

Présentation Git & Github

Eliesse HADJEM

8 janvier 2014

1 La gestion de version

- Concept
- Notions

2 Git

- Description
- Cycle
 - Create
 - Browse
 - Revert
 - Update
 - Branch
 - Commit
 - Publish

3 Github

- Descriptions
- Fonctionnalités

Une problématique

Comment permettre aux utilisateurs de partager l'information, tout en les empêchant de se marcher dessus ?

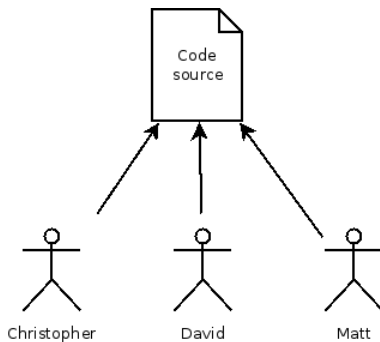


FIGURE: Différents Docteurs veulent accéder à la source

Une solution

La gestion de version (ou Version Control)

Définition

La gestion de version consiste à maintenir l'ensemble des version d'un ou plusieurs fichier.

FIGURE:

Une solution

La gestion de version (ou Version Control)

Définition

La gestion de version consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers.

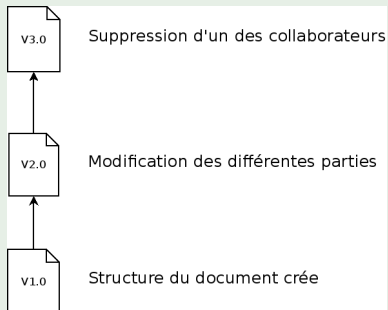


FIGURE: Différentes versions d'un fichier

Une solution

Le gestionnaire de version

- Un gestionnaire de version est un système qui enregistre l'évolution d'un fichier au cours du temps.
- Ce système permet de récupérer à tout moment une version antérieure du fichier.

Une solution

Le gestionnaire de version

- Un gestionnaire de version est un système qui enregistre l'évolution d'un fichier au cours du temps.
- Ce système permet de récupérer à tout moment une version antérieure du fichier.

Une solution

Exemple

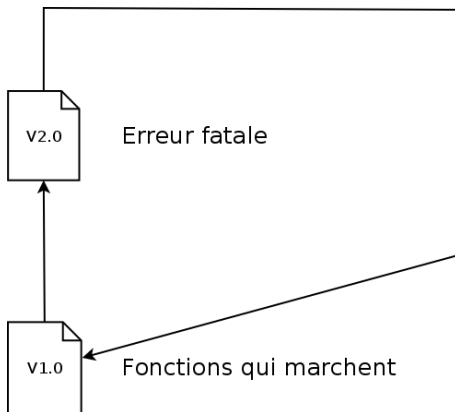


FIGURE: Récupération de la version 1.0 d'un fichier

Les types de gestionnaires de version

- Les gestionnaires centralisés
- Les gestionnaires décentralisés

Les types de gestionnaires de version

- Les gestionnaires centralisés : CVS, SVN
- Les gestionnaires décentralisés

Les types de gestionnaires de version

- Les gestionnaires centralisés
- Les gestionnaires décentralisés

Les types de gestionnaires de version

- Les gestionnaires centralisés
- Les gestionnaires décentralisés : GNU Arch, Mercurial, Git

Version

Définition

La version d'un fichier est l'avancement des modification d'un fichier qui a été validé par l'utilisateur.

Branche

Définition

Une branche est une version d'un projet que l'on souhaite continuer de manière indépendante



FIGURE: Un début de projet

Variables globales

La branche de développement par défaut

`master`

Le dépôt par défaut

`origin`

La branche courante

`HEAD`

Variables globales

La branche de développement par défaut

`master`

Le dépôt par défaut

`origin`

La branche courante

`HEAD`

Variables globales

La branche de développement par défaut

`master`

Le dépôt par défaut

`origin`

La branche courante

`HEAD`

Notation

Réprésentation de l'id d'un commit, d'une branche ou d'un tag

`$id`

Un fichier

`$file`

Une branche

`$branch`

Notation

Réprésentation de l'id d'un commit, d'une branche ou d'un tag

`$id`

Un fichier

`$file`

Une branche

`$branch`

Notation

Réprésentation de l'id d'un commit, d'une branche ou d'un tag

`$id`

Un fichier

`$file`

Une branche

`$branch`

Create

À partir de données existantes

```
cd ~/workspace/monprojet  
git init;  
git add.
```

À partir d'un dépôt existant

Le dépôt est sur la machine locale

```
git clone ~/existing/repo ~/new/repo
```

Le dépôt est sur une machine distante

```
git clone git://host.org/project.git  
git clone ssh://you@host.org/proj.git
```

Create

À partir de données existantes

```
cd ~/workspace/monprojet  
git init;  
git add.
```

À partir d'un dépôt existant

Le dépôt est sur la machine locale

```
git clone ~/existing/repo ~/new/repo
```

Le dépôt est sur une machine distante

```
git clone git://host.org/project.git  
git clone ssh://you@host.org/proj.git
```

Create

À partir de données existantes

```
cd ~/workspace/monprojet  
git init;  
git add.
```

