

第4章

Input : 高速インプット



4-1. Input は、書く前の戦闘準備

第3章で、旗、面、軸、マスを決めて、商品企画ブリーフにまとめました。

ここで言う「マス」は、今回書こうとしている本の対象となる、具体的な読者群のことです。

書きたいことの輪郭は、もう見えています。

ただ、ここで書き始める前に、もう一段、書く前のインプットを積んでおきます。

書く前のインプットの量と質が、書き上がる本の厚みを、そのまま決めます。

経験的な話ですが、私の中ではかなり堅い実感です。

もう1つ。

すでに自分が詳しいと思っているテーマでも、書く前にきちんと調べ直してください。

自分が知らなかった観点、まだ言葉にできていなかった切り口は、必ず出てきます。

マスが決まったから、書く前に動く

自分の頭の中のネタだけで書き始めると、「すでに自分が知っていること」しか書けない本になります。

読者がほしいのは、読者自身も薄々気づいているけれど、まだ言葉にできていないことです。

言い換えると、あなた自身の示唆や解釈です。

これは、書く前にインプットを丁寧に積まないと、なかなか拾えません。

私自身、最初の数冊で、やらかしたことがあります。

書く前のインプットを適当に済ませて、自分の知っているネタの組み合わせだけで書き上げてしまったのです…。

書き上がりは速かったのですが、出してみると、読者の心の真ん中には届きませんでした。

レビューも増えず、読まれたページ数(KENP)も伸びませんでした。

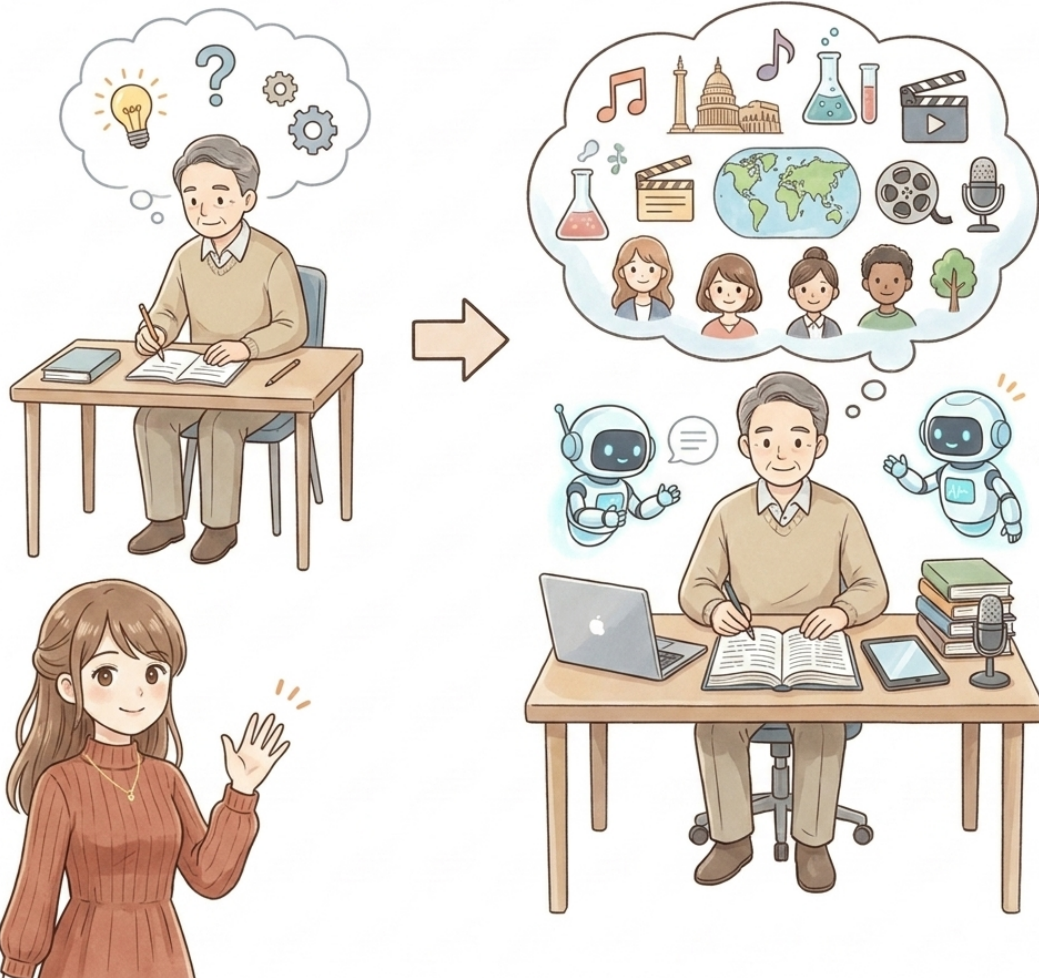
(お恥ずかしい限りです)

