



**GitHub**



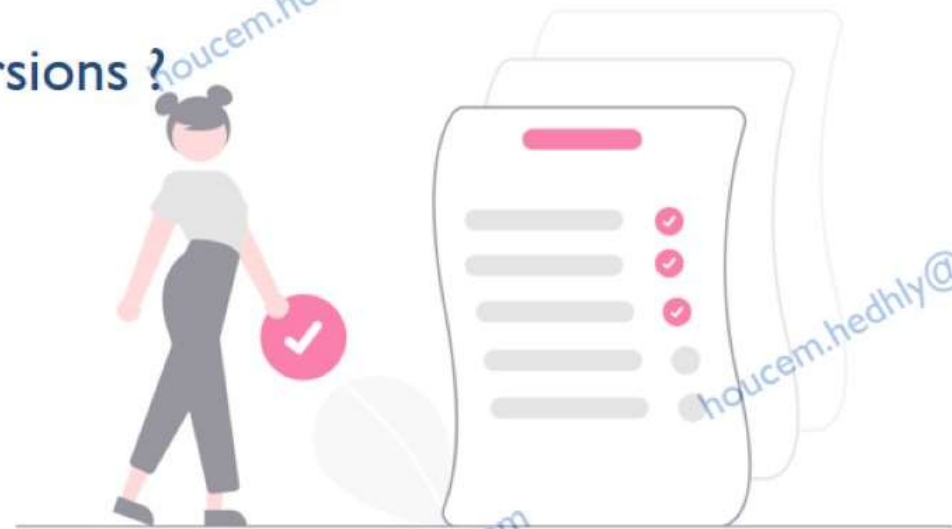
**git**

# INITIATION À GIT ET GITHUB

HOUCEM HEDHLY

# Plan

- Objectifs
- C'est quoi Git et le contrôle de versions ?
- Pourquoi Git est important ?
- C'est quoi GitHub ?
- Installation
- Workflow Git
- Les commandes de base Git



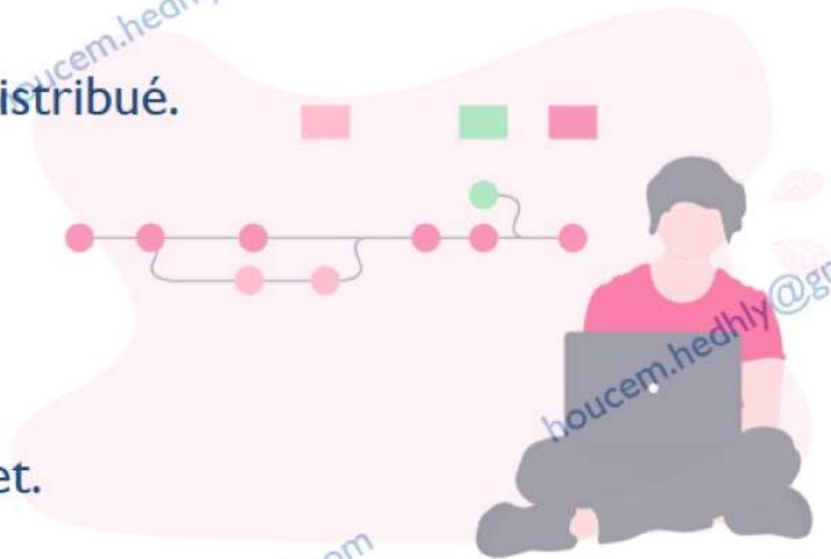
# Objectifs

- Comprendre le principe de contrôle de versions.
- Utiliser les commandes de base Git.
- S'initier à utiliser Git dans les projets personnels.
- S'initier à contribuer dans des projets utilisant le contrôle de versions.
- Utiliser un service cloud de serveur Git.



# C'est quoi Git et le contrôle de versions ?

- Git est un système de contrôle de versions distribué.
- Git est gratuit et open source.
- Garde la trace des projets et des fichiers.
- Enregistre les modifications.
- Historique des différentes versions d'un projet.

