

# 現場で神ってる！ エンジニア必携ツール10選

～Git・Slack・Dockerなど現場で使うツールの役割と使い方を徹底解説～



# 1. 本資料について

- ・エンジニアが現場でよく使う必須ツール10選を紹介
- ・Git・Slack・Docker など、開発・コミュニケーション・環境構築に欠かせないツールの役割と基本的な使い方を理解
- ・「なぜ必要か」を知ることで、単なる操作ではなく、仕事にどう活かすかをイメージできる

実務に入る前から環境を整えることで、スムーズにチームに  
溶け込めます！



## 2. ツール10選リスト(一覧表)

ツール名	役割・用途	主な利用シーン
① Git 	バージョン管理システム	ソースコードの変更管理
② GitHub/GitLab 	Gitリポジトリ管理・レビュー	チーム開発のコード共有
③ Slack 	チームコミュニケーションツール	チャット・情報共有
④ Docker 	コンテナ仮想化ツール	開発環境の構築・本番環境管理
⑤ Jira 	タスク管理・進捗管理ツール	プロジェクト管理
⑥ Visual Studio Code 	ソースコードエディタ	コーディング
⑦ Postman 	API開発・テストツール	APIの動作検証
⑧ Jenkins 	継続的インテグレーション(CI)ツール	自動ビルド・テスト
⑨ MySQL/PostgreSQL  	データベース管理システム	データの保存・管理
⑩ Zoom 	オンライン会議ツール	リモートミーティング

## 3. 各ツールの詳細解説

### ① Git: バージョン管理システムの基本

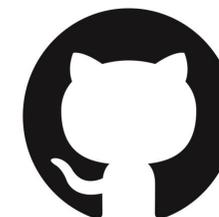
- **役割**: ソースコードの変更履歴を記録・管理し、複数人で安全に共同開発ができる
- **初心者向けポイント**:
  - コミット、プッシュ、プルの意味と使い方を覚える
  - ブランチを使って機能ごとに開発
  - 変更履歴の差分を見て問題点を発見
- **現場での活用イメージ**:
  - 複数人で同時にコードを書いても衝突を防ぐ
  - 過去のバージョンに戻せる安心感



## 3. 各ツールの詳細解説

### ② GitHub/GitLab: リポジトリ管理とレビュー

- **役割** : Gitリポジトリをオンラインで管理、プルリクエストでコードレビューも実施
- **初心者向けポイント** :
  - リポジトリのクローン・フォーク・プルリクエストの意味
  - Issues機能でバグ管理や課題の可視化
- **現場活用例** :
  - チームメンバー同士でコードをレビューし品質を担保
  - バグや新機能のタスク管理



## 3. 各ツールの詳細解説

### ③ Slack: コミュニケーションの中心

- **役割**: リアルタイムのチャットやファイル共有ができるチーム連絡ツール
- **初心者向けポイント**:
  - チャンネルごとに話題を分けて情報整理
  - メンションやリマインダー機能活用
  - Bot連携で作業自動化も可能
- **現場活用イメージ**:
  - 日常的な質問や雑談からプロジェクト連絡まで多用途で利用



## 3. 各ツールの詳細解説

### ④ Docker: 開発と本番の環境差をなくすコンテナ技術

- 役割: アプリケーションの動作環境をパッケージ化し、どこでも同じ環境で動かせる
- 初心者向けポイント :
  - Dockerイメージとコンテナの違い
  - Dockerfileで環境構築を自動化
  - Docker Composeで複数コンテナをまとめて管理
- 現場での活用 :
  - チーム全員が同じ環境で開発
  - 本番環境へのデプロイも効率化



## 3. 各ツールの詳細解説

### ⑤ Jira: タスクと進捗が見える化

- **役割**: プロジェクトのタスク管理、進捗把握、課題トラッキングに特化
- **初心者向けポイント**:
  - チケット(タスク)を作成し担当を割り当てる
  - スプリントやカンバンボードで進行管理
- **現場活用**:
  - 開発チームでタスク分担し納期を管理