逆に、書く前にインプットを丁寧に積んだ本は、書いている最中から「これは届く気がする」という体感が違います。

書き上げた本も、長く読まれます。

頭の中だけ

インプットで厚くなる



Input は、判断ではなく材料集め

Input は、刊行判断ではなく材料集めです。

ただし、集めた素材の中から「これは使えそう」「これは違う」と仕分ける作業は、この Input フェーズで行います。

「この本を書くべきか」「ここで撤退すべきか」のような判断は、この章では扱いません。

それは次の第 5 章「Value Proposition」で詰めます。

第 4 章でやることは、3 つだけ。

1 つ目、知らないことを減らす。

2 つ目、その領域の観点や切り口を、広く集める。

3 つ目、書く前に、材料をそろえる。

これだけです。

これは AI テーマに限った話ではありません

私のシリーズは AI 活用が多いので、この章の事例も AI に寄りがちです。

でも、英語学習でも副業でも、節約でも HSP でも、ダイエットでも恋愛でも、構造は変わりません。

(ただし、健康・医療・法律・投資など、読者の判断に大きな影響を与えるテーマでは、一般記事や SNS だけでなく、公式情報・専門家情報・最新の一次情報を必ず確認してください。)

マスが決まっているからこそ、そのマスの領域に深く潜って、観点と切り口を集める。

やることは、これだけです。

この章で AI の例が出てきたら、ご自身が書こうとしているマスに置き換えて読んでください。

インプットの量と質が、書いた本の厚みを決める

「もし 8 時間で木を切れと言われたら、最初の 6 時間は斧を研ぐのに使う」。

誰の言葉かは諸説ありますが、この発想は、Kindle 出版にもそのまま当てはまります。

本を書く時間が 8 時間あるなら、MTIV に 6 時間、S に 2 時間。

これくらいの配分が、私の実態です。

書く工程(S)は、MTIV が整っていれば、本当に短く済みます。

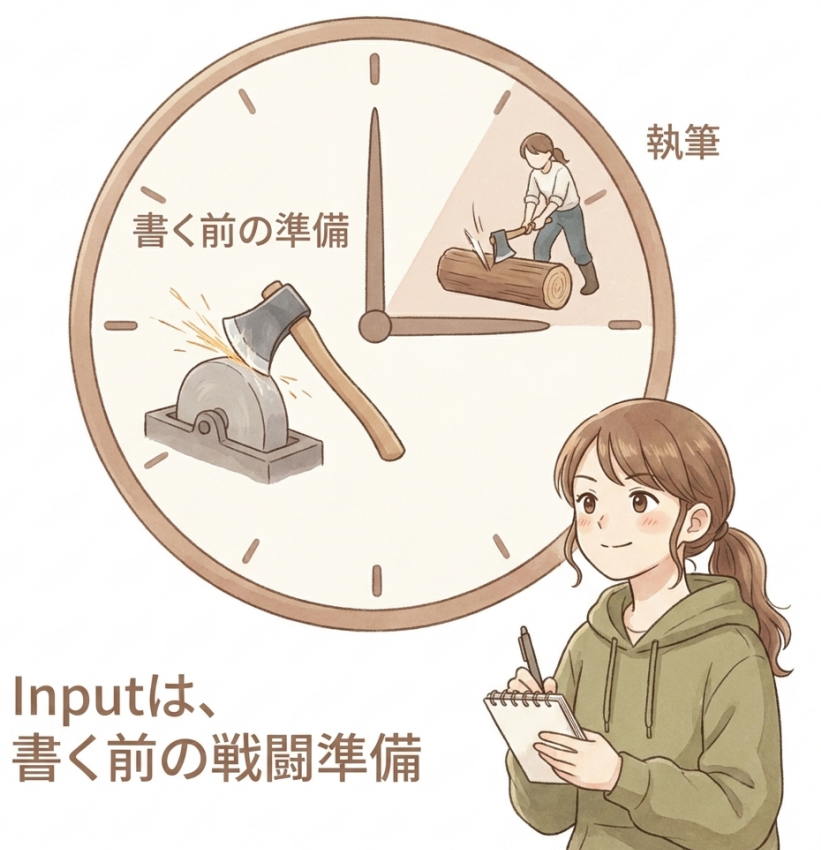
ただし、インプットを「丁寧にする」と、インプットの「集め方を非効率にする」ことは、別の話です。

