

Présentation Git & Github

Eliesse HADJEM

8 janvier 2014

1 La gestion de version

- Concept
- Notions

2 Git

- Description
- Cycles
- Commandes

3 Github

- Présentation
- Fonctionnalités
- Les outils

Section 1

La gestion de version

Une problématique

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

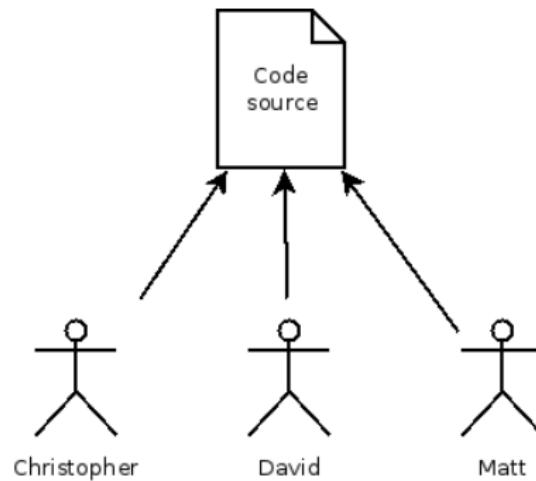


FIGURE: Différents Docteurs veulent accéder à la source

Une solution

La gestion de version (ou Version Control)

Définition

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

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.

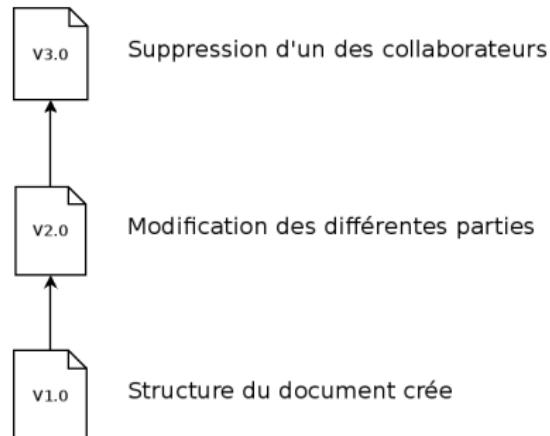


FIGURE: Différentes versions d'un fichier

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

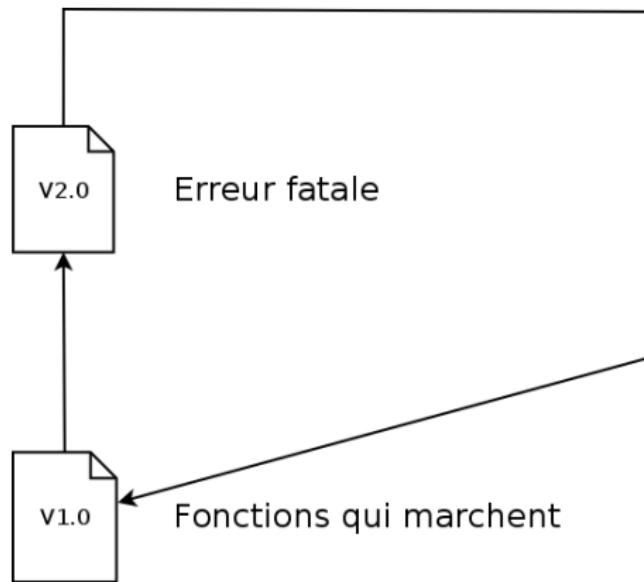


FIGURE: Récupération de la version 1.0 d'un fichier

Version

Définition

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

Jan 08, 2014	V1.0 hadeli authored 6 hours ago	64724e53e8 ↗ Browse code ↗
	Ajout de la structure des commandes hadeli authored 10 hours ago	950e271883 ↗ Browse code ↗
Jan 07, 2014	Ajout RevertFic hadeli authored a day ago	e997881fad ↗ Browse code ↗
	Ajout option overlay et src hadeli authored a day ago	6bfcc03723c ↗ Browse code ↗

FIGURE: Différentes versions d'un fichier

Branche

Définition

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



FIGURE: Un début de projet

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Avantages

Gestion de versions facilité

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Inconvénients

Travail hors connexion réseau

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Exemples

- CVS
- SVN
- ...

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Avantages

- Pas de dépendances à une seule machine
- Possibilité de travail hors connexion
- Possibilité de participer le projet sans demander les permissions aux responsables du projet
- Permet le travail en privé

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Inconvénients

- Clonage de dépôt plus long
- Pas de système de lock

Les types de gestionnaires de version

Les gestionnaires centralisés

Un seul dépôt de référence

Les gestionnaires décentralisés

Plusieurs dépôts de stockage

Exemples

- GNU Arch
- Mercurial
- Git

Section 2

Git

Généralités

- Système de gestion de version décentralisé
 - Logiciel libre
 - Crée par Linus Torvalds

Généralités

- Système de gestion de version décentralisé
- Logiciel libre
- Crée par Linus Torvalds

Généralités

- Système de gestion de version décentralisé
- Logiciel libre
- Crée par Linus Torvalds

