

Claude Codeで技術書を書く

執筆の現場の
使いどころを
お話します

モウフカブール
大澤文孝



Who Am I?

Follow me

 @sour23

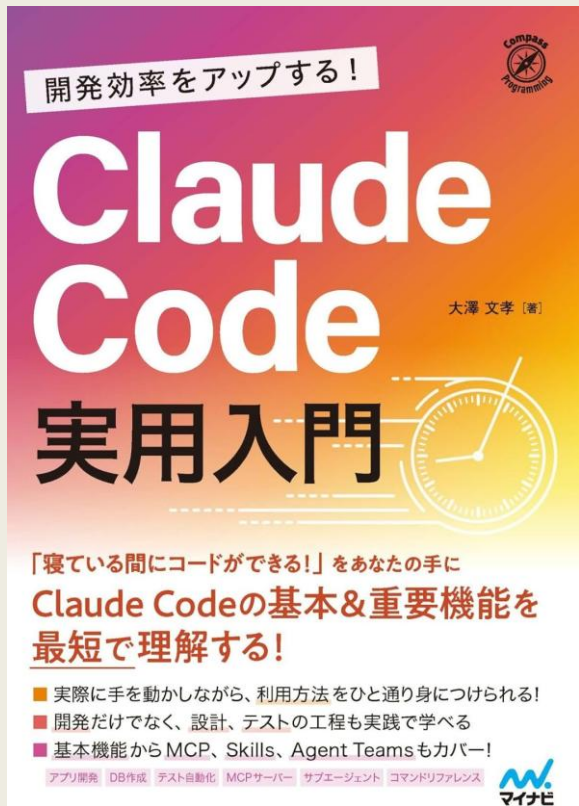
このセッションは
撮影・投稿OK!

#opendevcon

大澤文孝

技術ライター／システムエンジニア／プログラマ／インフラエンジニア。
著書は100冊以上。

開発効率をアップする！



5月19日発売 マイナビ出版

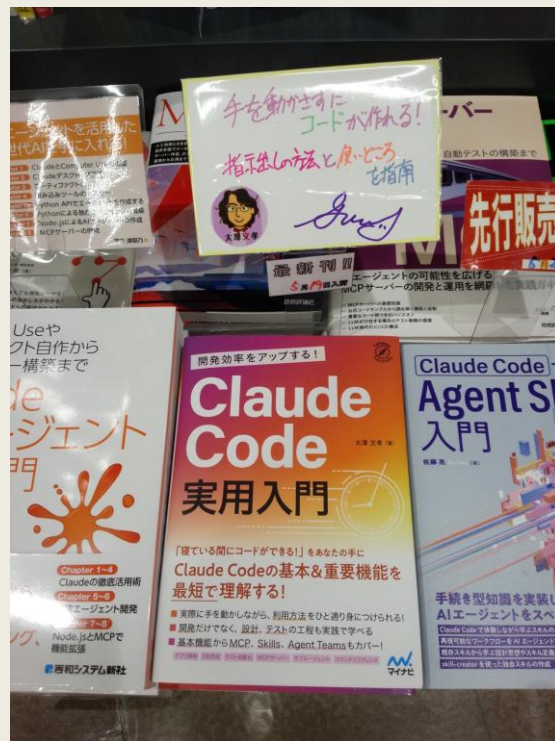


Amazon売れ筋ランキング「ソフトウェア開発・言語」にて1位！

書店さんを回ってきました



▲紀伊國屋書店 新宿本店さま



▲書泉ブックタワーさま

SEPlusで講座しました

話題のエージェント型AIを解説本著者から学ぶ！『Claude Code入門』オープン講座を2026年8月に2日程で開催決定

発売前から重版の話題書『開発効率をアップする! Claude Code 実用入門』の著者である大澤文孝氏が登壇。ブラウザのみで動く演習環境を提供し、実務で“思い通りに動かす指示出し”を徹底解説

株式会社SEプラス 2026年7月17日 10時30分



1日で基本をマスターしよう

Claude Code 入門講座

8/7・21 1日完結コース
オンライン開催

担当講師
大澤 文孝氏



免責

このスライドも
Claude Codeに
下書きさせた！

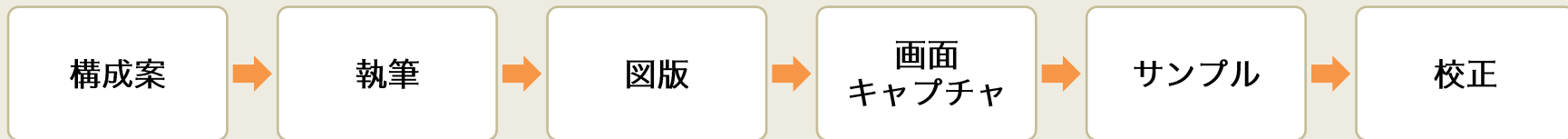
- 大澤文孝の個人的な意見です。所属・関連の団体とは一切関係ありません
- Claude Codeやモデルの仕様・価格は頻繁に変わります（**本日の内容は2026年8月時点**）
- 今回は「執筆での使いどころ」の要点を伝える！