集める段階は、徹底的に効率化していい。

読み込む段階は、丁寧に進める。

これが、自分出版社の編集長として、Input フェーズで意識すべきことです。

次の節から、私が普段使っている 6 つのインプット動線を、開いていきます。



4-2. インプットは 6 つの動線で持つ

6 つの動線で、書く前のマスを読み込む

先に、6 つの動線を並べておきます。

便宜上、自分の内側を見る動線を「動線 0」、外側に出る動線を「動線 1～5」と呼びます。

動線 0: 自分の本棚と読者の反応

動線 1: Google 検索と記事リサーチ

動線 2: Kindle Unlimited で関連書籍を流し読み

動線 3: X と Grok で対象アカウントを読み込む

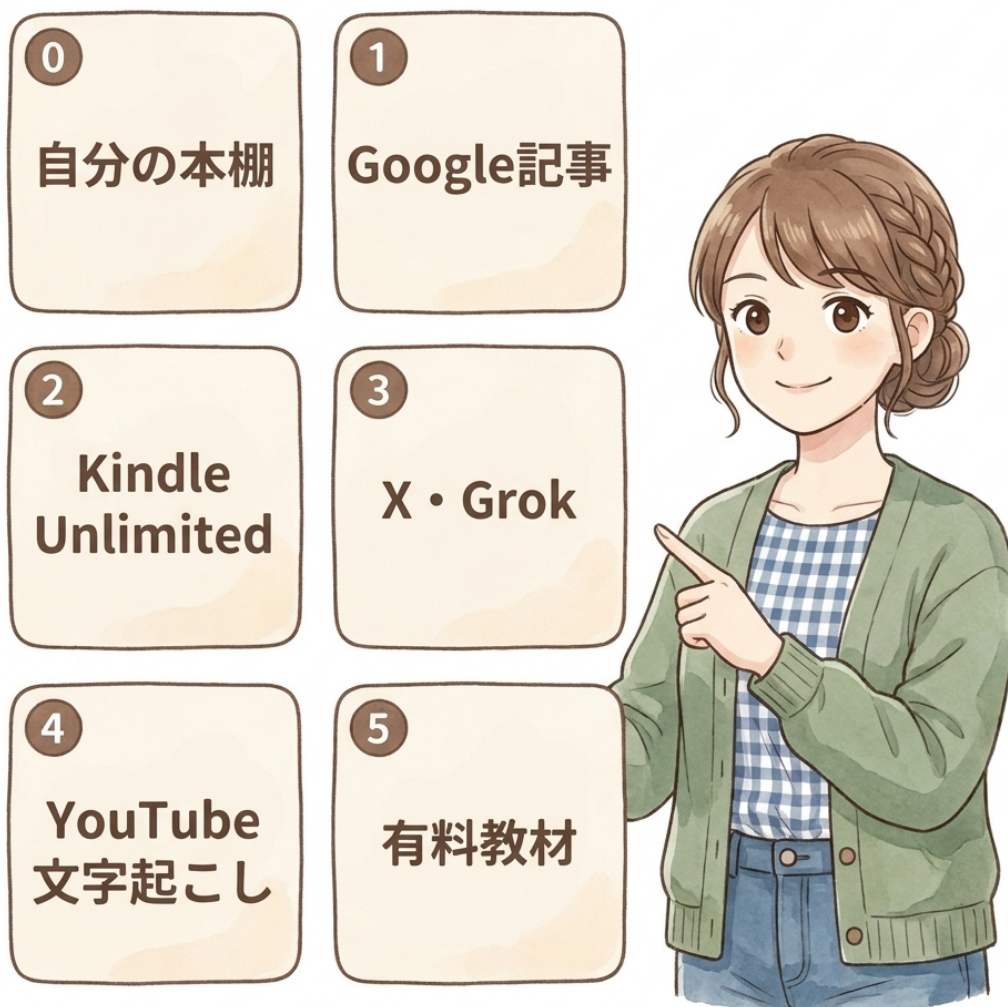
動線 4: YouTube を文字起こしして読む

動線 5: Brain や有料教材にも、必要に応じて投資する

動線 0 は、外部に出る前に、自分の本棚の中にすでにある一次情報から拾います。

動線 1 から 5 は、その後に外部に出ていく動線です。

順番に開いていきます。



自分の本棚と読者の反応から始める(動線 0)

外に出る前に、まず内側を見ます。

見るのは、3 つです。

1 つ目、過去の本のレビュー。

良かった点と、物足りなかった点。

2 つ目、SNS の反応。

読者の生の言葉。

3 丁目、KDP の数字で、長く読まれ続けているテーマ。

これらは、外部リサーチでは手に入りにくい、自分だけが持っている一次情報です。

マスにいる読者と、すでに何らかの形で接点を持っているからこそ、その接点の中に、次の本に使える材料がたくさん埋まっています。

ただし、レビューを見るときは、1 点だけ。

明らかに根拠のない辛口レビューは、本の材料にはなりません。

具体的に何が物足りなかったかが書かれていない、感情的な一言だけが並んでいるようなレビューは、深刻に受け止めなくて大丈夫です。

気にしすぎると、こちらの編集判断がブレます。

(とはいえ、★1 のレビューは凹みます。これは本当に凹みます。)

