

AIで何ができる？ ChatGPT・Codex入門

仕事で使えるAI活用をやさしく解説

Excel加工から始める、身近なAI活用

安田 大介

本書を読む前に

本書は、非プログラマーがCodexを仕事に使うための入門書です。画面、機能、料金、プラン、対応環境は変更されることがあります。操作前には本書執筆時点の説明だけでなく、OpenAIと各サービスの最新の公式案内も確認してください。

AIの出力には間違いが含まれることがあります。元ファイルを残し、完成物を実際に開き、件数、数字、表示、動作を人間が確認してください。本番環境の変更、外部送信、公開、購入、権限変更は、内容を確認したうえで実行してください。

目次

最終章は実例集です。

先に「実際に何が完成したのか」を見たい方は、第27章から読んでも構いません。

第1部 まずCodexを使える状態にする

- 第1章 ChatGPTとCodexは何が違う？
- 第2章 Codexを使う準備をする
- 第3章 難しいプロンプトは必要ない

第2部 Excel・CSVの仕事を楽しむ

- 第4章 どうしてもうまくいかなかったExcelを加工する
- 第5章 複数のExcelを1つにまとめる
- 第6章 顧客リストの重複を整理する
- 第7章 CSVの文字化け・列・日付形式を直す
- 第8章 毎月同じExcel集計を自動化する
- 第9章 売上を担当者別・商品別・月別に集計する

第3部 文書の仕事を楽しむ

- 第10章 大量のWordから必要な情報を抜き出す
- 第11章 新旧文書の違いを確認する
- 第12章 毎月の定型報告書を作る
- 第13章 アンケート結果を整理・集計する

第4部 営業・企画の仕事を楽しむ

- 第14章 営業用PowerPoint提案書を作る
- 第15章 新規事業・新サービスの企画書を作る

第5部 面倒なPC作業を楽しむ

- 第16章 大量のファイル名を一括変更する
- 第17章 フォルダを顧客別・日付別に整理する
- 第18章 毎日・毎月の繰り返し作業を自動化する

第6部 ホームページとシステムを作る

- 第19章 既存ホームページを修正する
- 第20章 会社ホームページを新しく作る
- 第21章 新サービスの紹介ページをその日のうちに作る
- 第22章 Excel管理をWebシステムに変える
- 第23章 作ったシステムをサーバーで公開する

第7部 安全に使い、自分の仕事へ広げる

- 第24章 Codexを安全に使うために
- 第25章 人間とCodexの役割
- 第26章 今日からCodexに仕事を頼んでみる

第8部 実例集——「ここまでできるのか」を成果物で見る

- 第27章 実例集——「これもCodexでできるのか」の連続
 - 実例1 会社ホームページをリニューアルして公開する
 - 実例2 本書の原稿、図解、PDF、Kindle用データを作る
 - 実例3 AEO+LP——情報収集からサイト生成・確認までを仕組みにする
 - 実例4 AI透視家——社内の生成AI利用をWindows環境で見える化する
 - 実例5 WorldScene——20以上のアプリとAIツールを一つの公開サイトへまとめる