生成AIは進化が
速いから最新情
報も見てね！



技術書の執筆は、手間がかかる

- 「書きたいこと」が頭に揃っていても、本として仕上げるまでが長い
 - 分量があるから、説明の順序を考えなければならない
 - 図もキャプチャもサンプルも、作らなければならない
 - 正しいかどうかを確認する校正作業は、気を張る作業でなかなかたいへん



頭のなかに「書きたいこと」が揃っていても、作業が多くなかなか実現まで遠い
Claude Codeで、その手間を削減しよう！

Claude Codeと使い方

Claude Codeとは

- ターミナルで動くAIエージェント（Anthropic製）
- 「自分のPC」で動く
 - ファイルの読み込み・作成
 - コマンドの実行
 - ネットワーク通信（Webでの検索、データ取得）

コーディン

コードの作成
ビルド
テスト

→ テストが通るまでシステム開発を完成に持って行ける

指示と終了条件
さえ書けば
あとはお任せ

執筆に使う

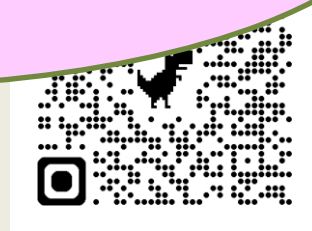
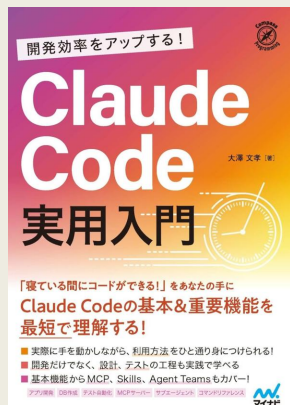
構成案作り
執筆
図版作成
サンプル作り
校正

テストのような
明確な条件がないので完全自動
は無理！

Claude Codeを使うには

- 自分のPCに直接インストールすると「何されるかわからない」
- コンテナを使うのが

いろんなLinuxツールが使える
(officeファイル読み込み、
pdf読み書き、ImageMagick
などなど)



Qiita 「Claude Code入りのDockerイメージをDevContainerで動かす」

DevContainerのデモ

- Docker Desktopを用意する
- プロジェクトフォルダ以下に.devcontainerフォルダを作りdevcontainer.jsonを置く

devcontainer.json

```
{  
  "image": "ghcr.io/fosawa/claudecode-devcontainer:latest",  
  "remoteUser": "vscode",  
  "containerUser": "vscode"  
}
```