Cycle de vie git

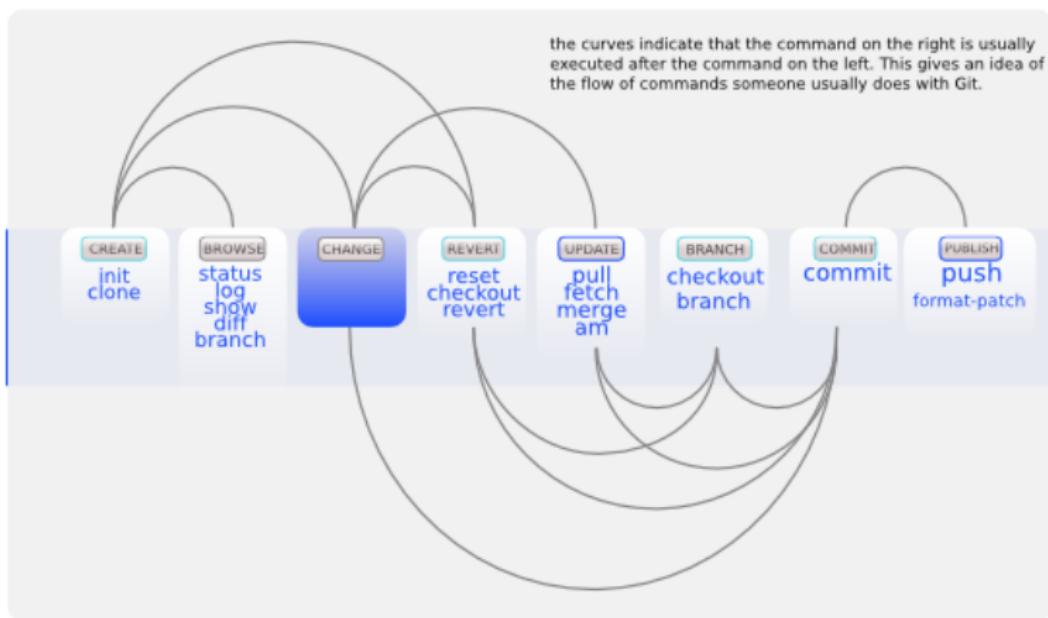


FIGURE: Cycle de vie Git

Cycle de vie d'un fichier

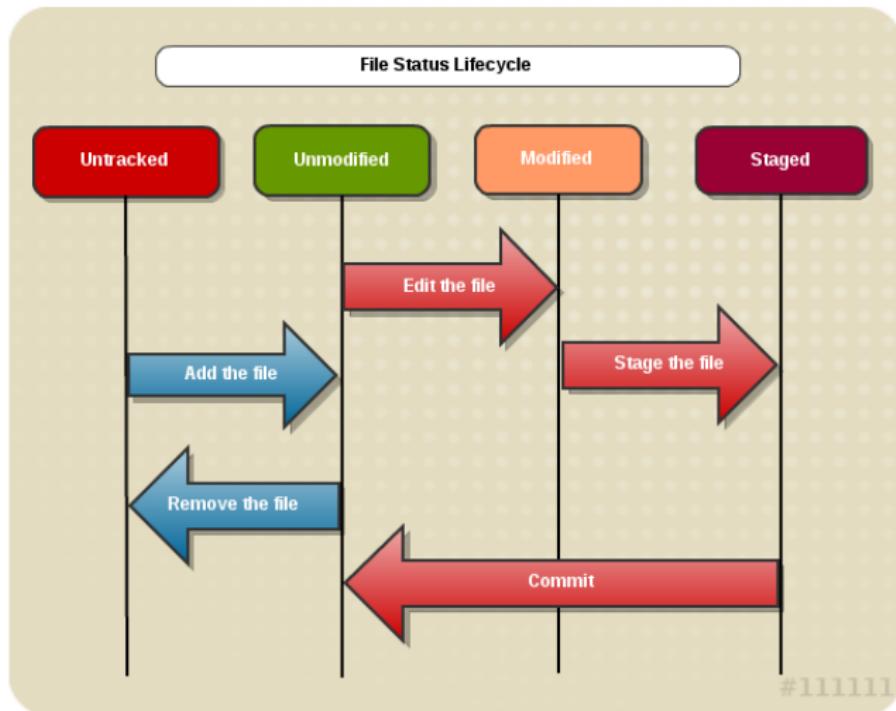


FIGURE: Cycle de vie Git d'un fichier

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`

Section 3

Github

Présentation

- Service web
 - Basé sur Git
 - Développé en Ruby on Rails & Erlang

Présentation

- Service web
- Basé sur Git
- Développé en Ruby on Rails & Erlang

Présentation

- Service web
- Basé sur Git
- Développé en Ruby on Rails & Erlang

Fonctionnalités

The screenshot shows a list of GitHub repositories. At the top, there are navigation links for Contributions, Repositories, Public Activity, and Edit Your Profile. Below that is a search bar with the placeholder 'Find a repository...', a 'Search' button, and a dropdown menu with options: All, Public, Private, Sources, Forks, Mirrors, and a green 'New' button. The main content area displays six repository cards:

- PresVCS**: Présentation des gestionnaires de version, plus particulièrement de git et github. Last updated 4 hours ago. Tags: TeX, 0 stars, 0 forks.
- WebMining**: Tp webmining Java. Last updated 2 days ago. Tags: Java, 0 stars, 0 forks.
- hadeli.github.io**: Last updated 4 days ago. Tags: 0 stars, 0 forks.
- BearDefender**: Projet "Algo & Java" portant sur la conquête du monde. Last updated 4 days ago. Tags: Java, 0 stars, 0 forks.
- BotTwitter**: Last updated 4 days ago. Tags: Java, 0 stars, 0 forks.
- FacebookBot**: Programme interagissant avec l'API Facebook. Ce programme permet diverses opérations avec un compte utilisateur. Last updated 4 days ago. Tags: Java, 0 stars, 0 forks.

FIGURE: Hébergements de projets

Fonctionnalités

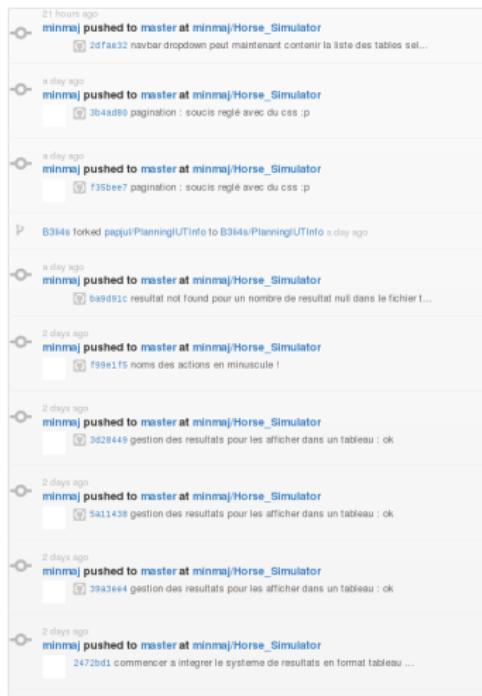


FIGURE: Suivi de personnes et de projets

Fonctionnalités



FIGURE: Graphes de réseau

Fonctionnalités

PUBLIC hadeli / **vmControl.sh** Created a month ago

Fonction shell pour permettre le lancement de toutes les Virtuals Machines liés à Virtual Box. Utilisez l'argument "start" pour les lancer. "stop" pour les éteindre

Gist Detail

Revisions 1

Download Gist

Clone this gist /hadeli/78660907

Embed this gist <script src="https://

Link to this gist https://gist.github.com/

vmControl.sh

Shell

```
1 function vmControl
2 {
3     vmsTemp=$(VBoxManage list vms | sed '$ s/ \\\\".*\\\\"|sed '$ s/\"/\\\"/|sed '$ s/\\\"/\"/')
4     for var in $vmsTemp
5     do
6         echo $var
7         if(test $1 = "start")
8             then
9                 echo Lancement de la virtual Box $var
10                vboxheadless -s $var
11            fi
12        if(test $1 = "stop")
13        then
14            echo 'Arret de la virtual Box '$var
15            VBoxManage controlvm $var poweroff;
16        fi
17    done
18 }
```

FIGURE: Système de pastebin

Fonctionnalités



FIGURE: Pages perso et Wiki disponible pour chaque dépôt

Windows

En ligne de commande

- *Git for Windows*
- *Tortoise Git*

Avec interface graphique

Windows

En ligne de commande

- *Git for Windows*
- *Tortoise Git*

Avec interface graphique

Windows

En ligne de commande

Avec interface graphique

Github for Windows

- Facile d'accès
- Possibilité d'écrire de longs messages de commit
- Affichage des diffs de manière ergonomique

Windows

En ligne de commande

Avec interface graphique

Github for Windows

- Facile d'accès
- Possibilité d'écrire de longs messages de commit
- Affichage des diffs de manière ergonomique

Windows

En ligne de commande

Avec interface graphique

Github for Windows

- Facile d'accès
- Possibilité d'écrire de longs messages de commit
- Affichage des diffs de manière ergonomique

Linux

En ligne de commande

Documentation Git

Avec interface graphique

Linux

En ligne de commande

Avec interface graphique

- *Git-cola*
- *Gitg*
- *SmartGit*
- *Git Eye*

Linux

En ligne de commande

Avec interface graphique

- *Git-cola*
- *Gitg*
- *SmartGit*
- *Git Eye*

Linux

En ligne de commande

Avec interface graphique

- *Git-cola*
- *Gitg*
- **SmartGit**
- *Git Eye*

Linux

En ligne de commande

Avec interface graphique

- *Git-cola*
- *Gitg*
- *SmartGit*
- ***Git Eye***

IDEs

La majorités des IDEs du marché ont une intégration de plusieurs VCS dont Git

IDEs

La majorités des IDEs du marché ont une intégration de plusieurs VCS dont Git

Exemples :

- PhpStorm, IntelliJ
- NetBeans
- Eclipse

Pour approfondir ...

Pour les commandes Git

[*Try Git*](#)

TP à faire

[*TP Git*](#)