

第2回 CODEX PRACTICE

Codex 実践ドリル

— つくる。試す。直す。



説明は20分。 残り70分は、自分で進む。

90min

REVIEW 20 / PRACTICE 70



聞く時間より、
手を動かす時間を長くする。

20min

昨日の復習+4つの道具

基本ループを思い出し、
AGENTS.md / Skill / Hook / MCPの役割をつかむ。

0:00-0:20

共通の土台を
そろえる

70min

3問を、自分の地点から進む

つくる → 試す → 直す → 完成証拠を残す。条件を満たしたら次へ。

初めて 演習①

変更できる 演習②

制作経験あり 演習③

0:20-1:30

自分のペースで
手を動かす

Codexの仕事は、5つの往復で進む。

操作を覚えるより、仕事の渡し方を思い出す。
昨日行ったことを5つに戻します。

- | | | | |
|----|---------|------------------------------|-------------|
| 01 | 場所を決める | 1つの目的=1つのプロジェクト。作業するフォルダを選ぶ。 | → 迷わず始められる |
| 02 | 材料を渡す | 読む資料、使う画像、作るファイル名を同じ場所へ置く。 | → 必要な情報がそろう |
| 03 | 目的を伝える | 「何を」「誰のために」「どんな形で」を短く指示する。 | → 作業の向きが決まる |
| 04 | 実行を承認する | Codexの計画と操作を読み、進めてよい範囲を判断する。 | → 安全に任せられる |
| 05 | 結果を返す | 動かして確認し、期待と違う点を次の指示として返す。 | → 往復するほど完成へ |