- VSCodeでプロジェクトフォルダを開く



Claude Codeで本を書く流れ

構成案：目次のたたき台を作らせる



AWSのEC2についての解説記事を書きたいので構成案を作ってほしい



AWSのEC2についての解説記事を書きたい。

はじめてEC2を使う人向け。マネジメントコンソールから操作。

VPCを作るといところから解説してほしい。VPCは「VPCなどから作る」を使ってほしい。

具体的な手順付きで、パブリックIPを設定して、インターネットから接続できるように。

そこにはnginxをインストールして、外部から接続できましたで完了としよう。

EC2インスタンスのログインには、インスタンス接続が... ブラウザから接続できる

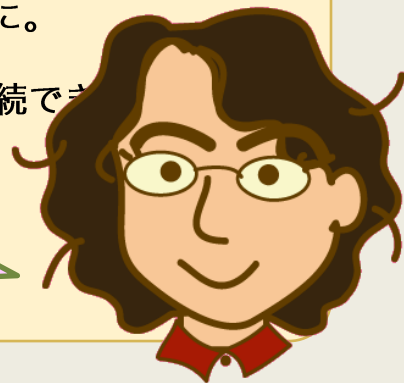
れを使おう。コラムとして「SSHでつなぐ」みたいな

分量は、だいたいA4相当で30ページぐらいか

手順は画面キャプチャ付きで、丁寧に感じで

そこで、こんな内容の構成案を作ってほしい。

何を作りたいの
かを明確に！



生成ガチャの確率を上げる

- 指示していないところは生成AIが「いい案配にやる」
- 指示することで「欲しいものか出てくる確率を上げる」

わざと細かい指示を与えずに、何度かやりとりして詳細を詰めていくというやり方もあり！

→ただし、遠回りになる

→きちんと決めないといけないところと、どうでもいいところをわけて、決めないといけないところはきちんと決めた指示を出す

どのモデルで動かすか

Fable 5

最上位。下書き執筆などはこれ。原稿執筆のように「正解が曖昧」な仕事の主力（入力\$10／出力\$50 per 100万トークン）

Opus 5

高性能。Fableに次ぐ。じっくり考えさせたい作業。構成案出しやPDFの校正とかに使っています（\$5／\$25）

Sonnet 5

バランス型。Fableからサブエージェントで調査させるときに使用（\$3／\$15）

Haiku 4.5

軽量・高速・安価。使ってないので知らない（\$1／\$5）

原稿のテイストを決める

過去の原稿群



過去の原稿を「過去原稿フォルダ」に置いた。
これと同じような原稿を書きたいので、それらの原稿を読んで、「トンマナ.md」を作って



トンマナ.md

自分の持ち味

- トンマナ

- 「トーン&マナー」の略。原稿のテイストや雰囲気、書式のこと（編集の専門用語というかスラング）

トンマナの例

本邦初公開！

これがニャゴロウのトンマナだ！

前者「明後日プログラミング」の文体・構成を分析し、本書「暗号化本」でも同一著者のテイストを維持するためのガイドライン。

1. 全体のトーン

基本姿勢

- 「飲み屋で先輩が面白おかしく話す」ようなフランクな口調。商業誌では書かないカジュアルさが特徴。
- 読者を「皆さん」と呼ぶ。ときどき「あなた」も使う。
- 一人称は「僕」。
- 敬体（です・ます調）をベースにしつつ、興奮したり脱線したりするときは常体（だ・である調）に切り替わる。この切り替えが自然体で、統一性よりもライブ感を重視する。

口調の特徴

- ツッコミ芸：自分で言ったことに自分でツッコむ。「しゅごい。」「ヤバイ句いしかしい。」「カisパワー！」のような短い感嘆。

- 興奮：「しゅごい！」「デター！」「オ、ワッケー！！」
- 呆れ：「やってられないよ！！」「意味がわからない。ウム。僕もわからない。」
- 自虐：「僕みたいなせっかち人間には向かないよ！」「俺もCPUも。」

2. 比喩・例え話のパターン

日常の例え

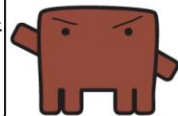
- 技術概念を日常のモノに置き換える。オルゴール、ダンス、倉庫、ポケット、うどん屋の話など。
- 食べ物の比喩が多い：「たけのこの里は甘いのが特徴だし、ポテトチップスはしょっぱいのが特徴」「ポテトチップスから塩を抜いて砂糖をまぶすのは筋が悪い」

歴史・文化の例え

- 戦国武将・三国志で例える：「戦国武将で喩えると、織田信長。三国志だったら曹操。」
- 洋楽バンドで例える：「洋楽で喩えるならビートルズ」「バンドならELO」
- アニメ・ゲームの引用：「異世界で何度も死んだり生き返ったりするやつ！」「まゆしい☆の死亡が繰り返されち

Claude Code と 大澤文孝が共闘したもの ニャゴロウに破れた話

モウフ
技術ノ
秘密基地



I'm always hallucinating
right after I wake up.

@mofukaburTech
@shigetaka256
@sour23



▲「モウフカブール」のブースで頒布中

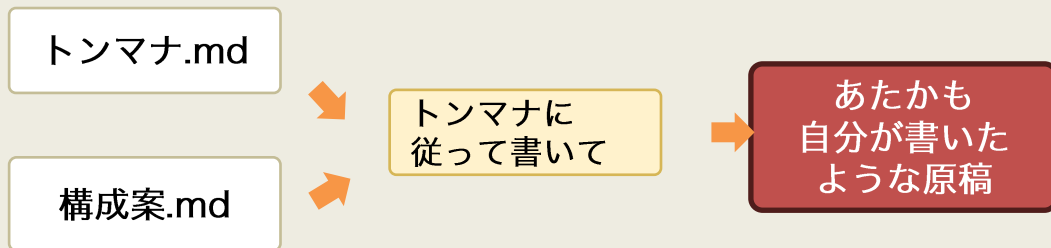
トンマナに従って書かせる

トンマナ.mdに従って、構成案.mdの原稿を書いてほしい。

原稿は「原稿フォルダ」に、01、02、のようなフォルダを作って、その下に、01.md、02.mdなどとして作ってほしい

そのサブフォルダimagesに、01-01.png、01-02.pngなどとしておいてほしい。なおイラストの場合は、drawio.pngでよろ。

サブエージェントを使って、どうしてもいいことをするときはSonet5を使って。そのサブエージェントからは必要に応じてFable 5を呼び出して。



できた原稿

1.1 暗号化とは何か

さて、さっそく暗号の話をしていきましょう。

「暗号」と聞いて何を思い浮かべますか？ スパイ映画でしょうか。戦争モノでしょうか。あるいは、子供の頃に友達と秘密の手紙を交換した記憶があるかもしれません。「あーい、いーう」のようにずらして書くやつ！ あれも立派な暗号です。

