
Formation Git - Maîtriser git en profondeur



Git, c'est ton super pouvoir.

Bien plus qu'un simple outil, Git est la fondation de tous tes projets.

Cette formation te plonge sous le capot pour comprendre vraiment comment Git fonctionne, éviter les pièges classiques et manipuler ton historique avec confiance.

Résultat : plus de sérénité, plus de maîtrise, et des projets qui avancent sans friction.

Durée: 2 jours (14 heures)

Niveau: Intermédiaire / Confirmé

Nature: Inter-entreprise et
Intra-entreprise

Objectifs pédagogiques:

À l'issue de la formation, tu sauras :

- **Comprendre** le fonctionnement interne de Git (objets, références, stockage)
- **Créer, gérer et manipuler** des branches en toute confiance
- **Configurer Git** proprement pour tes projets et ton environnement
- **Fusionner intelligemment** avec les bonnes stratégies (merge, rebase)
- **Lire et personnaliser** ton historique de commits comme un pro
- **Filtrer et explorer** l'historique efficacement pour retrouver n'importe quoi
- **Réécrire l'historique** sans risquer de tout casser
- **Annuler et restaurer** ton travail pour ne plus jamais avoir peur de te tromper
- **Collaborer efficacement** avec d'autres développeurs via pull, fetch, rebase, push
- **Gérer proprement les conflits et les synchronisations** entre plusieurs branches distantes

Public

Développeurs, Lead devs, ou Product Managers motivés par la technique

Pré-requis

- Tu as déjà utilisé Git en environnement pro (même un peu)
 - Tu sais taper des commandes dans un terminal
 - Tu as git d'installé sur ta machine
-

Programme de la formation

Git sous le capot

- Objets internes : blob, tree, commit, tag – et leur SHA-1
- Les secrets du dossier .git/
- Références : branches, tags, HEAD, références spéciales
- Pourquoi Git est **rapide, fiable et puissant** (et comment ça marche)

Branches & navigation

- Créer, renommer, supprimer une branche **sans crainte**
- Navigation dans l'historique : git switch, HEAD~, HEAD^, etc.
- Fusionner : commit de merge, --ff, --no-ff... toutes les subtilités
- Rebase simple ou interactif (squash, reorder, drop, fixup, etc.)
- Merge ou rebase ? **Choisir intelligemment selon le workflow**
- Conflits : pourquoi ils arrivent, et **comment bien les résoudre**

Analyser l'historique comme un pro

- Le cycle de vie d'un fichier : de `untracked` à `committed`
- `log`, `diff`, `blame`, `bisect`, `-G`, `-S` : **les bons outils au bon moment**
- Recherches fines, formats de logs personnalisés
- **Bonus** : trouver un commit fautif avec `git bisect`

Corriger ses erreurs & réécrire l'histoire

- Tout annuler : `commit(s)`, `merge`, `rebase...` sans peur
- `reset` : `--soft`, `--mixed`, `--hard` – **savoir quoi utiliser quand**
- `Reflog`, `HEAD@[1]` : retrouver ce que tu croyais perdu
- Modifier, fusionner, rejouer des commits : `revert`, `amend`, `cherry-pick...`
- Sauver une branche après un `reset` foireux ou un `rebase` raté

Git en équipe

- Que se passe-t-il vraiment quand tu `pull` ou `push` ?
- Gérer les remotes : cloner, ajouter, renommer, supprimer
- Branches distantes : création, suppression, gestion
- Travail collaboratif : `pull`, `fetch`, `merge`, `rebase...` **en sécurité**
- `--force-with-lease` : **forcer proprement**
- Ne plus paniquer quand un collègue `rebase` la branche distante

Stratégies de branches & workflows

- Pourquoi une stratégie ? **Sécurité, clarté, fluidité**
- **Git Flow** : robuste (`release`, `hotfix...`), parfois trop lourd
- **GitHub Flow** : simple et moderne (`feature branches` + `PR`)
- **GitLab Flow** : orienté environnement (`dev`, `staging`, `prod`)
- **Trunk Based Development** : une seule branche `main` + `feature flags`
- Comment choisir ? **Dépend de ton équipe, ton niveau, ton CI/CD**

Bonus :

- Arbres de commits illustrés
- Tableaux comparatifs
- Checklists : “Est-ce que Git Flow est fait pour moi ?”

Outils & productivité

- Améliore ton confort en ligne de commande : delta, git-extras, plugins Zsh...
- Découvre des outils visuels puissants : GitKraken, GitLens, SourceTree, Tower...
- Alias pratiques, configs utiles, tips de productivité : tout pour aller plus vite