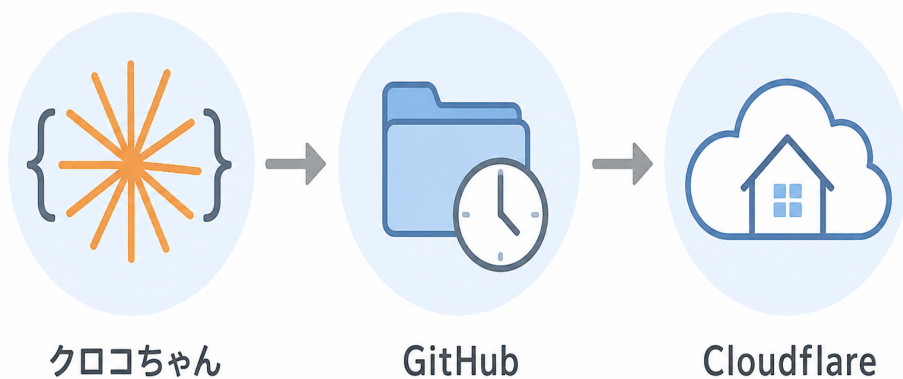


クロコちゃん × GitHub × Cloudflare 入門

非エンジニアのための15分概論

3つの相棒・非エンジニア向け入門



みやもん 学習用教材 v6
2026年5月

はじめに

このPDFのゴール

本書は「これら3つのツール、結局なに？」という疑問を、15分で「なんとなく分かった」状態にすることをゴールとしています。

想定読者

- ・パソコンで日本語入力ができる
- ・「ネット上のサービス」「Git」「サーバー」と言われるとぼんやりする
- ・自分の業務に活かせるか興味はある

読み方

図を中心に、流し読みで構いません。

わからない単語が出てきても止まらず、まずは雰囲気をつかむことを優先してください。

本書で扱わないこと

- ・実際の操作手順（インストール・コマンド入力など）
 - ・プログラムの中身の詳しい説明
- これらは別の資料で、手を動かしながら学ぶのが効率的です。

読了後のゴール

「これ自分の業務にも使えそう」と1つでも思えたら、本書の目的は達成です。

第1章 クロコちゃん (Claude Code)