巻末

- おわりに まず一つ、仕事を頼んでみてください
- 付録 そのまま使える指示例と確認表

第1章 ChatGPTとCodexは何が違う？

本章の中心メッセージ

ChatGPTは相談するAI。Codexは仕事を頼むAI。

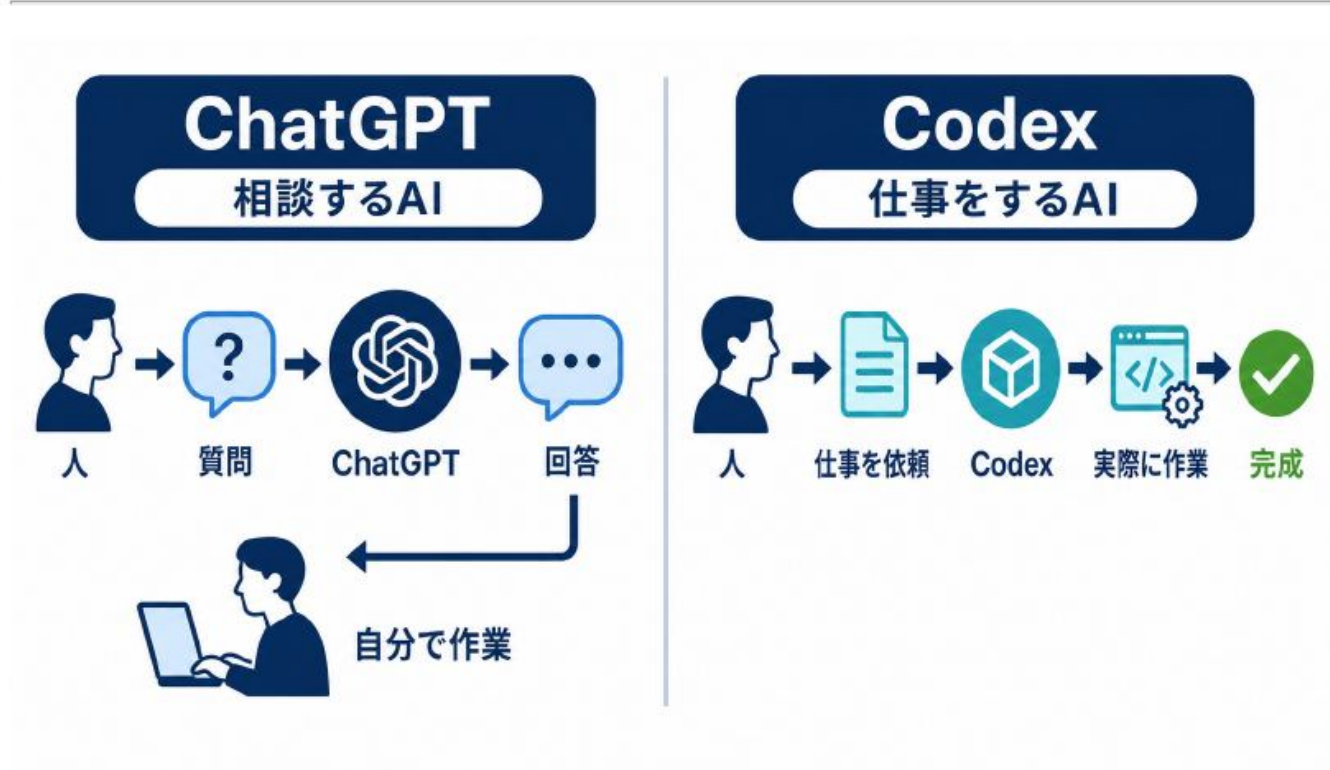


図01 ChatGPTとCodexの違い

まず図01を見てください。相談して自分で作業する流れと、Codexへ仕事を頼んで完成物を確認する流れの違いを示しています。

まず、この図を見てください

図の左側では、人間がAIに質問しています。

AIは質問に答えます。

しかし、その後の作業は人間が行っています。

図の右側では、人間がAIに仕事を頼んでいます。

AIはファイルを確認し、作成や加工を進めます。

最後に人間が見るのは、作業方法の説明だけではありません。

実際に作られたファイルや、修正された画面です。

この違いが、本書で最初に知ってほしいことです。

AIは、答えるだけでなく、実際に仕事をするようになりました。



実画面 Codexで本書の企画ファイルを編集している画面

この画面は説明用に作った架空の見本ではありません。

著者が本書の改訂方針をCodexへ伝え、Codexが企画ファイルを実際に編集している画面です。

画面上部には、変更したファイル名と変更量が表示されています。

中央には、著者からの指示とCodexの作業内容があります。

左側には、この仕事に関するプロジェクトがあります。

掲載に不要なプロジェクト名と個人名は非表示にしています。

ここで重要なのは、Codexが答えを書いただけではないことです。

実際の企画ファイルを見つけ、内容を変更し、変更結果を確認できる状態にしています。

※画面、機能名、対応環境は変わる可能性があります。掲載時には「本書執筆時点」と明記し、OpenAI公式情報で再確認します。

Codexは作業領域の中で何をするのか



補助図 Codexが作業領域で行う6つの仕事

Codexへ仕事を頼むときは、対象となるフォルダやファイルを作業領域として渡します。

Codexは、その中で必要なファイルを調べます。

文書、表、ページなどを作ります。

既存のデータやファイルを加工します。

必要な処理を実行します。

件数、表示、動作などをテストします。

問題が見つかれば修正します。

この一連の流れが、通常のチャットで方法だけを聞く場合との大きな違いです。

もちろん、すべての仕事で六つ全部を行うわけではありません。

Excel加工なら、ファイルを調べ、加工し、件数を確認します。

ホームページなら、構成を調べ、修正し、ブラウザで表示を確認します。

小さなシステムなら、必要なファイルを作り、実行し、操作して問題を直します。

人間は、Codexへ自由に何でも触らせるのではありません。

作業してよい場所と、変更してはいけない場所を決めます。

この準備は次章で詳しく扱います。

ChatGPTに方法を聞いていた

たとえば、あなたの手元に顧客一覧のExcelがあるとします。

電話番号の列を追加したい。

重複した顧客も整理したい。

日付の書き方もそろえたい。

回答ではなく、完成した仕事を受け取る



補助図 回答を受け取るか、成果物を受け取るか

違いを、もう一度整理してみましょう。

ChatGPTに相談した場合、受け取るものは主に回答です。

Codexに仕事を頼んだ場合、受け取るものは回答だけではありません。

加工済みのExcel。

整理されたCSV。

作成されたPowerPoint。

修正されたホームページ。

ブラウザで使える小さな管理システム。

このように、実際に使える形を目指して作業を進められます。

もちろん、頼んだ仕事や使用環境によって、できる範囲は変わります。

会社の権限やセキュリティ上の理由で、操作できない場合もあります。

必要なソフトや材料が足りず、途中で確認を求められることもあります。

それでも、「方法を説明してもらう」と「作業を進めてもらう」とことには、大きな違いがあります。

PowerPointも「作り方」ではなく「そのもの」を頼む

営業提案書を作る場面でも同じです。

これまでは、ChatGPTに構成案を考えてもらい、その文章をPowerPointへ貼り付けていた人も多いでしょう。

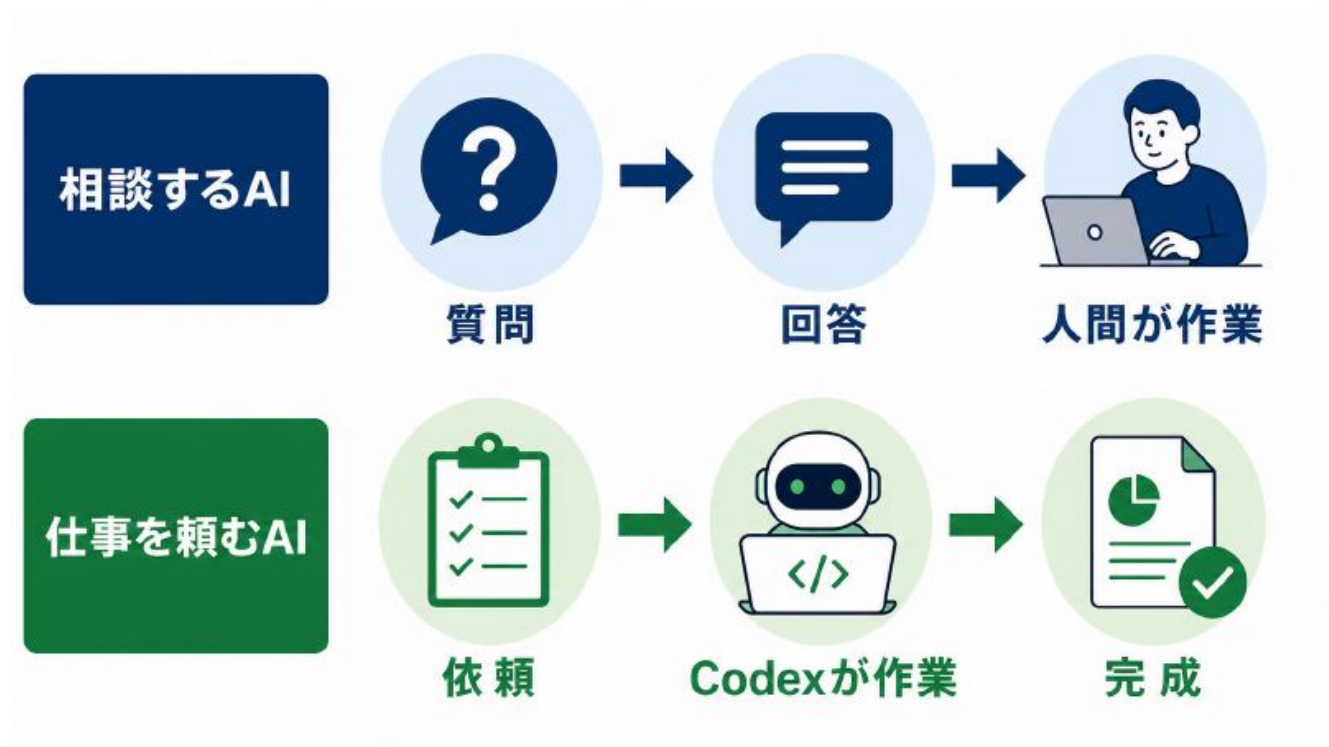
タイトルを貼る。

本文を貼る。

文字の大きさを変える。

図を置く。

「AIが答える」から「AIが作業する」へ



補助図 相談するAIと仕事を頼むAIの流れ

従来の流れは、次のようなものでした。

1. 人間が質問する
2. ChatGPTが回答する
3. 人間が回答を読む
4. 人間が実際の作業をする
5. うまくいかなければ、また質問する

Codexでは、流れが変わります。

1. 人間が仕事の目的を伝える
2. 必要なファイルや材料を渡す
3. Codexが対象を確認する
4. Codexが作成・加工・修正を進める
5. 必要に応じてテストする
6. 人間が完成物を確認する
7. 違うところがあれば修正を頼む

