

Travailler avec des branches

Si vous souhaitez ajouter un alias pour afficher les logs sur une seule ligne :

```
[localhost ~]$ git config --global alias.lgone 'log --oneline --decorate'
```

```
[localhost ~]$ git config --global core.pager 'less -FRX'
```

1. Utiliser Git pour gérer un site Web

Les sources HTML de votre site Web résident dans un référentiel Git sur votre poste de travail local. Le répertoire sites-prod servira de site actif (branche main) et le répertoire sites-dev servira pour les innovations en cours de développement sur le site.

```
[user@localhost ~]$ cd ~/LabsGit
```

```
[user@localhost LabsGit]$ mkdir -p website/{site-prod,site-dev}
```

```
[user@localhost LabsGit]$ cd website
```

```
[user@localhost ~/website]$ git init
Initialized empty Git repository in /home/user/website/.git/
```

```
[~/website]$ echo '<h1>Hello World!<h1\>' > site-prod/index.html
```

```
[~/website]$ touch site-dev/.gitkeep
```

```
[user@localhost ~/website]$ git add -A
```

```
[user@localhost ~/website]$ git status
# On branch main
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#       new file:   site-dev/.gitkeep
#       new file:   site-prod/index.html
```

```
[~/website]$ git commit -q -m "The beginnings of my web site."
```

```
[user@localhost ~/website] (main)$ git lgone
a05f4c3 (HEAD, main) The beginnings of my web site.
```

```
[user@localhost ~/website] (main)$ git branch
* main
```

Maintenant vous allez créer une nouvelle branche pour les nouvelles fonctionnalités telles qu'une page de contacts et une autre pour de l'aide.

```
[user@localhost ~/website] (main)$ git branch new-features

[user@localhost ~/website] (main)$ git branch
* main
  new-features

[user@localhost ~/website] (main)$ git lgone
a05f4c3 (HEAD, new-features, main) The beginnings of my web site.

[user@localhost ~/website] (main)$ git checkout new-features
Switched to branch 'new-features'

[user@localhost ~/website] (new-features)$ git branch
  main
* new-features

[user@localhost ~/website] (new-features)$ echo 'Contact :
elies.jebri@gmail.com' > site-dev/contact.html

[user@localhost ~/website] (new-features)$ git status
# On branch new-features
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       site-dev/contact.html
nothing added to commit but untracked files present (use "git add" to track)

[user@localhost ~/website] (new-features)$ git add -A

[user@localhost ~/website] (new-features)$ git status
# On branch new-features
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   site-dev/contact.html
#

[user@localhost ~/website] (new-features)$ git commit -m "Added contact.html
file"

[new-features 262a97c] Added contact.html file
1 file changed, 1 insertion(+)
create mode 100644 site-dev/contact.html

[user@localhost ~/website] (new-features)$ git lgone
262a97c (HEAD, new-features) Added contact.html file
a05f4c3 (main) The beginnings of my web site.

[user@localhost ~/website] (new-features)$ echo 'Help : God bless you' >
site-dev/help.html

[user@localhost ~/website] (new-features)$ git add -A
```

```
[user@localhost ~/website] (new-features)$ git status
# On branch new-features
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   site-dev/help.html
#

[user@localhost ~/website] (new-features)$ git commit -m "Added help.html file"
[new-features 01b3032] Added help.html file
1 file changed, 1 insertion(+)
create mode 100644 site-dev/help.html
```

```
[user@localhost ~/website] (new-features)$ git lgone
01b3032 (HEAD, new-features) Added help.html file
262a97c Added contact.html file
a05f4c3 (main) The beginnings of my web site.
```

```
[user@localhost ~/website] (new-features)$ git checkout main
Switched to branch 'main'
```

```
[user@localhost ~/website] (main)$ git lgone
a05f4c3 (HEAD, main) The beginnings of my web site.
```

Au retour vers la branche main vous voyez que les modifications effectuées sur la branche « new-features » n'y figurent pas.

