

# tf-util リポジトリ概要

作成者: システム v1.0, 2026-01-25

## はじめに

このドキュメントは、tfukuda675/tf-util リポジトリの構成と概要をまとめたものです。

## リポジトリ概要

tf-utilは、開発作業を支援する様々なユーティリティとツールを集めたリポジトリです。

### 主要な機能

- MCP (Model Context Protocol) サーバーの実装
- コンテナ環境でのドキュメント生成
- AsciiDocからPDF/HTML変換
- GitHub Actions self-hosted runner
- Kaggle機械学習ツール
- Vimカスタマイズ
- Python自動化スクリプト

## ディレクトリ構成

### 1. container\_for\_cloude\_code/

Claude Code開発用のコンテナ環境を提供します。

### プロジェクト一覧

#### asciidoc\_to\_html\_pad/

- **概要:** AsciiDocファイルからPDF/HTML生成システム
- **機能:**
  - Krokiを使用したMermaid図生成
  - PlantUML図生成
  - 日本語フォント対応 (Noto CJK)
  - LaTeX数式処理
- **主要ファイル:**
  - Dockerfile: AsciiDoctor環境定義
  - docker-compose.yml: Docker環境構成
  - build.sh: ビルドスクリプト
  - Makefile: コマンド実行用

### **asciidoc\_simple/**

- **概要:** シンプルなAsciiDoc変換プロジェクト
- **機能:**
  - PDF/HTML変換
  - 日本語フォント対応
  - LaTeX数式表記対応
- **特徴:** 軽量でシンプルな構成

### **double\_libe\_gantt\_chart/**

- **概要:** ガントチャート生成ツール
- **ファイル:** index.html, script.js

## **2. mcp\_server\_local\_file\_treat/**

MCPサーバーのローカルファイル処理実装です。

### **機能**

- ドキュメントディレクトリのファイル一覧取得
- ファイル内容の読み取り（テキスト、PDF対応）
- MCPプロトコルによる通信

### **主要ファイル**

- server.js: MCPサーバー実装
- Dockerfile: コンテナ環境
- package.json: Node.js依存関係
- STARC\_README.md: Verilog HDLコーディングガイドライン

### **STARC ガイドライン内容**

- Verilog HDL基本原則
- 命名規則
- モジュール設計
- ファイル管理規則

## **3. podman\_in\_podman\_github\_runner\_claude\_code/**

GitHub Actions self-hosted runnerをPodman上で実行する環境です。

### **構成要素**

- Dockerfile: Runner環境構築
- launch\_runner.sh: ランナー起動スクリプト

- `containers.conf`: Podman設定
- `docker-compose.txt`: 環境構成

#### 機能

- Podman rootless実行
- GitHub Actions統合
- コンテナ内でのPodman実行

### 4. kaggle/

Kaggle機械学習コンペティション用ツールです。

#### ファイル

- `baseline_notebook.ipynb`: ベースライン実装ノートブック
- `my_utils.py`: ユーティリティ関数

#### 機能

- Google Colab統合
- Kaggle API設定
- データ前処理
- モデル学習パイプライン
- BERT日本語モデル対応

### 5. scripts/src/

各種自動化スクリプト集です。

#### github\_python/

- `get_issue.py`: GitHub Issue取得
- `get_projects.py`: GitHub Projects管理

#### polars\_text\_processing/

- `polars_text_processing.py`: Polarsを使用したテキスト処理
- 高速データ処理
- 正規表現による抽出

### 6. vim/

Vimエディタのカスタマイズファイルです。

## 内容

- colors/PaperColor.vim: カラースキーム
- doc/indent\_guides.txt: インデントガイドドキュメント
- template/python.txt: Pythonテンプレート

## 7. ansible/

Ansible設定ファイルです。

### inventories/

- production環境設定
- staging環境設定（未使用）

## 技術スタック

### コンテナ技術

- Docker
- Podman
- docker-compose

### プログラミング言語

- Python 3.11+
- JavaScript/Node.js 22
- Bash/Shell

### ドキュメント生成

- AsciiDoctor
- asciidoctor-pdf
- Kroki（図表生成）
- LaTeX

### 機械学習

- TensorFlow
- PyTorch
- scikit-learn
- Transformers（BERT）

## その他ツール

- GitHub Actions
- Vim
- Ansible
- Polars

## 使用例

### AsciiDoc変換

```
cd container_for_cloude_code/projects/asciidoc_to_html_pad
make build
```

### GitHub Issues取得

```
cd scripts/src/github_python
./get_issue.py
```

### Polarsテキスト処理

```
cd scripts/src/polars_text_processing
./polars_text_processing.py --config config.yaml
```

## 環境構築

### 前提条件

- Docker/Podman
- Python 3.11以上
- Node.js 22以上

### セットアップ手順

1. リポジトリクローン

```
git clone https://github.com/tfukuda675/tf-util.git
cd tf-util
```

2. Docker環境構築

```
cd container_for_cloude_code/projects/asciidoc_to_html_pad
docker-compose up -d
```

3. Python環境

```
pip install -r requirements.txt
```

## プロジェクト特徴

### 1. 多機能性

様々な開発タスクを支援するツール群を提供

### 2. コンテナ化

Docker/Podmanによる環境の分離と再現性

### 3. 自動化

GitHub Actionsやスクリプトによる作業自動化

### 4. ドキュメント重視

AsciiDocによる高品質なドキュメント生成

### 5. 機械学習対応

Kaggleコンペティション用のツールとテンプレート

## まとめ

tf-utilリポジトリは、開発作業を効率化するための包括的なツールセットです。ドキュメント生成、コンテナ管理、自動化スクリプト、機械学習など、幅広い用途に対応しています。

## ライセンス

MIT License

## 作成日

2026年1月25日