

Git Version Control Associate

Prüfungscode IFW-GIT-200-01 · Fassung 1.0 · Stand 18.09.2026

Versionsverwaltung mit Git: Commits und Historie, Branches und Merges, Zusammenarbeit über Remotes und das Zurücknehmen von Änderungen.

Die Prüfung im Überblick

Bearbeitungszeit 1 Std. 30 Min.	Fragen 70
Bestehensgrenze 70 %	Fragetypen Einfach- und Mehrfachauswahl
Prüfungssprache Deutsch	Zertifikat gültig Unbegrenzt
Durchführung Online, zeitlich überwacht	Preis 49,00 €

Themenblöcke

Block	Thema	Gewichtung	Fragen
1	Grundlagen, Commits und Historie	33 %	23
2	Branches und Zusammenführen	27 %	19
3	Remotes und Zusammenarbeit	20 %	14
4	Änderungen zurücknehmen	20 %	14
Gesamt		100 %	70

Die Gewichtung gibt an, welchen Anteil ein Block an den Prüfungsfragen hat. Die Fragen werden beim Start zufällig und anteilig je Block aus dem Fragenpool gezogen.

Grundlagen, Commits und Historie

Die drei Bereiche eines Repositorys und der Umgang mit der Historie.

- 1.1** Arbeitsverzeichnis, Index und Repository unterscheiden
`git add, git commit, git commit -am`
- 1.2** Commits nachträglich ergänzen
`git commit --amend` vor dem Push
- 1.3** Dateien ignorieren
.gitignore, Muster und Ausnahmen mit !
- 1.4** Historie lesen
`git log --oneline`, Bereiche wie `main`, `feature`, `HEAD~n`
- 1.5** Unterschiede anzeigen
`git diff, --staged`, Vergleich zweier Commits
- 1.6** Stände markieren und ansteuern
annotierte Tags mit `-a`, Detached HEAD

Branches und Zusammenführen

Parallel arbeiten und Stände wieder zusammenbringen.

- 2.1** Branches anlegen und wechseln
`git switch -c, git checkout -b`
- 2.2** Stände zusammenführen
Fast-Forward und Merge-Commit
- 2.3** Konflikte auflösen
Konfliktmarkierungen, `git add`, Merge abschließen
- 2.4** Merge und Rebase abwägen
lineare Historie gegenüber unveränderter Historie geteilter Branches
- 2.5** Branches löschen
`-d` gegenüber `-D` und die Meldung „not fully merged“

Remotes und Zusammenarbeit

Mit entfernten Repositorys arbeiten und Änderungen im Team integrieren.

- 3.1** Mit Remotes arbeiten
`git fetch`, Remote-Tracking-Branches wie `origin/main`

- 3.2** Änderungen veröffentlichen
`git push`, abgelehnte Pushes, `--force-with-lease`
- 3.3** Fremde Änderungen integrieren
`git pull` und `git pull --rebase`
- 3.4** Über Pull Requests zusammenarbeiten
Feature-Branches, Review, Konflikte im eigenen Branch lösen

BLOCK 4

20 % · 14 Fragen

Änderungen zurücknehmen

Fehler korrigieren – je nachdem, ob die Commits schon geteilt wurden.

- 4.1** Branch-Zeiger zurücksetzen
`git reset` mit `--soft`, `--mixed` und `--hard`
- 4.2** Dateien wiederherstellen
`git restore` und `git restore --staged`
- 4.3** Commits rückgängig machen
`git revert` als Gegen-Commit für geteilte Branches
- 4.4** Änderungen beiseitelegen
`git stash`, `-u` für unversionierte Dateien

Institut für Webentwicklung · unabhängiges Lernangebot ohne Verbindung zu den genannten Technologieanbietern. Maßgeblich ist die Fassung auf der Zertifizierungsseite.