一言で言うと

日本語で話しかけたらプログラムを書いてくれる、専属の開発者

イメージ：英語が話せる外国人秘書

あなたは日本語で「Excelを業者ごとに集計するプログラムを書いて」と話すだけ。

クロコちゃんが裏で英語（プログラミング言語）に翻訳し、動くプログラムとして返してくれます。

結果が思ったものと違ったら「ここを直して」と日本語で指示し直せます。

日本語の指示 → 動くプログラム

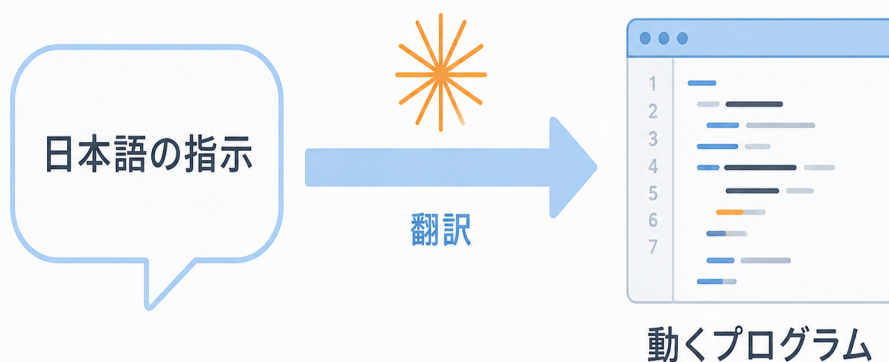


図1の1：日本語の指示 動くプログラム

クロコちゃんができる3つのこと

1. アプリや自動化プログラムを作る

例：請求書をPDFで生成するアプリ / 家計簿アプリ / LINEに毎朝通知を送るBot

2. データを整形・分析する

例：1000行のCSVを取引先別に集計 / メアドの一括整形 / PDFから文字を抜き出す

3. 既存ファイルを編集・修正する

例：エクセルのマクロを別形式に変換 / マニュアルの誤字一括修正

クロコちゃんができる3つのこと

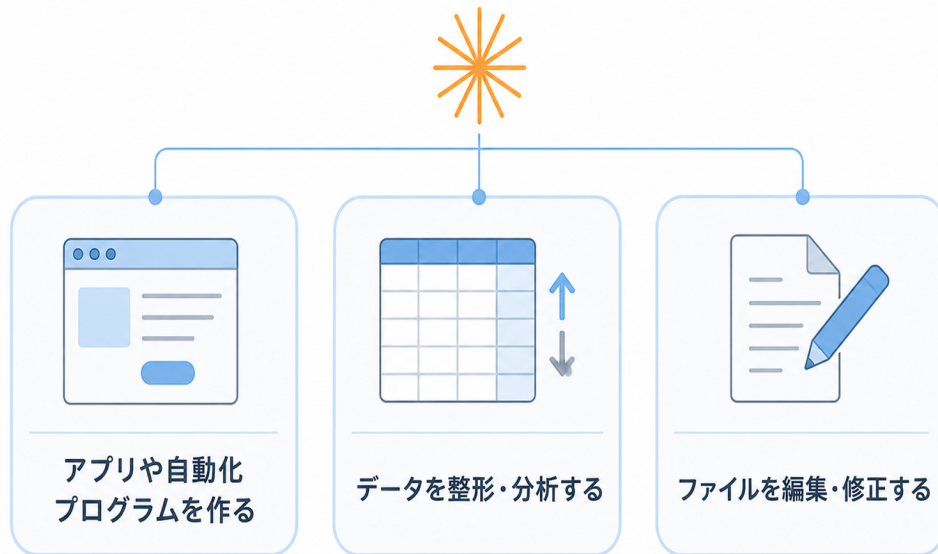


図1の2：クロコちゃんができることの3つの軸

業務活用ヒント

- ・ 毎週手作業でやっているExcel集計を自動化
- ・ 報告書のひな形を「文章入力 PDF出力」する仕組み化
- ・ 大量のメアド / リストの整形・突合

注意点

- ・ 完璧ではありません。間違える時もあります。最終チェックは人間が必須
- ・ パスワード / APIキー / 顧客の個人情報などの機密情報は渡さない

第2章 GitHub

一言で言うと

書いたプログラムを保管しつつ、過去に戻れる「タイムマシン付きの保管庫」

イメージ：銀行の口座のようなもの

書いた内容を「点」として記録します。入金（プログラム保存）するたびに残高（履歴）が積み上がり、いつでも昔の状態に戻せるのが特徴です。

「3日前の状態に戻したい」「あのバージョンでもう一度試したい」が実現できます。