À partir d'un dépôt existant

Le dépôt est sur la machine locale

```
git clone ~/existing/repo ~/new/repo
```

Le dépôt est sur une machine distante

```
git clone git://host.org/project.git  
git clone ssh://you@host.org/proj.git
```

Create

À partir de données existantes

```
cd ~/workspace/monprojet  
git init;  
git add.
```

À partir d'un dépôt existant

Le dépôt est sur la machine locale

```
git clone ~/existing/repo ~/new/repo
```

Le dépôt est sur une machine distante

```
git clone git://host.org/project.git  
git clone ssh://you@host.org/proj.git
```


Create

À partir de données existantes

```
cd ~/workspace/monprojet  
git init;  
git add.
```

À partir d'un dépôt existant

Le dépôt est sur la machine locale

```
git clone ~/existing/repo ~/new/repo
```

Le dépôt est sur une machine distante

```
git clone git://host.org/project.git  
git clone ssh://you@host.org/proj.git
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Browse

Connaitre les fichiers modifiés dans le dossier

```
git status
```

Connaitre les modifications sur les fichiers suivis

```
git diff
```

Connaitre l'historique des modifications

```
git log
```

Connaitre le détail d'un commit

```
git show $id
```

Connaitre toutes les branches locales

```
git branch
```

Revert

Repartir au dernier commit

```
git revert HEAD
```

Repartir à un commit donné

```
git revert $id
```

Retourner à un commit donné

```
git reset --hard
```


Revert

Repartir au dernier commit

```
git revert HEAD
```

Repartir à un commit donné

```
git revert $id
```

Retourner à un commit donné

```
git reset --hard
```

Revert

Repartir au dernier commit

```
git revert HEAD
```

Repartir à un commit donné

```
git revert $id
```

Retourner à un commit donné

```
git reset --hard
```

Revert

Repartir au dernier commit

```
git revert HEAD
```

Repartir à un commit donné

```
git revert $id
```

Retourner à un commit donné

```
git reset --hard
```

Revert

Repartir au dernier commit

```
git revert HEAD
```

Repartir à un commit donné

```
git revert $id
```

Retourner à un commit donné

```
git reset --hard
```

Un 'revert' peut être annulé. Un 'reset' NON !

Update

Connaitre les dernières modifications par rapport à origin

```
git fetch
```

Récupérer les dernières modifications par rapport à origin

```
git pull
```

Update

Connaitre les dernières modifications par rapport à origin

```
git fetch
```

Récupérer les dernières modifications par rapport à origin

```
git pull
```

Update

Connaitre les dernières modifications par rapport à origin

```
git fetch
```

Récupérer les dernières modifications par rapport à origin

```
git pull
```

Branch

Passer à une branche donnée

```
git checkout $id
```

Créer une nouvelle branche à partir de l'actuelle

```
git branch $branch
```

Créer une nouvelle branche à partir d'une branche donnée(\$other) et passer sur la nouvelle

```
git checkout -b $new_branch $other
```

Supprimer une branche

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


Branch

Passer à une branche donnée

```
git checkout $id
```

Créer une nouvelle branche à partir de l'actuelle

```
git branch $branch
```

Créer une nouvelle branche à partir d'une branche donnée(\$other) et passer sur la nouvelle

```
git checkout -b $new_branch $other
```

Supprimer une branche

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

Branch

Passer à une branche donnée

```
git checkout $id
```

Créer une nouvelle branche à partir de l'actuelle

```
git branch $branch
```

Créer une nouvelle branche à partir d'une branche donnée(\$other) et passer sur la nouvelle

```
git checkout -b $new_branch $other
```

Supprimer une branche

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

Branch

Passer à une branche donnée

```
git checkout $id
```

Créer une nouvelle branche à partir de l'actuelle

```
git branch $branch
```

Créer une nouvelle branche à partir d'une branche donnée(\$other) et passer sur la nouvelle

```
git checkout -b $new_branch $other
```

Supprimer une branche

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

Branch

Passer à une branche donnée

```
git checkout $id
```

Créer une nouvelle branche à partir de l'actuelle

```
git branch $branch
```

Créer une nouvelle branche à partir d'une branche donnée(\$other) et passer sur la nouvelle

```
git checkout -b $new_branch $other
```

Supprimer une branche

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

Commit

Valider des changements

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

L'option `-m` permet d'ajouter un message au commit

IMPORTANT

Les changements que l'on valide sont ceux que l'on a ajouté grâce à `git add $file`

On peut ajouter dans un commit tous les fichiers suivis avec l'option `-a`

Commit

Valider des changements

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

L'option `-m` permet d'ajouter un message au commit

IMPORTANT

Les changements que l'on valide sont ceux que l'on a ajouté grâce à `git add $file`

On peut ajouter dans un commit tous les fichiers suivis avec l'option `-a`

Commit

Valider des changements

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

L'option `-m` permet d'ajouter un message au commit

IMPORTANT

Les changements que l'on valide sont ceux que l'on a ajouté grâce à `git add $file`

On peut ajouter dans un commit tous les fichiers suivis avec l'option `-a`

Publish

Envoyer les changements sur le serveur

```
git push origin $branch
```


Publish

Envoyer les changements sur le serveur

```
git push origin $branch
```

Lol