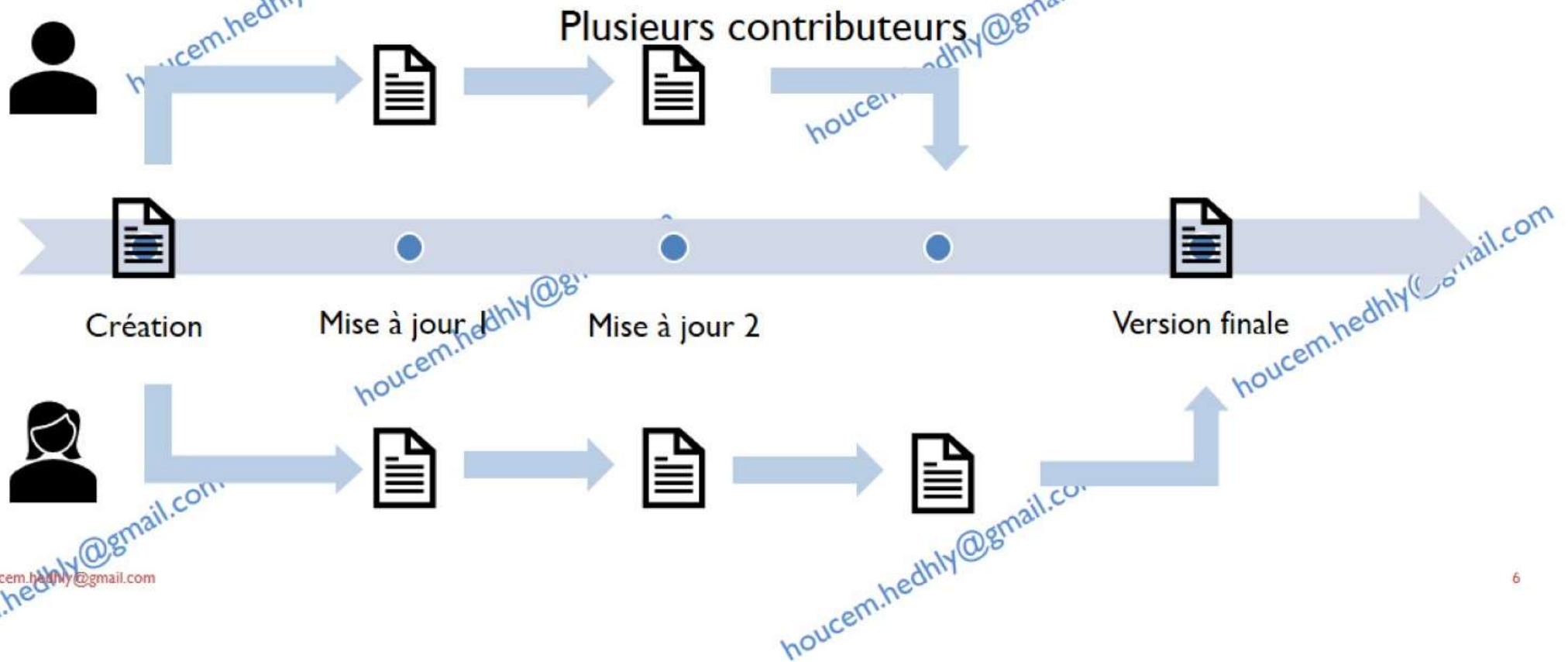


# C'est quoi Git et le contrôle de versions ?

## Cycle de vie d'un fichier



# C'est quoi Git et le contrôle de versions ?





## Pourquoi Git est important ?

- Qu'est-ce qu'on a modifié exactement ?
- Qui a fait la modification?
- Quand est-ce que une modification a été apporté ?
- Pourquoi on a fait la modification ?
- Rechercher et restaurer une ancienne version.



## Pourquoi Git est important ?

- Créé en 2005 par Linus Torvalds.
- Utilisé par la quasi totalité des entreprises dans le monde.
- Serveur Git privé / Plateforme commerciale d'hébergement.
- Utilisé par plus que 88% des développeurs dans le monde.