保管場所は ネット上 なので、自分のPCが壊れても中身は無事です。

GitHub = プログラムのタイムマシン

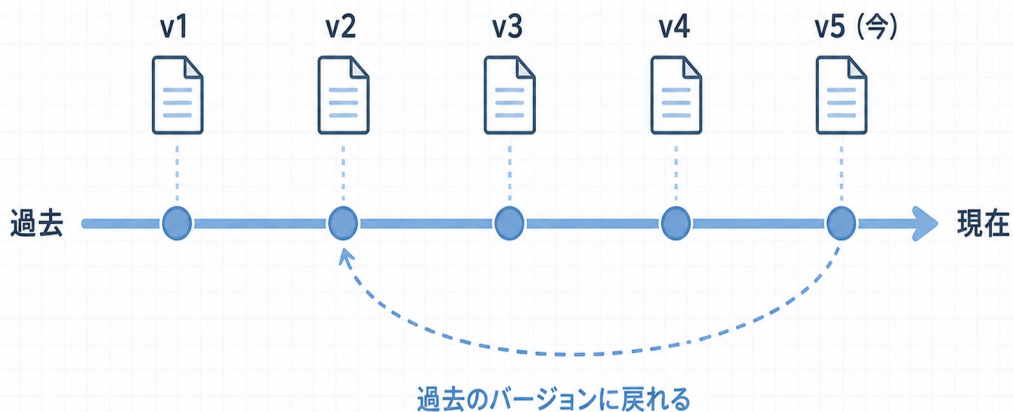


図2の1：GitHubは「プログラムのタイムマシン」

GitHubでできる3つのこと

1. プログラムの保管

自分のPCが壊れても、ネット上の保管庫に無事。引っ越しや故障で失う心配がない。

2. 履歴管理

「3日前の状態」「先週の状態」にいつでも戻れる。実験的な変更も気軽にできる。

3. 共有

自宅PCで書いて、会社PCで続きを書く。チームメンバーと同じプログラムを使う。

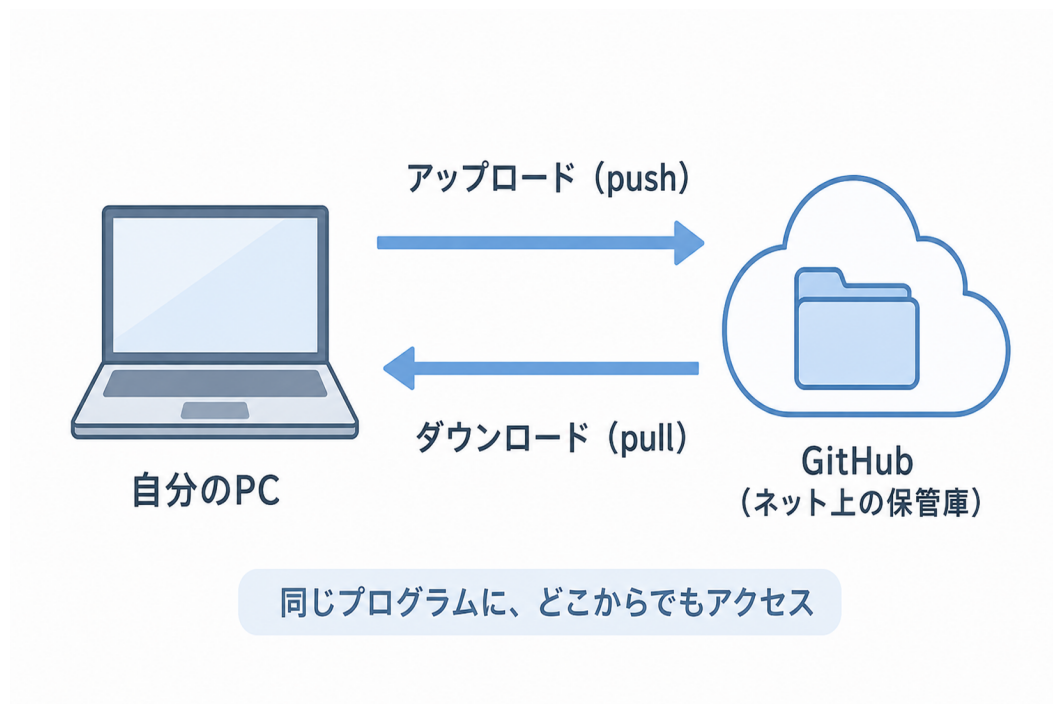


図2の2：自分のPC GitHub (ネット上の保管庫)

業務活用ヒント

- ・大事なExcelマクロやスプシのプログラムを 消えない場所 に保管
- ・社内マニュアルのバージョン管理 (誰がいつ何を変えたかが追える)
- ・チームでのプログラム共有・レビュー

注意点

- ・Public (公開) とPrivate (非公開) の設定を間違えない
- ・個人情報・機密情報を含むファイルは保管しない (特にPublicでは致命的)

もう一步 ローカルとリモートの関係

GitHubを使い始めると「自分のPCのファイル」と「GitHub上のファイル」の関係に戸惑うかもしれません。結論：「同じ中身を保ち続けている、別の場所にある2つのファイル」と考えてください。

1台のPCで作業する場合（多くの個人開発はこれ）

ローカルで修正 アップロード（push）。これだけで運用できます。
ローカル＝「実体」、GitHub＝「バックアップ＋履歴」の役割です。
ダウンロード（pull）はほぼ不要です。

会社PCと自宅PCの両方で触る場合

始める前にダウンロード（pull）で最新を取得 修正 アップロード（push）。
「始める前にpull、終わったらpush」を習慣にすると、内容のズレが起きません。

