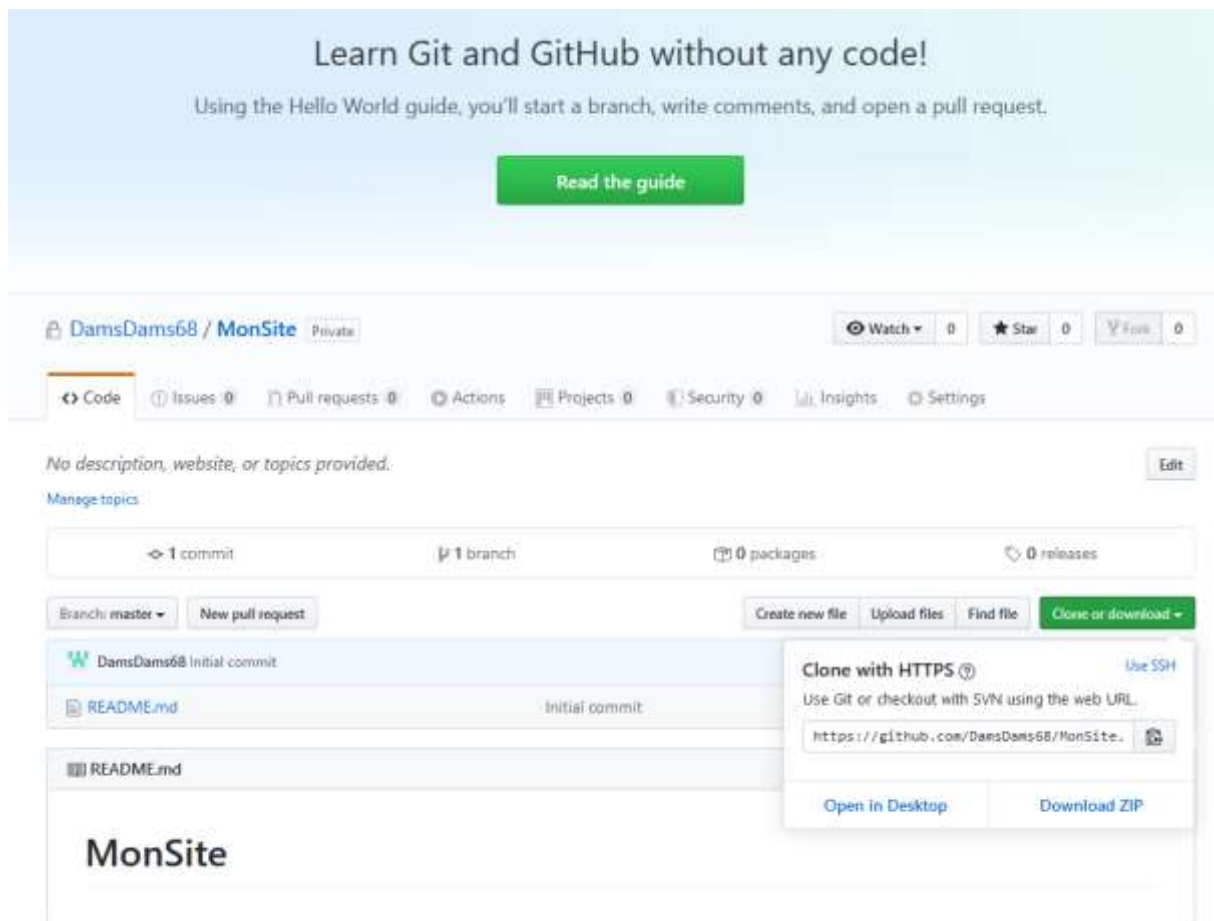
Add .gitignore: None ▾

Add a license: None ▾

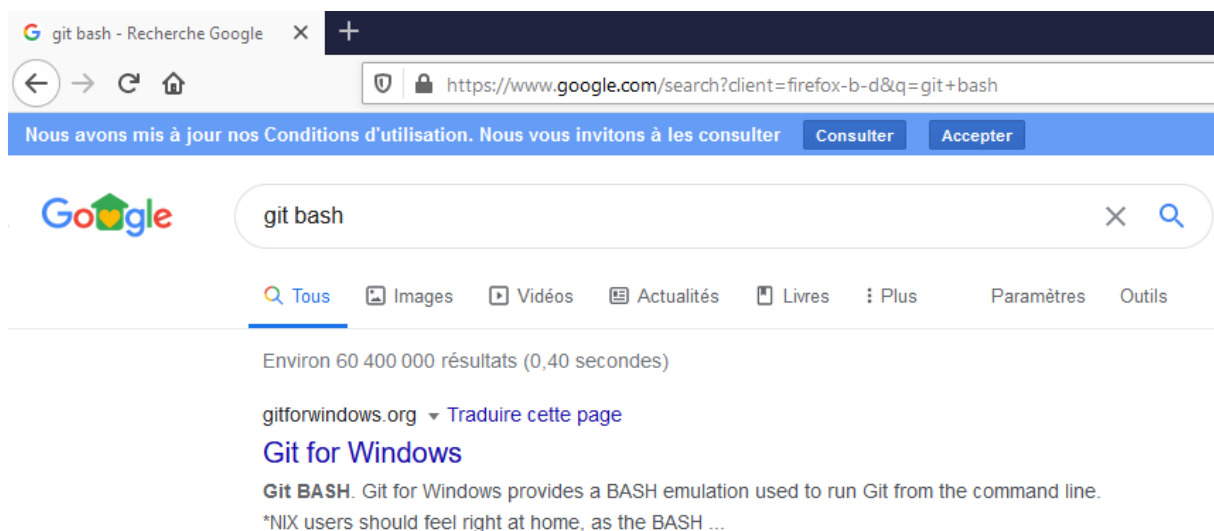


Create repository

Le repository est créé :



Etape 3 : Télécharger et installer GitBash



Une fois installé crée un dossier avec le nom de votre projet

Ouvrir GitBash puis naviguer jusqu'à votre dossier à l'aide de la commande cd

```
MINGW64:/c/Users/wonde/OneDrive/Documents/MonSite
wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd Documents/

wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd MonSite
bash: cd: MonSite: No such file or directory


wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd ..

wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd /c/Users/wonde/OneDrive/Documents


wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents
$ cd MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite
$ |
```

Sur github copié le lien du repository :

Clone with HTTPS  [Use SSH](#)

Use Git or checkout with SVN using the web URL.



[Open in Desktop](#) [Download ZIP](#)

Avec la commande git clone puis coller le lien pour cloner le repository dans votre dossier :

```
MINGW64:/c/Users/wonde/OneDrive/Documents/MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd Documents/

wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd MonSite
bash: cd: MonSite: No such file or directory

wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd ..

wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd /c/Users/wonde/OneDrive/Documents

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents
$ cd MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite
$ git clone https://github.com/DamsDams68/MonSite.git
```

Renseigner votre identifiant et votre mot de passe github

```
MINGW64:/c/Users/wonde/OneDrive/Documents/MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd Documents/

wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd MonSite
