

# Spickzettel: Git Dateisystem – Ordner & Dateien

## Grundstruktur eines Git-Repositories

```
projekt/
├── .git/           # Git-Verwaltungsordner (nicht verändern!)
├── src/            # Eigene Quelldateien, Code, Funktionen
├── libs/           # Eingebundene externe Bibliotheken
├── bin/            # Kompilierte Binärdateien
├── docs/           # Dokumentation (z.B. Markdown-Dateien)
├── tests/          # Testdateien (Unit-Tests, Integrationstests)
├── README.md       # Projektbeschreibung
└── .gitignore      # Ignorierte Dateien definieren
```

## Wichtige Dateien und Funktionen

Datei	Beschreibung
.gitignore	Definiert Dateien, die nicht versioniert werden
README.md	Kurze Projektbeschreibung
.git-Verzeichnis	Speichert Git-interne Verwaltung
libs/	Enthält externe Abhängigkeiten
bin/	Kompilierte ausführbare Dateien

## Inhalt des .git-Verzeichnisses

Ordner / Datei	Bedeutung
HEAD	Zeigt auf den aktuellen Branch
config	Repository-spezifische Einstellungen
description	Beschreibung für einige Git-Dienste
hooks/	Enthält Skripte für Automatisierung (Pre-/Post-Hooks)
info/	Enthält unversionierte Metadaten
logs/	Änderungsverlauf der Branches
objects/	Speichert Commits, Trees, Blobs (Dateiinhalte)
refs/	Enthält Referenzen auf Branches & Tags

## Standard .gitignore-Vorlage

```
# Logdateien
*.log

# Temporäre und generierte Dateien
/tmp
/build

# Systemdateien
.DS_Store
Thumbs.db
```

```
# Node-Abhängigkeiten
/node_modules

# Externe Bibliotheken (falls nicht versioniert)
/libs/

# Kompilierte Binärdateien
/bin/
*.exe
*.dll
*.so
*.o
*.a
```

## Hinweise

- **.git-Ordner nie manuell bearbeiten!**
- **Externe Bibliotheken (libs/) nicht versionieren, wenn über Paketmanager installiert**
- **Kompilierte Dateien nicht versionieren, sondern nur den Quellcode**
- Klar strukturierte Ordner erleichtern Zusammenarbeit

## Häufige Fragen

- **Sollte ich Bibliotheken und Abhängigkeiten in Git speichern?**
  - Nein, wenn sie über einen Paketmanager installiert werden können (z.B. pip, npm, maven). Versioniere nur die `package.json`, `requirements.txt` oder `pom.xml`.
- **Warum keine kompilierten Dateien in Git?**
  - Binärdateien erzeugen große Repos, sind nicht diff-fähig und lassen sich aus dem Quellcode neu generieren.
- **Was passiert, wenn ich versehentlich Dateien im .git-Ordner lösche?**
  - Kann Repository beschädigen. Lieber nicht verändern!
- **Kann ich Ordnerstruktur später noch anpassen?**
  - Ja, Änderungen möglich, aber mit Bedacht (Commit vorher erstellen).