ローカルとリモートの関係

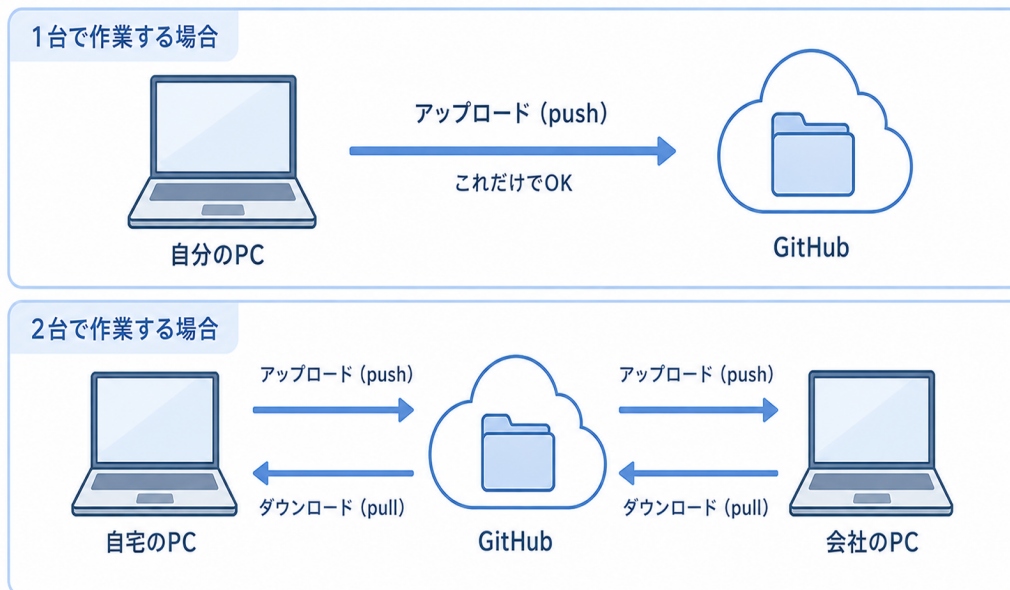


図2の3：1台で作業する場合と、2台以上で作業する場合

クロコちゃんて作業する時

クロコちゃんに作業を頼むときは、push / pull

もクロコ自身に頼めるので、みやもん本人がコマンドを覚える必要はありません。

「いま修正したやつGitHubに上げといて」と言えばOK。

第3章 Cloudflare

一言で言うと

作ったアプリをネット上で公開できる「家」を、無料で借りられる場所

イメージ：マンションの賃貸契約

Webアプリをインターネットに公開するには、本来「土地（サーバーと呼ばれる、ネット上の置き場所）」を借りる必要があります。

Cloudflareは 土地代 + 家の構造 を個人用なら基本無料で提供してくれます。しかも家の中に鍵（認証）をかけられて、特定の人だけが入れるように設定できます。