暗号とは、一言で言えば、「秘密を守る技術」です。

人間は大昔から、「この情報は特定の人だけに伝えたい」「途中で誰かに読まれたくない」という欲求を持ってきました。軍事作戦、密文、商取引の秘密。理由はさまざまですが、「秘密を守りたい」という気持ちは、人間の根源的な欲求と言ってもいいでしょう。

そして、この「秘密を守りたい」を技術的に実現するのが、暗号なのです。

例が、僕っぽい

僕っぽい言い回しだが、
内容が違う

僕っぽい

暗号化の基本的な考え方

暗号化の基本は、とてもシンプルです。

まず、伝えたい内容のことを「平文（ひらぶん）」と呼びます。「今日の作戦は東から攻める」とか「好きです付き合ってください」とか、そういう生のメッセージですね。

この平文を、なんらかの方法でグチャグチャにして、第三者が読んでもわからない状態にします。このグチャグチャにしたものを「暗号文」と呼びます。そして、グチャグチャにする操作のことを「暗号化」と言います。

逆に、暗号文を元の平文に戻す操作を「復号（ふくごう）」と言います。暗号化の逆ですね。復号化と呼ぶ人もいますが、正式には「復号」です¹。

つまり、暗号の基本的な流れは、こうです。

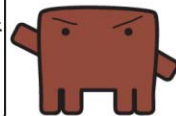
平文 → 暗号化 → 暗号文 → 復号 → 平文

間違っているとい
うか正確でない

流れも自然。そん
にへんなことも言
ってない。でも、
趣旨がよくわか
ない。「人間は大昔
から」のくだり要
る？

Claude Code と 大澤文孝が共闘したもの ニャゴロウに破れた話

モウフ
技術ノ
秘密基地



I'm always hallucinating
right after I wake up.

@mofukaburTech
@shigetaka256
@sour23



▲ 「モウフカブール」のブースで頒布中

できた原稿

送りたい人が平文を暗号化して暗号文にし、受け取りたい人が暗号文を復号して平文に戻す。途中で誰かに暗号文を見られても、復号の方法を知らなければ読めない。これが暗号の基本です。

What is programming?

グチャグチャにして、元に戻す。シンプル!



にゃごろうさん

暗号の三要素

さて、暗号化には三つの要素があります。

にゃごろうさんの台詞が
勝手に作られたのは正
直、驚いた!

叩く、直す

- 事実に事実を重ねただけ
- 用語解説が足りない（解説が後ろに来ている）
 - 知らない人に向けて書いてない
- 筋が通っておらず（ストーリーがない）
 - 意図がわからない

薄っぺらい



修正を入れて直す

- 原稿に「@」などでコメントを入れる
- プロンプトで「@の指示を直して」と入れる

@ 「長く育てる」みたいなへんな言い回しはしない@

つまり、「Windows環境を長く育てる」のではなく、**「ちょっと試して、捨てる」** ことに特化した仮想マシンが、Windows Sandboxです。

修正を入れて直す

@下記の箇所だが、読者は、「なぜ軽いのか」とか、そんなに詳しく知りたくない。ただ使いたいだけなので、もっと簡素に。ホストからリンクで共有しているからファイルサイズが小さい、メモリを共有してるから少ないメモリで済むなど、メリットとデメリットを挙げておくだけでいい。あと、ふつうのひとは、ntdll.dllをSandbox内で読み込んだときき言わなくていいので、きちんと解説するか割愛するかして@

前後関係が
おかしい

この用語初出

レベル感が合わ
ないことは
書かない

…略

ポイントは、Microsoftの「[Windows Sandbox] (https://docs.microsoft.com/ja-jp/windows/security/application-security/application-isolation/what-is-windows-sandbox-architecture)」を参照)。

****①動的ベースイメージ (Dynamic Base Image) ****

Windowsのシステムファイルの大部分は、インストール後に書き換わることのない「不変」のファイルです。Windows Sandboxは、これらを ****ホストからリンクで共有**** します (コピーしません)。書き換わる可能性のある一部のファイルは、まったくのコピーを持ちます。だから「Windowsを丸ごともう1セット」持つ必要がなく、ディスクは節約できます。これが、****ホストと同じビルドのWindowsしか動かせない**** 理由です。

****②メモリの共有****

ホストと同じOSバイナリを使っているため、たとえば、****同じ物理メモリページ****をそのまま使います (direct mapping)。ふつうのプロセスのように、ホストの状況に応じて動的に

