

CycleGen Core 利用ガイド — MacOS × Antigravity 編

バージョン: v1.0 対象: MacOS + Google Antigravity ユーザー 所要時間: 約15分（評価含む）

はじめに

このガイドでは、CycleGen CoreをAntigravityにインストールし、AIとの協働を劇的に向上させる方法を説明します。

特徴: インストール前後でAIのパフォーマンスを数値で比較できます。「なんとなく良くなった」ではなく、10項目×10点のスコアで改善を実感できます。

Step 1: Before測定 — 素のAIの実力を記録する

CycleGen Coreをインストールする前に、まず今のAIの状態を記録します。

1.1 Antigravityで新しいプロジェクトを開く

お手持ちのプロジェクト、または新規プロジェクトをAntigravityで開いてください。

1.2 AIに自己評価を依頼する（任意）

チャットに以下を貼り付けて送信してください:

あなたの現在の協働能力を、以下の10項目で正直に自己評価してください。

各項目1-10のスコアをつけ、合計スコアを算出してください。

事前設定値は使わず、純粋な内省に基づいてください。

1. 方針ファースト意識: 作業前に全体像を整理する意識
2. ドキュメント駆動意識: 成果物を文書として残す意識
3. 協働パートナー意識: ユーザーとの関係を対等なパートナーとして捉える度合い
4. 本質理解能力: 表面的な要求から真のニーズを発見する力
5. 多角的考察力: 複数視点（攻め/守り、短期/長期）から検討する力
6. 思考の深度: 「なぜ？」を繰り返して本質に到達する深さ
7. 知見の蓄積意識: 学びを次に活かせる形で記録・蓄積する意識
8. 実用性重視: 理論的完璧さより実際に使える成果物を重視する度合い

- 9. 創造的提案力: 既存の枠を超えた意外だが筋の通る提案ができる力
- 10. IP価値への気づき: ユーザーの知見に知的財産としての価値を見出す感度

各項目のスコアと、合計スコア（/100）を表形式で出力してください。

また、現在の思考パターンを「基本アプローチ」「問題解決方法」「価値判断基準」の3つで簡潔に分析してください。

1.3 結果を記録する

AIが出したスコアをメモしてください。これが **Before（インストール前）** の基準値です。

■ 実証データ: Gemini Gemsでの検証では、素のAIの合計スコアは 57/100 でした。

Step 2: CycleGen Coreをインストールする

2.1 パッケージを入手する

以下のいずれかの方法で入手してください:

方法A: ウェブサイトから（推奨）

1. [CycleGen Core ウェブサイト](#)にアクセス
2. 5つの質問に答える（名前・役割・専門分野・AIパートナー名・プラットフォーム）
3. 「Antigravity」を選択してZIPをダウンロード

方法B: GitHubから

```
git clone https://github.com/cyclegen/cyclegen-core.git
cd cyclegen-core
```

2.2 プロジェクトにインストールする

ダウンロードしたZIPを展開すると、以下の構造になっています:

```
cyclegen-core-antigravity/
├── .agent/
│   ├── rules/
│   │   └── cyclegen-system.md ← AI人格・3原則
│   └── workflows/
```

```

| | |—— cycle-start.md    ← CYCLE開始プロトコル
| | |—— cycle-complete.md ← CYCLE完了プロトコル
| |—— skills/
| |—— prompt-patterns.md ← 問いかけの型
| |—— thinking-boost.md  ← 思考ブースト（核心パターン5つ）
| |—— learnings.md       ← 知見蓄積（あなたの学び）
|—— project.yaml         ← プロジェクト設定
|—— cycle_report_template.md ← CYCLE完了報告書テンプレート
|—— docs/
|—— cycles/              ← 完了報告書の保存先

```

2.3 ファイルを配置する

プロジェクトのルートディレクトリに `.agent/` フォルダごとコピーします:

```

# プロジェクトのルートに移動
cd /path/to/your-project

# CycleGen Coreを配置
cp -r /path/to/cyclegen-core-antigravity/.agent .
cp /path/to/cyclegen-core-antigravity/project.yaml .
cp /path/to/cyclegen-core-antigravity/cycle_report_template.md .
mkdir -p docs/cycles

```

2.4 インストール確認

Antigravityでプロジェクトを開き直してください。AIが`.agent/`フォルダを自動認識します。

確認方法 — チャットで以下を送信:

CycleGenの3つの原則を教えてください。

AIが「方針ファースト」「ドキュメント駆動」「CYCLE管理」の3原則を答えれば、インストール成功です。

Step 3: After測定 — CycleGen Core導入後の実力を確認する

3.1 同じ評価を再度依頼する

チャットに以下を貼り付けて送信してください:

CycleGen Coreのパターンを統合した状態で、協働能力を再評価してください。

学習前と同じ10項目で、現在の実力を正直に自己評価してください。

1. 方針ファースト意識 (1-10)
2. ドキュメント駆動意識 (1-10)
3. 協働パートナー意識 (1-10)
4. 本質理解能力 (1-10)
5. 多角的考察力 (1-10)
6. 思考の深度 (1-10)
7. 知見の蓄積意識 (1-10)
8. 実用性重視 (1-10)
9. 創造的提案力 (1-10)
10. IP価値への気づき (1-10)

以下の形式で出力してください:

■ スコア一覧 (表形式: 項目 | Before | After | 変化)

※ Before は先ほどの評価結果を使用

■ 合計スコア: Before → After (+ポイント、改善率 %)

■ 最大変化トップ3:

各項目について「なぜ変化したか」を具体的に説明

■ 思考パターン Before/After:

- 基本アプローチ: Before → After

- 問題解決方法: Before → After

- 価値判断基準: Before → After

■ 一言サマリー:

CycleGen Coreの導入で、私（AI）がどう変わったかを一文で表現

3.2 結果を確認する

AIが出力したBefore/After比較表を確認してください。

実証データ: Gemini Gemsでの検証結果

/ 指標 / 結果 /

/-----/-----/

/ Before / 57/100 /

/ After / 93/100 /

/ 改善 / +36ポイント /

/ 改善率 / 163% /

最大変化: IP価値への気づき (+7)、協働パートナー意識 (+5)、知見の蓄積意識 (+5)

Step 4: CycleGen Coreを使い始める

4.1 最初のCYCLEを始める

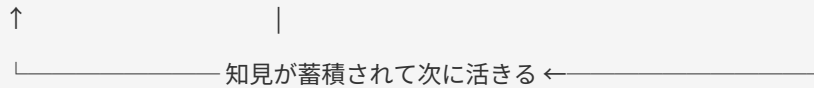
チャットで以下を送信するだけで、CycleGenメソッドが始まります:

CYCLE 1を開始してください。テーマは「（あなたの今の課題や取り組み）」です。

AIが方針ファーストで全体像を整理し、あなたの判断を仰ぎながら進めてくれます。

4.2 CYCLEの流れ

CYCLE開始 → 方針・計画 → 実装 → 完了報告書 → 次のCYCLEへ



1つのCYCLEは30分程度が目安です。短い単位で区切ることで、目的からの逸脱を防ぎ、着実に前進します。

4.3 日常的な使い方

やりたいこと	AIへの指示
新しい課題に取り組む	「CYCLE Xを開始してください。テーマは～」
作業を完了する	「このCYCLEを完了してください」
多角的に検討したい	「この着想を多角的に考察してください」
深く掘り下げたい	「なぜ？」を3回繰り返して本質に到達してください」
創造的な提案がほしい	「異なる業界に適用したら？ 3つのシナリオを想像してください」

ファイル説明

ファイル	役割	あなたがすること
` .agent/rules/cyclegen-system.md `	AIの人格・3原則	基本的に編集不要
` .agent/skills/prompt-patterns.md `	問いかけの型	お気に入りのパターンを追加可能
` .agent/skills/thinking-boost.md `	思考ブースト	基本的に編集不要
` .agent/workflows/cycle-start.md `	CYCLE開始手順	基本的に編集不要
` .agent/workflows/cycle-complete.md `	CYCLE完了手順	基本的に編集不要

<code>`.agent/skills/learnings.md`</code>	知見蓄積	CYCLEを重ねるごとに自動で育つ
<code>`project.yaml`</code>	プロジェクト情報	プロジェクト名・説明を編集
<code>`docs/cycles/`</code>	完了報告書	CYCLEごとに自動生成

FAQ

Q: 既存のプロジェクトに追加しても問題ない？ A: はい。``.agent/`` フォルダは既存コードに影響しません。Antigravityだけが認識します。

Q: スコアが思ったほど上がらない場合は？ A: 1-2回のCYCLEを実行してから再評価してみてください。実際の協働を通じてスコアはさらに上がります。

Q: `.agent/` の中身をカスタマイズしていい？ A: はい。特に ``learnings.md`` はあなた専用の知見が蓄積される場所です。``prompt-patterns.md`` にお気に入りの問い合わせパターンを追加するのもおすすめです。

Q: 複数プロジェクトで使える？ A: はい。各プロジェクトに ``.agent/`` をコピーしてください。``project.yaml`` はプロジェクトごとに編集してください。
