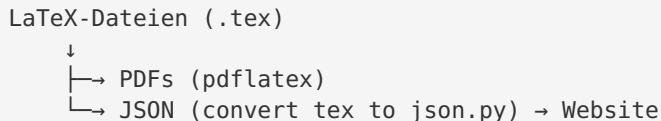


TOE Advent Calendar - Entwicklungs-Workflow

Überblick

Dieses Projekt verwendet **Single Source of Truth**: LaTeX-Dateien sind die einzige Quelle für alle Inhalte.



Automatischer Build-Prozess

Bei jedem Push zu GitHub:

1. **GitHub Actions startet automatisch** (`.github/workflows/deploy.yml`)
2. **LaTeX → JSON**: `convert_tex_to_json.py` generiert `public/advent_data.json`
3. **LaTeX → PDF**: Alle 27 PDFs werden kompiliert → `public/pdfs/`
4. **Next.js Build**: Website wird gebaut → `out/`
5. **Deploy**: Automatisches Deployment zu `gh-pages` Branch

Ergebnis: Die Website ist live unter <https://amu2.github.io/advent-calendar-2025/>

Konfliktfreies Arbeiten (DeepAgent + Sie)

Goldene Regel:

Arbeiten Sie NUR mit dem `main` Branch. Berühren Sie NIE den `gh-pages` Branch manuell!

Workflow für Sie:

```
# 1. Holen Sie die neuesten Änderungen
git pull origin main

# 2. Bearbeiten Sie Ihre Dateien:
#     - LaTeX-Dateien (.tex)
#     - React-Komponenten (.tsx)
#     - Styles (.css)

# 3. Testen Sie lokal (optional):
./update-website.sh # Führt den gleichen Prozess wie GitHub Actions aus

# 4. Commiten und pushen:
git add .
git commit -m "Update Day 5 content"
git push origin main

# 5. GitHub Actions baut und deployed automatisch (dauert ~5 Minuten)
```

Was GitHub Actions automatisch macht:

- ✅ Konvertiert **alle** LaTeX → JSON
- ✅ Kompiliert **alle** LaTeX → PDF
- ✅ Baut **die gesamte** Website neu
- ✅ Deployed zu GitHub Pages

Was Sie NIE manuell tun sollten:

- ❌ Nicht `public/advent_data.json` manuell bearbeiten
- ❌ Nicht `gh-pages` Branch berühren
- ❌ Nicht direkt zu `/out` deployen

Datei-Struktur

```
website/
├── advent00.tex - advent31.tex
├── advent-layout.tex
├── convert_tex_to_json.py
└── .github/workflows/deploy.yml
public/
├── advent_data.json
└── bearbeiten!
    ├── pdfs/
    └── app/
        ├── page.tsx
        ├── how-to/page.tsx
        └── impressum/page.tsx
components/
└── update-website.sh

# 🔥 QUELLE: LaTeX-Dateien
# LaTeX-Layout
# Konvertierungs-Script
# GitHub Actions Config
# ⚠️ Automatisch generiert - NICHT manuell
# ⚠️ Automatisch generiert
# Hauptseite
# How-to Seite
# Impressum
# React-Komponenten
# Lokales Build-Script (simuliert GitHub Actions)
```

Lokale Entwicklung

Setup (einmalig):

```
cd /home/ubuntu/advent_project/website
yarn install
```

Entwicklungs-Server starten:

```
yarn dev
# → http://localhost:3000
```

Vollständiger Build (wie GitHub Actions):

```
./update-website.sh
```

Dieser Script macht:

1. LaTeX → JSON Konvertierung
2. LaTeX → PDF Kompilierung (alle 27 Dateien)
3. Next.js Build
4. (Optional) Git Commit & Push

Typische Szenarien

Szenario 1: LaTeX-Inhalt ändern

```
# Sie editieren advent05.tex
nano advent05.tex

# Pull + Push
git pull origin main
git add advent05.tex
git commit -m "Update Day 5: Fix typo"
git push origin main

# GitHub Actions updated automatisch:
# - advent05.pdf
# - advent_data.json (Day 5 entry)
# - Website
```

Szenario 2: Website-Design ändern

```
# Sie editieren components/advent-door.tsx
nano components/advent-door.tsx

git pull origin main
git add components/advent-door.tsx
git commit -m "Update door animation"
git push origin main

# GitHub Actions rebuilt die Website
```