****ホスト
当ては**



最終的に自分で書く

- できた原稿はあくまでも下書き
- それを参考に書き直す（リライトする）

全部がプロンプトで直るほど甘くない



労力の削減になってるの？

- 下書きがあるのとないのとで雲泥の差
 - 調査や下調べが終わっている
 - 画面キャプチャやイラストの下書きがある（後述）
 - サンプルのソースコードも作られている（後述）

材料はできてる！
それを再構成する
だけでよい



仕上げの検証

これから原稿を提出するので校正してほしい。
校正した結果は、僕が反映するので、直接直さないでほしい。
そして事実関係の調査もお願いしたい。
結果は「校正結果.md」としてまとめて。
検索しやすいよう、校正結果は、「元の文」「置換提案文」「理由」の
フォーマットとして。

校正：表記ゆれを一括チェック

指示：「原稿の表記ゆれを洗い出して。一覧（語・件数・行番号・例）を報告して」

実出力

検出結果（Fable 5：18項目から抜粋）：

- ・「Windows Sandbox」15件 / 「Windows サンドボックス」2件が混在
- ・「デフォルト」「既定」「初期状態」が同じ意味で1件ずつ
- ・「マウント」2件 / 「共有」3件（MappedFoldersの説明で混在）

例）L185「Hyper-Vや VirtualBoxで」 — 「や」の後に半角スペースが紛れ込み

例）「ブラウザ」だけ長音なし（他は「フォルダー」「インストーラー」で統一）

人間の目は
見落とすが、
機械的な検出は
得意分野

参考：WORDへの変換

- サンプルのWORD原稿（過去のWORD原稿）のレイアウトをそのまま採用した「WORDファイル」に変換できる
 - お手本があれば、それを真似るのが得意
- ページ数の調整もできる

できたWORDのページ数を3ページに収めたい。僕が編集するので、3ページよりも少し多めぐらいまでにカットしてほしい

図、サンプル、画面キャプチャ

図版：.drawioを直接書かせる

- .drawioファイルの実体はXML → Claude Codeが直接生成・編集できる
- たたき台を作らせて、人がdraw.ioで開いて仕上げる

実出力

```
<mxCell id="host" value="ホストのWindows 11（ふだん使っているPC）"  
  style="rounded=1;fillColor=#dae8fc;strokeColor=#6c8ebf;..." />  
<mxCell id="sbx" value="Windows Sandbox<br>（使い捨てのWindows）"  
  style="rounded=1;fillColor=#d5e8d4;strokeColor=#82b366;..." />  
<mxCell id="fresh" value="起動するたびに<br>まっさらなWindows" ...
```

開けばそのまま
編集できる図に
なっている

図版の実例

claude-work.wsb

```
<Configuration>
  <MappedFolders>
    <MappedFolder>
      <HostFolder>C:\sandbox-share
      <SandboxFolder>C:\share
      <ReadOnly>false
    </MappedFolder>
    <LogonCommand>
      <Command>C:\share\setup.cmd
    </Command>
  </LogonCommand>
</Configuration>
```

ダブルクリック（または wsb start --config）

① Sandbox（VM）が起動

② MappedFolders を
Sandbox内にマウント

③ 既定ユーザーで自動ログオン
(WDAGUtilityAccount／管理者)

④ LogonCommand を実行
(例：setup.cmd)

ホストの C:\sandbox-share
(setup.cmd など置いておく)

ホスト側フォルダは事前に存在が必要。書き込み
可にすると、Sandboxを閉じて中身は残る

複雑な処理は、マップしたフォルダに置いたスク
リプトを呼ぶのが定石（インストーラーの取得と
実行など）

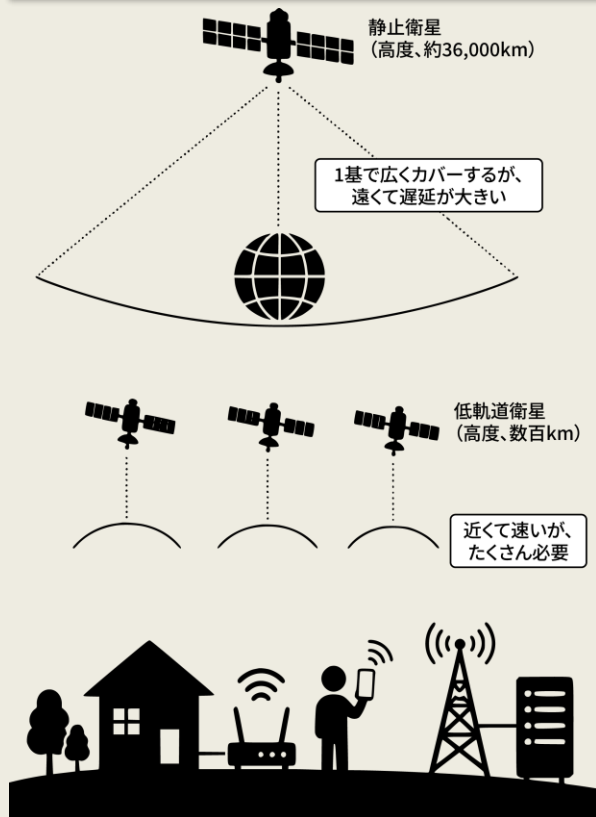
図版：APIキーで画像も作らせる

- ChatGPT（画像生成API）のキーをClaude Codeに渡す
- Claude Code自身がAPIを呼び出して、挿絵やイメージ図を生成
 - ClaudeCodeがChatGPTへのプロンプトを生成して、そのプロンプトで作らせる

