**Organisation de dossiers et GitHub**

Table des matières

[1. Organisation de dossiers 2](#_Toc91623542)

[2. GitHub 2](#_Toc91623543)

# 

## Organisation de dossiers

* Pour bien s'organiser dans les différentes tâches et missions lors de cette formation ou entreprises il est primordial d'avoir une bonne organisation pour bien se repérer dans les données et les travaux effectués ou en cours.
* Pour cela on crée un répertoire qu'on place dans les favoris et barre d'outils sur les pc qu'on utilise pour travailler.
* Dans ce répertoire (fbi pour mon cas) on crée différents dossiers tels que : cours, install, data, dev, deploy, util, autres, …
* On enregistrera tout ce qu'on fait dans les dossiers correspondants pour s'y retrouver.

## GitHub

* Github est une plateforme d'hébergement de code pour le contrôle de version et de collaboration. Elle permet à différentes personnes de travailler ensemble sur le même projet n'importe où.
* Dans notre cas, elle permet de synchroniser les versions et le contenu du répertoire "fbi" sur le post htcompass et le pc à la maison via la centrale de dépôt github.
* On crée le compte sur github.com avec un login et password. Et installer git bash ou la console.
* Ensuite, il faut connaître quelques commandes de base à taper sur la console pour gérer les versions:

- git init (pour créer un nouveau dépôt Git. Elle peut être utilisée pour convertir un projet existant, non versionné, en un dépôt Git ou pour initialiser un nouveau dépôt vide.)

- git status (repère et pour voir le statut des modifications)

- git add . (Pour ajouter les untracked files ou dossiers non suivis)

- git commit -am "commentaire" (pour enregistrer les modifications locales sur une branche. Dans notre cas c'est "master". Le commentaire sert d'un repère pour mieux s'y retrouver sur la branche)

- git push (pour publier un changement local vers le dépôt centralisé)

- git pull (pour aller chercher et télécharger le contenu d'un dépôt distant et mettre à jour le dépôt local)

- git log (pour afficher l'historique des modifications dans le répertoire)

- history (liste des commandes de base effectuées sur la console)

- git clone URL (pour cloner le répertoire sur un autre pc ou dans un autre emplacement sur le même pc)

- cd \_ (change directory, pour créer un sous repertoire contenant les fichiers)

* Raccourci utile:

- touches [Shift] + [Inser] pour coller un élément sur la console.