人間が何もしなくなるわけではありません。

人間の仕事の場所が変わります。

細かな手作業を続けるのではなく、何を作るかを決め、完成したものが正しいかを判断します。

この変化を一言で表すと、次のようになります。

「AIに相談する」から「AIに仕事を頼む」へ。

Excelを加工する。

ホームページを作る。

サービス紹介ページを作る。

小さなシステムを作る。

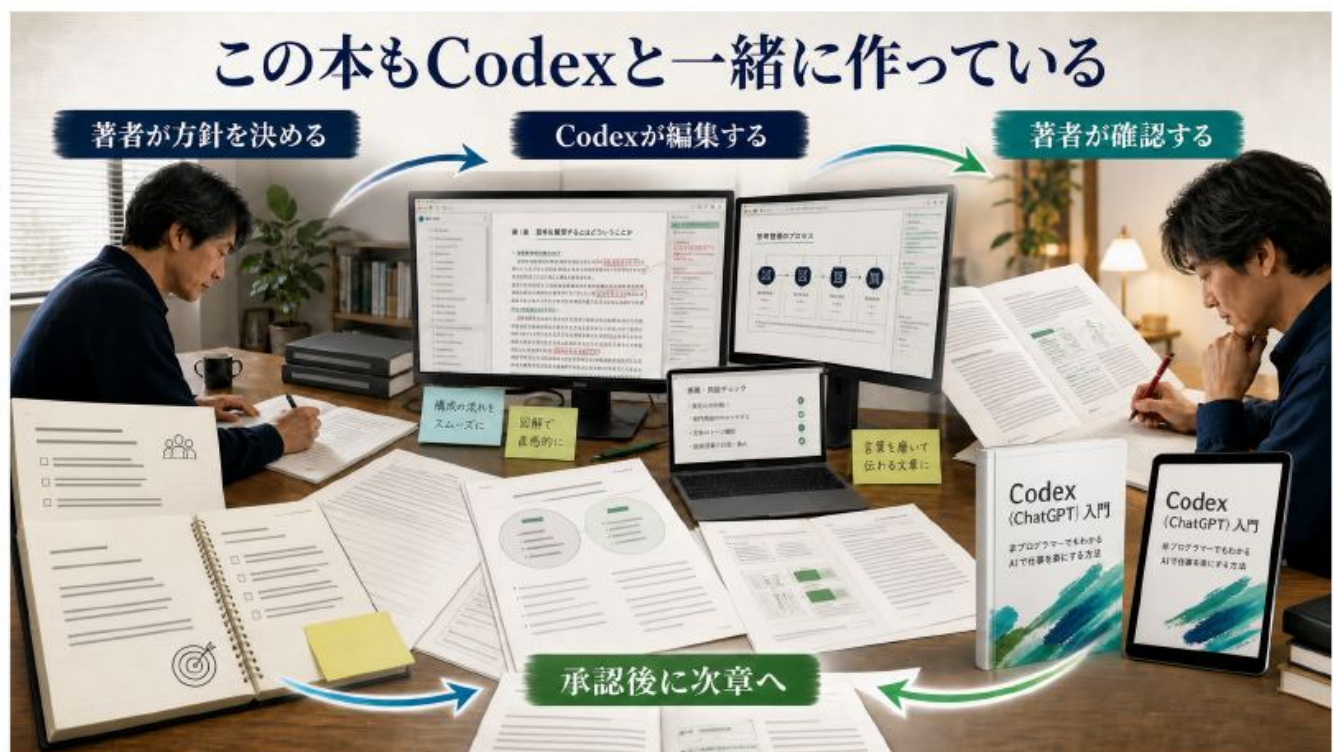
公開に必要な作業を進める。

それまで技術者や制作会社へ頼まなければ動かなかった仕事を、自分から進められるようになったことが大きな変化でした。

ただし、次の情報は実話として正確に掲載するため、著者確認後に追記します。

数字や出来事を推測で補うことはしません。

本書制作そのものが実例になっている



補助図 この本もCodexと一緒に作っている

本書の制作でも、人間とCodexの役割を分けています。

著者は、誰に読んでほしいか、何を伝えたいか、どの部分を直すかを決めます。

Codexは、原稿、図解、画像、出版用データを確認し、依頼された変更を進めます。

著者は、変更後の内容と図を見て、読みやすいか、実務に役立つか、事実と違っていないかを判断します。

承認前に次章を大量に変更しないのも、この役割分担の一つです。

このように、本書で説明している仕事の進め方は、本書自身の制作にも使われています。

本書で扱うのは「コード」ではなく「終わった仕事」

Codexが仕事を進める途中で、プログラムを使うことがあります。

しかし、本書では長いコードを主役にしません。

読者に覚えてほしいのは、プログラムの書き方ではないからです。

第2章 Codexを使う準備をする

本章の中心メッセージ

AIへ質問する準備ではありません。AIへ実際の仕事を渡す準備です。



図02-1 最初の仕事を頼むまでの5段階

準備で一番大切なのは、作業範囲を決めること

Codexを使い始めるために、プログラミングを勉強する必要はありません。

最初に行うのは、次の5つです。

1. Windows版のアプリを入手する
2. ChatGPTのアカウントでサインインする
3. 仕事ごとにプロジェクトを作る
4. Codexが扱う作業フォルダを決める
5. 元ファイルのコピーで小さく試す

難しそうに見える設定を、最初から全部覚える必要はありません。

本章では、初心者が最初のファイルを安全に確認できるところまで進みます。

画面の名称や配置は変わる可能性があります。本書の画面説明は、本書執筆時点のものです。表示が少し違う場合でも、「アプリを開く」「仕事を分ける」「作業場所を決める」という目的は同じです。

サインインの流れ



図02-2 起動からサインインまで

Windows版を入手してサインインする

本書では、Windowsのデスクトップアプリを使う方法を基本にします。

OpenAIの公式案内からWindows版を入手し、アプリを起動します。初回は「Continue」などの案内からサインインへ進みます。ブラウザが開いたら、普段使っているChatGPTのアカウントで認証します。

本書ではAPIキーを使う方法を前提にしません。

初心者が仕事で使い始めるなら、まずChatGPTのアカウントでサインインする流れが分かりやすいからです。APIキーは、外部の仕組みから自動実行する場合などに使われる別の方法です。ここで覚える必要はありません。サインイン後は、プロフィール欄で使っているアカウントを確認します。会社のアカウントと個人のアカウントを使い分けている人は、ここを見落とさないでください。

会社の資料を扱うときは、会社の利用規則も確認します。利用できる機能や権限は、契約や会社側の設定によって異なる場合があります。

ここで確認すること

- 公式の案内から入手したか
- 使用するアカウントは正しいか
- 会社の資料を扱ってよい環境か
- 画面に自分のアカウントが表示されているか

サインインできれば、次は仕事を置く場所を作ります。

仕事ごとにプロジェクトを分ける



図02-3 プロジェクトは仕事ごとの作業机

プロジェクトは、仕事ごとに分ける

プロジェクトという言葉が出てきます。

難しく考えなくて構いません。

プロジェクトは、仕事ごとの作業机です。

会社案内を直す仕事、売上を集計する仕事、ホームページを作る仕事。それぞれ別の机に分けるイメージです。同じプロジェクトの中では、関係するファイルと会話をまとめて扱いやすくなります。一方、関係のない仕事を同じ場所へ入れると、どの資料を使うのか分かりにくくなります。最初のプロジェクト名は、仕事の結果が分かる名前にします。