bash: cd: MonSite: No such file or directory

wonde@DESKTOP-IEMUSBP MINGW64 ~/Documents
$ cd ..

wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd /c/Users/wonde/OneDrive/Documents

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents
$ cd MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite
$ git clone https://github.com/DamsDams68/MonSite.git
Cloning into 'MonSite'...
fatal: HttpRequestException encountered.
  Une erreur s'est produite lors de l'envoi de la demande.
Username for 'https://github.com': |
```

Dans votre dossier un nouveaux dossier est créé .git et un fichier README.md

Ce PC > Documents > MonSite > MonSite			
Nom	Modifié le	Type	Taille
.git	30/04/2020 20:15	Dossier de fichiers	
README.md	30/04/2020 20:15	Fichier MD	

Placer les fichiers de votre projet dans votre dossier projet :

Ce PC > Documents > MonSite > MonSite			
Nom	Modifié le	Type	Taille
.git	30/04/2020 20:15	Dossier de fichiers	
index.html	30/04/2020 20:31	Firefox HTML Doc...	1 Ko
README.md	30/04/2020 20:15	Fichier MD	1 Ko
style.css	30/04/2020 20:32	Document de feui...	1 Ko

Pour envoyer vos fichiers de votre dossier à votre repository utiliser la commande `cd` pour vous placer dans le dossier de votre projet

```

MINGW64:/c/Users/wonde/OneDrive/Documents/MonSite/MonSite
wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd /c/Users/wonde/OneDrive/Documents/MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite
$ cd MonSite/

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ |

```

Utiliser la commande `git add .` pour ajouter les nouveaux fichiers à l'index

```
MINGW64:/c:/Users/wonde/OneDrive/Documents/MonSite/MonSite
wonde@DESKTOP-IEMUSBP MINGW64 ~
$ cd /c:/Users/wonde/OneDrive/Documents/MonSite

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite
$ cd MonSite/

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ git add .

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ |
```