無料でアプリを公開できる場所

図3の1：Cloudflareは「無料で借りられるネット上の家」

Cloudflareでできる3つのこと

1. Webアプリの公開

作ったアプリを「自分専用URL」で公開できる。個人用なら基本¥0。

2. 鍵付き（Access認証）

特定のメールアドレスだけが入れる設定が、追加料金なしで使える。

3. データの保管

アプリで使うデータも一緒に置ける。家計簿の数字や、業務記録など。

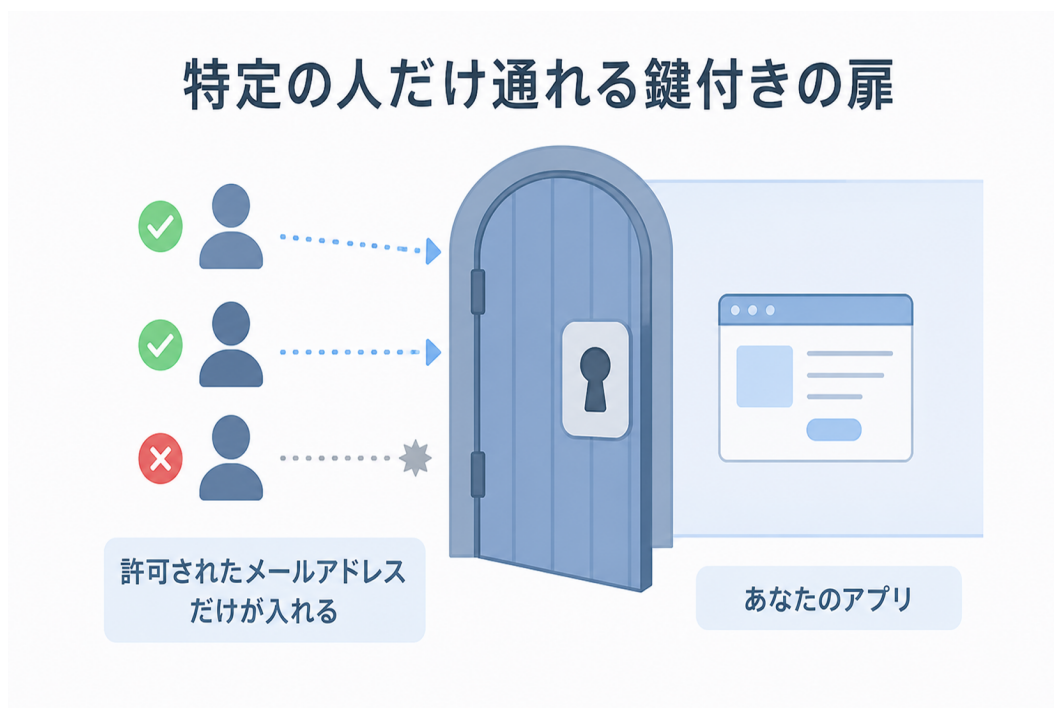


図3の2：Access認証 = 鍵のかかった扉

業務活用ヒント

- ・社内ツールを「うちの部署だけ使えるアプリ」として公開
- ・個人副業のサービスをコストゼロでスタート
- ・自分専用URLで「いつでもどこでもアクセスできる業務ツール」

注意点

- ・無料の範囲を超えると課金される（個人利用なら基本超えない）
- ・Access認証は必須に設定しないと「誰でもアクセス可能」になる
- ・URLを知っている人なら誰でもアクセスできる、ということを理解する

第4章 3つを組み合わせるとこう動く

ここまで紹介した3つのツールは、それぞれ単独でも使えますが、組み合わせると本領を発揮します。

非エンジニアの開発ループ

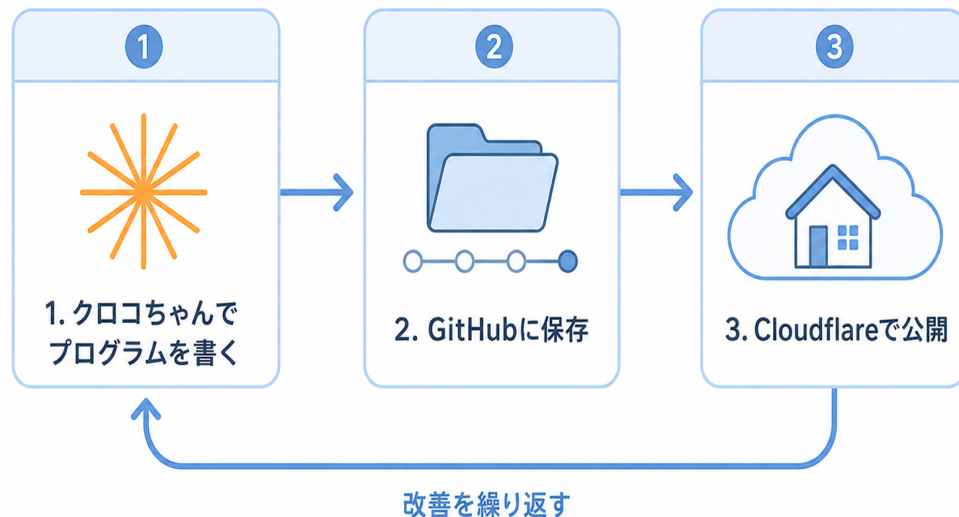


図4：3ツールの連携フロー

例：個人用の請求書アプリを作る場合

1. クロコちゃんと話して「請求書PDF生成アプリ」を作る
日本語で要件を伝えるだけ。プログラムはクロコが書く。
2. GitHubに保存（ネット上の保管庫へ）
明日も来週も同じプログラムが使える。途中の変更履歴も残る。
3. Cloudflareにアップロード
自分専用URLで世界中どこからでも使える。Access認証で「自分だけ」に。