# Pourquoi Git est important ?

**In case of fire**



**1. git commit**



**2. git push**



**3. leave building**

# C'est quoi GitHub?

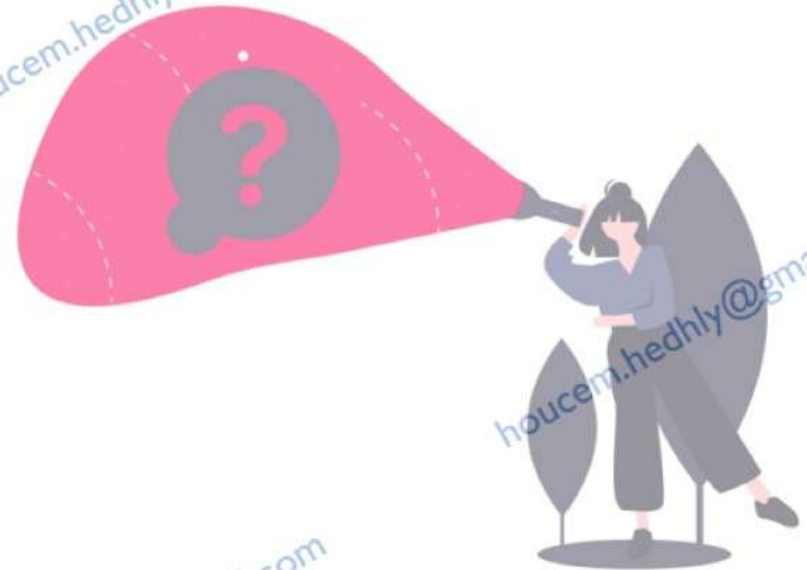
- Plateforme web d'hébergement.
- Serveur Git comme un service.
- Serveur Git distant avec une interface Web très conviviale.
- Fondée en 2007. Acheté par Microsoft en 2018 pour \$7.5B.
- La plus grande part de marché par +40M de comptes utilisateurs.



# C'est quoi GitHub?

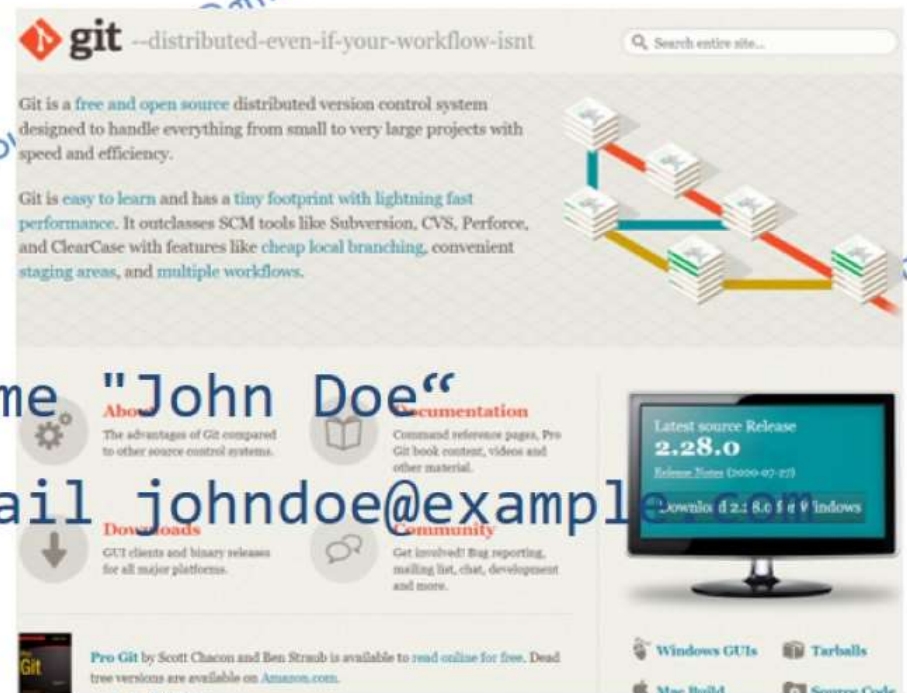
Exemples d'alternatives à GitHub:

- GitLab.
- BitBucket.
- SourceForge.
- LaunchPad.
- Google Cloud Source Repositories
- AWS CodeCommit

