

*Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayant droit ou ayant cause est illicite. Il en est de même pour la traduction, l'adaptation ou la transformation, l'arrangement ou la reproduction par un art ou un procédé quelconque,*



# Versioning

## Travaux Pratiques

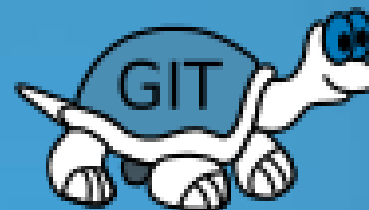
Naby Daouda Diakite



GitLab



GitHub



# Programme

- I. Utilisation des commandes de base via Git Bash
- II. Utilisation des commandes de base via TortoiseGit
- III. Simulation du travail en équipe via TortoiseGit
- IV. Cas d'étude (Git Bash)

## TP - Partie 1

### Utilisation des commandes de base via Git Bash (1)

- **Positionner chaque personne sur un repo**
  - [https://gitlab.dtek.fr/formation/git-tp-\[01 à 12\].git](https://gitlab.dtek.fr/formation/git-tp-[01 à 12].git)
- Commandes à voir
  - Récupération d'un repo en local
  - Ajout d'un nouveau fichier
  - Publication des modifications
  - Mise à jour de la version locale

## TP - Partie 1

### Utilisation des commandes de base via Git Bash (2)

- Commandes à voir
  - Historique des modifications
- Création d'une branche
  - Ajout d'un nouveau fichier
  - Publication des modifications
- Merge d'une branche
- Suppression d'une branche

## TP - Partie 2

### Utilisation des commandes de base via TortoiseGit

- **Positionner chaque personne sur un repo**
  - [https://gitlab.dtek.fr/formation/git-tp-\[01 à 12\].git](https://gitlab.dtek.fr/formation/git-tp-[01 à 12].git)
- Commandes à voir
  - Récupération d'un repo en local
  - Ajout d'un nouveau fichier
  - Publication des modifications
  - Mise à jour de la version locale

## TP - Partie 3

### Simulation du travail en équipe via Tortoise Git

- **Positionner tout le monde sur le même repo**
  - <https://gitlab.dtek.fr/formation/git-tp-all.git>
- Commandes à voir
  - Récupération d'un repo en local
  - Ajout d'un nouveau fichier / personne
  - A tour de rôle
    - Mise à jour de la version locale
    - Publication des modifications
  - Historique des modifications

## TP - Partie 4

### Cas d'étude (Git bash) (1)

- **Positionner chaque personne sur un repo**
  - [https://gitlab.dtek.fr/formation/git-tp-\[01 à 12\].git](https://gitlab.dtek.fr/formation/git-tp-[01 à 12].git)
- Commandes à voir
  - Récupération du repo en local
  - Supprimer tous les fichiers du repo et publier ces changements
  - Ajouter un fichier « readme.md » décrivant le projet en quelques lignes sur la branche « master », puis publier le fichier
  - Créer une branche « develop » pour les versions en cours de développement à partir de la branche « master »

## TP - Partie 4

### Cas d'étude (Git bash) (2)

- Commandes à voir
  - Créer une branche « feature/authenticate » pour la gestion de la connexion à partir de la branche « develop »
  - Créer un répertoire « src » et y ajouter un fichier « user.ts » dans lequel vous ajouterez la déclaration d'une classe typescript avec la description ci-dessous. Puis publier l'ensemble sur le repo distant « feature/authenticate »
    - Description de la classe « user »
      - Attributs : id (number) / lastName (string) / firstName (string) / age (number)
      - Méthodes : displayInfos():string ➔ affiche les informations



## TP - Partie 4

### Cas d'étude (Git bash) (3)

- Commandes à voir
  - Créer une branche « feature/order » pour la gestion des commandes à partir de la branche « develop »
- Dans le répertoire « src », ajouter un fichier « order.ts » dans lequel vous ajouterez la déclaration d'une classe typescript avec la description ci-dessous. Puis publier l'ensemble sur le repo distant « feature/order »
  - Propriétés de la classe « order »
    - Attributs : `userId (number)` / `productReference (string)` / `quantity (number)`
    - Méthodes : `process():void` ➔ affiche « Order has been handled »

## TP - Partie 4

### Cas d'étude (Git bash) (4)

- Commandes à voir
  - Merger la branche « feature/order » dans la branche « develop »
  - Merger la branche « develop » dans la branche « master »
  - Merger la branche « master » dans la branche « feature/authenticate »

## TP - Partie 4

### Cas d'étude (Git bash) (5)

- Commandes à voir
  - Sur la branche « feature/authenticate », ouvrir le fichier « user.ts » et y ajouter à la fin un commentaire :  
« //TODO: This feature could evolve after client's feedbacks ». Puis, publier ces changements.
  - Merger la branche « feature/authenticate » dans la branche « develop »
  - Merger la branche « develop » dans la branche « master »

## TP - Partie 4

### Cas d'étude (Git bash) (6)

- Commandes à voir
  - Supprimer les branches « feature/authenticate » et « feature/order » sur le repo distant
  - Sur la branche « feature/authenticate » en local, ouvrir le fichier « user.ts » et supprimer le commentaire : « //TODO: This feature could evolve after client's feedbacks ». Puis, publier ces changements sur une branche du repo distant.

## TP - Partie 4

### Cas d'étude (Git bash) (7)

- **Positionner chaque personne sur un repo**
  - [https://gitlab.dtek.fr/formation/git-tp-\[01 à 12\].git](https://gitlab.dtek.fr/formation/git-tp-[01 à 12].git)
  - ➔ Echanger son repo avec son voisin pour la suite
- **Commandes à voir**
  - Merger la branche « feature/authenticate » dans la branche « develop »
  - Merger la branche « develop » dans la branche « master »
  - Supprimer les branches « feature/authenticate »