

# **EINFÜHRUNG IN GIT & GITHUB**

**PICOTALK  
05.08.2024**

**M D**

# AGENDA

- Einführung zu Git & GitHub
- Terminal oder GUI-App?
- Git-Befehle (Terminal)
- Anwendungsszenarien/Hands-On
- Installation & Literatur

# EINFÜHRUNG IN GIT & GITHUB

- GIT: System zur Versionierung von Textdateien
- Eignet sich perfekt für Software/Quelltext, aber nicht für proprietäre Dateien (Office, JPG, etc.)
- Kann lokal oder auf Server laufen, auch Raspberry Pi
- Komplette per Terminal bedienbar, es gibt aber auch GUI-Apps (z.B. GitHub Desktop)
- GitHub: Webseite zum Archivieren und Zusammenarbeit von Code-Projekten
- Versionierung erfolgt über die GIT-Integration
- Basisversion (nur öffentliche Projekte) kostenlos
- Join GitHub · GitHub
- Alternative: GitLab
- Webseite von Git: Git (git-scm.com)

# EINFÜHRUNG: VOKABELN

- **Repository (Repo):** Ordner, der die Projektdateien enthält. **Remote-Repo** ist das Repo auf einem Server (z.B. GitHub)
- **Branch:** Unabhängige Kopie eines Projekts. Engl. für „Zweig“. Jedes Repo hat einen Haupt-Branch, in dem alles gespeichert wird. Weitere Branches eignen sich z.B. Software mit einigen Änderungen zu testen, ohne das alte zu verlieren
- **Commit:** Momentaufnahme des Projekts zu einer bestimmten Zeit. Jeder Commit wird kommentiert
- **History:** Besteht aus allen vorherigen Commits
- **Clone:** Ein Klon ist eine Kopie des Projekts von einem Server (z.B. GitHub) auf den eigenen Computer. Beide Kopien sind mit einander verbunden, sodass lokale Änderungen auf den Server synchronisiert werden können
- **Fork:** Ist eine Kopie des Projekts einer anderen Person, die unabhängig vom Original daran weiterarbeiten kann. Änderungen im Fork können ins Original übernommen werden.
- **Pull:** „Ziehen“, also integrieren von Änderungen im Remote-Repo auf die eigene Festplatte.
- **Push:** „Drucken“, also das ändern des Remote-Repo mit den Änderungen auf der eigenen Festplatte
- **Pull request:** Bitte um Übernahme der eigenen Änderungen (Fork) ins Original
- Quelle: [git vocabulary cheatsheet \(github.com\)](https://github.com/git/vocabulary)

# TERMINAL ODER GUI-APP?

- Gute GUI-Apps kosten Geld
- Grundsätzlich nicht nötig, wenn man die Befehle für das Terminal kennt
- Für uns: Desktop-Client von GitHub: [GitHub Desktop](#) | [Simple collaboration from your desktop](#)
- Bei Bedarf: [Git - GUI Clients](#) ([git-scm.com](#))
- Raspberry Pi als Git-Server: [Den Raspberry Pi als Git-Server nutzen - Raspberry Pi Geek](#) ([raspberry-pi-geek.de](#))

# **GIT-BEFEHLE (TERMINAL)**

- Mischung aus Linux und dem Git-Vokabluar
- Liste: [Git - Reference \(git-scm.com\)](https://git-scm.com)

# ANWENDUNGSSZENARIEN

- PicoBello Code-Beispiele
- PicoTalk Code-Beispiele
- Eigene Projekte für andere Pico-Nutzer
- Vorteile: Immer erreichbar, immer Aktuell, immer mit Zugriff auf Nutzer-Erweiterungen

# HANDS-ON AUF GITHUB

- <https://github.com/>



# INSTALLATION & LITERATUR

- Zur Installation einfach der Anleitung folgen:
- [Git - Git installieren \(git-scm.com\)](https://git-scm.com)
- [Git Lernen in 30 Minuten - Anfänger Tutorial \(2022\) \(lerneprogrammieren.de\)](https://lerneprogrammieren.de)