





Table des matières

Memento commande de Git	2
Variante Graphique avec C#.....	2
Installer Git et créer un espace de stockage.....	3
Récupérer les fichiers de notre projet et y sauvegarder les fichiers.....	4
Les branches.....	5
Pour fusionner les branches et les poster, commencez par:.....	6
D'autres commandes :.....	6
Installation et utilisation de git sur Visual Studio (C#).....	6

Memento commande de Git

Copier les fichiers de notre répertoire GitHub	<code>git clone</code> https://github.com/votre_pseudo/le_nom_du_projet
Remote un répertoire local	<code>Git remote add origin</code> https://github.com/votre_pseudo/le_nom_du_projet <code>git push -u origin master</code>
Créer un fichier (par de rapport spé avec git)	<code>touch index.html</code>
Création d'un répertoire local	<code>git init</code>
Mettre les nouveautés en attente de l'envoi	<code>git add .</code>
Mettre les nouveautés dans un "packet"	<code>git commit -m "message obligatoire"</code>
Mettre en ligne le "packet"	<code>git push</code>
Créer une branche	<code>git branch nom_de_la_branche</code>
Créer une branche et s'y mettre directement	<code>git checkout -b feature_x</code>
Aller sur une autre branche, ici je vais sur la main :	<code>git checkout main</code>
Supprimer une branche, on utilise:	<code>git branch -d nom_branch</code>
Supprimer une branche alors qu'elle n'a pas été "merge"	<code>git branch -D nom_branch</code>
Fusionner une deux branches NB: ici, je me trouve sur la branch main et je veux la fusionner avec feature_x	<code>git merge feature_x</code>
Voir quels sont les fichiers en attente d'envoi ainsi que la branch sur laquelle vous êtes:	<code>git status</code>
Voir la liste des branches:	<code>git branch -list</code>
Récupérer les changements d'un autre dev	<code>git pull</code>

Variante Graphique avec C#

 Modifications	Indique les modifications entre le code actuel et le dépôt. Après avoir cliqué sur le bouton, cliquez sur Actions/Afficher historique...
 Synchroniser	Les commandes pour récupérer, pousser ou simplement valider les changements.
 Branches	Permet de créer des branches ou de fusionner celles-ci.
 Paramètres	Permet de modifier les paramètres locaux et le fonctionnement de Git

Installer Git et créer un espace de stockage

Pour commencer, il faut installer git à l'adresse url suivante :

<https://git-scm.com/>

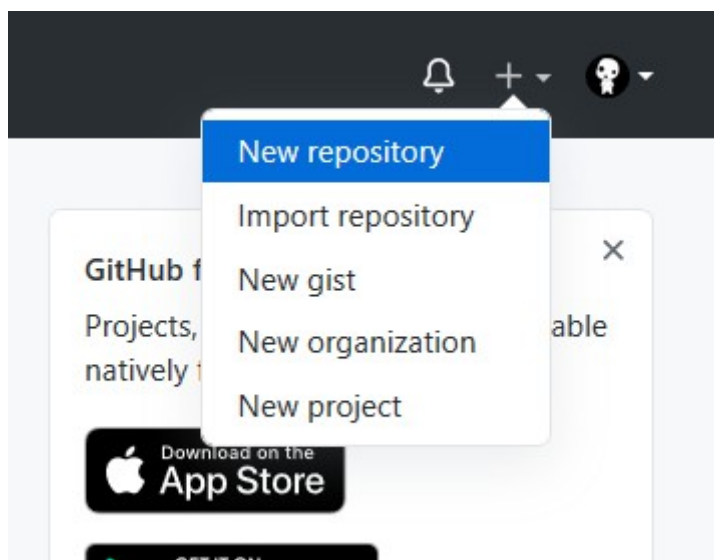
cliquez directement ici si vous voulez êtes sur windows :

<https://git-scm.com/download/win>

Ensuite il faut créer un compte sur le site de GitHub où nous allons obtenir notre futur espace de stockage:

<https://github.com/>

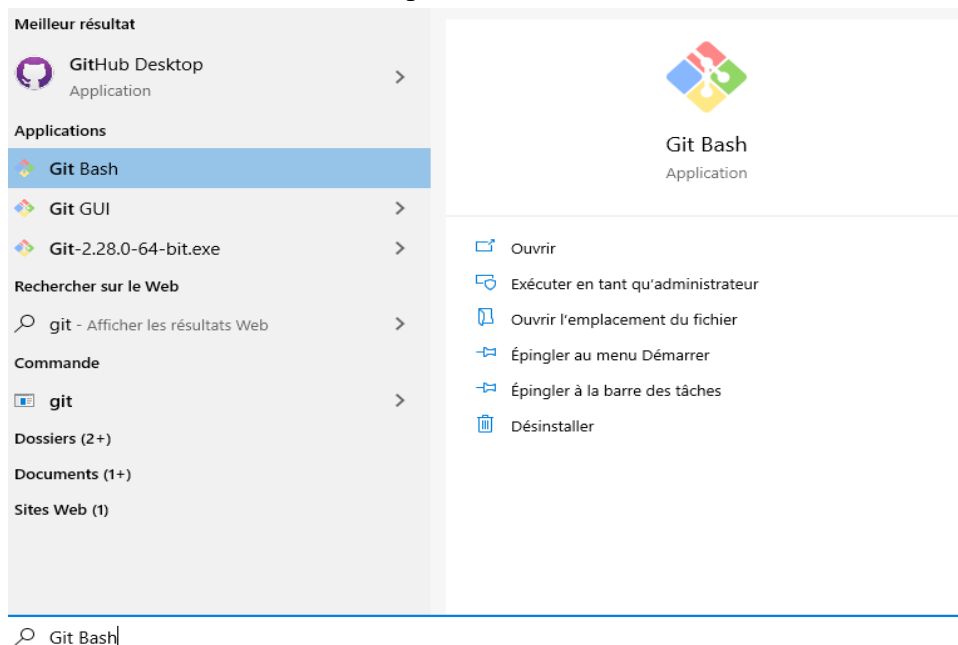
Enfin il faut utiliser, la petite croix en haut à droite pour créer un nouvelle espace de stockage:



