ENSEA

Beyond Engineering

OPTION MAKER

Git & Github

ENSEA

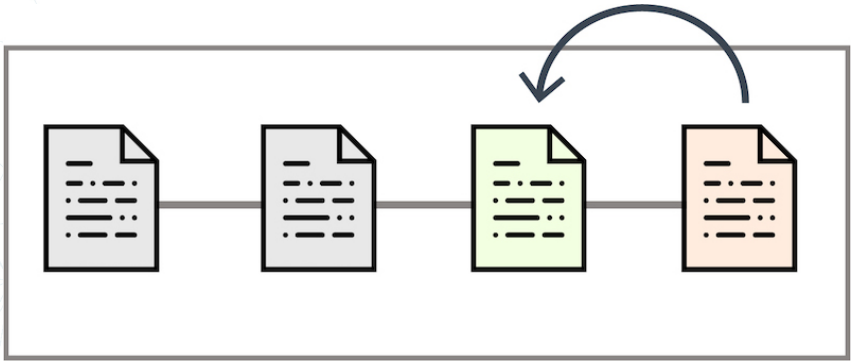
Nicolas Papazoglou nicolas.papazoglou@ensea.fr

15 janvier 2025

Versioning, pourquoi ?

ENSEA
Beyond Engineering

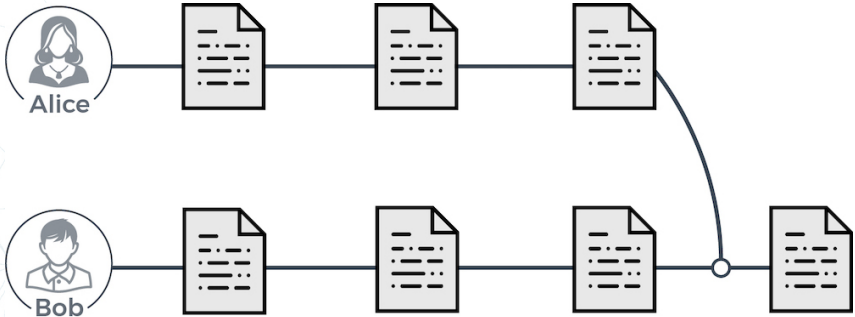
Le droit à l'erreur



Retour à une version précédente de vos codes.

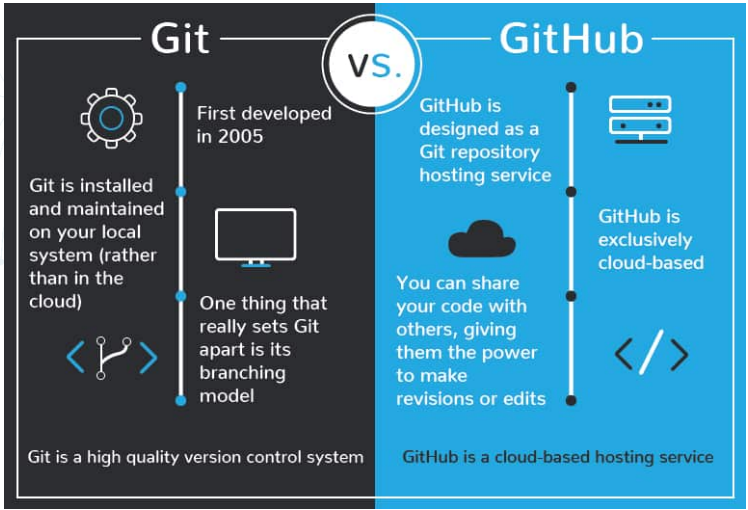
ENSEA
Beyond Engineering

Working together



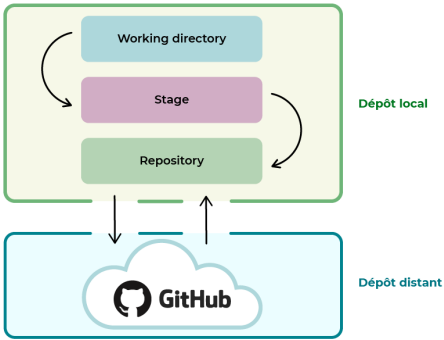
Travail sur le même projet sans géner le travail de l'autre.

Git & Github : differences



ENSEA
Beyond Engineering

Working flow



- Le répertoire de travail : Cette zone correspond au dossier du projet sur votre ordinateur.
- Stage ou Index : Cette zone est un intermédiaire entre le répertoire de travail et le dépôt. Il représente tous les fichiers modifiés que vous souhaitez voir apparaître dans votre prochaine version de code.
- Le Dépôt : Lorsque vous créez de nouvelles versions d'un projet, c'est dans cette zone qu'elles sont stockées.

ENSEA
Beyond Engineering

Git quick start guide

ENSEA
Beyond Engineering

Installer git

- Windows : https ://gitforwindows.org/
- Linux : sudo apt install git
- Mac os : brew install git

ENSEA
Beyond Engineering

Git Configuration

Name and email configuration :

