

EINRICHTEN

<code>git config --global user.name "Name"</code>	Name für Commits setzen
<code>git config --global user.email "mail"</code>	E-Mail für Commits setzen
<code>git init</code>	neues Repository im Ordner anlegen
<code>git clone URL</code>	Repository von einem Server kopieren

TÄGLICHER ABLAUF

<code>git status</code>	Zustand anzeigen (geändert, vorgemerkt)
<code>git add DATEI</code>	Änderung für den nächsten Commit vormerken
<code>git add .</code>	alle Änderungen im Ordner vormerken
<code>git commit -m "Text"</code>	vorgemerkte Änderungen speichern
<code>git diff</code>	Änderungen ansehen (noch nicht vorgemerkt)
<code>git diff --staged</code>	vorgemerkte Änderungen ansehen
<code>git log --oneline --graph</code>	Verlauf kompakt mit Verzweigungen

BRANCHES

<code>git branch</code>	Branches auflisten
<code>git switch -c name</code>	neuen Branch anlegen und wechseln
<code>git switch name</code>	zu einem Branch wechseln
<code>git merge name</code>	Branch in den aktuellen übernehmen
<code>git branch -d name</code>	gemergten Branch löschen

MIT DEM SERVER

<code>git remote -v</code>	verbundene Server anzeigen
<code>git fetch</code>	Neues holen, ohne zu mischen
<code>git pull</code>	Neues holen und übernehmen
<code>git push</code>	Commits hochladen
<code>git push -u origin name</code>	Branch erstmals hochladen und verknüpfen

RÜCKGÄNGIG MACHEN

<code>git restore DATEI</code>	Änderung in der Datei verwerfen
<code>git restore --staged DATEI</code>	Vormerkung zurücknehmen
<code>git revert HASH</code>	Commit durch einen Gegen-Commit aufheben
<code>git reset --soft HEAD~1</code>	letzten Commit zurücknehmen, Änderungen bleiben
<code>git stash / git stash pop</code>	Änderungen zwischenlagern / zurückholen

Vorsicht: `git reset --hard` und `git restore` verwerfen Änderungen unwiderruflich. Dateien, die nicht ins Repository sollen (Passwörter, Zugangsdaten, `node_modules`), gehören in die Datei `.gitignore`.