

使い方マニュアル

Claude Code 画像生成 接続キット v1.0

APIキーを1つも使わずに、Claude Codeから画像を作れるようにします。このマニュアルは、**Claude Code**を触りはじめたばかりの方を想定して書いています。

もくじ

1. これは何をするものか
 2. 4つの経路から1つ選ぶ
 3. 用意するもの
 4. 入れる（3分）
 5. はじめて動かす
 6. 1枚作ってみる
 7. まとめて何十枚も作る
 8. 画像の中に日本語を入れたいとき
 9. 出来上がりを確かめる
 10. つまずき集（エラーが出ない4つ）
 11. 終わったら閉じる
 12. よくある質問
-

1. これは何をするものか

Claude Codeは、文章を扱う道具です。**自分では絵を描けません。**

では「Claude Codeで画像を作った」というとき、何が起きているのか。やり方は大きく2つに分かれます。

ひとつめは、**画像生成のAPIを呼ぶやり方**。OpenAIのAPIキーを設定して、1枚作るたびにお金がかかります。試し刷りを重ねると、月末の請求で驚きます。

ふたつめが、このキットのやり方です。すでにブラウザでログインしているChatGPTの画面を、Claude Codeに代わりに操作させます。あなたが手で「○○の画像を作って」と打ち込む作業を、そのまま自動でやらせるだけです。

だから、追加のお金はかかりません。ChatGPTの月額の中で終わります。

```
あなた
|
| 「○○の画像を作って」
v
Claude Code
|
| ブラウザを操作する
v
Chrome (ChatGPT)
|
| 画像ができる
v
指定したフォルダに保存
```

「代わりに操作させる」やり方は、4通りあります。次の章で1つ選んでください。

2. 4つの経路から1つ選ぶ

どれを選んでも、できることは同じです。環境と、作る枚数で決まります。

	経路	必要なもの	向く場面	速さ
①	Codex CLI	Codex CLIとログインだけ	月に数枚。すぐ始めたい	中
②	Chrome DevTools	専用プロファイルのChrome	何十枚も作る。止めて再開したい	速い
③	Claude in Chrome	Chrome拡張機能だけ	端末の設定を触れない	遅い
④	Playwright	Node と Playwright	Claude Codeの外で自動化したい	速い

迷ったら②を選んでください。量を作るときに一番安定する経路で、このマニュアルも②を前提に書いています。

会社の端末などでChromeの設定を触れないなら③。拡張機能を入れるだけで動く経路です。

月に数枚しか作らないなら①。ただし①は、画像生成がChatGPTの枠を 通常のメッセージより3～5倍速く消費します（OpenAIの公式ヘルプに明記されています）。

Claude Codeを立ち上げずに定期実行したいなら④。画面の構造が変わったら自分で直す前提です。

経路ごとのセットアップ手順

`setup/` フォルダに、経路ごとの手順書が1本ずつ入っています。その中身をまるごとコピーして、Claude Codeに貼り付けてください。あとは自動で進みます。

```
setup/セットアップ①（Codex CLI方式）.md
setup/セットアップ②（Chrome DevTools方式）.md
setup/セットアップ③（Claude in Chrome方式）.md
setup/セットアップ④（Playwright方式）.md
```

あとから別の経路に乗り換えられます。同時に複数入れておいても問題ありません。

3. 用意するもの

	説明
Claude Code	CLI / Cursor / VS Code拡張 のどれでも構いません
Google Chrome	普段使っているもので大丈夫です
Node.js 18以上	<code>node -v</code> で確認できます
ChatGPTのアカウント	画像生成が使えるプラン（Plus / Team / Enterprise など）

OpenAIのAPIキーは要りません。用意しないでください。

Node.js が入っているか確かめる

ターミナル（Cursorなら下の黒い画面）で、こう打ちます。

```
node -v
```

`v18.0.0` のように数字が出れば大丈夫です。`command not found` と出たら、nodejs.org から入れてください。

4. 入れる（3分）

4-1. スキルをコピーする

Claude Codeで使っているプロジェクトのフォルダを開きます。その中に `.claude` というフォルダがあるはずです（無ければ作ります）。

このキットの `skills/image-gen-connect-ss` を、フォルダごと次の場所にコピーします。

```
<あなたのプロジェクト>/claude/skills/image-gen-connect-ss/
```

コマンドでやるなら、プロジェクトのフォルダの中で次を実行します。

```
mkdir -p .claude/skills  
cp -r <このキットの場所>/skills/image-gen-connect-ss .claude/skills/
```