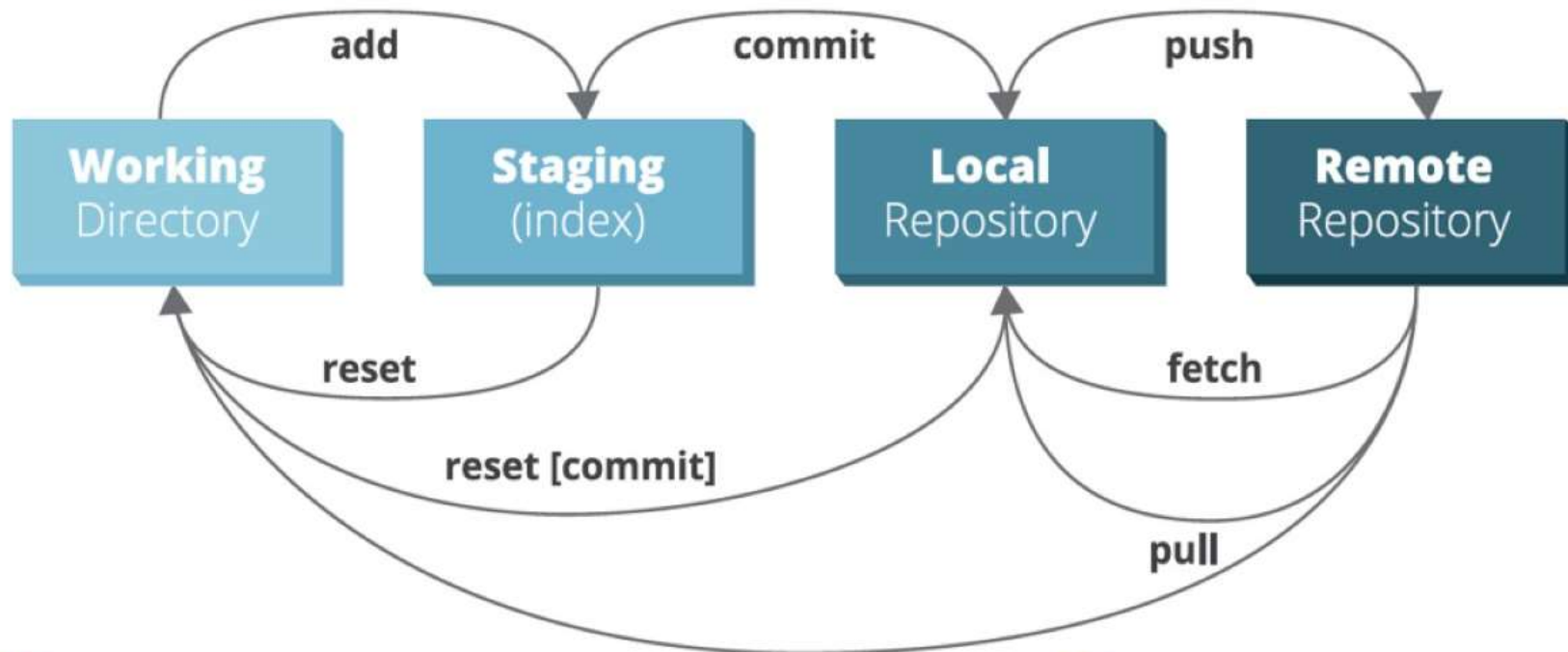


# Installation

- <https://git-scm.com/>
- `$ git --version`
- `$ git config --global user.name "John Doe"`
- `$ git config --global user.email johndoe@example.com`



# Workflow Git





# Les commandes de base Git

## Créer un repository

- Initier un nouveau repository local:

```
$ git init [project name]
```

- Télécharger depuis un repository existant distant:

```
$ git clone my_url
```



# Les commandes de base Git

## Observer un repository

- Lister les nouveaux et les fichiers modifiés non encore « commités » :

```
$ git status
```

- Afficher l'historique des changements:

```
$ git log
```



# Les commandes de base Git

## Enregistrer un changement

- Ajouter un fichier dans la zone de transit (avant le commit):

```
$ git add [file]
```

- Ajouter tous les fichiers dans la zone de transit:

```
$ git add .
```

- Faire le commit des fichiers dans la zone de transit:

```
$ git commit -m "commit message"
```



# Les commandes de base Git

## Synchronisation avec le dépôt distant

- Chercher les dernières modification dans l'origine (origin) sans faire le merge:

```
$ git fetch
```

- Chercher les dernières modification dans l'origine (origin) et faire le merge:

```
$ git pull
```

- Envoyer les modifications faites en local vers l'origine:

```
$ git push
```





# Les commandes de base Git

## Travailler avec les branches

- Lister toutes les branches locales:

```
$ git branch
```

- Lister toutes les branches locales et distantes:

```
$ git branch -av
```

- Créer une nouvelle branche locale nommée my\_branch:

```
$ git branch my_branch
```





# Les commandes de base Git

## Travailler avec les branches

- Basculer dans la branche nommée my\_branch:

```
$ git checkout my_branch
```

- Merger la branche nommée branch\_a dans la branche nommée branch\_b:

```
$ git checkout branch_b
```

```
$ git merge branch_a
```

- Supprimer la branche nommée my\_branch:

```
$ git branch -d my_branch
```



# Les commandes de base Git

## En cas de besoin

- Utiliser le Help:

```
$ git command --help
```

- Visiter le site <https://training.github.com/>
- Utiliser Google.



Merci pour votre attention