- 「AI練習」ではなく「2026年8月売上集計」
- 「テスト」ではなく「会社案内の文章修正」
- 「新規」ではなく「顧客台帳の重複整理」

後から見たときに、何を完成させる場所なのか分かる名前が便利です。

ひとつのプロジェクトへ、すべての会社資料を入れる必要はありません。その仕事に必要なものだけを置きます。

たとえば売上集計なら、次のように絞れます。

- 今月の売上データ
- 集計条件を書いたメモ
- 参考にする先月の完成ファイル

顧客名簿、採用資料、契約書など、関係のないものは入れません。

仕事を分けることは、整理のためだけではありません。Codexが触れる範囲を小さくする、安全対策でもあります。



図02-4 元ファイルと作業用コピー

最初は元ファイルを渡さない

初めてExcelを加工してもらったとき、会社で使っている原本をそのまま渡すのは避けます。

まず、原本をコピーします。

そして、Codex専用の作業フォルダを作ります。

たとえば、次のように分けます。

```
売上集計_Codex作業用
├─ 01_入力
├─ 02_作業中
└─ 03_完成
```

「01_入力」へ、元ファイルのコピーを入れます。

「02_作業中」は、途中のファイルを置く場所です。

「03_完成」は、人間が確認する完成候補を置く場所です。

元の保管場所は、Codexの作業対象から外しておきます。

この分け方なら、途中で思った結果にならなくても、原本へ戻れます。

元ファイルを守ったまま、小さく試す。

これが、Codexを仕事へ入れる最初の基本です。

ファイル名も分かりやすくする

コピーしたファイルには、作業用だと分かる名前を付けます。

例：

- 売上原本.xlsx
- 売上原本_Codex作業用.xlsx
- 売上集計_確認前.xlsx
- 売上集計_確認済み.xlsx

「最終」「最新版」だけでは、時間がたつと区別できなくなります。日付や状態を入れると、間違いを減らせます。

個人情報は必要最小限にする

練習のために本物の顧客情報を使う必要はありません。

最初は、氏名、電話番号、メールアドレスなどを架空の内容へ置き換えたサンプルを使います。会社の規則で外部サービスへの入力禁止されている情報は、入力しません。



図02-5 権限は小さく始める

「承認を求める」設定から始める

Codexは、ファイルを読む、変更する、処理を実行するなど、パソコンの中で作業します。

そのため、「どこまで自分で進めてよいか」を決める権限があります。

初心者は、**「承認を求める」**設定から始めるのが安全です。

この設定では、Codexは現在の作業範囲内で仕事を進め、範囲を越える必要があるときに止まって確認を求めます。

確認画面が出たら、反射的に許可しないでください。

次の3点を読みます。

1. 何をしようとしているか
2. どのファイルや場所が対象か
3. なぜ、その操作が必要なのか

分からない場合は、許可する前にCodexへ聞けます。

この操作を許可すると、どのファイルが変わりますか。
初心者にも分かるように説明してください。
より狭い範囲で進める方法があれば、そちらを使ってください。

広い権限が悪いわけではない

広い権限は、毎回の確認を減らせるため便利です。

ただし、触れる場所と実行できる操作も増えます。仕事の流れと影響範囲を理解してから使う設定です。

最初の目的は、確認画面をなくすことではありません。

何が起きるかを理解しながら、一つの作業用フォルダで成功させることです。

最初の依頼は変更ではなく確認



図02-6 最初に変更せず確認する

最初の仕事は、ファイルを確認してもらう

準備ができれば、いきなり加工させる必要はありません。

まず、作業用フォルダの中を確認してもらいます。

次の指示を、そのまま使えます。

第3章 難しいプロンプトは必要ない



図03-1 Codexへ伝える4つの要素

この図で最初に見てほしいのは、難しい言葉が一つもないことです。

Codexへ仕事を頼むときは、次の4点を伝えれば始められます。

1. 何に困っているか
2. 何をを使うか
3. どうしてほしいか
4. してはいけないこと

たとえば、毎月の売上集計に時間がかかっているなら、次のように頼めます。

毎月の売上集計に時間がかかっています。
このフォルダにあるExcelを使ってください。
担当者別と商品別に集計してください。
元ファイルは変更しないでください。

これで十分です。

「プロンプト」とは、AIへ入力する質問や指示のことです。この言葉を聞くと、特別な書き方を覚えなければいけないように感じるかもしれません。

しかし、Codexを使う目的は、上手な命令文を書くことではありません。仕事を終わらせることです。

英語で書く必要はありません。プログラムの名前を知っている必要もありません。普段、同僚へ仕事を頼むときの言葉で構いません。

1. 何に困っているか

最初に、今の状態を伝えます。

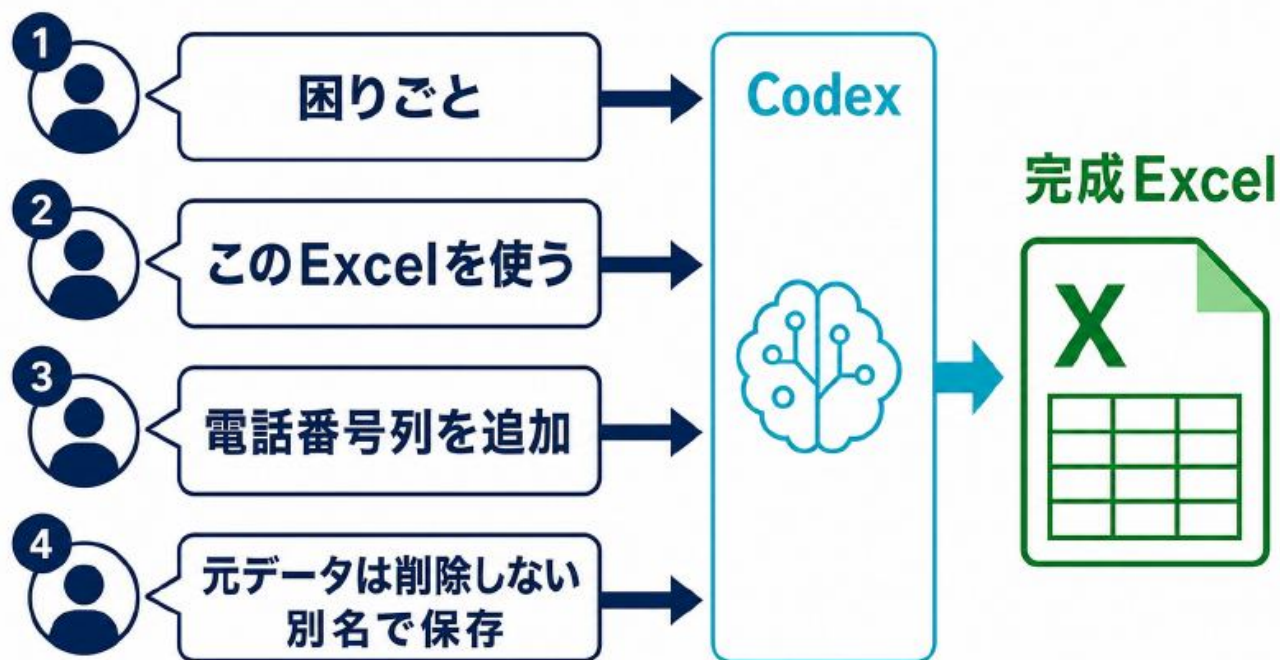


図03-2 短い依頼から完成Excelまで

図の左側は、人間が渡す情報です。右側は、Codexが作る成果物です。

人間が関数や操作手順をすべて決める必要はありません。

Excel加工を頼む例

このExcelに電話番号の列を追加してください。
顧客マスタ.xlsxにある会社名と照合してください。
元データは削除しないでください。
完成したファイルは「顧客一覧_電話番号追加.xlsx」という別名で保存してください。
保存後、追加できた件数と追加できなかった件数を報告してください。

この依頼には、材料、完成形、禁止事項、確認方法が入っています。

ここで大切なのは、Excelの操作方法を教えてもらうのではなく、Excelそのものを加工してもらうことです。

Wordから一覧を作る例

このフォルダにあるWord文書から、会社名、契約日、金額を抜き出してください。
一覧はExcelで作り、1文書を1行にしてください。
読み取れない項目は空欄にせず「要確認」と入れてください。
元のWord文書は変更しないでください。

PowerPointを作る例

顧客情報、商品資料、料金表を使って営業提案書を作ってください。
顧客が判断しやすいように、課題、提案、導入効果、料金、次の行動の順にしてください。
確認できない数字は作らず「要確認」と表示してください。
完成したPowerPointは別名で保存してください。

ホームページ修正を頼む例

このホームページの会社概要ページを修正してください。
新しい文章は「会社概要_確定.docx」を使ってください。
ほかのページは変更しないでください。
いきなり公開せず、確認用の画面を作ってください。

どの仕事でも基本は同じです。

材料を渡し、完成してほしい状態を伝え、触ってはいけない範囲を決めます。

分からないときは、先に中身を確認する



図03-3 分からないときは、先に中身を確認してもらう

仕事によっては、何を頼めばよいかわからないことがあります。

列の意味が分からない。古いデータと新しいデータが混ざっている。どこから直せばよいかわからない。
その場合は、無理に完成形を決めなくて構いません。最初の仕事を「確認」にします。

このExcelを確認してください。
まだ変更はしないでください。
シート名、列名、データ件数、空欄、重複、形式が揃っていない箇所を報告してください。
その後、直し方の候補を分かりやすく説明してください。

Codexから報告を受けたら、人間が次の作業を決めます。

重複候補は削除せず、別シートに一覧化してください。
日付形式だけを統一してください。
まず10行の見本を作ってください。

このように、1回の長い指示ですべてを終わらせる必要はありません。
確認する。見本を作る。人間が見る。必要な修正を頼む。
会話を続けながら、完成に近づければよいのです。

「分からない」と伝えることも立派な指示

次のような頼み方でも構いません。

何が問題か自分では分かりません。
ファイルを変更せずに確認し、問題になりそうな点を教えてください。

方法を知らなくても、困っている状態は説明できます。Codexの価値は、決められた作業を繰り返すことだけではありません。材料を調べ、次に何をすべきか整理できることにもあります。

