

Résumé des commandes de Git (basique)

SECTION INFORMATIQUE DE L'INSTITUT SAINT-MICHEL DE VERVIERS (ISMV)

Prérequis

- Avoir installé Git (<https://git-scm.com/downloads>)
- Avoir un terminal (invite de commande) où Git est fonctionnel
- Avoir un compte sur une plateforme collaborative de code
Exemple : [GitHub](#), [GitLab](#) et [Bitbucket](#)

Configuration

Cette étape est requise pour pouvoir utiliser Git et créer des commits.

Modifier le nom d'affichage de Git

```
$ git config --global user.name "John Doe"
```

Modifier l'adresse e-mail de Git

```
$ git config --global user.email "john@doe.com"
```

Créer un repository (dépôt) local (sur votre ordinateur)

Via un terminal, situez-vous dans un dossier où sera stocké votre code et exécutez la commande :

```
$ git init
```

Astuce : `$ cd <chemin du répertoire>`, à l'aide d'un explorateur de fichier, faites un glisser-déposer de votre dossier dans la fenêtre du terminal après avoir tapé la commande `cd`. Le chemin du fichier devrait apparaître.

Créer un repository (dépôt) distant (sur une plateforme collaborative de code)

Rendez-vous sur une plateforme comme GitHub, GitLab ou encore Bitbucket, créez un compte et/ou connectez-vous. Une fois connecté, créez un dépôt :

Sur GitHub : <https://help.github.com/en/github/getting-started-with-github/create-a-repo>

Sur GitLab : <https://docs.gitlab.com/ee/gitlab-basics/create-project.html>

Sur Bitbucket : <https://www.atlassian.com/fr/git/tutorials/learn-git-with-bitbucket-cloud>

Connexion à un repository (dépôt) distant

Pour connecter votre dépôt local à un dépôt distant, il faut copier l'URL de votre dépôt distant (disponible sur la plateforme collaborative de code) et la remplacer dans la commande suivante (en ajoutant ".git" à la fin) :

```
$ git remote add origin https://github.com/johndoe/hello-world.git
```

Vérifier le statut des fichiers d'un repository (dépôt)

Affiche l'arborescence de votre dépôt en montrant les fichiers ajoutés, supprimés ou modifiés.

```
$ git status
```

Ajouter ou supprimer des fichiers à un commit

Ajouter un ou plusieurs fichier(s)/répertoire(s) au commit courant.

À noter que lorsque vous ajoutez le fichier, vous enregistrez un instantané du/des fichier(s), c'est-à-dire que vous enregistrez la version actuelle du/des fichier(s). Si vous le(s) modifiez, vous devrez les ajouter à nouveau.

```
$ git add <fichier/répertoire>
```

Supprimer un ou plusieurs fichier(s)/répertoire(s) au commit courant.

```
$ git rm <fichier/répertoire>
```

Astuces :

- **Ajouter tous les fichiers à un commit**
Lorsque vous souhaitez ajouter l'ensemble des fichiers, vous pouvez remplacer le <fichier/répertoire> par un "." (un point).
- **Exclure des fichiers/répertoires du suivi de version**
Si certains de vos fichiers ou répertoires ne doivent pas être ajoutés aux commits (y compris lorsque vous exécutez la commande `$ git add .`), vous pouvez créer un fichier nommé ".gitignore" et y lister les fichiers ou répertoires comme suit :

```
nodes_modules  
.DS_STORE  
fichier_à_exclure.html
```

Créer un commit

Enregistre les modifications, c'est-à-dire les fichiers ajoutés au préalable via la commande git add

```
$ git commit -m "Mon message"
```

Envoyer un/plusieurs commit(s) vers un repository (dépôt) distant

Envoie le(s) commit(s) local/locaux vers le dépôt distant.

```
$ git push origin master
```

Une fois la commande exécutée, vous devriez voir les modifications sur le dépôt présent sur la plateforme collaborative de code de votre choix.

Astuce : ajouter le drapeau `--set-upstream` ou `-u` à la commande permet d'éviter de devoir spécifier à chaque fois "(...) origin master". Ce qui donne cette commande :

```
$ git push -u origin master
```

Ainsi, la prochaine fois il ne sera plus nécessaire de mentionner "(...) origin master".

Mettre à jour un dépôt local

Met à jour l'historique de version du dépôt local et télécharge les dernières modifications depuis un dépôt distant.

```
$ git pull origin master
```

Télécharger un repository (dépôt)

Télécharge un projet et son historique de versions depuis le dépôt distant. À noter que les projets que vous téléchargez via la commande ci-dessous appliqueront automatiquement le [remote](#).

```
$ git clone https://github.com/johndoe/hello-world.git
```

DISCLAIMER : Ce document ne reprend pas l'ensemble des commandes de Git mais uniquement celles qui sont essentielles pour une utilisation simple de l'outil. Il a pour objectif d'être facile à comprendre afin de permettre une première approche sans entrer trop dans les détails. Par conséquent, certaines notions ne sont pas abordées ici comme les branches, clés SSH, ...

Ce travail est sous licence [Attribution-ShareAlike 4.0 International \(CC BY-SA 4.0\)](#).

