WELT 11: KOMMANDOZENTRALE · MISSION 11.4

Git für Nicht-Nerds: Die Datei von gestern ist weg

7:32 Uhr, die KI hat Ihr Arbeitsblatt «verbessert» und das Original ist weg. Mit sechs Git-Befehlen holen Sie es zurück und synchronisieren künftig Schule und Zuhause.

Zur Mission

DAUER 60 Min. KOSTEN Gratis WERKZEUGE Git, Terminal, VS Code, ChatGPT DATENAMPEL Grün – unbedenklich

Das Wichtigste

- Git = Zeitmaschine für Ordner: Jede gesicherte Version bleibt abrufbar.
- Die 6 Befehle: git status · git add . · git commit -m "Text" · git log --oneline · git restore datei · git pull / git push.
- Kern: einpacken (add), Foto machen (commit). Ungesichertes kann Git nicht retten.
- Zurückholen: git restore datei (letzter Stand) oder git restore --source=KENNUNG datei (ältere Version).
- Sync zwischen Geräten: Pull am Anfang, Push am Ende.
- GitHub: Gratiskonto mit privaten Repositories. Codeberg: gemeinnützig, für offen lizenziertes Material.
- Nie ins Repository: Namen, Noten, Absenzen, Zugangsdaten – auch nicht privat. .gitignore zuerst anlegen.
- Agenten nur in Git-Ordnern arbeiten lassen: vorher committen, danach git status (auch neue Dateien!) und git diff – Änderungen an gesicherten Dateien sind rücknehmbar.

Ihr Produkt

Ihr **Projektordner mit Zeitmaschine** (Git, inklusive .gitignore für Vertrauliches), der **Git-Spickzettel** mit den 6 wichtigsten Befehlen zum Ausdrucken und – wer mag – ein **privates Online-Repository** für den Sync zwischen Schule und Zuhause.

Abnahme: Sie haben das überschriebene Arbeitsblatt mit git restore in unter drei Minuten zurückgeholt, eine ältere Version mit git log gefunden und wiederhergestellt, Ihr Projektordner hat eine .gitignore, die vertrauliche Ordner ausschliesst, und Sie haben den Sync mit pull und push ausprobiert – oder können ihn ohne Online-Kopie in drei Sätzen erklären.

So gehen Sie vor

1. Hat Ihr Gerät schon eine Zeitmaschine?
2. Zeitmaschine einschalten: das erste Foto
3. Die Katastrophe – Stoppuhr: 3 Minuten
4. Zeitreise: zurück zu Version 1
5. Sync: Schule und Zuhause
6. Der Datenschutz-Riegel: .gitignore

Start-Prompt

Ich bin Lehrperson an einer Schweizer Berufsfachschule und arbeite neu mit Git für mein Unterrichtsmaterial.
Erstelle mir 5 Regeln für gute Commit-Beschreibungen (git commit -m "...") mit je einem schlechten und einem guten Beispiel aus dem Schulalltag (Arbeitsblätter, Quiz, Bewertungsraster).
Sag mir auch, wie oft ich committen soll: nach jeder Änderung, jeder Lektion oder jedem Tag? Begründe.
Schweizer Rechtschreibung (immer «ss», kein Eszett), kurz.

Qualitätscheck

- Die Rettung mit git restore hat geklappt, und Sie können erklären, wann Änderungen unwiederbringlich verloren wären (nie gesichert).
- git log --oneline zeigt mindestens zwei Versionen mit verständlichen Beschreibungen.
- Der Projektordner hat eine .gitignore, und git status zeigt den Ordner vertraulich nicht an.
- Ein Online-Repository mit Unterrichtsmaterial ist privat – oder bewusst öffentlich mit offener Lizenz und ohne vertrauliche Inhalte.
- In keinem Repository liegen Namen, Noten oder Zugangsdaten.

Hängen Sie fest? Frag die KI

Ich bin Lehrperson an einer Schweizer Berufsfachschule und habe noch nie mit Git gearbeitet. Erkläre mir mit einem Bild aus dem Alltag (z. B. Fotoalbum oder Spielstand speichern), was Git macht und warum es für Unterrichtsmaterial nützlich ist. Sag mir auch, was Git NICHT ist (z. B. kein Backup auf einem anderen Gerät, solange es nur lokal läuft).

Klappt nicht? Git ist freie Software und kostet nichts. Für den Sync zwischen Geräten genügt ein Gratiskonto bei GitHub (private Repositories inklusive) – oder Sie lassen den Sync weg und nutzen Git nur lokal als Zeitmaschine. Wer das Terminal meidet: VS Code hat eine eingebaute Git-Ansicht (Quellcodeverwaltung) mit denselben Funktionen als Knöpfe.

Stand 19.11.2026 · CC BY 4.0 Alpasana GmbH · Werkzeuge ändern sich – prüfen Sie selbst.