Cisco ENAUTO 1.1: Utilize Common Version Control Operations with Git  
*Exam Blueprint Topic – Automating Cisco Enterprise Solutions*

## 🔹 Überblick

Im Rahmen der ENAUTO-Zertifizierung müssen Sie grundlegende Git-Befehle und Versionierungskonzepte beherrschen. Git ist das dominierende verteilte Versionskontrollsystem und wird verwendet, um Codeänderungen effizient nachzuverfolgen, zu verwalten und im Team zu bearbeiten.

## 🔹 Wichtige Git-Befehle

| **Befehl** | **Beschreibung** |
| --- | --- |
| git init | Erstellt ein neues lokales Repository |
| git clone <url> | Klont ein bestehendes Remote-Repository |
| git add <datei> | Stellt Dateien für den Commit bereit |
| git commit -m "Nachricht" | Speichert Änderungen lokal mit Nachricht |
| git push | Überträgt Commits zum Remote-Repository |
| git pull | Holt und integriert Änderungen vom Remote |
| git status | Zeigt aktuellen Repo-Status |
| git diff | Zeigt Unterschiede zwischen Dateien/Commits |

## 🔹 Branching und Merging

* **Branch erstellen**  
  git checkout -b feature-xyz  
  → Erstellt und wechselt zu einem neuen Branch
* **Branch zusammenführen (merge)**  
  git merge branchname  
  → Integriert Änderungen von anderem Branch in den aktuellen
* **Konflikte beim Mergen**
  + Git markiert Konfliktstellen mit <<<<<<<, =======, >>>>>>>
  + Benutzer entscheidet manuell, was übernommen wird
  + Danach: git add und git commit

## 🔹 Konfliktmanagement & Tools

* **Automatischer Konflikt-Resolver:**  
  git merge -X theirs branch → Nimmt automatisch die "andere" Version
* **Grafische Tools:**
  + git mergetool öffnet z. B. meld, kdiff3, vimdiff
  + In **VS Code** steht ein visueller Merge-Editor zur Verfügung
* **Rebase vs Merge:**
  + git rebase branch = "Lineare" Historie
  + Bei Konflikten: git rebase --continue

## 🔹 Empfehlenswerte Ressourcen

| **Thema** | **Link** |
| --- | --- |
| Git Grundlagenbuch | <https://git-scm.com/book/en/v2> |
| Git für Anfänger (GeeksForGeeks) | https://www.geeksforgeeks.org/git/git-tutorial |
| Konfliktlösung mit Merge | https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts |
| VS Code Git Integration | https://code.visualstudio.com/docs/sourcecontrol/overview |

## 🔹 Praxistipps

1. Übe clone, add, commit, push regelmäßig.
2. Arbeite mit zwei Branches – erzeuge absichtlich Konflikte.
3. Nutze git diff und status, um Änderungen vor dem Commit zu prüfen.
4. Löse Konflikte zuerst manuell, dann mit Tools.
5. Nutze GitHub oder GitLab für Remote-Repositories.