

Tag 2: Git - Grundlagen

Ziele des Tages

- Git installieren und konfigurieren
- Grundlegende Befehle kennenlernen und praktisch anwenden
- Funktionsweise des lokalen Git-Repositories verstehen

1. Einführung in Git

Git ist:

- Ein verteiltes Versionskontrollsystem (DVCS)
- Ursprünglich von Linus Torvalds entwickelt
- Lokal und dezentral (kein zentraler Server erforderlich)

Vorteile:

- Versionskontrolle
- Verzweigungen (Branches) und paralleles Arbeiten
- Vollständige Historie lokal verfügbar

2. Git installieren & konfigurieren

Installation (Debian/Ubuntu):

```
sudo apt install git
```

Konfiguration:

```
git config --global user.name "Dein Name"
```

```
git config --global user.email "dein@example.com"
```

```
git config --global core.editor "nano"
```

3. Git-Grundbefehle praktisch anwenden

Git-Workflow (lokal):

```
git init
```

```
git status
```

```
git add <datei>
```

```
git commit -m "Nachricht"
```

Beispiel:

```
mkdir git-test && cd git-test
```

```
git init
```

```
echo "Hallo Welt" > hallo.txt
```

```
git status
```

```
git add hallo.txt
```

```
git commit -m "Erster Commit"
```

```
git log
```

4. Repositories klonen und aktualisieren

Remote-Befehle:

```
git clone https://github.com/benutzer/projekt.git
```

```
cd projekt
```

```
git pull
```

```
git push
```

Tipp: Für GitHub-Übungen ggf. SSH-Key erzeugen

5. Verständnisfragen / Mini-Quiz

1. Was ist die Aufgabe von git add?
2. Welche Information zeigt git status?
3. Unterschied Arbeitsverzeichnis vs. Staging-Bereich?
4. Wie klonet man ein Repository?
5. Wozu dient git commit -m?

Wiederholung für morgen:

- Wiederhole init, add, commit, status, log
- Vorbereitung auf Tag 3: Branching & Merging