Utiliser la commande git commit pour faire un commit :

```
wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ git add index.html

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ git add style.css

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ git commit -a -m 'premier commit'
[master 903f050] premier commit
2 files changed, 14 insertions(+)
create mode 100644 index.html
create mode 100644 style.css

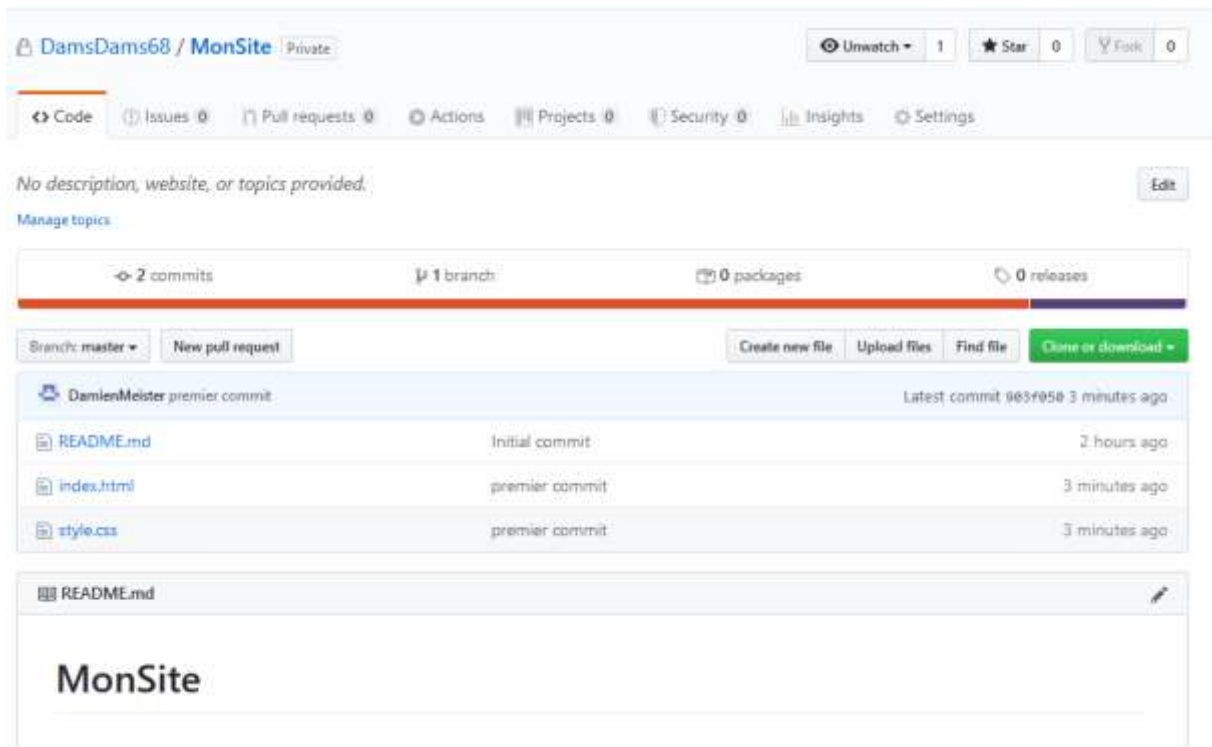
wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ |
```

Utiliser la commande git push pour envoyer vos fichiers :

```
$ git push
fatal: HttpRequestException encountered.
  Une erreur s'est produite lors de l'envoi de la demande.
Username for 'https://github.com': damien.meistertzheim@outlook.fr
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 427 bytes | 427.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/DamsDams68/MonSite.git
  4feba6f..903f050  master -> master

wonde@DESKTOP-IEMUSBP MINGW64 ~/OneDrive/Documents/MonSite/MonSite (master)
$ |
```

Vos fichiers sont bien envoyés sur votre repository GitHub :



Memento des principales commandes Git :

- **Git config**

L'une des commandes git les plus utilisées est **git config**. On l'utilise pour configurer les préférences de l'utilisateur : son mail, l'algorithme utilisé pour diff, le nom d'utilisateur et le format de fichier etc. Par exemple, la commande suivante peut être utilisée pour définir le mail d'un utilisateur:

```
MINGW64:/c/Users/wonde
wonde@DESKTOP-IEMUSBP MINGW64 ~
$ git config --global user.email sam@google.com
```

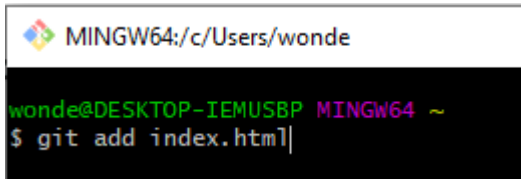
- **Git init**

Cette commande est utilisée pour créer un nouveau dépôt GIT :

```
MINGW64:/c/Users/wonde
wonde@DESKTOP-IEMUSBP MINGW64 ~
$ git init
```