このループの威力

このループが回ると、思いついた業務改善を数時間～半日でアプリ化できるようになります。アプリは消えず、どこからでも使え、必要なら他人とも共有できる。「副業で稼ぐ」も「会社で時短する」も、同じ仕組みで実現可能です。

もう一步 クロコちゃんはどこまで自動化してくれる？

「Cloudflareでアプリを公開する」と一言で言いましたが、本来はいくつかの手続き が必要です。クロコちゃんはどこまで代行してくれて、どこは自分でやる必要があるかを整理しておきます。

クロコちゃんに任せられること

- ・設定ファイル (wrangler.jsonc など) の作成・編集
- ・公開コマンド (wrangler deploy など) の実行
- ・エラーが出た時の修正・リトライ
- ・コード修正からの再公開までの全工程

人間 (みやもん) がやる必要があること

- ・Cloudflareのアカウント作成 (初回1回・5分)
- ・初回認証のブラウザでの「許可」ボタン押し (1回)
- ・Access認証で「誰を許可するか」のメアド指定 (初回設定時)

クロコちゃんはどこまで自動化してくれる？



面倒な部分は全部クロコ、押す系の作業だけ自分

図4の2：クロコちゃんの自動化範囲と、自分でやる部分の対比

棲み分けのイメージ

イメージ：「面倒な部分は全部クロコちゃん、押す系の作業だけ自分」。
コマンドを覚える必要も、設定ファイルの中身を理解する必要もありません。

補足：GitHubとCloudflareの2つの繋ぎ方

実は連携には2パターンあります。

A. 直接公開（個人開発の入門におすすめ）

ローカルから直接Cloudflareにアップロード。GitHubは「履歴管理 + バックアップ」の役割。

B. GitHub連携の自動公開（中級・自動化の極み）

ローカル → GitHub → Cloudflareが自動検知して公開。ただし設定がやや複雑。

本書では A を前提に話しています。慣れてきたらBに移行する選択肢があります。

第5章 あなたの業務、こう使えるかも

ここまで読んでくれたあなたへ、業務活用の「種」を5つ提示します。

3つの相棒、こんな業務に使える



図5：業務活用のヒント

ヒント例

- 毎週やっているExcel集計をアプリ化
ククロコ + Cloudflareで、誰でもブラウザで使えるツールに
- 古いマクロを今風のWebアプリに作り直す
別端末・スマホからも使えるように
- 大事な設定ファイルを消えない場所に保管
GitHubで履歴管理 + ネット上にバックアップ
- 部署内だけで使えるミニツールを公開
Cloudflare Accessで、メアド指定の認証付き
- 副業のサービスを無料で開始
Cloudflareで個人用なら¥0で公開可能

本書のゴール（再掲）

上記のうち1つでも「自分の業務にも使えそう」と思えたら、本書の目的は達成です。次は、その1つを実際に試してみましょう。

おわりに 次のステップ

本書は「触る前のイメージ作り」を目的にしました。
ここから先、実際に手を動かしていく順番のおすすめは以下の通りです。

ステップ1：クロコちゃんから始める

まずはクロコちゃんに「自分の業務の困りごと」を1つ話してみる。「これ作れる？」と聞くだけで、できるかどうかの感触がつかめます。

ステップ2：GitHubは「保管したい」と思った時

最初の数個のアプリは、自分のPCに保存だけでも十分。
「あのプログラムもう一度使いたい」「壊れたら困る」と思った時に、GitHubの必要性が腹落ちします。
もし既にいくつかアプリ・自動化プログラムを作っている場合は、今すぐ移行を始めるのがおすすめ。

ステップ3：Cloudflareは「公開したい」と思った時

アプリを「自分のPC以外でも使いたい」「他人に使ってもらいたい」と思ったら、Cloudflareの出番。

最後にひとこと

大事なのは、3つを同時に学ばないこと。

1つずつ「困った時に必要な道具を取り出す」感覚で、十分に進められます。

質問があれば、本書を読んだ後にクロコちゃんへ直接ぶつけてください。

「これ、自分の で使える？」「この理解で合ってる？」が、最も早い学習方法です。