参考にするのは、具体的な不満が書かれているレビュー、もしくは「ここが良かった」と具体的な観点が書かれているレビューです。

補足ですが、動線 0 は、ある程度の本棚がある方向けの動線です。

まだ数冊しか出していない方は、動線 1 から始めて大丈夫です。

本棚が育ってきたら、また動線 0 に戻ってきてください。

本棚が動線 0 で機能し始めると、外部リサーチの精度がぐっと変わります。

「あ、この観点は前の本のあのレビューでも見たな」「あの質問とつながるな」と、外と内が結びついていきます。

Google 検索と記事リサーチ(動線 1)

動線 1 は、調査部のデジタル社員に、記事を集めてもらう作業です。

「このマスの読者にとって役立ちそうな記事を、まず 20～30 本集めてください」と依頼します。

方向性が合っていれば、そこから 50 本、100 本へ広げていきます。

集めるときのプロンプトの作り方にはコツがあります。

これは次の 4-3 で独立して扱います。

調査部のデジタル社員に任せきりにせず、地味な Google 検索の目視確認も併用します。

100 本のリストを眺めたあとで、Google で関連キーワードを検索して、自分の目で記事を選ぶ作業を、必ずやります。

自分の目で記事のタイトルや構成を眺めることで、「あ、この切り口は思いつかなかった」「この記事の見出しの並びは強い」と気づくことがあるからです。

整理されたリストだけを見ていると、こういう発見の角度が落ちます。

集める部分は効率化していい。

ただし、選ぶ部分は手作業で残しておく。

私はこれが、安定的な運用だと感じています。

英語学習でも、副業でも、節約でも、構造は変わりません。

「30 代社会人 独学 英語」「会社員 副業 月 3 万円」「一人暮らし 節約」のようなキーワードで、自分の目で記事のタイトルを並べてみる。

これだけで、マスの読者がどんな悩みを抱えていて、どんな事例にすでに触れているかが、見えてきます。

集める＝効率化



選ぶ＝手作業



Kindle Unlimited で関連書籍を流し読み(動線 2)