- **Git add**

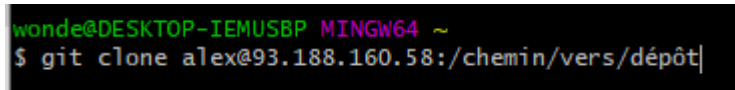
La **commande git add** peut être utilisée pour ajouter des fichiers à l'index. Par exemple, la commande suivante ajoutera un fichier nommé index.html dans le répertoire local de l'index:



```
MINGW64:/c/Users/wonde  
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git add index.html
```

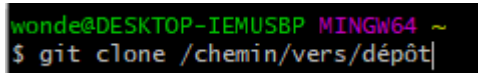
- **Clone git**

La **commande git clone** est utilisée pour la vérification des dépôts. Si le dépôt se trouve sur un serveur distant, utilisez:



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git clone alex@93.188.160.58:/chemin/vers/dépôt
```

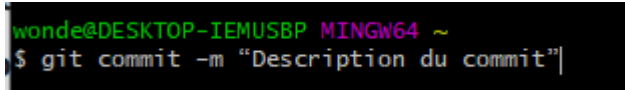
Inversement, si une copie de travail d'un dépôt local doit être créée, utilisez:



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git clone /chemin/vers/dépôt
```

- **Git commit**

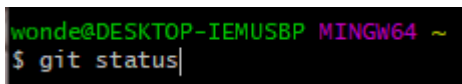
La **commande git commit** permet de **valider les modifications apportées** au HEAD. Notez que tout commit ne se fera pas dans le dépôt distant.



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git commit -m "Description du commit"
```

- **Git status**

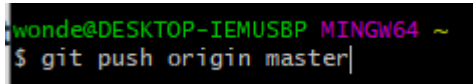
La **commande git status** affiche la liste des fichiers modifiés ainsi que les fichiers qui doivent encore être ajoutés ou validés. Usage:



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git status
```

- **Git push**

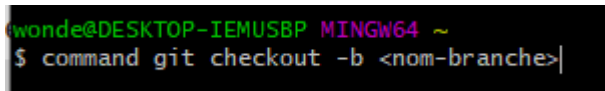
Git push est une autre commandes GIT de base. Un simple push envoie les modifications locales apportées à la branche principale associée :



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git push origin master
```

- **Git checkout**

La **commande git checkout** peut être utilisée pour créer des branches ou pour basculer entre elles. Par exemple nous allons créer une branche:



```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ command git checkout -b <nom-branche>
```

Pour passer simplement d'une branche à une autre, utilisez:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git checkout <nom-branche>|
```

- **Git remote**

La **commande git remote** permet à un utilisateur de se connecter à un dépôt distant. La commande suivante répertorie les dépôts distants actuellement configurés:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git remote -v|
```

Cette commande permet à l'utilisateur de connecter le dépôt local à un serveur distant:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git remote add origin <93.188.160.58>|
```

- **Branche git**

La **commande git branch** peut être utilisée pour répertorier, créer ou supprimer des branches. Pour répertorier toutes les branches présentes dans le dépôt, utilisez:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git branch|
```

Pour supprimer une branche:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git branch -d <nom-branche>|
```

- **Git pull**

Pour fusionner toutes les modifications présentes sur le dépôt distant dans le répertoire de travail local, la commande pull est utilisée. Usage:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git pull|
```

- **Git merge**

La **commande git merge** est utilisée pour fusionner une branche dans la branche active. Usage:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git merge <nom-branche>|
```

- **Git diff**

La **commande git diff** permet de lister les conflits. Pour visualiser les conflits d'un fichier, utilisez

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git diff --base <nom-fichier>|
```

La commande suivante est utilisée pour afficher les conflits entre les branches à fusionner avant de les fusionner:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git diff <branche-source> <branche-cible>|
```

Pour simplement énumérer tous les conflits actuels, utilisez:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git diff |
```

- **Git reset**

Pour réinitialiser l'index et le répertoire de travail à l'état du dernier commit, la commande **git reset** est utilisée :

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git reset --hard HEAD|
```

- **Git rm**

Git rm peut être utilisé pour supprimer des fichiers de l'index et du répertoire de travail. Usage:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git rm nomfichier.txt|
```

- **Git fetch**

Git fetch permet à un utilisateur d'extraire tous les fichiers du dépôt distant qui ne sont pas actuellement dans le répertoire de travail local. Exemple d'utilisation:

```
wonde@DESKTOP-IEMUSBP MINGW64 ~  
$ git fetch origin|
```