4-2. Claude Code を再起動する

再起動しないとスキルが読み込まれません。いったん閉じて開き直してください。

4-3. 読み込まれたか確かめる

Claude Codeにこう聞きます。

```
image-gen-connect-ss というスキルは入っていますか
```

「入っています」と返れば完了です。

5. はじめて動かす

まず、選んだ経路の手順書（ `setup/` の中）をコピーして、Claude Codeに貼り付けてください。そのあと、こう言います。

```
image-gen-connect-ss を使って、画像生成の環境をセットアップして
```

以下は②Chrome DevTools方式を選んだ場合の流れです。①③④は各手順書に書いてあります。

すると、**画像生成専用のChromeが1つ開きます**。普段使っているChromeとは別物です。ブックマークも履歴も共有しません。普段のChromeには一切影響しません。

開いた画面にChatGPTのログインが出るので、**1回だけログインしてください**。このログイン状態は、この専用Chromeにだけ残ります。次からは自動で通ります。

なぜ専用のChromeが要るのか

普段使いのChromeをそのまま使おうとすると、**接続を拒否されます**。最近のChromeは、普段使いのプロファイルに外から操作されることを許していません。

もうひとつ理由があります。**自動で操作している最中に、あなたが同じタブを触ると両方が壊れます**。別のウィンドウにしておけば、その心配がありません。

6. 1枚作ってみる

```
image-gen-connect-ss で「夕暮れの商店街を歩く猫」の画像を作って、images/neko.png  
に保存して
```

Claude Codeが専用Chromeを操作して、画像を作って保存します。**1枚で60～120秒かかります**。待っている間、画面は動かないように見えますが、動いています。

保存されたら、そのファイルを開いて確かめてください。

7. まとめて何十枚も作る

1枚ずつ頼むのは大変なので、先に一覧を作ります。

`images.json` のような名前でファイルを作り、こう書きます。

```
[
  { "prompt": "夕暮れの商店街を歩く猫", "out": "images/img_01.png" },
  { "prompt": "朝の海辺で本を読む人", "out": "images/img_02.png" },
  { "prompt": "机の上のコーヒーとノートPC", "out": "images/img_03.png" }
]
```

そのうえで、Claude Codeにこう言います。

image-gen-connect-ss で images.json の画像を作って。
out のファイルがまだ無いものだけを、1枚ずつ順番にお願いします。

ここが一番大事です

「まだ無いものだけ」と必ず言ってください。

30枚頼んで20枚目で止まったとき、この言い方をしていれば、もう一度同じことを頼むだけで21枚目から再開できます。

言っていないと、最初の20枚をもう一度作り直すことになり、時間もChatGPTの枠も無駄です。

守ること

	なぜ
1枚ごとに新しいチャットに移る	同じチャットで4枚以上続けると、送信ボタンが押せなくなります
同時に走らせるのは2本まで	3本以上にすると、ChatGPT側が一時的に受け付けなくなります

8. 画像の中に日本語を入りたいとき

図解のように絵の中に日本語の文字を入れる場合は、書き方にコツが要ります。

だめな書き方

比較表の図解を作って。タイトルは「4つの経路」で、左に「APIキーを使う」右に「ブラウザを操作する」と入れて、青系で.....

1行の長い文にすると、日本語が崩れます。これが文字化けの一番の原因です。

いい書き方

4つに分けます。

```
* Subject: (Comparison infographic with Japanese text. Title reads "4つの経路".)
* Layout: (Two columns. Left column reads "APIキーを使う". Right column reads "ブラウザを操作する".)
* Visuals: (Right column highlighted in indigo. Left column soft gray. White background. Generous margins.)
* Style: (Flat vector design, clean Japanese typography, bold sans-serif fonts, white background, high resolution 4k.)
```

上に指定した日本語の文字列だけを、一字一句そのまま正確に描いてください。
指定していない文字・英単語・記号・ロゴを一切足さないこと。

中に入れる日本語は、英語に訳さずそのまま書きます。ここを訳すと、訳された文字が絵に入ります。

9. 出来上がりを確かめる

「成功しました」というログを信じないでください。開いて目で見てください。

機械では見つからない事故が2つあります。

その1: 別の作業の画像が混ざる

同じChromeを別の作業でも使っていると、画像が入れ替わることがあります。ハッシュの重複チェックで見つかりません。別の画像なので、当然ハッシュは違うからです。

見つけ方は、**全部を1枚のシートに並べて、それぞれの寸法と縦横比を一緒に表示**することです。指定していない比率が混ざっていれば、並べた瞬間に浮きます。

その2: 指定と違う絵が出る

全部の画像に共通の指定（「薄暗い室内の照明で」など）を付けているとき、その指定と中身が噛み合わない1枚だけが負けます。

「薄暗い室内の照明」を付けたまま「夕暮れの商店街」を頼むと、**室内の絵が出てきます**。共通の指定を使うときは、屋外・俯瞰など**その照明では成立しない絵**を先に洗い出して、その1枚だけ文を差し替えてください。