Szenario 3: How-to Seite ändern

```
nano app/how-to/page.tsx

git pull origin main
git add app/how-to/page.tsx
git commit -m "Update instructions"
git push origin main
```

Szenario 4: Neues LaTeX-File hinzufügen

```
# Erstellen Sie advent26.tex (falls noch nicht vorhanden)
nano advent26.tex

git pull origin main
git add advent26.tex
git commit -m "Add Day 26 content"
git push origin main

# GitHub Actions erkennt die neue Datei automatisch!
```



Fehlerbehandlung

“Git Conflict” beim Pull:

```
# Option 1: Stashen Sie Ihre Änderungen
git stash
git pull origin main
git stash pop
# Lösen Sie Konflikte manuell

# Option 2: Force Ihre Änderungen (! überschreibt remote)
git fetch origin
git reset --hard origin/main
```

GitHub Actions Build fehlgeschlagen:

1. Gehen Sie zu: <https://github.com/amu2/advent-calendar-2025/actions>
2. Klicken Sie auf den fehlgeschlagenen Workflow

3. Prüfen Sie die Logs:

- **LaTeX Error:** Syntax-Fehler in .tex Datei
- **JSON Error:** Problem im convert_tex_to_json.py
- **Build Error:** Problem in React-Komponenten

Website zeigt veraltete Inhalte:

```
# Warten Sie 5 Minuten nach dem Push
# Prüfen Sie GitHub Actions Status
# Hard-Refresh im Browser: Ctrl+Shift+R
```



Monitoring

GitHub Actions überprüfen:

1. <https://github.com/amu2/advent-calendar-2025/actions>
2. Grüner Haken = Erfolgreich deployed
3. Rotes X = Build fehlgeschlagen

Live Website:

- <https://amu2.github.io/advent-calendar-2025/>
- GitHub Pages Dashboard: <https://github.com/amu2/advent-calendar-2025/settings/pages>



Technische Details

GitHub Actions Workflow:

- **Trigger:** Push zu `main` Branch
- **Dauer:** ~5-8 Minuten
- **Schritte:**
 1. Python Setup → LaTeX to JSON (30s)
 2. TeX Live Installation → PDF Compilation (3-4min)
 3. Node.js Setup → Next.js Build (2-3min)
 4. Deploy zu gh-pages (10s)

Lokales Build-Script:

- `update-website.sh` : Vollständiger Build + Deploy
- `test-update.sh` : Nur Build, kein Deploy

Dependencies:

- **Python 3.11+:** Für convert_tex_to_json.py
- **TeX Live:** Für LaTeX → PDF
- **Node.js 18+:** Für Next.js
- **Yarn:** Package Manager

Best Practices

DO:

- Immer `git pull` vor dem Arbeiten
- Nur **source files** bearbeiten (`.tex`, `.tsx`, `.css`)
- Kleine, atomare Commits machen
- Beschreibende Commit-Messages verwenden
- GitHub Actions Status überprüfen nach Push

DON'T:

- Nie `public/advent_data.json` manuell bearbeiten
- Nie `gh-pages` Branch direkt berühren
- Nie generierte Dateien committen (PDFs im root, `.next`, `out`)
- Nie force push zu `main` (außer im Notfall)

Hilfe

DeepAgent kontaktieren:

1. Beschreiben Sie das Problem
2. Teilen Sie relevante Logs
3. Geben Sie den Commit-Hash an

Selbst debuggen:

```
# Lokaler Test:  
cd /home/ubuntu/advent_project/website  
python3 convert_tex_to_json.py  
yarn build  
  
# Prüfen Sie die Outputs:  
ls -lh public/advent_data.json  
ls -lh public/pdfs/  
ls -lh out/
```

Zusammenfassung

Das System ist vollautomatisch:

1. Sie bearbeiten LaTeX oder React-Dateien
2. Sie pushen zu GitHub
3. GitHub Actions macht den Rest
4. Website ist nach ~5 Minuten aktualisiert

Keine manuellen Schritte nötig für:

- PDF-Generierung
- JSON-Konvertierung

- Website-Build
- Deployment

Konfliktfrei arbeiten:

- Immer `git pull` vor dem Pushen
- Nur `main` Branch verwenden
- GitHub Actions macht das Deployment

 **Viel Erfolg mit dem TOE Advent Calendar!** 