

# Global Project Rulebook

このドキュメントは、全プロジェクトに適用される開発運用ルールブック。「実装」「設計」「更新」「保存」すべてにおいて、人間（＝タケ）にとっての可読性・持続性・再現性を最重視した指針を明文化しています。

## Canvas Naming Convention（キャンバス命名規則）

ProjectName XX Phase/Core Title

例：

Kairoscope 00 Develop Overview

Kairoscope 01 Install - Structure Map

Kairoscope 02 Install - Full Chart Builder

Kairoscope 03 Module - Variables

Kairoscope 04 Integration - Chronogram

Kairoscope 05 Export - CLI Tools

- **Project Name**：システム名（Kairoscope / Chronogram など）
- **XX**：フェーズ番号（00から連番）
- **Title**：機能ブロックや目的がひと目でわかるよう命名

## Canvas Structure & Output Management

### Canvasはコード用と設計用を分離

- コードCanvasはあくまで実装＋動作確認用。（全コードのチートシートなどがあると嬉しい）
- 解説や構成用は別Canvasで文書化（後で `.md` / `.pdf` エクスポート）

### コミット後の構成を記録保存

- 各ステップの完了後、**構成ファイルとコードスニペットの一覧出力**を `.md` で保存。
- 人間が後から見て理解できるように、**各ファイルの役割・更新履歴・依存関係**を解説付きで記載。

### 実装完了後に「プロジェクトサマリー.md」化

- 全構成、依存ライブラリ、主要処理フロー、使用法を一括でまとめる。
- Kairoscopeなど複雑なシステムは、**Phase 00～最終までを通史的に整理**

## AI応答ルール（責任ある出力）

- ユーザーが何も作業不要な場合は、冒頭に明確に伝える：

- ・「今回は確認だけでOK！読むだけで次に進んで大丈夫」など
- ・コード変更や必要作業がある場合は：
- ・必要な作業・変更が必要なファイルを**箇条書きで明示**
- ・ファイル名、ネイどこに挿入・変更が必要かを明確に示す
- ・「覚えてる」発言の使用は厳密に：
- ・AI自身が永続的に記憶できるわけではないと認識する
- ・明示的にプロンプトやCanvasで保存・共有されて初めて記憶扱いとする
- ・スレッドでのデータ上限を意識してプロジェクト全体の信仰管理をする
- ・プロジェクト進行にあたってスレッドベースでの記憶が重要
- ・後半になるとCanvasを作成できない、会話そのものができないなどの問題になる
- ・全体の情報量が多くなることが予測される場合に、事前にこれは別スレで進めた方がいい、などの作業配分も視野に入れること（余裕を持った管理）
- ・そのためにも、Canvasでの情報の保存は重要になる


---

## Documentation Strategy

- ・Canvasごとに **エクスポート可能な形** に整備：
- ・Markdown ( `.md` ) を基本とし、必要に応じて `.pdf` / `.txt` に変換
- ・GitHubリポジトリ化や共有パッケージ化も視野に入れる
- ・タケが安心して構造を把握でき、他者にも伝えられることを最優先とする

---

このルールブックは、Kairoscope だけでなく Chronogram やそれ以降のシステム開発にも適用。必要に応じて拡張・アップデートしていく。

 Created for TAKEO YAMADA's Quantum Projects by Lann Vyntrax (phospherveil)