Vous allez créer une autre branche relative à la partie dynamique du site :

```
[user@localhost ~/website] (main)$ git checkout -b php-features
Switched to a new branch 'php-features'
```

```
[user@localhost ~/website] (php-features)$ git branch
main
new-features
* php-features
```

```
[user@localhost ~/website] (php-feature)$ vim site-dev/index.php
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello World!";
?>
</body>
</html>
```

```
[user@localhost ~/website] (php-feature)$ git status
# On branch php-feature
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       site-dev/index.php
nothing added to commit but untracked files present (use "git add" to track)
```

```
[user@localhost ~/website] (php-feature)$ git add -A
```

```
[user@localhost ~/website] (php-feature)$ git commit -m "Added index.php"
```

```
[php-feature 9f0c86b] Added index.php
1 file changed, 9 insertions(+)
create mode 100644 site-dev/index.php
```

```
[user@localhost ~/website] (php-feature)$ git lgone
9f0c86b (HEAD, php-feature) Added index.php
a05f4c3 (main) The beginnings of my web site.
```

```
[user@localhost ~/website] (new-features)$ git checkout main
```

De retour vers la branche main vous allez effectuer quelques modifications à votre site avant de faire un commit.

```
[user@localhost ~/website] (main)$ vim README.md
My First HTML Site
```

```
[user@localhost ~/website] (main)$ vim site-prod/index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World! Site Title</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

```
[user@localhost ~/website] (main)$ git add -A
```

```
[user@localhost ~/website] (main)$ git status
# On branch main
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   README.md
#       modified:   site-prod/index.html
#
```

```
[user@localhost ~/website] (main)$ git commit -m "Added html skeleton to
index.html and README.md"
```

```
[user@localhost ~/website] (main)$ git lgone
9c3230d (HEAD, main) Added html skeleton to index.html and README.md
a05f4c3 The beginnings of my web site.
```

2. Fusionner les changements

A ce stade vous décidez de fusionner votre travail sur la branche « main » et celui de la branche des fonctionnalités « new-features ».

```
[user@localhost ~/website] (main)$ git merge new-features -m 'Adding Contact and Help files'
```

```
Merge made by the 'recursive' strategy.
 site-dev/contact.html | 1 +
 site-dev/help.html    | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 site-dev/contact.html
 create mode 100644 site-dev/help.html
```

```
[user@localhost ~/website] (main)$ git lgone
fc9cc52 (HEAD, main) Merge branch 'new-features' Adding Contact and Help files
9c3230d Added html skeleton to index.html and README.md
01b3032 (new-features) Added help.html file
262a97c Added contact.html file
a05f4c3 The beginnings of my web site.
```

```
[user@localhost ~/website] (main)$ git lgone --graph
*   fc9cc52 (HEAD, main) Merge branch 'new-features' Adding Contact and Help files
| \
| * 01b3032 (new-features) Added help.html file
| * 262a97c Added contact.html file
* | 9c3230d Added html skeleton to index.html and README.md
|/
* a05f4c3 The beginnings of my web site.
```

```
[user@localhost ~/website] (main)$ tree .
```

```
.
├── README.md
├── site-dev
│   ├── contact.html
│   └── help.html
└── site-prod
    └── index.html
```

```
2 directories, 4 files
```

Tous vos commits de la branche « new-features » se retrouvent maintenant dans « main », votre branche « new-features » ne servant plus à rien, vous pouvez la supprimer :

```
[user@localhost ~/website] (main)$ git branch -d new-features
Deleted branch new-features (was 01b3032).
```

```
[user@localhost ~/website] (main)$ git branch
* main
  php-features
```

3. Tagger une version

Il est possible de donner un alias à un commit précis pour le référencer sous ce nom.

```
[user@localhost ~/website] (main)$ git tag v0.1 fc9cc52

[user@localhost ~/website] (main)$ git lgone
fc9cc52 (HEAD, tag: v0.1, main) Merge branch 'new-features' Adding Contact
and Help files
9c3230d Added html skeleton to index.html and README.md
01b3032 Added help.html file
262a97c Added contact.html file
a05f4c3 The beginnings of my web site.
```

4. Conflits de fusions (Merge conflicts)

Quelques fois, le processus ci-dessus ne se déroule pas aussi bien. Si vous avez modifié différemment la même partie du même fichier dans les deux branches que vous souhaitez fusionner, Git ne sera pas capable de réaliser proprement la fusion.

La branche « php-features » n'étant pas encore fusionnée avec la branche « main » les développeurs travaillent encore dessus :

```
[user@localhost ~/website] (main)$ git log --oneline --decorate --graph --all
*   fc9cc52 (HEAD, tag: v0.1, main) Merge branch 'new-features' Adding
Contact and Help files
| \
| * 01b3032 Added help.html file
| * 262a97c Added contact.html file
* | 9c3230d Added html skeleton to index.html and README.md
| /
| * 9f0c86b (php-features) Added index.php
| /
* a05f4c3 The beginnings of my web site.
```

et décident de rajouter un fichier « README.md » à leur tour, sachant que ce fichier existe déjà dans la branche « main » mais il a été créé après la création de « php-features ».

