

Bloc1

Développer la présence en ligne de l'organisation

4h

Travail collaboratif sur Git/GitHub

Objectifs pédagogiques :

Pré-requis :



- Disposer d'une connexion internet et de droits administrateur sur son poste de travail.

- Comprendre l'intérêt professionnel et académique de Git/GitHub.
- Manipuler les commandes de base (init, clone, add, commit, push, pull).
- Créer un support réutilisable (tuto/aide-mémoire) pour la classe.
- Collaborer via un dépôt partagé (simulation de travail d'équipe).

Phase 1 : Découverte et recherche (1h)

Consigne : "Par groupes de 2, explorez Git et GitHub pour répondre aux questions suivantes. Vous présenterez vos trouvailles à la classe en 5 min max par groupe."

Questions à investiguer:

1. Pourquoi utiliser Git/GitHub ?
2. En entreprise ?, En cours ?
3. Qu'est-ce qu'un dépôt (repo) ? Un commit ? Une branche ?
4. Schéma simple à dessiner pour expliquer.
5. Quelles sont les commandes Git de base ? (à tester en local ou via GitHub Desktop).
6. Quelle est la différence entre Git et GitHub ?

Ressources autorisées :

- [Documentation officielle Git](#) (simplifiée).
- [GitHub Guides](#) (tutoriels visuels).
- [Git immersion](#)
- Vidéo courte : "[Git & GitHub Crash Course](#)" ([Traversy Media](#)) (15 min).

Livrable :

Un support visuel (Google Slides, canva, ou fichier Markdown) pour la présentation.

Et l'IA ?

"Vous pouvez utiliser l'IA (ChatGPT, Bing, GitHub Copilot, etc.) **uniquement** pour :

1. **Comprendre un concept** (ex: 'Explique-moi comme à un débutant ce qu'est un conflit de merge').
2. **Débloquer une erreur** (ex: 'Pourquoi git me dit "fatal: not a git repository" ?').
3. **Améliorer vos explications** (ex: 'Rédige une analogie pour expliquer les branches Git').

Il est interdit de :

- Copier-coller des réponses sans les adapter.
- Demander à l'IA de faire le travail à votre place (ex: 'Écris-moi un README.md complet').
- Utiliser l'IA pour générer du code sans le comprendre.

Bloc1

Développer la présence en ligne de l'organisation

Phase 2 : Pratique guidée (1h30)

Consigne : "Passons à la pratique ! Chaque groupe doit :

1. *Créer un compte GitHub (si ce n'est pas déjà fait).*
2. *Créer un dépôt (repo) nommé SIO-Git-Tuto-VotreGroupe.*
3. *Ajouter un fichier README.md avec :*
 - Une explication **en vos mots** de l'utilité de Git.
 - La liste des commandes de base (avec des exemples).
4. *Faire un commit/push de ce fichier.*
5. *Inviter un autre groupe à contribuer (pull request) en ajoutant une section "Bonnes pratiques".*

Étapes :

1. **Installer Git** : [Lien pour Windows/Mac/Linux](#).
2. **Configurer Git** (en ligne de commande) :
3. **Créer un repo sur GitHub** (via l'interface web).
4. **Cloner le repo en local** :
5. **Modifier le README.md**,
6. **Collaborer** : Un groupe fait un fork du repo d'un autre groupe, ajoute une section, et crée une **pull request**.

Outils alternatifs :

- Pour les réticents à la ligne de commande : [GitHub Desktop](#).
- Éditeur Markdown : [StackEdit](#) ou VS Code.

Bloc1

Développer la présence en ligne de l'organisation

Phase 3 : Création d'un aide-mémoire collaboratif (1h)

Consigne : "Maintenant, compilez vos trouvailles dans un **guide commun** pour la classe. Chaque groupe est responsable d'une section."

Organisation :

1. **Sur le repo commun** : SIO-2025-Git-CheatSheet.
2. **Répartition des thèmes entre groupes** (exemples) :
 - Installation et configuration de Git.
 - Création repo + premier commit.
 - Travail (clone, commit, push, pull...).
 - Collaboration (fork, pull request, résolution de conflits).
 - Bonnes pratiques (messages de commit, .gitignore).
 - Travail sur branches (intérêt et modalités)
3. **Format imposé** : Fichier README.md avec :
 - Titres clairs (##, ###).
 - Exemples de code dans des blocs
 - Captures d'écran si utile.

Validation :

- Les pull requests seront mergées **uniquement si** :
 - Le contenu est original (pas de copier-coller).
 - Les commandes sont testées et fonctionnelles.
 - La syntaxe Markdown est correcte.

Bloc1

Développer la présence en ligne de l'organisation

Phase 4 : Restitution et débat (30 min)

1. **Présentations éclair** (5 min/groupe) :
 - Chaque groupe présente **une commande ou un concept** (ex: "Pourquoi git pull avant de pousser ?").
2. **Débat** :
 - "*GitHub est-il indispensable pour un développeur, un admin réseau ?*" (arguments pour/contre).
 - "*Quels pièges avez-vous rencontrés ?*" (ex: conflits de merge, erreurs de chemin).
3. **Tour de table** :
 - "*Une chose que vous avez apprise aujourd'hui.*"
 - "*Une question qui reste sans réponse.*"