**Compte rendu**

**TD1 : Utilisation de Git**

**Configuration globale de Git**

1. Configurer le nom et l'adresse e-mail pour Git :

git config --global user.name "votre nom"

git config --global user.email "votre adresse mail"

1. Enregistrer les identifiants pour ne pas saisir le mot de passe à chaque fois :

git config --global credential.helper store

**Mise en place d'un dépôt local**

1. Initialiser un dépôt local Git :

git init

1. Vérifier les fichiers/répertoires non suivis par Git :

git status

1. Ajouter tous les fichiers pour les suivre :

git add …..

1. Vérifier que les fichiers sont indexés :

git status

1. Faire un premier commit :

git commit -m "Initialisation du dépôt TypeScript"

1. Vérifier le commit et l'état du dépôt local :

git log ( q pour quitter )

**Mise en place d'un dépôt distant**

1. Créer un projet privé sur GitLab :**td-git**.
2. Associer le dépôt local au dépôt distant :

git remote add origin https://gitlab.univ-lorraine.fr/e27844u/td-git

1. Vérifier la connexion au dépôt distant :

git remote -v

1. Pousser les modifications locales sur le dépôt distant :

git push -u origin master

1. Vérifier que les deux dépôts sont synchronisés :

git status

**Finalisation du TD**

1. Dernier commit du compte-rendu :

git add compte\_rendu.docx

git commit -m "Rendu final du compte-rendu"

git push

**TD2 : Qualité de développement**

**Clonage d'un dépôt distant**

* git clone <URL> : permet de créer une copie locale d'un dépôt distant.

**Collaboration et gestion des conflits**

* git pull : met à jour le dépôt local avec les dernières modifications du dépôt distant.
* git merge <branche> : fusionne une branche distante avec la branche locale, utilisé pour résoudre les conflits après un pull.
* git merge --abort : annule une tentative de fusion si un conflit est détecté.
* git add <fichier> : après avoir résolu un conflit, permet d'ajouter le fichier pour indiquer que le conflit est résolu.
* git commit -a -m "<message>" : enregistre les changements et résolutions de conflits dans l'historique des commits.
* git log : permet de visualiser l'historique des commits.

**Gestion des erreurs :**

**Problème de divergence** :

* Si le dépôt distant a été mis à jour pendant que vous travaillez en local, il faut faire un pull avant de push :
* git pull
* git push

**Conflit lors d'un pull** :

* Si des changements locaux n'ont pas été commités, il faut les commit avant de faire un pull :
* git commit -a -m "Sauvegarde avant pull"
* git pull

**Règles importantes**

* **Règle d'or** :  
  Toujours faire un git pull avant un git push.  
  Toujours faire un git commit avant un git pull.  
  La séquence idéale : commit → pull → push.

**TD3 : Débug**

**Débogage avec des points d'arrêt**

* Un point d'arrêt permet d'arrêter l'exécution du programme à un moment précis afin d'inspecter les valeurs des variables en temps réel.
* Il est utilisé pour suivre l'évolution des variables à l'intérieur d'une boucle par exemple.
* Observer grâce au débogueur que des variables du même nom peuvent exister dans des contextes différents.