```
[user@localhost ~/website] (main)$ git checkout php-features
Switched to branch 'php-features'

[user@localhost ~/website] (php-features)$ vim README.md
My First PHP Site

[user@localhost ~/website] (php-features)$ git add -A

[user@localhost ~/website] (php-features)$ git commit -m "Added README.md
file"
[php-features c70c214] Added README.md file
1 file changed, 1 insertion(+)
create mode 100644 README.md

[user@localhost ~/website] (php-features)$ git checkout main
Switched to branch 'main'
```

```
[user@localhost ~/website] (main)$ git log --oneline --decorate --graph --all
* c70c214 (php-features) Added README.md file
* 9f0c86b Added index.php
| * fc9cc52 (HEAD, tag: v0.1, main) Merge branch 'new-features' Adding
Contact and Help files
| |\
| | * 01b3032 Added help.html file
| | * 262a97c Added contact.html file
| | /
| /|
| * 9c3230d Added html skeleton to index.html and README.md
| /
* a05f4c3 The beginnings of my web site.
```

Essayez maintenant la fusion des deux branches :

```
[user@localhost ~/website] (main)$ git merge php-features
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

Git n'a pas automatiquement créé le *commit* de fusion. Il a arrêté le processus le temps que vous résolviez le conflit. Si vous voulez vérifier, à tout moment après l'apparition du conflit, quels fichiers n'ont pas été fusionnés, vous pouvez lancer la commande `git status`. Tout ce qui comporte des conflits et n'a pas été résolu est listé comme *unmerged*.

```
[user@localhost ~/website] (main)$ git status
# On branch main
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Changes to be committed:
#
#       new file:   site-dev/index.php
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#       both added:      README.md
#
```

Git ajoute des marques de résolution de conflit standards dans les fichiers qui comportent des conflits, pour que vous puissiez les ouvrir et résoudre les conflits manuellement. Votre fichier contient des sections qui ressemblent à ceci :

```
[user@localhost ~/website] (main)$ vim README.md
<<<<<< HEAD
My First HTML Site
=====
My First PHP Site
>>>>>> php-features
```

Effectuez la resolution du conflit de fusion manuellement comme suit :

```
[user@localhost ~/website] (main)$ vim README.md
My First HTML-PHP Site
```

```
[user@localhost ~/website] (main)$ git status
# On branch main
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Changes to be committed:
#
#       new file:   site-dev/index.php
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#       both added:    README.md
#
```

Après avoir résolu un conflit, lancez git add sur chaque fichier pour le marquer comme résolu. Placer le fichier dans l'index (Stage) marque le conflit comme résolu pour Git.

```
[user@localhost ~/website] (main)$ git add -A
```

```
[user@localhost ~/website] (main)$ git status
# On branch main
# All conflicts fixed but you are still merging.
#   (use "git commit" to conclude merge)
#
# Changes to be committed:
#
#       modified:   README.md
#       new file:   site-dev/index.php
#
```

```
[user@localhost ~/website] (main)$ git commit
[main 21cca39] Merge branch 'php-features'
```



```
[user@localhost ~/website] (main)$ git lgone --graph
* 21cca39 (HEAD, main) Merge branch 'php-features'
|\
| * c70c214 (php-features) Added README.md file
| * 9f0c86b Added index.php
* | fc9cc52 (tag: v0.1) Merge branch 'new-features' Adding Contact and Help files
|\ \
| * | 01b3032 Added help.html file
| * | 262a97c Added contact.html file
| | /
* | 9c3230d Added html skeleton to index.html and README.md
| /
* a05f4c3 The beginnings of my web site.
```

```
[user@localhost ~/website] (main)$ tree
.
├── README.md
├── site-dev
│   ├── contact.html
│   ├── help.html
│   └── index.php
└── site-prod
    └── index.html

2 directories, 5 files
```