GIT

Le versionning pour les Nuls

1. Principes fondamentaux

Git est un outil gestionnaire de version, qui permet l'**historisation** des modifications sur un projet, mais également la gestion de potentiels **conflits** lorsque l'on travaille en groupe sur un même projet. Il permet la gestion de différentes versions d'un projet, d'isoler des développements sur un projet même quand le projet est déjà livré, permet de **restaurer** des versions précédentes si on a mis en prod une bêtise…

Bref, c'est un outil purement et simplement **indispensable**, que l'on travaille seul ou à plusieurs.

Il s'articule autour de plusieurs concepts importants à comprendre, dont voici une liste non-exhaustive :

- Un **dépôt**, ou un **repository** en anglais, est l'endroit où sont stockés les fichiers sur lesquels on travaille. Il se découpe en plusieurs parties : un dépôt **distant**, c'est-à-dire stocké sur GitHub, et un ou plusieurs dépôts **locaux**.
- Le **dépôt distant** est le référentiel de notre projet. C'est lui qui détiendra la vérité pour tous les autres : tout doit transiter par ce dépôt distant si l'on veut historiser les modifications. On ne développe **pas** sur le dépôt distant.
- Le ou les dépôts **locaux** sont des **clones** du dépôt distant sur lequel nous allons pouvoir développer. Il faut visualiser la chose comme ceci : On part d'un dépôt distant que l'on **clone** en local, puis on fait notre développement en local avant de le renvoyer vers le dépôt distant.
- Un **commit** est l'action d'historiser des modifications que l'on vient de faire sur un dépôt local. Tant que je n'ai pas **commit** ces modifications, personne d'autre ne pourra les voir sur le dépôt distant : elles sont sur le dépôt local seulement et ne sont pas encore historisées.
- Un **push** est l'action de **pousser** les commits effectués depuis un dépôt local vers le dépôt distant. C'est comme ça que l'on va rendre ces modifications visibles par les autres utilisateurs du dépôt.
- Un **pull** est l'action de **tirer** les modifications depuis un dépôt distant vers un dépôt local. Par exemple si une modification a été mise en ligne sur le dépôt distant, je veux bénéficier de la même modification sur mon dépôt local, alors je vais faire un **pull** de ces modifications depuis le dépôt distant vers mon dépôt local. On ne fait jamais de **pull** directement sur le dépôt distant

2. Installation et usage

Note : Pour utiliser GIT, vous allez avoir besoin d'un compte GIT.

Une fois que ce compte est créé, passons à la suite :

Pour installer GIT pour Windows, se rendre ici :

https://git-scm.com/install/windows

Une fois que cela est installé, vous devriez avoir accès à l'alias « git » dans le terminal **Powershell**

Il faudra alors dire à votre terminal d'utiliser votre compte GIT par défaut pour toutes les actions. C'est comme cela que vous serez authentifié à chaque commit / push / pull etc… Nous allons donc exécuter ces deux commandes (remplacer les emails et pseudonymes par les vôtres) :

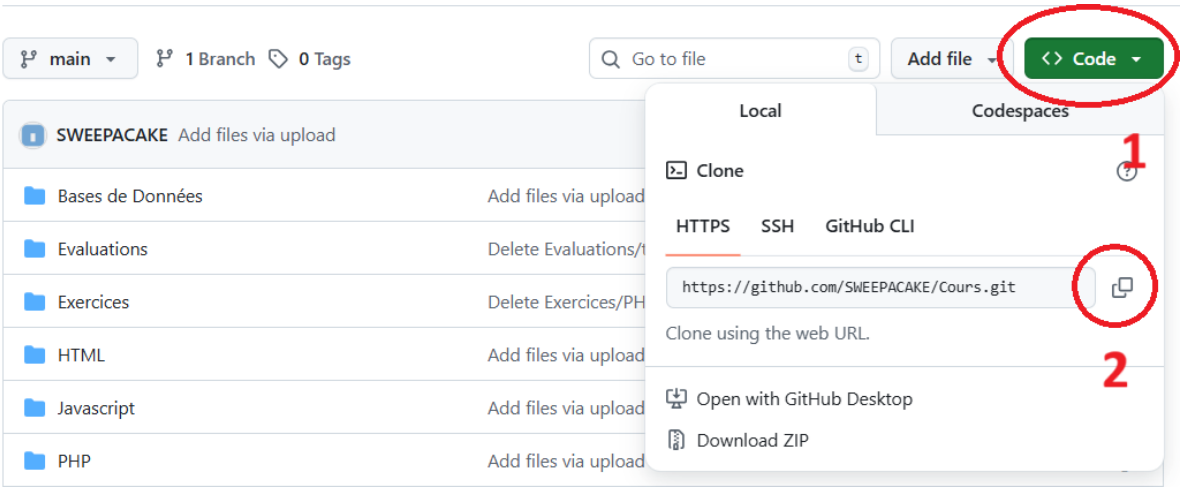
```
git config --global user.name "John Doe"

git config --global user.email johndoe@example.com
```

3. Créer un dépôt

Maintenant que la configuration est faite, vous pouvez utiliser GIT presque comme bon vous semble.

Admettons que vous voulez créer un dépôt de toute pièce. Même si vous avez déjà du développement de commencé dans votre répertoire, mais qui n'est pas encore sur Git, ce n'est pas grave. Rendez-vous dans votre dossier de travail en utilisant la commande **cd**, puis :

- Utilisez la commande **`git init`**. Cela va créer un dossier .git dans votre répertoire, nécessaire pour stocker la configuration du dépôt et son historique.
- Utilisez la commande **`git add .`** Avec le point. Ce point signifie « tout ce qui est dans ce répertoire ». Cette commande va donc dire à Git de commencer à suivre tous les fichiers qui sont dans ce répertoire.
- Ensuite on va se rendre sur GitHub, et créer notre nouveau **dépôt distant** à la main. Une fois notre dépôt créer, dans le bouton à droite, en vert, nous allons avoir ceci :



Cliquer sur le bouton 1, en vert, puis sur le petit picto dans le cercle 2. Cela va vous permettre de **copier** le lien de votre dépôt. Une fois ce lien copier, vous allez pouvoir retourner dans votre Powershell et taper :

```
git remote add origin LELIENQUEVOUSAVEZCOPIE
```

Cela va permettre à GIT de faire la correspondance entre votre dépôt local et le dépôt distant.

- Enfin, nous allons pouvoir faire notre premier commit. Ce commit va envoyer tous les fichiers qui ont été ajouter avec **`git add .`** avec un message si on précise « -m »
  `git commit -m 'Premier commit'`
- Enfin, une fois que le commit est fait, on va pouvoir le pousser sur le dépôt distant avec :
  `git push -u origin master`

  Vérifiez bien que vous retrouvez ces modifications sur le dépôt distant

4. Récupérer un dépôt distant

Heureusement, si vous arrivez sur un projet en cours avec du dev déjà commencé et tout de déjà créé, la procédure est **beaucoup plus simple**. Pour cela, il suffit en effet de créer un dépôt local **supplémentaire** sur votre machine, avec la commande suivante que vous exécuterez une fois placé dans votre répertoire de travail :

```
git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```

Cet URL que vous allez copier est obtenue exactement de la même façon que dans l'étape décrite au dessus : on va se rendre sur GitHub, sur le dépôt en question, ouvrir la petite fenêtre « Code <> » et cliquer sur le petit picto qui nous permet de copier le lien.