

Neu: switch und restore

<code>switch main</code>	zum Branch wechseln
<code>switch -c neu</code>	Branch anlegen und wechseln create
<code>switch -</code>	zurück zum vorigen Branch <i>wie cd -</i>
<code>switch --detach a1b2c3</code>	Commit ansehen, ohne Branch
<code>restore datei</code>	Änderung verwerfen
<code>restore --staged datei</code>	aus dem Index nehmen
<code>restore -s HEAD~1 datei</code>	Stand vor dem letzten Commit source

Früher erledigte `checkout` beides. `switch` fasst nur Branches an, `restore` nur Dateien.

Alt gegen neu

<code>checkout main</code>	heute <code>switch main</code>
<code>checkout -b neu</code>	heute <code>switch -c neu</code>
<code>checkout -- datei</code>	heute <code>restore datei</code>
<code>reset HEAD datei</code>	heute <code>restore --staged datei</code>

Alltag

<code>status -sb</code>	Stand kurz, mit Branch short , branch
<code>add -p</code>	Änderungen stückweise vormerken patch
<code>commit -m "Text"</code>	Commit anlegen message
<code>commit --amend --no-edit</code>	letzten Commit ergänzen
<code>diff</code>	was noch nicht vorgemerkt ist
<code>diff --staged</code>	was in den Commit geht
<code>log --oneline --graph --all</code>	Verlauf als Baum

Branches

<code>branch</code>	lokale Branches
<code>branch --show-current</code>	aktueller Branch
<code>branch -m neuer-name</code>	Branch umbenennen move
<code>branch -d alt</code>	gemergten Branch löschen, <code>-D</code> erzwingt delete
<code>push -u origin HEAD</code>	Branch erstmals pushen upstream

Abgleich mit dem Server

<code>fetch --prune</code>	holen, gelöschte Remote-Banches entfernen
<code>pull --rebase</code>	holen, eigene Commits oben drauf
<code>push --force-with-lease</code>	überschreiben, nur wenn niemand dazwischen gepusht hat
<code>remote -v</code>	Server-Adressen anzeigen verbose

Nie `--force` ohne `-with-lease`: das überschreibt auch fremde Commits.

Zwischenablage

<code>stash push -m "Text"</code>	Änderungen beiseitelegen
<code>stash list</code>	beiseitegelegtes anzeigen
<code>stash pop</code>	zurückholen und entfernen

Rückgängig

<code>reset --soft HEAD~1</code>	letzten Commit auflösen, Änderungen bleiben
<code>revert a1b2c3</code>	Gegen-Commit, sicher bei gepushten Commits
<code>reflog</code>	wo HEAD überall war, rettet fast alles
<code>clean -n</code>	zeigt, welche neuen Dateien <code>-f</code> löscht <code>-n</code> = nur zeigen

Commits aufräumen

<code>commit --fixup a1b2c3</code>	Korrektur für einen älteren Commit
<code>rebase -i --autosquash main</code>	Korrekturen einsortieren interaktiv
<code>rebase --abort</code>	Rebase abbrechen

Mehrere Arbeitskopien

<code>worktree add ../fix fix</code>	Branch in eigenem Ordner
<code>worktree list</code>	alle Arbeitskopien
<code>worktree remove ../fix</code>	Ordner wieder entfernen

Praktisch für einen schnellen Fix, ohne die laufende Arbeit zu stashen.

Einstellungen, einmalig

<code>push.autoSetupRemote true</code>	<code>push</code> ohne <code>-u</code> beim ersten Mal
<code>pull.rebase true</code>	<code>pull</code> rebased immer
<code>fetch.prune true</code>	<code>fetch</code> räumt immer auf
<code>rerere.enabled true</code>	gelöste Konflikte wiederverwenden reuse recorded resolution
<code>init.defaultBranch main</code>	neue Repos starten mit main
Setzen mit <code>config --global schlüssel wert</code>	