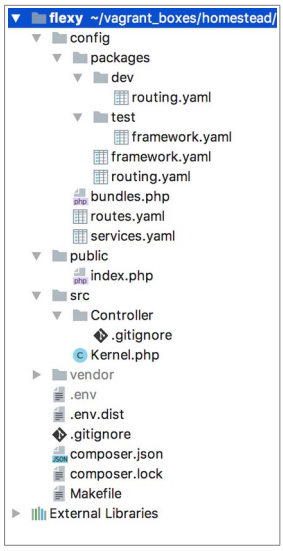
Séance 1 : GIT

Introduction



Problématique 1 : Comment travailler en FTP à plusieurs ? Risque de modifier les mêmes fichiers. Git détecte les différences avant de mettre à jour un fichier.

Problématique 2 : J'ai modifié une dizaine de fichiers dans un gros projet, comment mettre en ligne toutes les modifications sans rien oublier. Git détecte les changements et n'oublie aucun fichiers.

Problématique 3 : J'ai un projet stable et je veux le faire évoluer ; mais je risque de tout casser. Git enregistre l'historique des modifications dans chaque fichier et permet de revenir en arrière.

Programme pédagogique

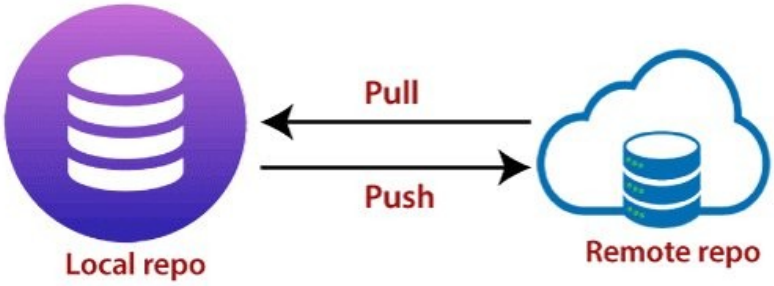
Travailler en local

Apprentissage des commandes de bases

À chaque fois que vous validez ou enregistrez l'état du projet dans Git, il prend un instantané du contenu de votre espace de travail.

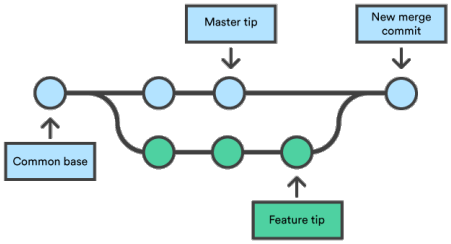
Travailler avec des dépots distants

Récupérer du code source et pousser des modifications



Travailler sur différentes branches

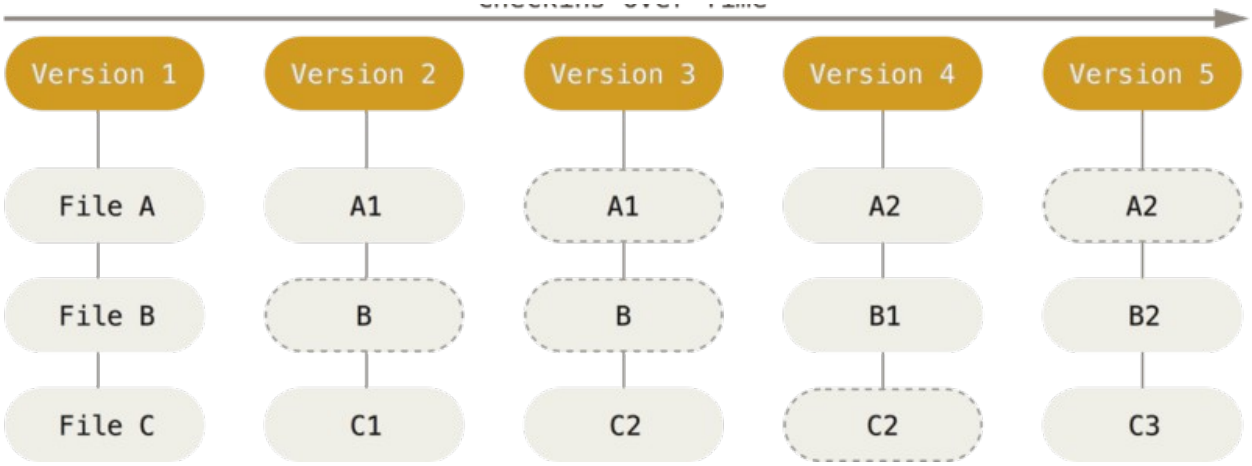
Git permet également de travailler sur des branches parallèles. Pratique pour faire évoluer son code.