## 3. 各ツールの詳細解説

### ⑥ Visual Studio Code: 多機能で人気のコードエディタ

- **役割**: ソースコードの編集、デバッグ、拡張機能でカスタマイズ可能
- **初心者向けポイント**:
  - プラグイン導入で言語補完やLintを活用
  - Git連携でコード管理も
- **現場活用**:
  - ほぼ全員が使う標準エディタ



## 3. 各ツールの詳細解説

### ⑦ Postman: APIの開発・テスト支援

- **役割** : REST APIのリクエスト送信・レスポンス確認をGUIで簡単に行うツール
- **初心者向けポイント** :
  - GET/POSTなどHTTPメソッドの使い分け
  - JSONレスポンスの見やすい表示
- **現場活用** :
  - APIの動作確認や開発中の動作検証に必須



## 3. 各ツールの詳細解説

### ⑧ Jenkins: 自動ビルド・テストツール(CI/CD)

- **役割**: コードの自動ビルド、テスト、デプロイを自動化するツール
- **初心者向けポイント**:
  - ジョブ作成でビルドフローを設定
  - 成果物の自動生成・配布が可能
- **現場活用**:
  - 人手を減らしミスを減らす開発体制の基盤



# Jenkins

## 3. 各ツールの詳細解説

### ⑨ MySQL/PostgreSQL: 代表的なデータベース管理システム

- **役割**: データの保存・検索を効率的に管理するソフトウェア
- **初心者向けポイント**:
  - SQLでデータ操作(SELECT、INSERT、UPDATE、DELETE)
  - テーブル設計の基本
- **現場活用**:
  - Webアプリのデータ管理に不可欠



## 3. 各ツールの詳細解説

### ⑩ Zoom: リモートワークに必須のオンライン会議ツール

- **役割**: 音声・映像会議、画面共有、録画などを簡単に実施
- **初心者向けポイント**:
  - ミュート・ビデオオンオフの切替え
  - ブレイクアウトルームで小グループ会話
- **現場活用**:
  - リモート面談やスクラムミーティングで日常的に使用



## 4. まとめ

ツールは“使うこと”が目的ではなく、チームで効率よく開発するための手段です。

基本を押さえれば、未経験でも現場でのスタートダッシュが切れます。

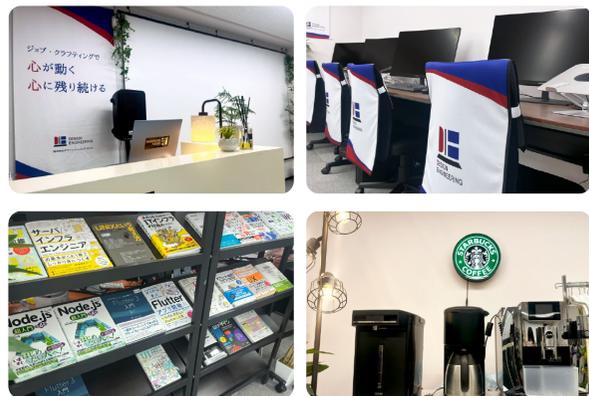
「知っている」から「使える」へ。

今日から一歩ずつ実践してみましよう！





LINE公式アカウントにて  
最新情報配信中！



無料で自由に使える  
学習&カフェスペース開放中！

変化を楽しみ、自分らしく未来へ。

デザインエンジニアリングは、  
挑戦するエンジニアの一步を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、  
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、  
毎日が少しずつ楽しくなる未来へ。  
未経験でも大丈夫。あなたの最初の一步を、心からお待ちしています！



URL: <https://design-engineering.jp/>



イベント・セミナー開催中！

カジュアル面談・エントリーは  
こちらから！



LINE ID: @749gaovb