動線 2 は、Kindle Unlimited で関連書籍を 10 冊くらい流し読みする作業です。

私の感覚では、まったく未知のテーマでも、関連書籍を 10 冊ほど流し読みすると、そのテーマの大まかな輪郭は見えてきます。

もちろん、専門性の高いテーマでは、それだけで十分とは言えません。

ただ、書く前に市場の地図を作る目的なら、まず 10 冊をざっと読むだけでも、かなり見える景色が変わります。

10冊の本に共通して書かれている観点。

逆に、ある本にしか書かれていない独特の切り口。

読者からのレビューで繰り返し言われている不満。

これらを並べていくと、マスの読者が「すでに知っていること」と「まだ言葉にできていないこと」の輪郭が、立体的に見えてきます。

流し読みのときは、「あ、この観点あるな」と思った瞬間に、一言だけメモします。

長文のメモは要りません。

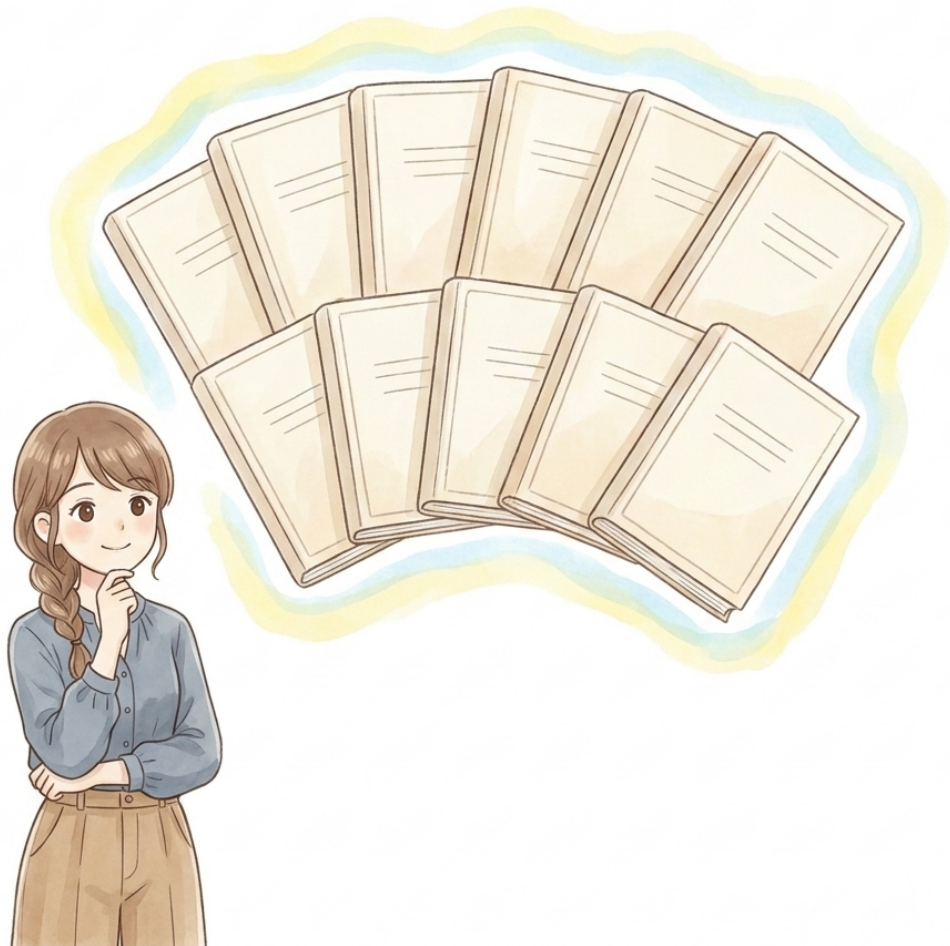
これも、テーマは選びません。

HSPの本を書こうとしているなら、HSP関連の本を10冊。

お金の本を書こうとしているなら、家計やお金関連の本を10冊。

Kindle Unlimitedに登録しているなら、追加コストはゼロでこれができます。

関連書を10冊、流し読み



X と Grok で対象アカウントを読み込む(動線 3)