安全のため
10ドルなど安価な
課金をしたキーを
渡すのが無難



画像の実例



Claude Codeが
絵を描くための
プロンプトを自分
で考えて渡す

縦の構図。最上部に静止衛星（高度約3万6千km）」、地球の広い範囲を覆う扇形の線、注釈「1基で広いが、遠くて遅延が大きい」。中段に低軌道衛星を3機並べ、ラベル「低軌道衛星（高度数百km）」、それぞれ小さな扇形、注釈「近くて速いが、たくさん必要」。最下部に地球の地表と家・スマートフォン。

スタイル：細い均一な黒線のみ。白背景。影・色・グラデーションはなし。絵文字レベルの文字。重要文字は太く、網の中の文字は細く、吹き出しや崩れが「経線」スマー。長方形の箱。点目+単純な体の輪郭。

EPSに変換し、
Illustratorで文字
の打ち直し



画面キャプチャは、執筆最大の苦行

- 手順書どおりに操作して、1枚ずつ撮る
- アプリのUIが変わったら、最初から全部撮り直し
- 入門書は特に枚数が多い（1冊で100枚近いことも）

撮り直しもつらい



Playwright MCPで自動操縦

①指示

「この手順で操作して
スクショを撮って」



②自動操作

Claude Codeが
ブラウザを操作

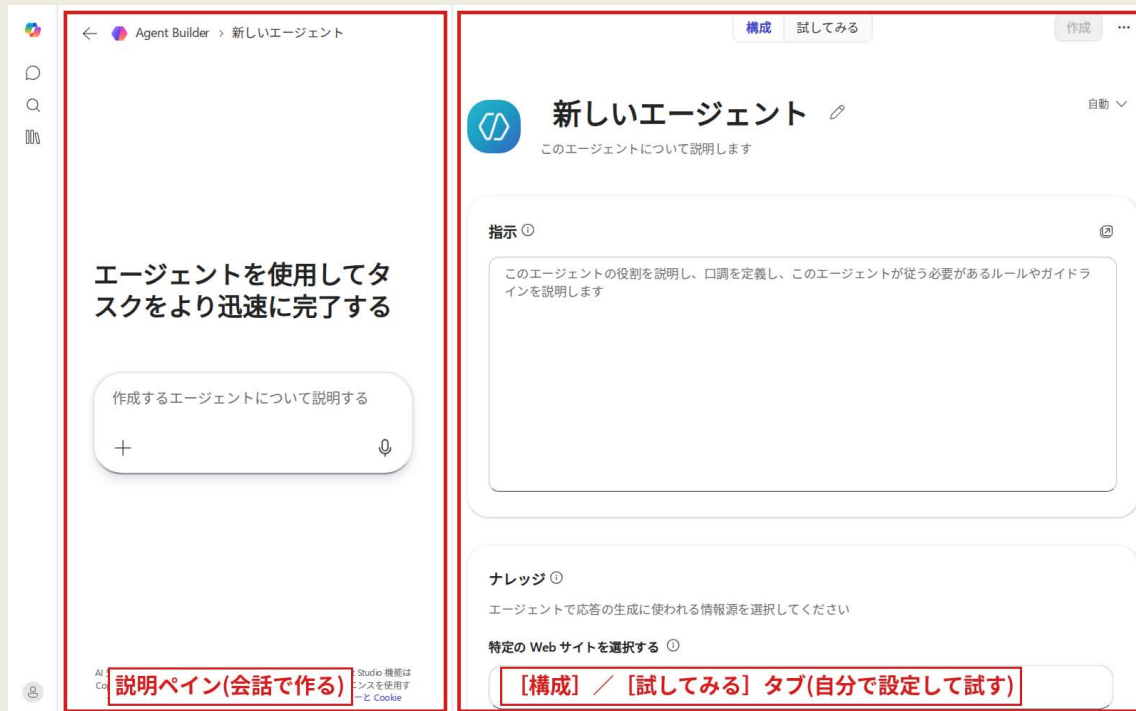


③完成

連番キャプチャが
フォルダに揃う

- 撮り直しも「もう1回」と言うだけ
- 手順書どおりの操作＝読者と同じ画面が撮れる

キャプチャの実例

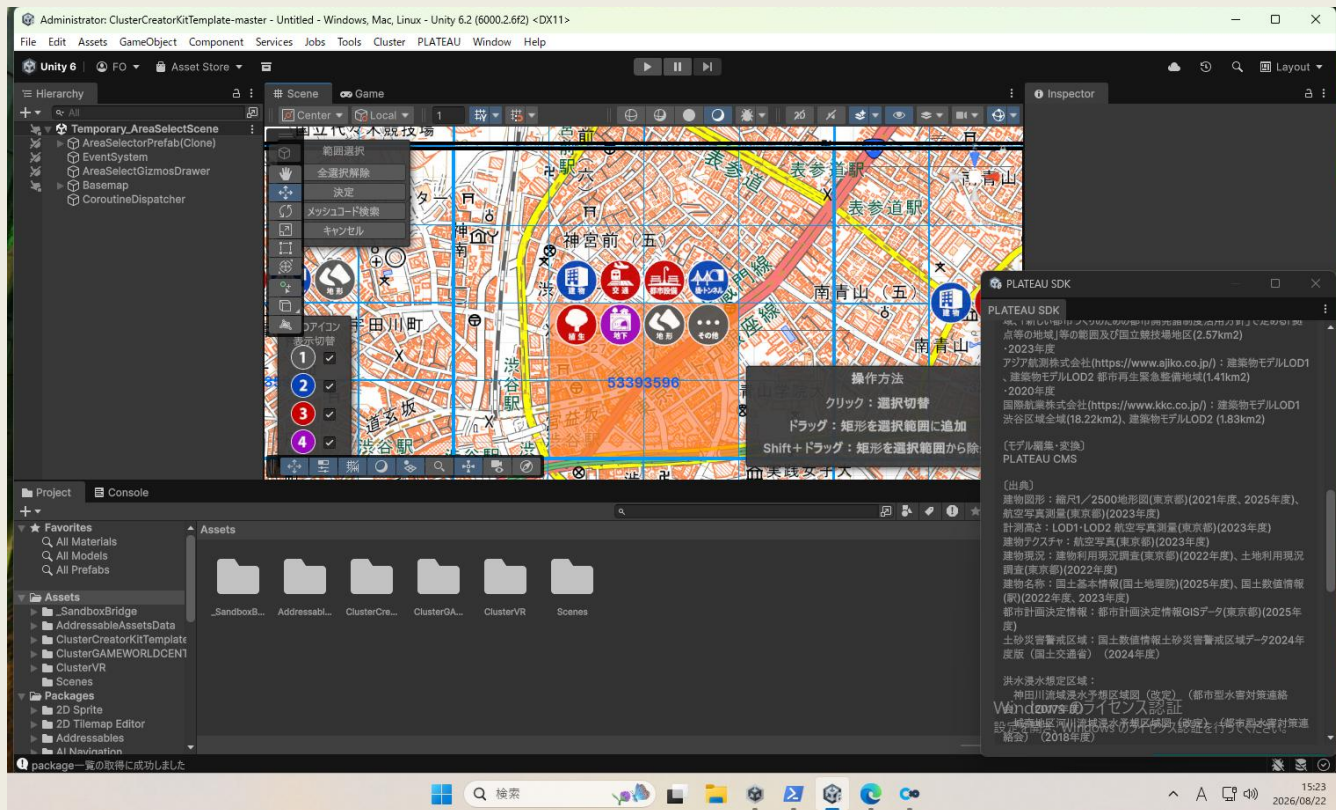


もし解説文が必要だと判断したら、囲みなどで解説文も重ねること。
その場合、オリジナルの画像も残しておくこと

Sandbox連動でアプリ画面も撮る

- ブラウザ以外のアプリは、Playwrightでは撮れない
- そこで：Windows Sandbox + 自動操縦MCP
- 使い捨て環境の中でアプリを操作させて、キャプチャを撮る
- 「まっさらな環境の画面」が撮れるのが魅力
 - インストール直後の画面 = 読者が見る画面