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
git config --list
```

ENSEA
Beyond Engineering

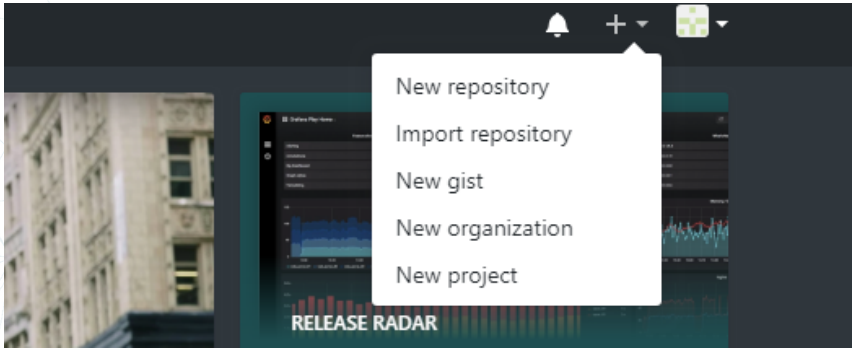
Github quick start guide

ENSEA
Beyond Engineering

Create account on github

ENSEA
Beyond Engineering

First repository



ENSEA
Beyond Engineering

Public ou Privé

Dépôt public :

- Accessible à tous sur Internet.
- Excellent choix pour les projets open source, car il encourage la collaboration et la contribution d'une communauté mondiale.
- Les référentiels publics permettent une plus grande visibilité, permettant aux collaborateurs potentiels de trouver plus facilement et de s'engager dans votre travail.

Dépôt privé

- Limité à un groupe sélectionné d'individus.
- Pour les projets qui impliquent du code propriétaire, des données confidentielles ou des éléments sensibles.
- Les référentiels privés offrent un environnement sécurisé pour la collaboration au sein d'une équipe de confiance tout en gardant le projet caché aux yeux du public.
- Nécessite un abonnement GitHub payant, sauf si vous êtes étudiant.

ENSEA
Beyond Engineering

Les fichiers importants

- readme.md : Indique les informations clés de votre projet : description, environnement à utiliser, dépendances possibles et copyright. Il est affiché directement sur la page web de github.
- gitignore : Il s'agit d'un fichier qui vous permet d'ignorer certains fichiers de votre projet Git, très utile lorsque vous ne voulez pas synchroniser des fichiers temporaires. Il existe des templates pour chaque type de projet en fonction des langages utilisés.

ENSEA
Beyond Engineering

Premier exemple

ENSEA
Beyond Engineering

Create local repository

```
johndoe@computer:~ cd Documents/PremierProjet
johndoe@computer:~/Documents/PremierProjet git init
Initialized empty Git repository in /home/johndoe/Documents/PremierProj
```

ENSEA
Beyond Engineering

Index your files

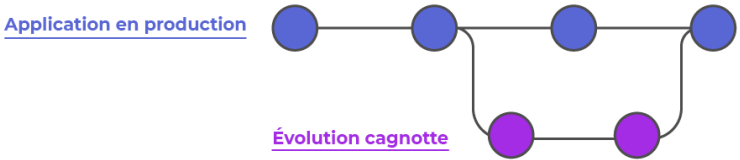
```
git add index.html styles.css
git commit -m "Ajout des fichiers html et css de base"
git remote add origin https://github.com/<user>/<repos>.git
git branch -M main
git push -u origin main
```

ENSEA
Beyond Engineering

Branch

ENSEA
Beyond Engineering

Branch



Créer une nouvelle branche

```
git branch
git branch myNewBranch
git checkout myNewBranch
git commit -m "some important comments"
```

Supprimer une branche

```
git branch -d myOldBranch
```

ENSEA
Beyond Engineering

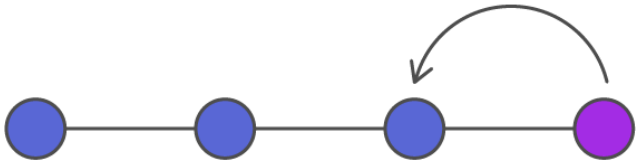
Merge

Fusionner myNewBranch into the main branch

```
git checkout main
git merge myNewBranch
```

ENSEA
Beyond Engineering

Reset



ENSEA
Beyond Engineering

Reset

Revenir à une version spécifique :

- Syntaxe : git reset [–hard] <commit>
- Retour à '3050fc0' tout en gardant les changements dans le répertoire de travail :

  `git reset 3050fc0`
- Retour à 'c0d30f3' et suppression des changements :

  `git reset --hard c0d30f3`

ENSEA
Beyond Engineering

Reset HEAD

Retour n version en arrière

- Syntaxe : git reset [–hard] HEAD <n>
- Retour de 5 commits en arrière tout en gardant les changements dans le répertoire de travail :

  ```
  git reset HEAD~5
  ```

- Retour de 3 commits en arrière et suppression des changements :

  ```
  git reset --hard HEAD~3
  ```

ENSEA
Beyond Engineering

Clone

- Cloner un répertoire existant :

  `git clone https://github.com/<user>/<repos>.git`
- Mettre à jour un repertoire déjà cloné :

  `git pull [origin main]`

ENSEA
Beyond Engineering

Ce que vous devez retenir : Git puis Github

Répertoire local créé en premier :

```
git init
git remote add origin https://github.com/<user>/<repos>.git

<<<coding>>>

git add .
git commit -m "important comment"
git push
```

ENSEA
Beyond Engineering

Ce que vous devez retenir : Github puis Git

Répertoire github créé en premier :

```
git clone https://github.com/<user>/<repos>.git

<<<coding>>>

git add .
git commit -m "important comment"
git push
```

ENSEA
Beyond Engineering

Pour aller plus loin : clé SSH

- Authentification par clé SSH :
  - Plus sécurisé,
  - Possible depuis un serveur distant,
  - Authentification par id/password non valible avec github,
  - Création obligatoire d'un token
- Procédure :
  - Génération de la clé ssh :

```
johndoe@computer:~ ssh-keygen
```

  - Récupération de la clé publique :

```
johndoe@computer:~ cat .ssh/id_rsa.pub
```

  - Copie de la clé publique sur github :
    - Settings
    - SSH and GPG keys
    - New SSH key
- Changement :
  - URL de la forme git@github.com :<user>/<repo>.git

ENSEA
Beyond Engineering