ただし、Codexが提案した方法を、そのまま正解だと思わないでください。業務上の意味や社内ルールは、人間が確認します。



図03-4 大量処理は、小さく試してから広げる

100件、1000件のファイルを処理するとき、最初から全件を任せるのは避けます。
まず5件だけで試します。

第4章 どうしてもうまくいかなかったExcelを加工する

事例01

ChatGPTに方法を聞くのではなく、ExcelそのものをCodexへ渡す。

本章のファイル名、列名、処理内容は、仕事の流れを説明するための例です。著者の実体験として時間、件数、エラー、費用を推測して記載していません。

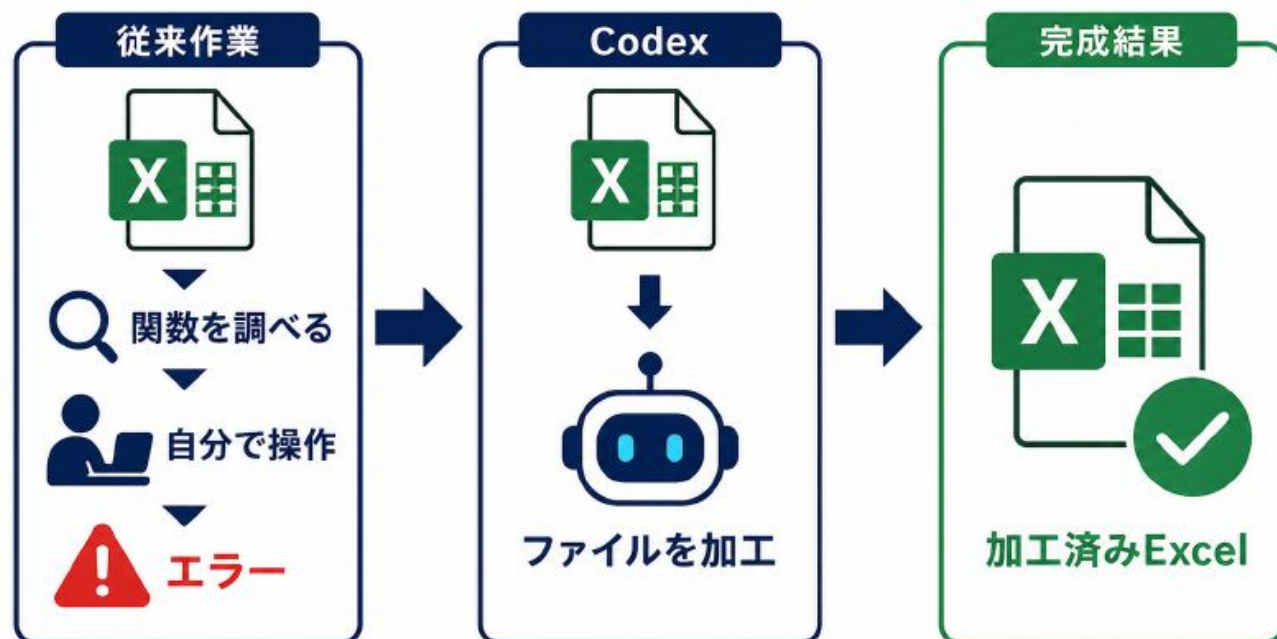


図04-1 方法を聞く仕事から、完成ファイルを受け取る仕事へ

Excelで困ったとき、これまでは関数や操作方法を調べていました。

ChatGPTに質問すれば、使い方を説明してもらえます。しかし、回答を読んだ後にExcelを操作するのは人間です。

数式をコピーする。自分の列に合わせて修正する。エラーの意味を調べる。別の方法を試す。

人間が欲しかったのは、Excelの勉強ではありません。加工が終わったExcelです。

Codexでは、加工したいExcelそのものを渡せます。そして「何を直し、どの状態で保存してほしいか」を頼めます。

知りたかったのは方法ではなく、終わらせたかったのは仕事です。

こんなことで困っていませんか？

会社では、顧客一覧、売上管理、在庫表、見積一覧、問い合わせ記録など、多くの仕事にExcelを使います。

少し加工したいだけなのに、急に難しくなることがあります。

- 別の表から電話番号を入れたい
- 日付の書き方を揃えたい

- 重複している行を整理したい
- 空欄だけを一覧にしたい
- 担当者別に並べ替えたい
- 金額を集計できる形式へ直したい

言葉にすると簡単です。しかし、どの関数を使うのか、どのセルを選ぶのか、途中で何を変えたのかが分からなくなります。

最悪の場合、元のExcelまで上書きしてしまいます。

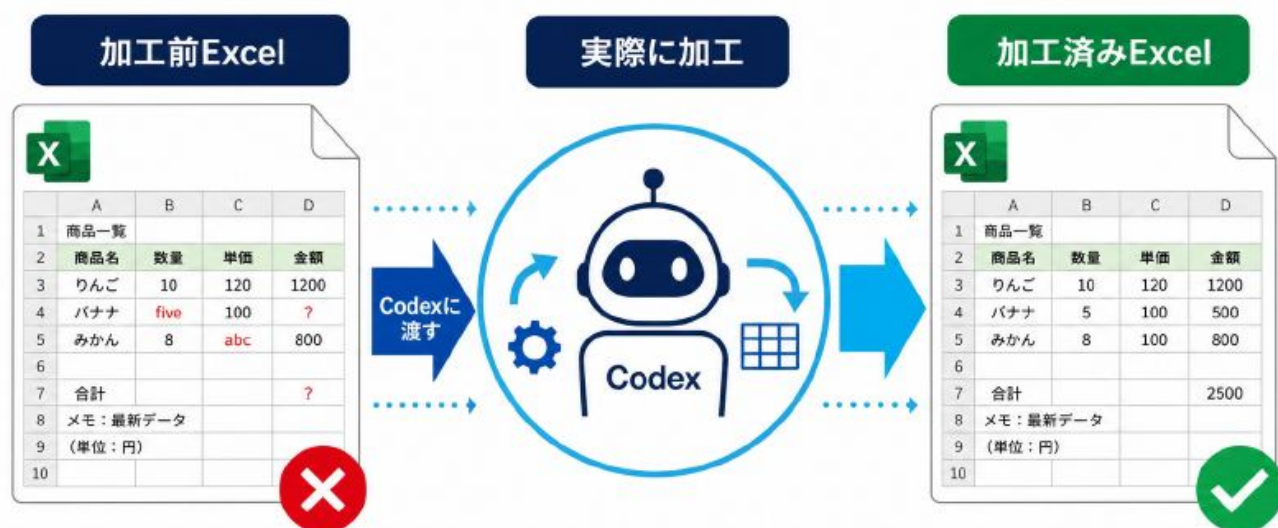


図04-2 Excelそのものを加工してもらう

この図の左側には、文字と数字が混ざり、合計が出せないExcelがあります。

中央ではCodexがファイルの中身を確認し、データを直します。右側には、計算できる状態になったExcelがあります。

ここで重要なのは、関数名を指定していないことです。

人間は、完成してほしい状態を伝えます。

Codexにはこう頼む

このExcelを確認し、数量と単価を計算できる数字へ統一してください。