動線 3 は、第 2 章 2-5 節のマーケ部のフェーズで見つけたトレンドメーカー候補のアカウントを、Input フェーズでもう一段深く読み込む作業です。

マーケ部のときの目的は、「動いている熱を見立てる」ことでした。

Input フェーズでは、目的が変わります。

マスの領域で、どんな観点や事例がすでに語られているのか、その領域の知識そのものを自分の中に蓄えるのが目的です。

私はここでは Grok をよく使います。

「このアカウントの過去半年の投稿の傾向を整理してください」「この発信者がよく扱っているテーマと、主張のパターンをまとめてください」と依頼します。

Grok で対象アカウントの投稿傾向を確認できる場合は、そのまま過去投稿の傾向を整理してもらいます。

ただし、アカウントの公開状況や Grok 側の仕様によって、直接読み込める範囲には限界があります。

その場合は、自分で気になる投稿の URL やテキストを集めて、Grok や Claude に渡して整理してもらいます。

大事なのは、特定ツールに丸投げすることではなく、対象アカウントの発信傾向を、自分の本の材料として読み解くことです。

英語学習であれば、英語学習の発信で伸びているアカウント。

副業であれば、副業の実践報告をしている会社員のアカウント。

HSP であれば、HSP の当事者として発信しているアカウント。

対象アカウントを 2～3 人読み込むと、その領域でいま語られている観点や、よく出てくる事例が、立体的に見えてきます。

YouTube を文字起こしして読む(動線 4)

動線 4 は、YouTube の動画を、文字起こしして読む作業です。

ポイントは、「動画を見る」のではなく、「文字起こしを読む」こと。

要約ではなく、発信者がどんな文脈で、どんな事例を語っているかまで読み取るイメージです。

その領域で語られている具体的な事例や切り口は、要約には乗ってこないことが多いので、フル文字起こしの方が、書く前の素材としては密度が高くなります。

文字起こしを取り出す方法は、2 つあります。

1 つは、Tactiq という無料サイトを使う方法。

もう 1 つは、Claude Code と、YouTube 音声を文字起こしするツールを組み合わせ、複数本を一気通貫で処理する方法。

2 つの違いと、それぞれの使い方は、次の 4-4 節で詳しく扱います。

Brain や有料教材も、選んで投資する(動線 5)

