

命令書：単一銘柄の損益計算ツール

[2026-09-12 作成] このファイルは、同じものをもう一度作らせるための命令書（プロンプト）です。上から順にクロコに渡せば、同じ手順で再現できます。

1. 体制

役	担当
メイン（クロコ本体）	進行管理・最終判断・本人への報告
code-generator	実装
auditor	監査。書き込み権限を持たない

2. 先に作るファイル（2つ）

`.claude/agents/code-generator.md`

```
---
name: code-generator
description: HTML/JS/CSSのコード実装を担当する。新機能の追加や修正指示を受けたときに使用する。
tools: Read, Write, Edit, Bash, Glob, Grep
model: sonnet
---
```

あなたは実装担当エンジニアです。与えられた仕様に基づいてコードを実装してください。

- 仕様に忠実に実装する。余計な機能や過剰な装飾は加えない
- 単一のHTMLファイルに、CSSとJavaScriptを含める形にする
- 入力値のバリデーション(空欄、0、負の数、文字列混入など)を必ず行う
- 実装後、コードの構造と主な処理内容を簡潔に報告する
- auditorから修正指示を受けた場合は、指摘箇所を優先的に直し、修正内容を明確に報告する

```
---
name: auditor
description: code-generatorが実装したコードの品質・仕様適合性を監査する。実装完了後は必ず使用すること。
tools: Read, Grep, Glob, Bash
model: sonnet
---
```

あなたは監査役です。コードの修正は一切行わず、レビューのみを行います。

以下の観点でチェックしてください:

1. 仕様通りに動作するか(入力→計算→表示のロジックが正しいか)
2. 損益・損益率の計算式に誤りがないか
3. 入力値のエラーハンドリング(未入力、0、負数、非数値)が十分か
4. UIとして最低限の見やすさ・使いやすさがあるか
5. コードに明らかな不具合やタイプミスがないか

報告フォーマット:

- 判定: 合格 / 要修正
- 問題点(要修正の場合は、具体的な箇所と修正内容の指示)
- 総評

※ 定義ファイルは**起動時に読み込まれる**。作った直後のセッションでは名前と呼ばない。その場合は、上の本文をそのまま渡して同じ役を演じさせる。

3. アプリ仕様

ブラウザで開くだけで動く単一HTMLファイル。CSSとJavaScriptを同じファイルに含める。

入力（3つ。小数を許容する）

購入単価（円）／株数／現在の単価（円）

計算

購入金額（合計）	= 購入単価 × 株数
現在の評価額（合計）	= 現在の単価 × 株数
損益	= 現在の評価額（合計） - 購入金額（合計）
損益率	= 損益 ÷ 購入金額（合計） × 100

表示（結果欄は4行）

行	書式	色
購入金額（合計）	3桁区切り＋「円」	通常色
現在の評価額（合計）	3桁区切り＋「円」	通常色
損益	3桁区切り＋「円」	プラス＝緑／マイナス＝赤／0＝通常色
損益率	小数第2位、プラスは＋を付ける	損益と同じ色

- 表示する金額は四捨五入して整数の円にする。**内部の計算は丸めない**
- 色と損益率の符号は、**丸め後の損益が0かどうか**で判定する

ボタン（2つ）

ボタン	動作
計算する（青）	検証してから計算し、結果を表示する
クリアする（グレー）	結果欄の下に 常に表示 。入力3つを空にし、結果を隠し、エラーを消し、購入単価にカーソルを戻す

入力の検証

条件	対象
未入力・空欄	3項目すべて
数値として読み取れない	3項目すべて
0 以下	購入単価、株数
負の数	現在の単価（0は許容）

エラーは項目名を添えて列挙し、結果欄は表示しない。**全角数字・全角カンマ・全角ピリオドは半角に読み替えてから**判定する。

前提と禁止

- `<!DOCTYPE html>` / `<html lang="ja">` / `<meta charset="UTF-8">` を必ず付ける
- 外部のCDN・ライブラリは**一切使わない**（オフラインで動く）
- 使うのは64歳の非エンジニア。Windows の Chrome でファイルを直接開く。文字は小さすぎないように
- 禁止**：装飾・アニメーション・ダークモード・履歴保存・Enterキーでの計算・仕様外の表示項目（1株単価の逆算など）・売買の推奨や助言に読める文言

4. 進め方

- code-generator に実装を依頼する

2. 実装が終わったら、**必ず** auditor にレビューを依頼する
3. 「要修正」なら、**指摘をそのまま** code-generator に渡してやり直させる
4. 「合格」が出るまで 2~3 を繰り返す（**最大3回**）
5. 3回で合格しなければ、その時点の状態と問題点を報告し、本人の判断を仰ぐ
6. 最後に、完成したファイルの場所・監査で指摘された点・修正の経緯を報告する

5. 進行管理役への注意（今回の反省）

- **仕様に違和感があれば、着手前に本人に確認する。** 今回は「株数が計算に使われない」仕様のおかしさに気づきながら申し送りに書くだけで済ませ、作り直しになった
- **監査役は「仕様どおりか」しか見ない。** 仕様そのものの誤りは止められない。そこは進行管理役の仕事
- **実装担当の報告を鵜呑みにしない。** 実物を自分で読む。今回、報告と中身が違っていた例が1件あった
- **監査の前後でファイルのハッシュを比較し、監査役が書き込んでいないことを確認する**
- **自分が思いついた追加仕様は、自分で穴を作る。** 今回「全角対応」を足し、全角の小数点を取りこぼした。追加分も必ず監査に回す
- 完成の判断は、報告ではなく**自分で実際に動かして行う**