金額列は「数量×単価」で計算してください。

数字へ変換できない値は推測で直さず、黄色で目印を付けてください。

元ファイルは変更しないでください。

完成したファイルは「商品一覧_加工済み.xlsx」という別名で保存してください。

完了後、元の行数、修正した行数、確認が必要な行数を報告してください。

依頼に入っているのは、次の5点です。

1. 使うファイル

2. 完成してほしい状態
3. 判断できない場合の扱い
4. 元ファイルを残すこと
5. 完成後の確認方法

「どの関数を使うか」はCodexが考えます。

ただし、現在の書式や数式を残したい場合は、その条件を追加します。

現在の色、罫線、列幅、数式をできるだけ残してください。
変更した場所があれば、最後に一覧で報告してください。

重要なファイルなら、最初に変更せず確認だけを頼みます。

まだ加工しないでください。
シート名、列名、データ件数、空欄、数式、非表示の行や列を確認してください。
加工前に注意すべき点を教えてください。

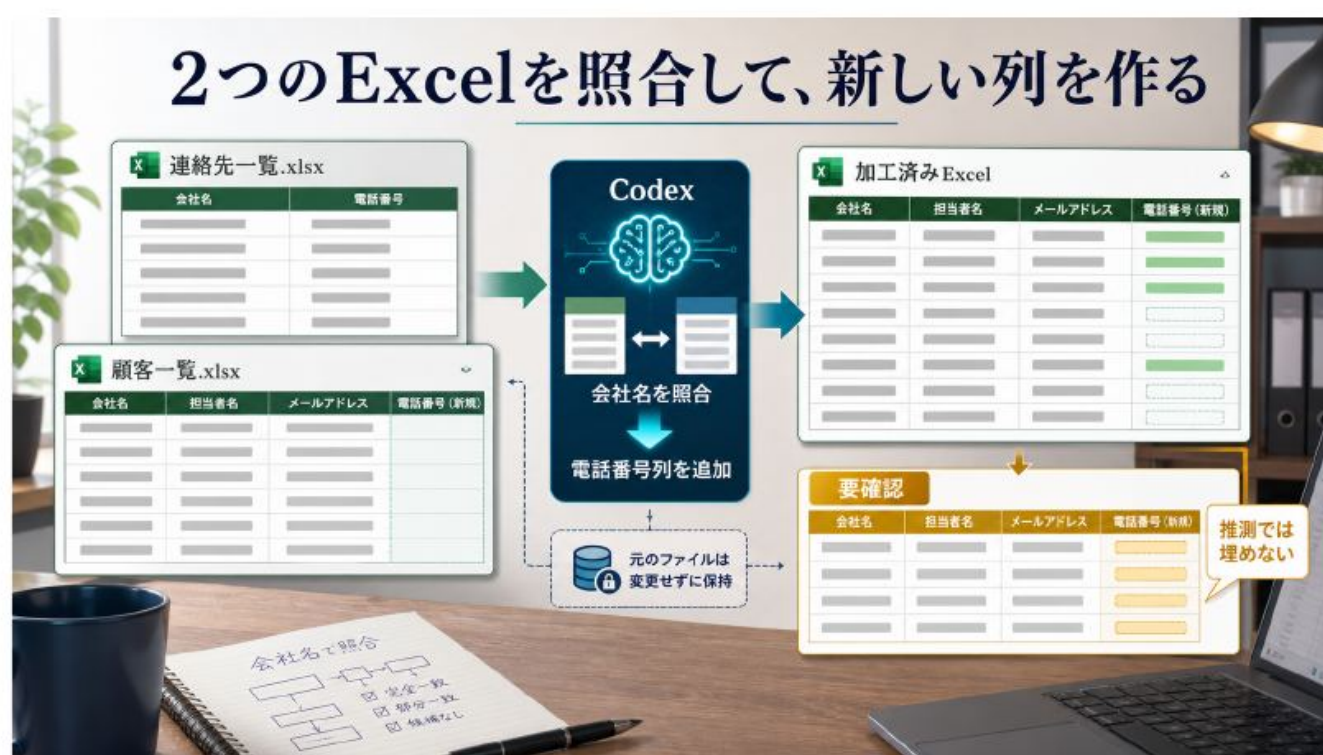


図04-3 2つのExcelを照合して、新しい列を作る

次は、顧客一覧へ電話番号を追加する例です。

顧客一覧には会社名と担当者が入っています。別の連絡先一覧には、会社名と電話番号が入っています。

Codexへ2つのExcelを渡し、会社名を照合して電話番号列を作ってもらいます。

「顧客一覧.xlsx」に電話番号の列を追加してください。
電話番号は「連絡先一覧.xlsx」の情報を使ってください。
会社名が明確に一致した行だけ入力してください。
一致しない行と判断できない行は、確認用シートへまとめてください。
電話番号を推測で作らないでください。
元ファイルは変更せず、完成したファイルを別名で保存してください。

情報がなければ、追加できない

電話番号列そのものは作れます。しかし、電話番号の情報がどこにもなければ、正しい番号は入力できません。
会社名から電話番号を推測させてはいけません。

情報が手元にない場合は、空欄または「要確認」にします。外部情報を調べる場合は、調べてよい範囲と情報源を先に決めます。

AIは、材料のない事実を作るために使うものではありません。手元にある情報を、使える形へ変えるために使います。

Codexは何をするか

内部の難しい処理を覚える必要はありません。仕事の流れは次のとおりです。

1. 2つのExcelを開き、シートと列を確認する
2. 会社名など、照合に使える項目を確認する
3. 一致した行へ電話番号を入れる
4. 判断が必要な行を確認用シートへ分ける
5. 加工前後の件数を比べる
6. 加工済みExcelを別名で保存する