動線 5 は、Brain や有料教材を、必要に応じて買う作業です。

無料の情報だけでは情報が不足していると感じたら実践してください。

私自身、書く前の Input フェーズで、想定しているテーマに沿った Brain を買うことはあります。

普通に有料の書籍を Amazon で買うこともあります。

ただ、第 2 章 2-5 節でお伝えしたとおり、無料公開情報の中にも、公式発表や実践者本人の検証ログのように、一次性の高い情報はあります。

「必要だと感じたものに、自分のペースで投資する」というスタンスで OK です。

英語学習を書こうとしているなら、英語学習関連の書籍。

副業を書こうとしているなら、副業の有料教材。

HSP を書こうとしているなら、HSP 関連の書籍やサービス。

テーマに合わせて、必要なときに、必要なものを買う。

無料情報も、有料情報も、必要に応じて使い分ける。

これが、動線 5 の姿勢です。

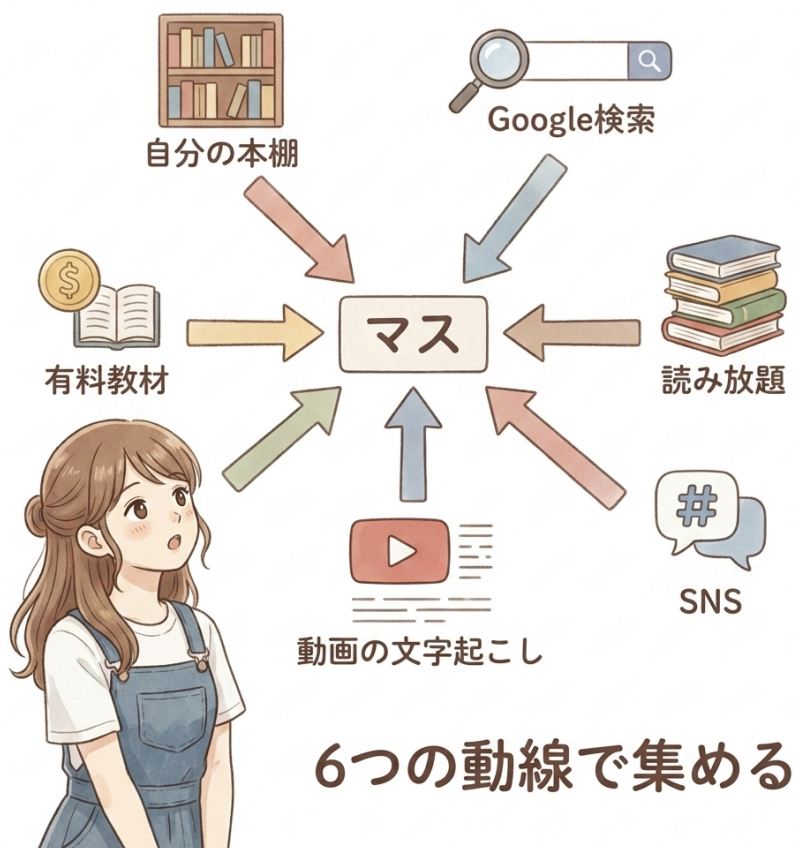
以上 6 つの動線が、普段私が使っているインプット動線です。

動線 0 で内側を見て、動線 1 から 5 で外側を読み込む。

さて、ここまでは、「動線そのもの」の話でした。

動線 1 (Google 検索と記事リサーチ) と動線 3 (X と Grok) では、「調査部のデジタル社員に、何を、どう投げるか」というプロンプトの作り方が、結果の質を大きく左右します。

次の 4-3 節では、その「マスから逆算する調査プロンプト」の作り方を、テンプレートと一緒に扱います。



4-3. 調査プロンプトは、マスから逆算する

4-2 でお伝えした 6 つの動線のうち、動線 1 (Google 検索と記事リサーチ) と動線 3 (X と Grok) では、調査部のデジタル社員に「何を、どう投げるか」が、集まる素材の質を大きく左右します。