見るときは「きれいな絵か」ではなく「この文章の隣に置いて正しいか」を見てください。

10. つまづき集（エラーが出ない4つ・②Chrome DevTools方式）

よくある4つは、どれもエラーメッセージが出ません。だから気づきにくいのです。

10-1. フォルダ名に空白が入っている

専用Chromeを置く場所の名前に空白があると、指定が途中で2つに割れます。

`Debug ChatGPT` という名前なら、`Debug` までのフォルダの指定として読まれ、余った `ChatGPT` は住所として別のウィンドウに開きます。

症状: 起動したはずなのに、つながらない。**直し方:** フォルダ名に空白を入れない。このキットの起動スクリプトは、最初からそうしてあります。

10-2. 専用Chromeが落ちているのに、成功したように見える

長く動かしていると、専用Chromeが落ちることがあります。落ちた状態で生成を頼むと、**1枚ずつ0秒で空振りして、しかも成功したように返ってきます**。

ログには「終わりました」と並び、フォルダには1枚も増えていません。

直し方: 生成の前に、毎回これを実行します。

```
node .claude/skills/image-gen-connect-ss/scripts/check.mjs
```

OK と出れば生きています。 **つながりません** と出たら、起動し直してください。

「終わりました」を完了の合図にしないでください。合図はフォルダにできたファイルです。

10-3. タブを固定していない

自動操作は、既定では「最初に見つけたChatGPTのタブ」を掴みます。同じChromeを別の作業でも使っていると、**両方が同じタブを取り合います**。

直し方: 作りはじめる前に、自分専用のタブを1枚作ります。

```
node .claude/skills/image-gen-connect-ss/scripts/check.mjs --new-tab
```

表示されたタブIDを、Claude Codeに伝えてください。1枚しか作らないときも省略しないでください。

10-4. 最小化したら急に遅くなった

裏で動かしたいので最小化したいところですが、最小化するとChromeが「見えていない」と判断して、描画の速度を落とします。

直し方: このキットの起動スクリプトには、それを止める指定が入っています。自分でChromeを起動している場合は、次の3つを付けてください。

```
--disable-backgrounding-occluded-windows  
--disable-renderer-backgrounding  
--disable-background-timer-throttling
```

この指定は起動するときにしか効きません。すでに開いているChromeに後から足すことはできません。

11. 終わったら閉じる（②のとき）

②を使ったときは、作業が終わったら専用Chromeを閉じてください。①③④では要りません。

```
node .claude/skills/image-gen-connect-ss/scripts/check.mjs --quit
```

開けばなしにすると、その端末の中からは操作を受け付ける状態が続きます。危険というほどではありませんが、閉じておくに越したことはありません。

ログインごと消したいときは、専用のプロファイルフォルダを削除します。場所は起動したときに画面に表示されています。

12. よくある質問

Q. OpenAIのAPIキーは本当に要りませんか。 要りません。用意しないでください。必要になるのは、他の人に配るツールを作るときだけです。

Q. 普段使いのChromeは大丈夫ですか。 影響しません。専用のプロファイルを別に作るので、ブックマークも履歴もログインも共有しません。

Q. ChatGPTの無料プランでも動きますか。 画像生成が使えるプランが必要です。このキットは「人が手でやる操作を代わりにやる」だけなので、**手でできないことはできません**。

Q. 何枚まで作れますか。 ChatGPTのプランの枠しだいです。枠を使い切ると、生成を断られます。そのときは翌日に回すか、枚数を減らしてください。

Q. 途中で止まりました。最初からやり直しますか。 やり直す必要はありません。「まだ無いものだけを作った」と、もう一度同じことを頼んでください。続きから再開します。

Q. 生成した画像に、頼んでいない英語が入りました。 プロンプトの最後に「指定していない文字・英単語・記号・ロゴを一切足さないこと」と書いてください。それでも入るときは、1行の長い文になっていないか確かめてください。

Q. Macでも動きますか。 動きます。②の起動は `setup-chrome.sh` を使ってください。それ以外は同じです。

Q. 経路はあとから変えられますか。 変えられます。同時に複数入れておいても構いません。 `setup/` の別の手順書をClaude Codeに貼り付ければ、その経路が足されます。

Q. どの経路を使っているか分からなくなりました。 Claude Codeに「いまでの経路で画像を作る状態ですか」と聞いてください。スキルのStep 1に、経路ごとの確認方法が書いてあります。

株式会社スターストリーム・スタジオ