Concepts

Historique des modifications

Avec git, vous pouvez enregistrer les modifications que vous faîtes de votre travail.



Comment sont enregistrées les modifications ?

Les modifications ne sont pas enregistrées au fur et à mesure que vous modifier vos fichiers. Les modifications sont faites lorsque vous le décidez. Lorsque vous souhaitez enregistrer vos modifications, il faut :

- d'abord **indexer (suivre)** les fichiers concernés
- ensuite, lorsque tous les fichiers ont été indexés, **valider**

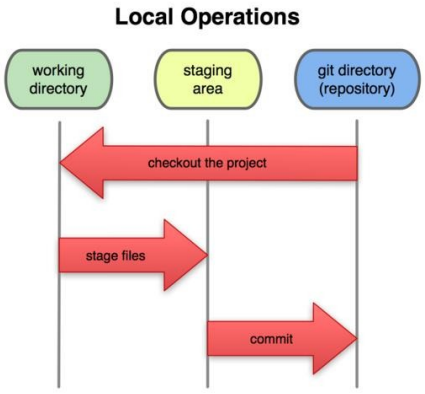
Par analogie, l'enregisrement de l'état de vos fichiers est comme une photo de famille : chacun prend place, et lorsque tout le monde est prêt : la photo est prise

Les 5 Zones

Par conséquent,

les fichiers « existent » dans 3 « zones » différentes :

- Le répertoire de travail : **Working directory**
- La zone d'index : **Staging Area**
- Le répertoire (dépôt) git : **Git Directory**



La première manipulation à connaître : placer les fichiers dans la zone d'index puis dans le répertoire git

A ces 3 zones, s'ajoute 2 autres :

- le(s) dépôts distants : **Remote repository**
- la remise : **Stash**

Installation

Télécharger le soft : https://gitforwindows.org/ et l'exécuter.

Premières commandes

Les commandes qui vont suivre doivent être saisies dans un terminal (Gitbash, vscode, oh-my-zsh…). Il faudra d'abord vous placer dans le répertoire de travail.

Créer un nouveau dossier et ouvrir un terminal

git config

```
git config --global user.name "Mon nom"
git config --global user.email "mon@email.fr"
```

git init

Cette commande doit être exécutée dans votre dossier de travail. Elle initialise git. Concrètement : elle crée un dossier .git qui va enregistrer l'historique.

git status

Cette commande permet d'afficher l'état de votre application :

- Sur quelle branche vous êtes (master par défaut)
- Si les fichiers sont « suivis », et s'ils sont modifiés

git add

Cette commande permet de placer les fichiers dans la zone « staging ».

```
git add index.html
git add *.html
git add .
git add –all
```

alternative :

```
git stage
```

Ouvrir votre dossier dans VSCode et créer 2 fichiers : index.php et styles.css. Lancer les commandes git status, git add . Et de nouveau git status.

git commit

Cette commande permet de déplacer les fichiers depuis la zone « stage » vers la zone de «depot». Il faut mettre un message de commit.

```
git commit -m « message de commit »
```

En retour, on peut voir les 7 premiers caractères de l'identifiant du commit (Ici : 5b445ea), ainsi que d'autres informations

Comitter vos modifications, puis ajouter un fichier script.js et commiter les nouvelles modifications.

git log

Cette commande permet de lister tous les commits enregistrés.

```
git log --oneline
git log -p index.php
```

git tag

Cette commande permet d'ajouter une étiquette pour marquer un commit

```
git tag v1.0
```

git diff

Cette commande permet de voir toutes les différences entre le répertoire de travail et le dépôt.