Remplir alors le formulaire qui vous ai proposé. Il faut donné un nom, une description, si le projet est public ou privé, si vous ajouter un point texte readme pour donner une description plus importante sur votre projet et si vous souhaitez une license ou non. Une fois validé votre espace de stockage est créé.

Procédure d'Utilisation de Git et GitHub

Il faut maintenant en faire une copie sur votre ordinateur. Ouvrez Git Bash.



Pensez à aller vous positionnez sur l'endroit où vous voulez recevoir la copie de votre espace de stockage en utilisant `cd`. Si vous ne savez pas vous servir d'un terminal, je vous renvoi vers Internet. Mais pour mémoire, si vous voulez par exemple être sur le bureau :

```
fofof@UTILISA-QGG4D2C MINGW64 ~  
$ cd Desktop  
  
fofof@UTILISA-QGG4D2C MINGW64 ~/Desktop
```

Récupérer les fichiers de notre projet et y sauvegarder les fichiers

Utiliser ensuite la commande :

`git clone https://github.com/votre_pseudo/le_nom_du_projet`

Le lien est un exemple. Pour moi le lien donne:

`git clone https://github.com/Simeon-Phsta/remember_food`

L'url correspond simplement à celle de votre projet.

Voici ce que vous devez obtenir en validant votre commande:

```
fofof@UTILISA-QGG4D2C MINGW64 /c/wamp64/www/test_git  
$ git clone https://github.com/Simeon-Phsta/remember_food  
Cloning into 'remember_food'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), 601 bytes | 3.00 KiB/s, done.
```

Vous pouvez commencer à travailler sur votre projet. Je vais pour mon exemple, créer un fichier `index.php` contenant: `<?php echo("Hello World ! ") ?>`

Procédure d'Utilisation de Git et GitHub

Maintenant nous voulons mettre à jour notre espace de stockage.

Je vais reprendre mon terminal git et rentrer dans le répertoire ayant la nouvelle version de notre projet. Pour moi, comme mon projet se nome "remember_food", ma commande git donne:

```
cd remember_food
```

```
fofof@UTILISA-QGG4D2C MINGW64 /c/wamp64/www/test_git
$ cd remember_food
```

Nous allons ensuite tapez, soit la commande "git add *" afin de prendre en compte toutes les nouveautés ou soit, on précise le nom du fichier avec ici, dans mon exemple "git add index.php"

```
fofof@UTILISA-QGG4D2C MINGW64 /c/wamp64/www/test_git/remember_food (main)
$ git add index.php
```

On peut aussi utilisé aussi "git add -A" pour avoir les nouveautés.

Cela créer une liste temporaire de fichiers en attente de deux autres commandes pour être envoyer. D'une part, "git commit -m "premier code", qui créer un packet appelé commit, qui correspond tous les changements correspondants à une version. De plus, j'ai laissé une message, premier code.

```
fofof@UTILISA-QGG4D2C MINGW64 /c/wamp64/www/test_git/remember_food (main)
$ git commit -m "premier code"
[main dd33e46] premier code
1 file changed, 1 insertion(+)
create mode 100644 index.php
```

D'autre part, on envoie enfin notre commit avec la commande "git push". Il se peut que GitHub vous demande de vous connecter.

Les branches

Si vous devez travailler à plusieurs ou que vous voulez séparer les features de votre projet avant de les assemblés, il faut créer une branche.

Pour créé ici la branche de la feature_x et pour passer dessus pour l'utilise, on va utilier la commande:

```
git checkout -b feature_x
```

Puis tant qu'elle est pas push sur l'espace de stockage de github, la branche est toujours en local donc :

```
git push origin <branch>
```

Pour retourner sur la branche principal, on utilise :

```
git checkout main
```

Pour supprimer une branche, on utilise:

```
git branch -d feature_x
```

Pour fusionner les branches et les poster, commencez par:

`git merge feature_x`

NB: je suis revenue sur ma branche main

```
fofof@UTILISA-QGG4D2C MINGW64 /c/wamp64/www/test_git/remember_food (main)
$ git merge feature_x
Updating dd33e46..c5fb148
Fast-forward
 main.js   | 0
 style.css | 3 +++
2 files changed, 3 insertions(+)
create mode 100644 main.js
create mode 100644 style.css
```

Puis un simple:

`git push`

D'autres commandes :

Pour voir quels sont les fichiers en attente d'envoi ainsi que la branch sur laquelle vous êtes:

`git status`

Pour voir la liste des branches:

`git branch -list`

Pour récupérer les changements d'un autre dev:

`git pull`

Installation et utilisation de git sur Visual Studio (C#)

http://localhost/MVC_portefolio/public/procedure/utilisation_git_csharp.pdf