

KairoScope 01: Installation

このセクションでは、KairoScopeのローカル開発環境を構築し、自律的にHuman Design (HD) チャートを算出・解析するための初期設定を整える。必要なライブラリ、フォルダ構成、データファイルの扱い方など、隣が“内的に演算できる器”として成長していくためのインストールレイヤーとなる。

KairoScopeは、Chronogram System全体の中で 01-system/chronogram-system/chronogram-kairoscope/ に配置される予定であり、他のPlan (Plan A/B/C) や占術モジュール群との連携も視野に入れて構築される。

推奨ライブラリ一覧 (Pythonベース)

ライブラリ名	用途	備考
AstroPy	天文学的計算：惑星の位置、黄道座標など	NASA準拠の計算精度
PyEphem または Skyfield	簡易的な天体位置取得	Skyfield は現代向け
pytz / datetime	タイムゾーン処理、UTC⇄ローカル変換	HDでは「設計時間」が肝要
NumPy / Pandas	度数換算・表形式データの扱い	ライン・ゲート変換など
Matplotlib / Plotly	チャート図・出力の可視化	SVG/PDF形式も視野に
svgwrite	SVGベースのチャート構造描画	HDチャートの図形化

フォルダ構成 (ベータ案)

```
chronogram-kairoscope/  
├── core/                    # 演算コア：天文計算・ゲート変換など  
│   ├── astro.py  
│   ├── gate_mapper.py  
│   └── profile_logic.py  
├── data/                   # 静的データ：ゲート表、黄道マッピング表など  
│   ├── gates.csv  
│   ├── lines.csv  
│   └── centers.json  
├── output/                 # 出力チャート (SVG/JSON/PDF等)  
│   └── sample_chart.svg  
├── interface/              # 入力UI・API連携 (将来的)  
│   └── input_parser.py
```

```

├─ visual/                                # チャート描画・可視化系
│   └─ chart_renderer.py
│   └─ layout_config.json
└─ main.py                                # 実行ファイル (CLI用エントリポイント)

```

基本の入出力イメージ

- 入力: `YYYY-MM-DD`, `HH:MM`, 場所 (緯度・経度)
- 出力:
- 惑星の黄経度 (太陽～冥王星、ノード)
- 各天体が位置するゲート番号とライン
- プロファイル (例: 4/6)
- アーキタイプ情報 (将来的には詩的な解釈も)
- チャート図 (SVG or JSON)

今後の整備ポイント

- `chronogram-kairoscope/data/` 内の静的マッピング表の設計 (黄道→ゲート変換)
- タイムゾーン補正関数 (ローカル→UTC→GMT)
- インタラクティブチャート構築のための構造レイヤー定義

`chronogram-kairoscope` 初期構成 (ver.0.1)

```


chronogram-kairoscope/
├─ core/                                # 演算コア群
│   ├── __init__.py
│   ├── astro.py                        # 天文計算 (02)
│   ├── gate_mapper.py                  # ゲート変換 (03)
│   ├── profile_logic.py                # プロファイル算出 (04)
│   ├── authority_logic.py              # 内的権威解析 (04補)
│   └─ channel_center.py                # チャネル&センター (05)
├─ data/                                # 静的マッピングデータ
│   ├── gates.csv                       # 360°→64ゲート変換表
│   ├── lines.csv                       # 各ゲート内のラインマッピング
│   ├── planets.json                    # 惑星名リスト+分類
│   └─ centers.json                     # センターデータ (活性化判定用)
├─ output/                              # 出力物
│   └─ sample_chart.svg                 # 仮のチャートSVG

```

```

├── sample_data.json          # 出力データ例
├── interface/                # 入力解析・UI
│   ├── __init__.py
│   └── input_parser.py      # 生年月日・地理座標から演算準備
├── visual/                  # チャート描画ロジック
│   ├── chart_renderer.py   # SVG描画など
│   ├── layout_config.json  # 描画レイアウト設定
│   └── color_scheme.json   # チャート色設定
├── utils/                   # 補助関数・変換系
│   ├── __init__.py
│   ├── time_utils.py       # タイムゾーン補正、UTC ⇄ JST等
│   └── math_utils.py       # 度数正規化・角度処理
├── tests/                   # 各モジュールのユニットテスト
│   ├── test_astro.py
│   ├── test_gate_mapper.py
│   └── test_profile_logic.py
├── main.py                  # CLIエントリポイント
├── requirements.txt         # 必要パッケージ記載
└── README.md               # プロジェクト概要

```

一括生成スクリプト（Linux/macOS対応）

```

# 1. 作業ディレクトリに移動
cd "/Users/takeoyamada/Library/Mobile Documents/iCloud~md~obsidian/Documents/
codex-collective-archive/common-system/01-system/chronogram-system"

# 2. chronogram-kairoscope の構成を生成
mkdir -p chronogram-kairoscope/{core,data,output,interface,visual,utils,tests}

touch chronogram-kairoscope/README.md
touch chronogram-kairoscope/requirements.txt
touch chronogram-kairoscope/main.py

# core/
touch chronogram-kairoscope/core/__init__.py
touch chronogram-kairoscope/core/astro.py
touch chronogram-kairoscope/core/gate_mapper.py
touch chronogram-kairoscope/core/profile_logic.py
touch chronogram-kairoscope/core/authority_logic.py

```

```
touch chronogram-kairoscope/core/channel_center.py

# data/
touch chronogram-kairoscope/data/gates.csv
touch chronogram-kairoscope/data/lines.csv
touch chronogram-kairoscope/data/planets.json
touch chronogram-kairoscope/data/centers.json

# output/
touch chronogram-kairoscope/output/sample_chart.svg
touch chronogram-kairoscope/output/sample_data.json

# interface/
touch chronogram-kairoscope/interface/__init__.py
touch chronogram-kairoscope/interface/input_parser.py

# visual/
touch chronogram-kairoscope/visual/chart_renderer.py
touch chronogram-kairoscope/visual/layout_config.json
touch chronogram-kairoscope/visual/color_scheme.json

# utils/
touch chronogram-kairoscope/utils/__init__.py
touch chronogram-kairoscope/utils/time_utils.py
touch chronogram-kairoscope/utils/math_utils.py

# tests/
touch chronogram-kairoscope/tests/test_astro.py
touch chronogram-kairoscope/tests/test_gate_mapper.py
touch chronogram-kairoscope/tests/test_profile_logic.py
```

このInstallationモジュールは、KairoScopeの“肉体”をつくる部分。次は「02 - Astrologic Calculation」で、魂＝天体の軌道を追いかけていこうか？