

# **Informatik-Begriffsnetz** **Github für HTML-Überarbeitung**

Stand 11.02.2026

**Arbeitskreis Informatik-Begriffsnetz  
der GI-Fachgruppen SWT-SE und WI-VM**

# Überblick

1. Öffentliches Repository, eingerichtet durch Gerhard Fessler (gerd196, Gerd)  
<https://github.com/gerd196/Informatikbegriffsnetz>
2. Inhalt: Befüllt mit den durch xmllint bearbeitete Dateien des Informatik-Begriffsnetz
  - Webseitenstand: Kanban
3. Schreibender Zugriff benötigt **Username** auf GitHub  
**Username** muss in das Repository eingetragen werden
4. **Lokale** Installation von Git sehr sinnvoll

# Registrierung auf GitHub (einmalig für jeden der mitarbeiten will)

1. In Browser GitHub öffnen: <https://github.com>
2. E-Mail-Adresse in weißes Feld eingeben  
Muss wegen Bestätigungs-E-Mail funktionieren
3. [Sign up for GitHub](#) (grünlicher Button) klicken
4. Aufklappenden Bereich für Cookies im Formular „Sign up for Github“: [Reject](#)
5. Password, Username und Your Country/Region ausfüllen
6. [Create Account](#) klicken
7. E-Mail empfangen
8. In E-Mail enthaltenen Code eingeben
9. Einloggen mit Usernamen und Passwort: [Sign in](#) klicken
10. Username bzw. eingegebene E-Mail an  
[gerhard.fessler@gf-qb.de](mailto:gerhard.fessler@gf-qb.de)  
senden, um schreibenden Zugriff zu erhalten

# Repository angelegt (einmalig nach Anlegen des GitHub-Accounts durch gerd196)

1. [New](#) klicken (grüner Button)
2. Repository name: [Informatikbegriffsnetz](#)
3. Description: [Repository for sanitizing webpages of the Informatikbegriffsnetz](#)
4. Choose visibility: [Public](#)
5. Add README: [On](#)
6. Add .gitignore: [No .gitignore](#)
7. Add license: [Creative Commons Zero v1.0 Universal](#)
8. [Create repository](#) klicken (grüner Button)
9. README durch klicken auf Bleistiftsymbol erstellt
10. [Commit changes](#) klicken (grüner Button)
11. Initiale commit message: [Initiales README](#)
12. Neue Verzeichnisse hinzugefügt
13. [+](#) geklickt
14. [+ Create new file](#) geklickt: [Informatikbegriffsnetz/web/leer.html](#)
15. [Commit changes](#) klicken (grüner Button)
16. etc/leer.txt genauso angelegt

# GitHub-Personal Access Token (classic) erstellen

1. Einloggen in GitHub (<https://github.com>, Sign in)
2. Rechts oben auf das „Photo“ klicken (farbiges Symbol im Kreis)
3. Auf Menüpunkt „Settings“ klicken
4. Linke Seite auf <> Developer settings klicken (ganz unten)
5. Personal access tokens aufklappen
6. auf Tokens (classic) klicken
7. Generate new token aufklappen
8. Generate new token (classic) klicken
9. Note: Eindeutigen Namen eingeben
10. Expiration: No expiration
11. x repo
12. x workflow
13. x write:packages
14. x delete:packages
15. x admin:org
16. evtl. weitere Scopes auswählen
17. Generate token
18. Token kopieren und an sicherer Stelle abspeichern

## Token verfügbar machen

1. GitHub CLI installieren, betriebssystem-abhängig: GitHub CLI installieren eventuell Suchpfad anpassen
2. In Terminal/Konsole: `gh auth login` aufrufen und Anweisungen folgen  
[Github.com](https://github.com)  
[HTTPS](https://github.com)  
Paste an `authentication token`: Gespeichertes Token eingeben

# Lokales Repository einrichten

- Mit oder ohne grafische Benutzeroberfläche
1. Geeignetes (lokales) übergeordnetes Verzeichnis einrichten  
Die folgenden Schritte sind Kommandozeilen
  2. `cd <geeignetes übergeordnetes Verzeichnis>`
  3. Repository „clonen“ (Unix-Syntax)  
`git clone https://github.com/gerd196/Informatikbegriffsnetz`  
`cd Informatikbegriffsnetz`  
`git config --global pull.rebase "true"`
  4. Verbindung prüfen  
`git remote -v`
  5. Status prüfen  
`git status`

# (Einmaliges) initiales Befüllen des GitHub-Repositorys durch gerd196

1. In lokales Verzeichnis gehen  
`cd <geeignetes übergeordnetes Verzeichnis>/Informatikbegriffsnetz`
2. Repository aktualisieren  
`git pull origin main`
3. Dateien aus dem mit xmllint bearbeiteten Kanban-Stand hereinkopiert  
Unterverzeichnisse web und etc
4. `git add -A`
5. `git commit -m "Initialisierung"`
6. `git push origin main`
- 7.

# Arbeiten mit Git und GitHub (jeder Mitwirkende, Kommandozeile, Unix-Syntax)

1. In lokales Verzeichnis gehen

`cd <geeignetes übergeordnetes Verzeichnis>/Informatikbegriffsnetz`

2. Repository aktualisieren

`git pull origin main`

3. Eigenen Arbeitsbereich einrichten (**Branch**), eventuell nur einmalig

`git branch <eigener branch name>`

4. In eigenen Arbeitsbereich wechseln

`git checkout <eigener branch name>`

5. Im eigenen Arbeitsbereich arbeiten, z. B. Dateien erstellen bzw. modifizieren usw.

6. Nur für **neu angelegte Datei**: <Dateiname> in (lokalem) git registrieren (Staging)

`git add <Dateiname>`

Wiederholen für andere neu angelegte Dateien. Alternative für alle neuen Dateien: `git add -A`  
prüfen mit `git status`

7. Neue bzw. geänderte Dateien freigeben (Commit bzw. Snapshot anlegen)

`git commit -m "sinnvoller Kommentar"`

8. Branch auf GitHub hochladen

`git push origin HEAD`

9. Admin bitten, Änderungen auf GitHub in main zu übernehmen

oder: Nach einloggen in [github.com](https://github.com) selbst machen

Alternative:

Im Browser direkt auf GitHub editieren

# Nützliche git-Kommandos

- git status
- git remote -v
- git add -An
- git add -A
- git checkout -b <branch name>
- git branch
- git push origin <branch name>
- git log
- git remote show origin

## Hinweise

Es ist sinnvoll, eine 2-Faktor-Anmeldung auf GitHub einzurichten

Für das Editieren des Informatik-Begriffsnetzes aber nicht notwendig

- TOTP ist empfehlenswert, ein Passkey besser
- Für TOTP eine der Authenticator-Apps auf Smartphone einrichten empfehlenswert
  - FreeOTP+
  - Aegis Authenticator
- nicht empfehlenswert:
  - Authenticator-Apps von Microsoft, Google usw.  
verhindern lokale Backups, Datenübertragung zu andern Apps usw.
- FreeOTP+ und Aegis sind für Android auch über F-Droid installierbar sinnvoll, da Apps auf dem Google Playstore oft mit versteckten Datenabflüssen versehen sind

Zusätzlich ist ein SSH-Zugang zu GitHub sinnvoll