

イケてるターミナルを使おう！

M240285 前土井光章

はじめに

プログラミングをする上で、ターミナルはエディタ等に比べて「とりあえず動けばいい」と軽視されがちです。しかし、開発時に必ず使うツールだからこそ、見た目・操作感・機能の違いは無視できません。そして何より、イケてるターミナルを使うとそれだけでテンションが上がりますし、プログラミングがデキる人になった気分にもなります。

この記事では、私がおすすするいくつかのターミナルの紹介と、私が普段実際に使用しているWezTermの設定ファイルを共有します✨

そもそもターミナルを新しくインストールする必要はあるの？

結論から言うと、**必須ではありません**。macOSには標準で「ターミナル」、Windowsでも標準で「コマンドプロンプト」や「Power Shell」が用意されており、基本的なコマンド操作はすべて可能です。

ただし、標準ターミナルでは見た目の自由度や機能が限定的です。

開発に欠かせないツールだからこそ、自分の作業スタイルや好みに合わせて新たな選択肢を検討してみるのはいかがでしょうか？

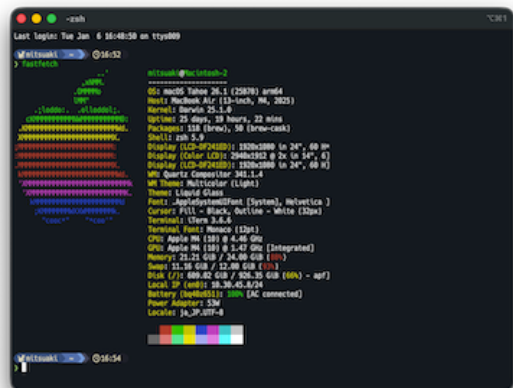
どんな種類があるの？

以下に、私が実際にメインで使用したことのあるターミナルを紹介します。どれもおすすめできるターミナルです。

※以下の画像はフォントやテーマを統一していません。あくまで雰囲気を感じ取っていただければと思います。それぞれ見た目は柔軟にカスタマイズ可能です。

iTerm2 (Mac)

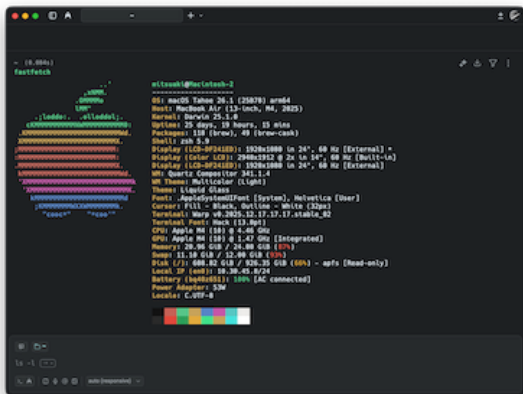
- 割と昔から存在しており、安定感がある（私が学部生の時にメインで使用していました）
- 日本語の記事も多く、設定が GUI 中心でわかりやすく、カスタマイズの幅も広い。



Warp (Mac, Linux, Windows)

- モダンなUIが特徴のRust製ターミナル
- 独自思想が強く、AIがデフォルトで組み込まれていたり、コマンドがブロック単位になっているなど、他のターミナルとは異なる点が多い

- 最近はAI機能をさらに強化し、コードエディタとしての利用も想定されているらしい



Ghostty (Mac, Linux)

- 比較的最近リリースされたOSネイティブなターミナル
- シンプル・ミニマルで非常に高速
- macOSの見た目と相性◎
- おぼけのアイコンが可愛い🐼 (ghostty +boo というコマンドを打つと、おぼけのアニメーションが見れます)
- テキストベースで設定可能。設定は数行ででき、とても簡単。
- 機能や拡張性は限定的だが、十分使える



WezTerm (Mac, Linux, Windows)

- 私が現在メインで使用しているターミナル
- Rust製。GPU レンダリングによる高速描画
- Luaというプログラミング言語で設定を記述
- カスタマイズをたくさんしたい人にとにかくおすすめ！

↓2026年1月現在、私が使用しているWezTermの外観。結構カスタマイズしました。



私がWezTermを使用している理由

私がWezTermを使用しているのは、主に次の点です。

- 見た目やキーバインドを**Lua**というプログラミング言語で柔軟に記載できる
 - 設定に変数やIF文を用いることが可能 → より柔軟にカスタマイズが可能
 - テキストファイルなので、設定をGitで管理可能
 - テキストファイルなので、他の人に設定を共有しやすい
- ターミナルの出力をVimキーバインドでコピーできるコピーモードがある
 - キーボードから手を離さずに過去の出力をさかのぼったり、コピーが可能

カスタマイズが豊富なので、使いながら少しずつ設定を調整し、自分好みに環境を育てていくことが可能です🌱

私の設定ファイル

私が2026年1月現在、WezTermに設定している内容を公開します！
もし興味があれば、以下のGitHubのリポジトリを参考にしてください！

<https://github.com/mohachi8/bug2026>

主なこだわりポイント

- 見た目
 - シンプルな見た目にするために、新規タブボタンや削除ボタン、ウィンドウ操作のボタンなど、キーボードショートカットで操作できるものは全て表示していません
 - 背景を少し透過し、ぼかしを入れています
 - ウィンドウの境界がわかるように境界線を入れています
 - ターミナルのベル🔔の音を消して、カーソルの見た目が一瞬変わるようにしています
- リーダーキーを用いたキーバインド
 - リーダーキーとは、複数のキー操作をまとめられる親となるキーです
 - 私は **Ctrl + Space** に設定しています。USキーボードの方はIME切り替えの割り当てをOFFにする必要があります。
 - 例えば、**リーダーキー → z** を入力すると分割したペインを拡大できるように割り当てています
- モード切り替え
 - Vimのモードと同じように、文字入力以外のモードがあります
 - **リーダーキー → c** でコピーモード、**リーダーキー → r** でペインのサイズ調整に割り当てています
- ステータスバーに現在のモードや状態を表示
 - コピーモード中やリーダーキーを押した時などはアイコン表示

シンプルかつ、現在の状態が把握できる見た目にし、過去の出力のコピーやペインのサイズ変更まで全てキーボードで行えるようにしています

おまけ：プロンプト設定

ターミナルをより良い見た目にするためには、プロンプトの設定も不可欠です。ターミナルにおけるプロンプトとは、コマンド入力の待ち受け状態を示す表示（>> や % など）のことです。

デフォルトは、Macでは `user@MacBook-Air ~ %`、Windowsでは `PS C:\Users>` のように表示されますね。

私は以下のような見た目になっています👁️



これは**Starship**というパッケージをインストールすることで簡単に変更できます。
設定はtomlという形式のファイルに書くのですが、公式サイトにもいくつかテンプレートが用意してあるので、まずはそれを使用するのが良いかもしれません。こちら私の設定をGitHubリポジトリで公開しています。

まとめ

今回の記事で、ターミナルにも様々な選択肢があるということをお伝えできれば幸いです。
ここでは紹介していないターミナルもたくさんあるので、ぜひ、あなたにぴったりのターミナルを探してみてください！

イケてるターミナルを使って楽しく開発しましょう！

Happy Hacking!!