



GUIDE PRATIQUE

Git au quotidien : le guide de référence pratique

Configuration, branches, rebase, résolution de conflits, récupération d'erreurs. Les commandes et workflows utilisés en équipe, expliqués clairement.



00 Introduction

Pourquoi ce guide, comment l'utiliser, ce que tu vas apprendre

01 Configurer Git comme un pro

gitconfig, alias, SSH, identité, outils

02 Le workflow quotidien

add, commit, push, pull, status, diff — les vrais réflexes

03 Branches : créer, naviguer, merger

Branching model, feature branches, merge vs rebase

04 Résoudre les conflits sans paniquer

Comprendre un conflit, le résoudre, les outils visuels

05 Réécrire l'historique proprement

amend, rebase interactif, squash, fixup

06 Stash, cherry-pick et bisect

Stash et cherry-pick au quotidien, bisect pour chasser les régressions

07 Git log et la recherche dans l'historique

log, blame, show, reflog — retrouver n'importe quoi

08 Pull requests et code review

Workflow PR, conventions, revue de code efficace

09 Les erreurs courantes et comment les corriger

reset, revert, recover — le filet de sécurité Git

10 .gitignore et hygiène de dépôt

Fichiers à ignorer, nettoyage, bonnes pratiques

– Conclusion & cheat sheet

Récapitulatif des commandes essentielles

Git, c'est le socle invisible de tout projet pro

Tu connais git add et git commit. Mais en entreprise, ça ne suffit pas. Ce guide te donne les vrais réflexes.

Chaque développeur professionnel utilise Git quotidiennement. Pourtant, la majorité des tutoriels s'arrêtent aux bases : init, add, commit, push. En situation réelle, tu dois gérer des branches, résoudre des conflits, réécrire un historique, retrouver un bug dans 2000 commits.

Ce guide couvre **les commandes et workflows que les développeurs expérimentés utilisent au quotidien**, celles qu'on apprend généralement sur le tas, en production, parfois dans la douleur.

Pas de théorie abstraite sur les DAG ou les objets internes de Git. Juste ce qui te sert quand tu codes, quand tu collabores, et quand quelque chose tourne mal.

Ce que tu vas apprendre

- Configurer Git avec les bons alias et les bonnes options pour gagner du temps chaque jour
- Maîtriser le workflow branche / commit / PR utilisé dans toutes les équipes pro
- Résoudre les conflits de merge calmement et efficacement
- Réécrire l'historique pour des commits propres et lisibles
- Utiliser stash, cherry-pick et bisect dans les situations qui les nécessitent
- Retrouver n'importe quel changement dans l'historique avec log, blame et reflog
- Corriger les erreurs courantes sans perdre de travail

PRÉ-REQUIS

Tu as déjà installé Git et tu sais faire un `git init`, `git add`, `git commit`. Si ce n'est pas le cas, commence par les bases avant de lire ce guide.