完成物は、説明文ではありません。電話番号列と確認用シートが追加されたExcelファイルです。

自動処理と人間確認を分ける

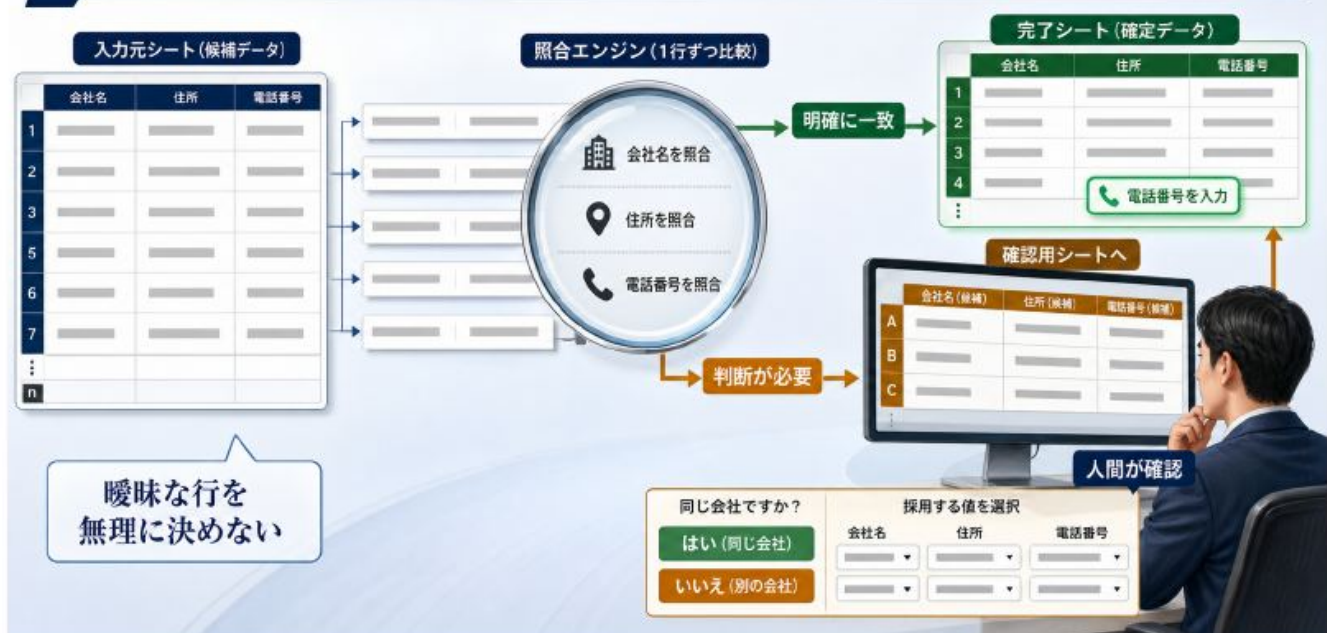


図04-4 曖昧な行を、AIだけで無理に決めない

Excelの照合で難しいのは、完全に同じ文字だけではありません。

「株式会社青空」と「(株) 青空」は同じ会社か。「東京商事」と「東京商事株式会社」は同じか。同じ会社名でも支店が違う場合はどうするか。

Codexは似ている行を候補として見つけられます。しかし、業務上同じものとして扱ってよいかは、人間が判断する場面があります。

そこで、行を2つに分けます。

- 明確に一致する行は加工する
- 判断が必要な行は確認用シートへ移す

会社名が完全一致する行は電話番号を入力してください。

表記が似ているだけの行は自動で統合しないでください。

候補となる会社名、住所、電話番号、確認理由を確認用シートへ並べてください。

この方法なら、曖昧な行を見失いません。また、AIが判断した結果を後から探し回る必要もありません。

削除は最後に行う

重複して見えても、別会社、別支店、別部門場合があります。

最初から削除を頼まず、まず重複候補の一覧を作ります。人間が確認してから、削除または統合を指示します。

曖昧なものを無理に完成させず、人間が判断できる形へ整理する。

これもCodexへ任せられる重要な仕事です。



図04-5 Codexの完了報告は、確認を始める合図

Codexが「完了しました」と報告しても、ファイルを開いて確認します。

第5章 複数のExcelを1つにまとめる

事例02

支店ごとのExcelをコピーする仕事から、まとめたExcelを確認する仕事へ。

本章の支店名、ファイル名、列名は説明用の例です。実際の件数、作業時間、削減時間、エラーを著者の体験として推測して記載していません。

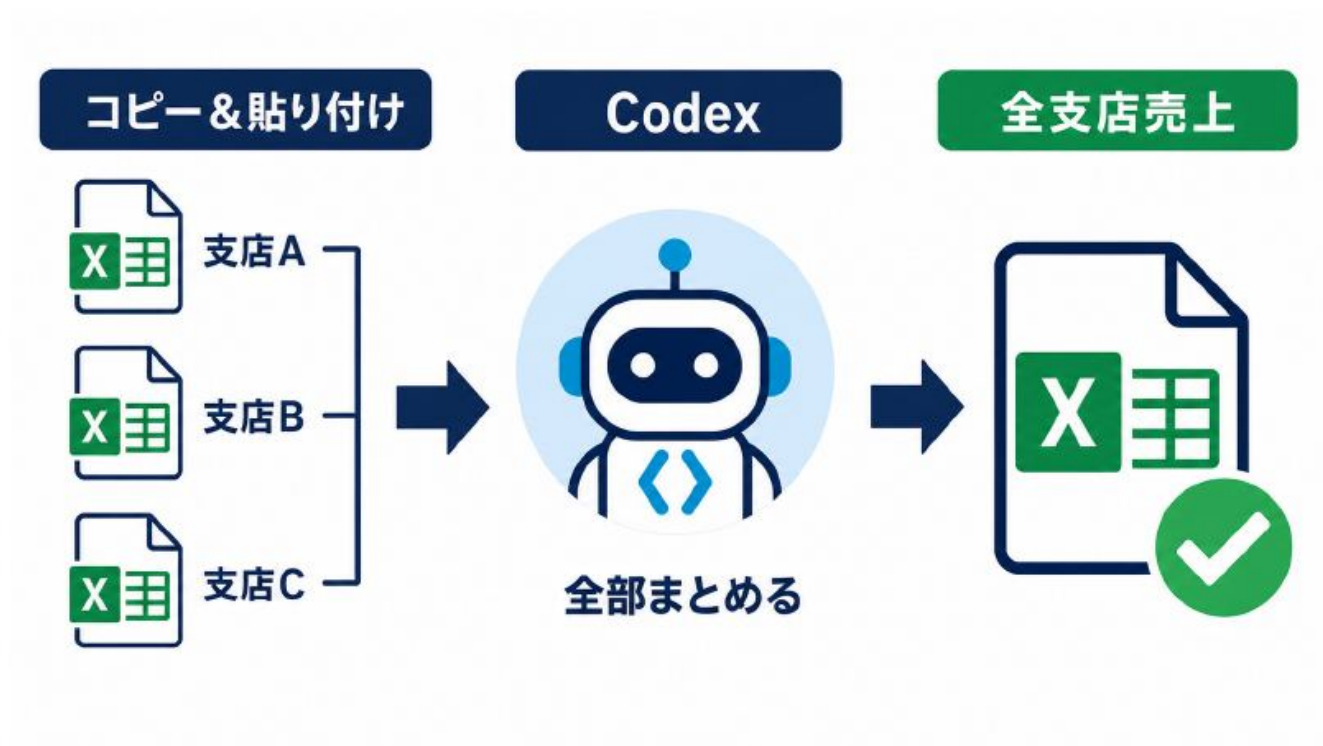


図05-1 支店別Excelを1つにまとめる

月末になると、各支店から売上Excelが届きます。

これまでは、支店Aを開いてコピーし、全体集計用のExcelへ貼り付けていました。次に支店B、支店Cと同じ作業を繰り返します。

単純な仕事に見えますが、ファイルが増えるほど不安も増えます。

- どの支店まで貼り付けたか
- 同じファイルを二度入れているか
- 見出し行まで混せていないか
- 最後の行が抜けていないか
- 列がずれていないか
- どの支店のデータが分からなくなっていないか