任せるのは作業。決めるのは人間。

Codexに全部を丸投げしない。
判断は人間、実行はCodexで分けます。

人間が決める

DECIDE

01

目的

誰の、何をよくしたいか

02

完成条件

何ができれば合格か

03

任せる範囲

読んでよいもの、変更してよいもの

04

最終判断

本当に使えるか、他人へ渡せるか

Codexに任せる

EXECUTE

01

材料を読む

資料や既存ファイルから状況を把握する

02

進め方を考える

作業を分解し、計画と選択肢を出す

03

成果物をつくる

ファイル作成、修正、整理を実行する

04

検証して報告する

条件を試し、失敗と未確認点を返す

判断 → 作業を渡す

失敗を返すたび、成果物は**完成**へ近づく。

1回動いたら終わりではありません。
条件を変えて試すところまでが作業です。



POINT

**1回 動いた だけでは
「完成」 ではない。**

空欄・長い文字・別の人への操作。
いくつかの条件で試して、
はじめて「使える」になります。

4つの道具は、効く場面が違う。

名前を覚えるより、「いつ使うか」で選ぶ。
約束・手順・自動確認・外部接続の4役です。

01 **AGENTS.md**
約束を固定

毎回守る仕事のルールを、プロジェクトに残す。

WHEN
仕事を始めるたび
日本語で説明／元ファイルを上書きしない

02 **Skill**
手順を固定

一度成功した進め方を、別の素材でも再現する。

WHEN
その作業をするとき
PDFから根拠付き要約を作る

03 **Hook**
タイミングを固定

決まった瞬間の確認を、自動で走らせる。

WHEN
保存・変更の直後
必須見出し／HTMLエラーを確認

04 **MCP**
接続先を固定

外部の情報やサービスへ、許可された範囲でつなぐ。

WHEN
外の情報が必要なとき
公開情報を読み、確認後に保存

AGENTS.mdは、毎回の約束を固定する。

同じお願いを何度も書かず、プロジェクトの仕事ルールとして残します。

AGENTS.md

このプロジェクトで毎回守ること

RULE 日本語で説明する

RULE 成果物は指定フォルダへ保存する

RULE 元ファイルを上書きしない

RULE 不明な情報を推測しない

RULE 完了前にチェック項目を確認する

APPLIES TO このプロジェクトの全作業
人が書き換えるまで、毎回読み直す

毎回

AGENTS.mdがない

作業のたびに、同じ注意をプロンプトへ書き直す。

- 日本語で
- 上書きしない
- 確認して

1回

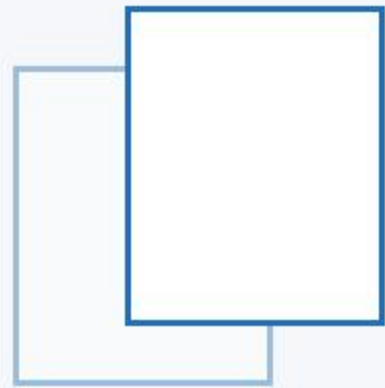
AGENTS.mdがある

一度書けば、Codexが毎回の作業で自動的に読み直す。

- 継続する
- チームで共有
- 漏れを減らす

Skillは、成功した手順を次でも再現する。

成果物そのものではなく、
うまくいった進め方を保存します。



01

入力を決める

どんな素材を受け取り、何を前提にするか。



02

手順を並べる

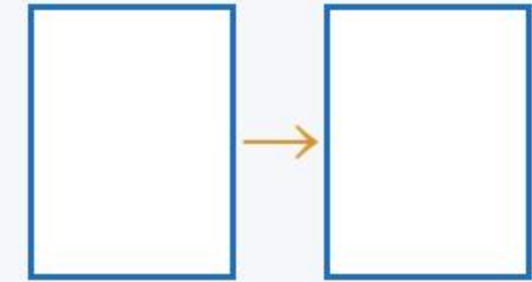
読む、抽出する、整理する、出力する。



03

確認を入れる

出典、欠落、形式など、合格条件を置く。



04

別素材で試す

違うPDFでも、同じ構造の結果になるか。

REUSE

PDF Aで成功した手順 → PDF Bでも同じ型で実行

成功を偶然で終わらせない

Hookは、忘れやすい確認を自動化する。

毎回思い出す代わりに、
決まったタイミングで確認を走らせます。



注意を、自動で戻す

人が忘れても、仕組みは忘れない。

01

作業前

必要なファイルや設定がそろっているか確認。

02

変更直後

必須見出し、ファイル名、HTMLエラーを自動検知。

03

作業終了時

チェック結果と未完了項目を記録して残す。

TRY ONCE

意図的に1回失敗させ、Hookが検知できれば完成。

MCPは、Codexを外のサービスにつなぐ。

つなぎたいサービスの「MCPサーバー」を1つ足すと、Codexがそのサービスを直接つかえるようになります。

WHAT IS MCP

MCP

Codexと外部サービスをつなぐ、共通の接続口。

つなぎ方は同じ

どのサービスも、決まった1つのやり方でつなぐ。

サーバーを足すだけ

使いたいサービスの「MCPサーバー」を追加する。

SERVICES WITH MCP

つなげる先の例

 Google ドライブ 共有フォルダの資料を読む	 Google カレンダー 空いている予定を読む
 Slack チームの会話を確認する	 Notion メモや資料を読み書きする
 GitHub コードや課題を管理する	 Figma デザインを読み取る

SAFETY FIRST	最初は「読み取り専用」から	保存・変更の前に、人が確認
--------------	---------------	---------------

固定したいものから、使う道具を選ぶ。

道具の名前から考えない。
何を変わらなくしたいかを先に問います。

? 今回、固定したいのはどれですか？

PROJECT RULE 毎回守る 仕事の約束 ↓ AGENTS.md 言語、保存先、上書き禁止、確認項目。	REUSABLE FLOW 何度も使う 仕事の手順 ↓ Skill 入力、工程、出力、合格条件の型。	AUTO TIMING 確認を走らせる 決まった瞬間 ↓ Hook 作業前、変更後、終了時の自動確認。	EXTERNAL ACCESS 外の情報へ つながる接続口 ↓ MCP サービス、情報、読み書きの権限。
--	--	---	--