

Git Cheatsheet: die wichtigsten Befehle

PRAXISLEITFADEN | Stand: 27. Oktober 2026 | Kostenlos

1. Repository prüfen

Vorgehen: Zeige Status, aktuellen Branch und jüngste Commits, bevor du Änderungen vornimmst.

Beispiel: git status

git branch --show-current

git log --oneline -5

Prüfkriterium: Du kennst den Arbeitsstand und uncommittete Dateien.

Vermeiden: Blind reset oder checkout ausführen.

2. Änderungen gezielt committen

Vorgehen: Prüfe den Diff und stage nur gewünschte Dateien.

Beispiel: git diff

git add src/app.php

git diff --cached

git commit -m "Fix navigation"

Prüfkriterium: Commit enthält ausschließlich beabsichtigte Änderungen.

Vermeiden: Secrets oder Build-Artefakte versehentlich committen.

3. Branch sicher aktualisieren

Vorgehen: Hole Remote-Änderungen und prüfe vor Merge oder Rebase Konflikte.

Beispiel: git fetch origin

git switch feature/wissen

git merge origin/main

Prüfkriterium: Konflikte sind aufgelöst und Tests laufen.

Vermeiden: Ungeprüft force-push auf gemeinsam genutzte Branches.

4. Änderungen rückgängig machen

Vorgehen: Unterscheide Arbeitsverzeichnis, Staging und veröffentlichte Commits.

Beispiel: git restore --staged datei.php

git restore datei.php

git revert <commit>

Prüfkriterium: Der gewünschte Zustand ist wiederhergestellt, Teamhistorie bleibt konsistent.

Vermeiden: git reset --hard ohne Backup einsetzen.

5. Fehler finden

Vorgehen: Nutze Log, Diff und Reflog zur Diagnose verlorener Referenzen.

Beispiel: git reflog
git show <commit>

Prüfkriterium: Gesuchter Commit ist identifiziert, bevor repariert wird.

Vermeiden: Reflog als unbegrenztes Backup missverstehen.

Fachquellen

Git Reference – <https://git-scm.com/docs>

Onlineversion: <https://www.cyberknecht.de/wissen/cheatsheets/git-cheatsheet.html>