Capsule 2 – Commandes de base et bonnes pratiques de collaboration

Hubert Cadieux et Arnaud Beaulé

Introduction (30 sec)

Bonjour tout le monde !

Dans la première capsule, nous avons vu ce qu'étaient Git et GitHub, pourquoi ces outils sont utiles en sciences sociales, et nous avons appris le vocabulaire de base. Maintenant, il est temps de passer à la pratique !

Dans cette deuxième capsule, nous allons voir concrètement comment collaborer avec Git et GitHub. Pour illustrer tout ça, nous allons suivre l'exemple d'Hubert et d'Arnaud, deux étudiants en sciences sociales qui travaillent ensemble sur un projet d'analyse sur le développement économique et social des pays.

À travers leur collaboration, vous allez apprendre les commandes essentielles, comprendre le flux de travail collaboratif, et découvrir les bonnes pratiques qui vous feront gagner du temps et éviter les problèmes.

> **Note**
>
> Pour cette capsule, on pourrait partager l'écran en changeant la couleur du header selon l'étudiant : une couleur pour Hubert, une autre pour Arnaud, avec une icône différente pour chacun. Ça rendrait clair quand on passe d'un étudiant à l'autre.

Démarrer un projet collaboratif avec Git et GitHub (2 min)

Alors, commençons par la première étape. Dans tout projet collaboratif avec Git et GitHub, quelqu'un doit d'abord créer le dépôt principal sur GitHub. C'est ce qu'on appelle "initialiser le projet".

(Note affichée au bas de l'écran : Rappel du vocabulaire vu dans la capsule 1)

ÉCRAN D'HUBERT

Hubert prend l'initiative de créer le projet. Il se rend sur github.com et navigue vers l'organisation où il veut héberger leur projet collaboratif *(Est-ce qu'on crée une organisation FSS? Est-ce qu'on montre comment créer sa propre organisation? Est-ce qu'on fait l'exemple dans un repo personnel?)*.

Il clique sur l'onglet `Repositories` (traduction de dépôt) et puis sur `New repository`.

Il remplit ensuite le formulaire avec les informations essentielles:

- un nom de dépot minuscules, sans accents et sans espace
- le choix entre privé ou public
- le choix d'un fichier readme.

Les autres

README.md

.gitignore

Commandes Git essentielles pour collaborer (2 min)

- **`git clone` : cloner un dépôt distant sur son ordinateur**
- **`git pull` : récupérer les dernières modifications du dépôt distant**
- **`git push` : envoyer ses modifications vers le dépôt distant**

Flux de travail collaboratif sur GitHub (2 min 30)

- **Création de branches pour développer de nouvelles fonctionnalités ou corriger des bugs**
- **Pull requests : proposer, discuter et intégrer des modifications dans la branche principale**
- **Résolution de conflits lors des fusions**

Bonnes pratiques de collaboration (2 min 30)

- **Toujours travailler sur des branches dédiées**
- **Écrire des messages de commit clairs et explicites**
- **Utiliser un fichier README et documenter le projet**
- **Revue de code : demander et donner des retours via les pull requests**
- **Protéger la branche principale (main/master) contre les modifications directes**

Choix entre interface graphique (GUI) et terminal (CLI) (1 min)

- **Avantages de la GUI (GitHub Desktop) pour débuter**
- **Importance de connaître les commandes de base en terminal pour mieux comprendre le fonctionnement de Git**

Fonctionnement général du versionnage (1 min 30)

- **Historique des modifications**
- **Revenir à une version antérieure**
- **Suivi des contributions individuelles**