HEAD

HEAD représente l'endroit où l'on se trouve. En général, on se trouve au niveau du dernier commit.

Outils

Git dans VSCode

Télécharger l'extension git history

GitKraken

https://www.gitkraken.com/

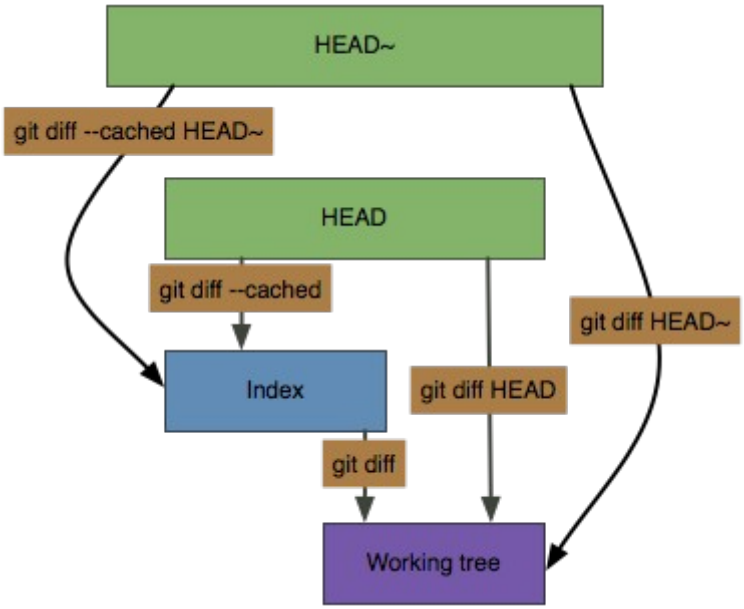
Git dans Sublime Text

https://www.sublimemerge.com/

Commandes avancées

Visualiser les différence entre 2 versions

```
git diff
```



Annuler un commit

Pour annuler le dernier commit :

```
git revert HEAD
```

Pour annuler un commit spécifique :

```
git revert 5b445ea
```

git revert créer un nouveau commit pour annuler un commit précédent

Annuler des changements dans l'index

Avec git reset

```
git reset HEAD
```

NB : git reset peut être utilisé dans certains cas pour supprimer complètement un commit.

Retirer des fichiers de l'index

git rm [--cached]

Cette commande permet de retirer des chier de l'index

```
git rm index.php
```

Annuler des changements dans le répertoire de travail

Avec git reset

```
git reset HEAD --hard
```

Avec git clean

```
git clean -f
```

Remiser des fichiers

Pour mettre de côté les modifications en cours dans le répertoire de travail :

```
git stash
```

Pour les récupérer :

```
git stash pop
```

Ignorer des dossiers ou des fichiers

Jusqu'à présent, nous avons vu que les fichiers de notre répertoire de travail sont de 2 types :

- **fichier non-suivi**
- **fichier suivi**

Nous allons voir maintenant un autre type de fichier :

- **fichier ignoré**

  On peut vouloir ignorer des fichiers (configs à la bdd) ou des dossiers (uploads de fichiers)
  Pour Ignorer des dossiers ou des fichiers, on crée un fichier .gitignore dans lequel on renseigne les chemins tous les fichiers qui ne seront pas suivi.
  Syntaxe : https://www.atlassian.com/fr/git/tutorials/saving-changes/gitignore
  NB : Commiter un dossier vide avec .gitkeep

Exercice

L'objectif est de créer un programme en php qui permettra d'uploader des photos de vacances. Le programme sera développé et testé sur votre serveur local. Les images uploadées ne devront pas être commitées. Seuls les fichiers php, html, css et js seront commités.
Dans un nouveau répertoire, iInitialiser git, créer un fichier .gitignore et renseigner le.

Pour aller plus loin

S'entraîner

http://gitimmersion.fr/

Documentation

https://git-scm.com/book/fr/v2

https://ndpsoftware.com/git-cheatsheet.html#loc=index

Apprendre

https://delicious-insights.com/fr/articles/apprendre-git/

https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github

https://grafikart.fr/formations/git

https://delicious-insights.com/fr/articles/git-stash/