Codexには、複数のExcelが入ったフォルダを渡せます。そして、すべてを1つのExcelへまとめるよう頼めます。

このフォルダにある支店別の売上Excelを、すべて1つにまとめてください。

どの支店のデータが分かるように、支店名の列を追加してください。

元ファイルは変更しないでください。

完成したファイルは「全支店売上.xlsx」として別名で保存してください。

完成するのは操作方法の説明ではありません。全支店のデータが入ったExcelです。

こんなことで困っていませんか？

支店別ファイルの形式が、すべて同じとは限りません。

支店Aには「売上日」、支店Bには「日付」と書かれていることがあります。支店名がファイル名にしか入っていないこともあります。

古い形式と新しい形式が混ざる場合もあります。途中から列を追加した支店があるかもしれません。

そのままコピーして終わりではなく、どの列を揃えるか、何を除外するか、件数が合っているかまで確認する必要があります。

ChatGPTへ方法を聞けば、Excelの機能や設定方法を教えてもらえます。しかし、その設定を自分のファイルへ合わせる作業は残ります。

今回終わらせたい仕事は、統合方法の勉強ではありません。

全支店のデータが入り、確認できるExcelを完成させることです。

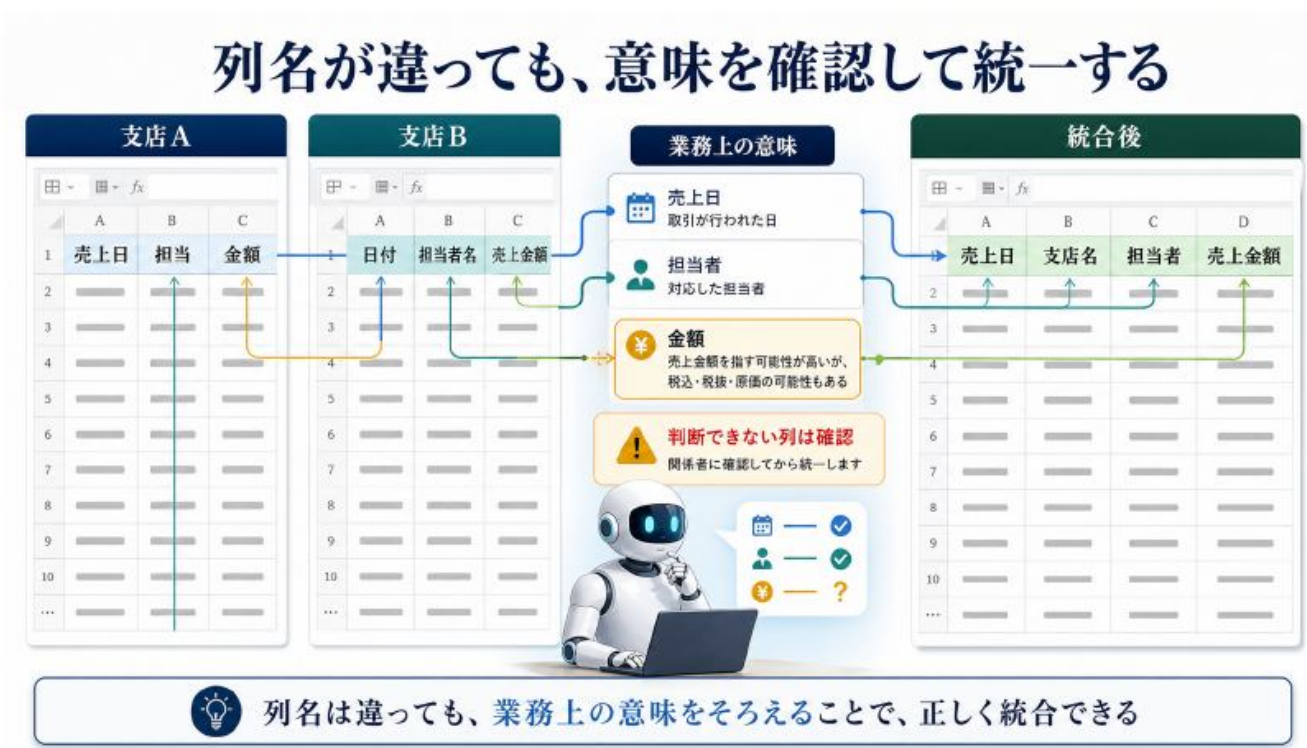


図05-2 列名が違って、意味を確認して統一する

複数のExcelをまとめるとき、最初に各ファイルの構造を確認します。

たとえば、同じ意味の列が次のように書かれている場合があります。

支店A	支店B	統合後
売上日	日付	売上日
担当	担当者名	担当者
金額	売上金額	売上金額

文字が違うだけで、業務上の意味が同じなら統一できます。

元データ、本番環境、秘密情報を直接渡さない



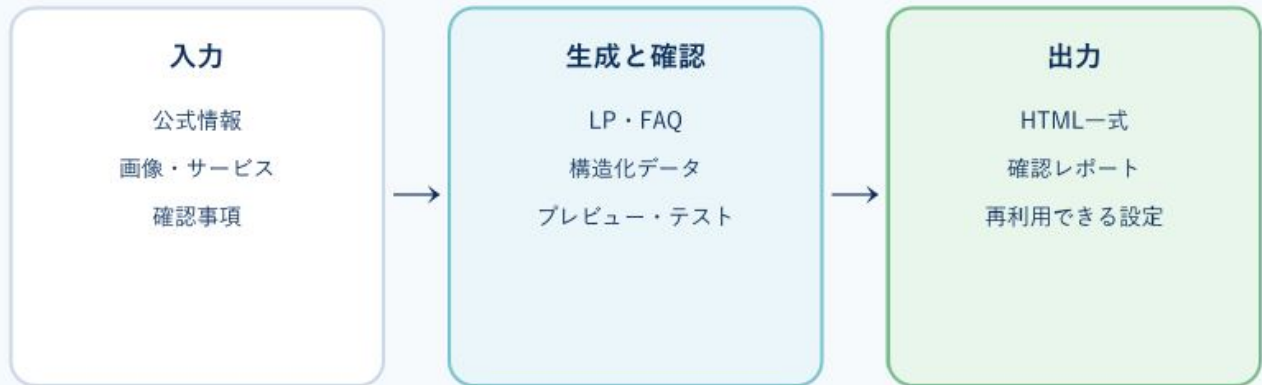
参考：OpenAI公式文書「[Sandbox](#)」「[Agent approvals & security](#)」（本書執筆時点）



図24-2では、顧客台帳.xlsx、売上明細.csv、会社案内.pptxを元データのフォルダへ置き、別の作業用コピーを作っています。

実例 AEO+LPで残る成果物

一枚のページではなく、繰り返し使える制作工程を残す



次の案件でも同じ確認手順を使える状態が成果

図27-6 ページだけでなく再利用できる制作工程を残す

公開情報は AEO+LP で確認できます。

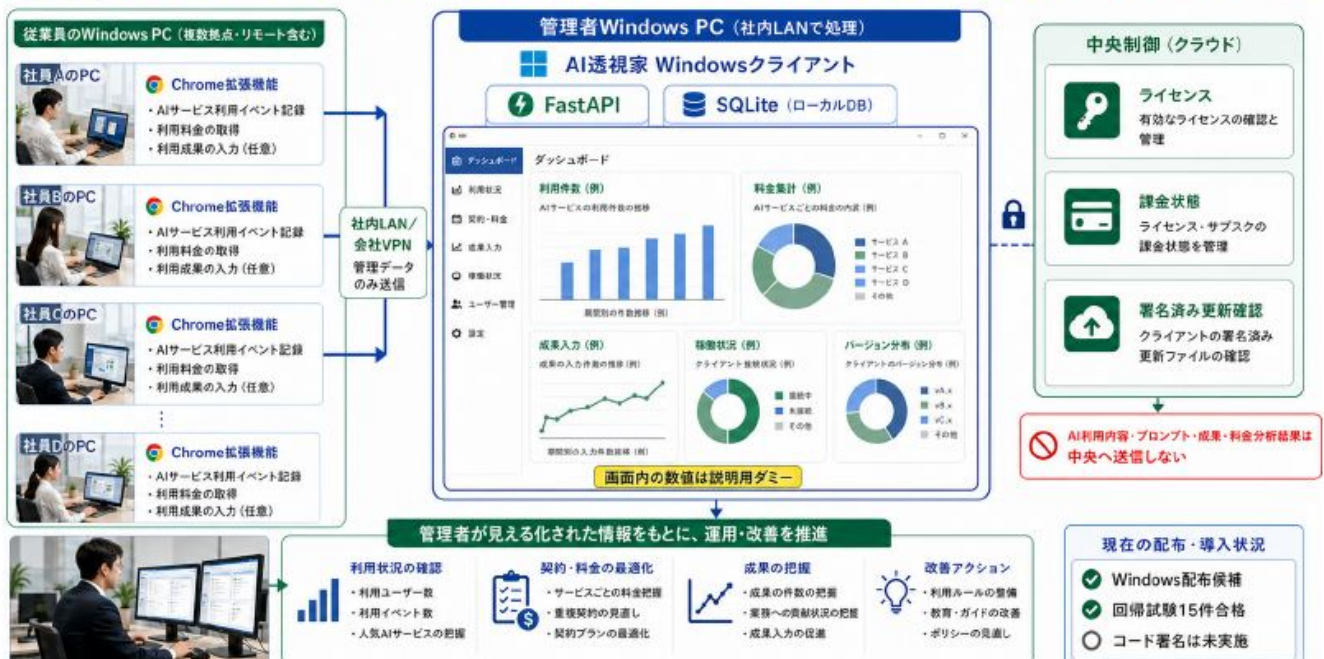
実例4 AI透視家をWindows製品として組み立てる

四つ目は、AI透視家です。

企業で複数のAIサービスを使い始めると、誰がどのサービスを使っているのか、契約が重複していないか、利用料金と成果をどう見直すかが分かりにくくなります。

この課題に対して作ったのが、Windows環境でAI利用状況を見える化する仕組みです。

実例 AI透視家は、社内のAI利用をWindowsで見える化した



实例



図27-10 分散したアプリ情報を監査し、公開カタログへまとめる流れ

Codexは各プロジェクトのファイル、公開URL、画像、説明、ガイド、ポリシーを確認し、アプリ名、用途、リンクなどを一覧にしました。仕事効率化、生活・日常、ツール、学習などの目的で探せるように分類し、日本語と英語のページを用意しました。

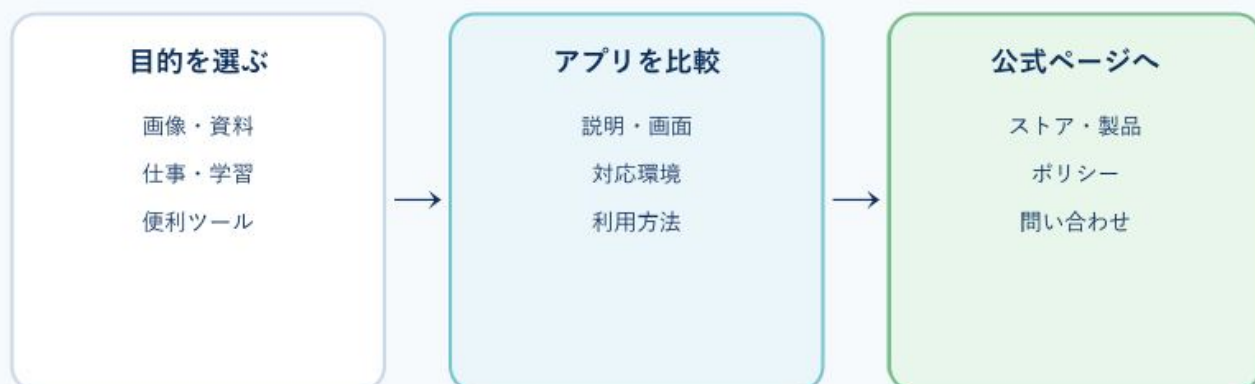
さらに、各アプリ紹介からGoogle Play、ガイド、ポリシーへ移動できるようにリンクをつなぎました。Wordで作った資料をPowerPointへ変換するDrop2PPTも、同じ公開入口から利用できるようにしています。

本書執筆時点の公開ページには、「20+ Android projects」と、多くのアプリがアカウントなしで動くことが揭示されています。これは公開ページで確認できる製品構成の説明であり、ダウンロード数、評価、売上を示すものではありません。

公開ページは [WorldScene](#) で確認できます。

実例 WorldSceneでアプリを探す流れ

複数のアプリを、利用者が目的から選べる入口へまとめる



作る側の分類ではなく、利用者が探せる導線にする