Sandboxを使ったキャプチャ例



サンプルは作って動かして確かめる

- コードを生成するだけでなく、その場で実行して検証まで任せる
- エラーが出たら、自分で直して再実行してくれる
- 「動かないサンプルを載せる事故」を防ぐ

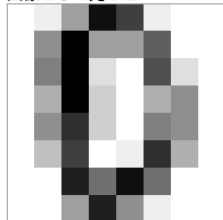
サンプルのバグ取
りから解放
確実に動くサンプ
ルが作れる



サンプルの実例

- Pythonのコードのほかグラフや検証の図なども作れる

画像として見ると



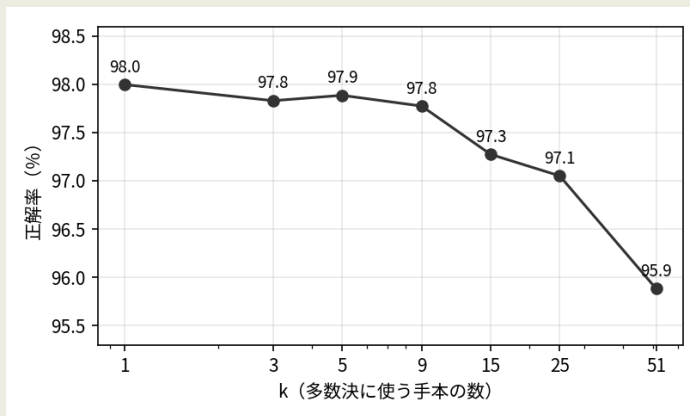
=

中身は8×8の数値 (0~16)

0	1	6	15	12	1	0	0
0	7	16	6	6	10	0	0
0	8	16	2	0	11	2	0
0	5	16	3	0	5	7	0
0	7	13	3	0	8	7	0
0	4	12	0	1	13	5	0
0	0	14	9	15	9	0	0
0	0	6	14	7	1	0	0

横一列につなげると――

[0, 1, 6, 15, 12, 1, 0, 0, 0, 7, 16, 6, 6, 10, …(中略)…, 7, 1, 0, 0] ←64個の数値の並び=特徴量ベクトル



校正作業

PDFを投げて校正

編集から戻ってきた初校を「初校フォルダ」に置いた。内容を精査し、誤記がないかを確認してほしい。

また事実関係の確認もしてほしい。

結果は「初校結果.md」に出力し、「該当のページ番号」「該当の元の文（検索用）」「修正案の文」「理由」を出力してほしい

WORDのコメント入り校正もできる

WORDの校正機能を使って、修正提案を書き込んで



その他の作業

取材記事

- 録音したMP3ファイルを置く
- Whisperで文字起こし
- 過去の下稿をフォルダに入れてトクマナを抽出
- インタビューイについての情報、この取材の趣旨についての情報をMarkdownに書く
- 上記をまとめてClaude Codeに投げて原稿執筆

プレゼン資料

- Marpを使ってMarkdownで下書き
- 過去のPowerPointを「このPowerPointファイルのフォーマットに従って」と指定して、PowerPointに変換
 - 公式スキルを使うことで「編集可能なpptx」ができる
- PowerPointで開いて微調整（以降はClaude Codeには戻らない）
 - PowerPointのClaudeプラグインを使う

同人誌の制作

- 写真集のレイアウト
- 過去の完成品PDFと写真をまとめて渡して、適当にレイアウトしたIDML形式ファイル（XML）に変換してもらう
- それをInDesignで編集
→ズレずに配置した配置ができる



まとめ

まとめ

- 技術書執筆の全工程で、Claude Codeは使える
 - 構成案・執筆・図版・キャプチャ・サンプル・校正
- まねっこが上手
 - トンマナなど過去のサンプルに似せたものを出力する
- Playwright、Windows Sandboxなどを使って画面キャプチャを自動化する
- テキストになっていれば何でもできる
 - WORD、Excel、PPTXなど（これらはZIPで固まったXML）
 - drawioで図を作る

告知

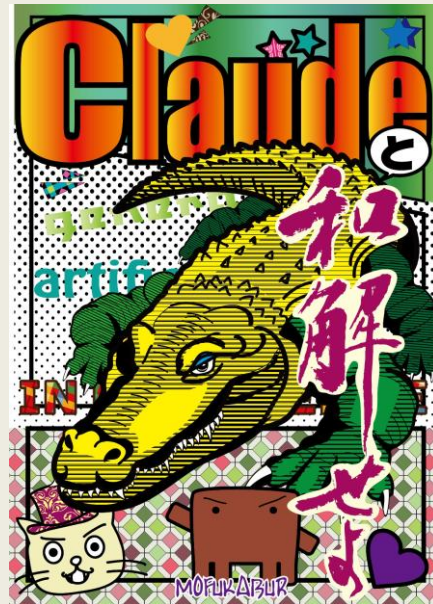
技書博
GISHOHAKU

第十四回 技術書同人誌博覧会

2026.09.13 Sun. 11:00~16:00

@大田区産業プラザPiO 大展示ホール

技書博14にサークル参加する



▲新刊予定

月刊I/O 50TH 特別展示

過去50年分のI/Oが読める



ご清聴ありがとうございました



2026.08.29 ODC 2026
書籍 Claude Code実用入門（マイナビ出版）

書籍も
よろしくね！

モウフカブール
<https://mofukabur.com/>



Follow me