ここでは、私が普段使っている「マスから逆算する調査プロンプト」の作り方を、テンプレートと記入例で渡します。

めちゃくちゃ時間をかけて集めたら、海外記事だらけだった日

最初に、私の失敗エピソードを 1 つ。

AI 活用シリーズの本を書こうとして、調査部のデジタル社員に「AI 活用について役立つような記事を 100 本集めてください」と頼んだことがあります。

時間はかなりかかりました。

トークン消費も、なかなかの量。

ようやく出来上がったリストを開いてみたら、半分以上が海外サイトの英語記事でした。

一応、AI 活用には違いない。

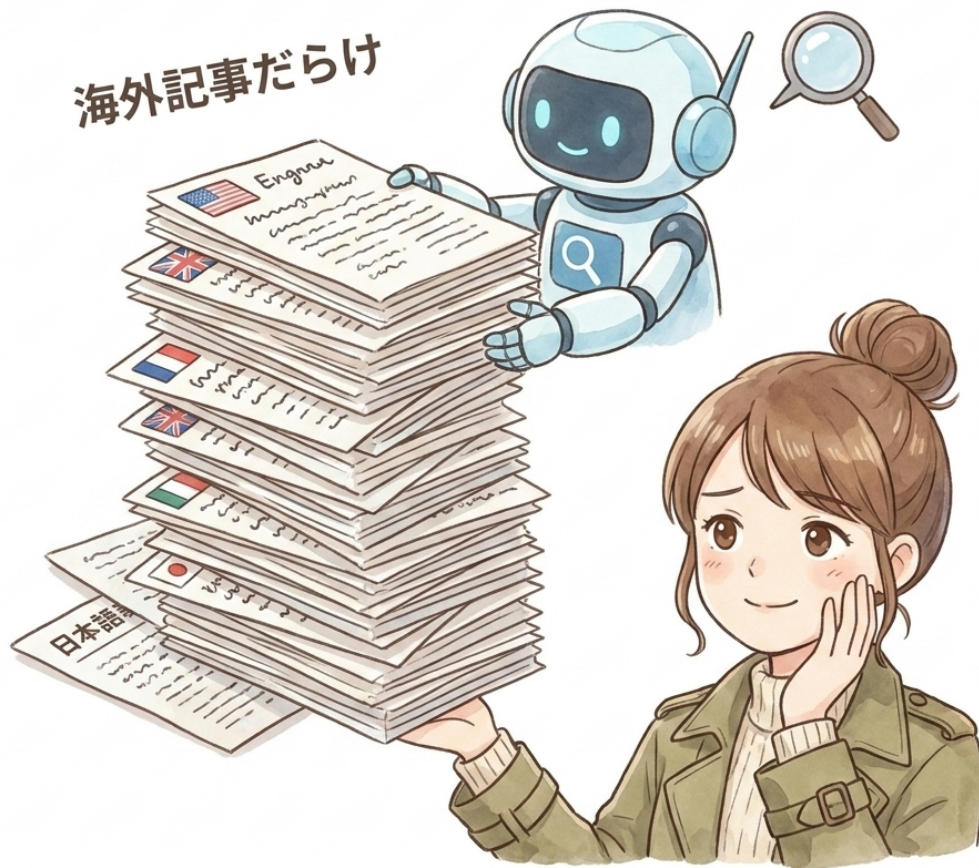
技術的には正確な内容も多い。

でも、私の本の読者は、日本在住の会社員の方が中心です。

「日本のオフィスでこう動く」「日本の中小企業ではこんな運用が現実的」のような、現場感のある記事じゃないと、本に落とせる素材にはなりません。

このときは、『ああ、トークンをかなり溶かしてしまった……』と、普通に頭を抱えました。

テーマだけで投げると…



失敗の原因は「テーマだけで投げる」こと

なぜこうなったのか。

プロンプトに、「マス」と「日本語サイト」が入っていなかったからです。

私が渡したのは、「Claude の AI 活用について」というテーマだけでした。

これだと AI にとっては、海外の研究記事もニュースリリースも、AI 活用の範疇です。

むしろ、データ量の関係で、海外記事の方が拾われやすい構造になっています。

「マス」を渡していないと、AI はどの読者に向けて集めればいいのか分からない。

「日本語サイトで」と書いていないと、英語圏のソースも普通に混ざってきます。

これは、AI テーマだけの話ではありません。

英語学習でも、副業でも、「テーマだけで投げる」と、似たような脱線が起きます。

英語学習なら、TOEIC900 点向けの記事と中学英語のやり直しが混ざる。

副業なら、株式投資の話と内職の話が混ざる。

調査の前に、もう一段、プロンプトの設計をやっておくと、ここが変わります。

マスから逆算する、汎用プロンプトのテンプレート

対策はシンプルです。

マスから逆算してプロンプトを組みます。

第 3 章で決めた商品企画ブリーフを開いて、棚の段・読者の現在地・読者が欲しい変化・テーマを、すべてプロンプトに織り込みます。

加えて、「日本語記事で」「優先順位順に」「整理項目を指定」を、毎回入れます。

記事検索は、慣れている方は 100 本一気に実施しても構いません。

ただ、最初は 20～30 本で十分です。

方向性が合っていることを確認してから、50 本、100 本へ広げていくほうが安全です。

テンプレートはこれです。

私は今、(棚の段)で、(読者の現在地)の人が、(欲しい変化)に近づくための本を書こうとしています。

テーマは(テーマ)です。

この読者にとって役に立つ可能性が高い日本語記事を 20～100 本探し、優先順位順に並べてください。

優先順位は、「この読者マスへの近さ」「日本の読者にとっての再現性」「具体事例の多さ」「新しさ」の順で判断してください。

各記事について、URL、タイトル、誰向けか、見えそうな観点、注意点を整理してください。

一般論ではなく、この読者マスに役立つものを優先してください。

()の中に、自分のブリーフから言葉を持ってきて埋めるだけです。

商品企画ブリーフを北極星として置いた意味は、ここにつながります。

本を書くたびに、マスを言語化し直す必要がない。

ブリーフから言葉を写すだけで、プロンプトが組み上がります。

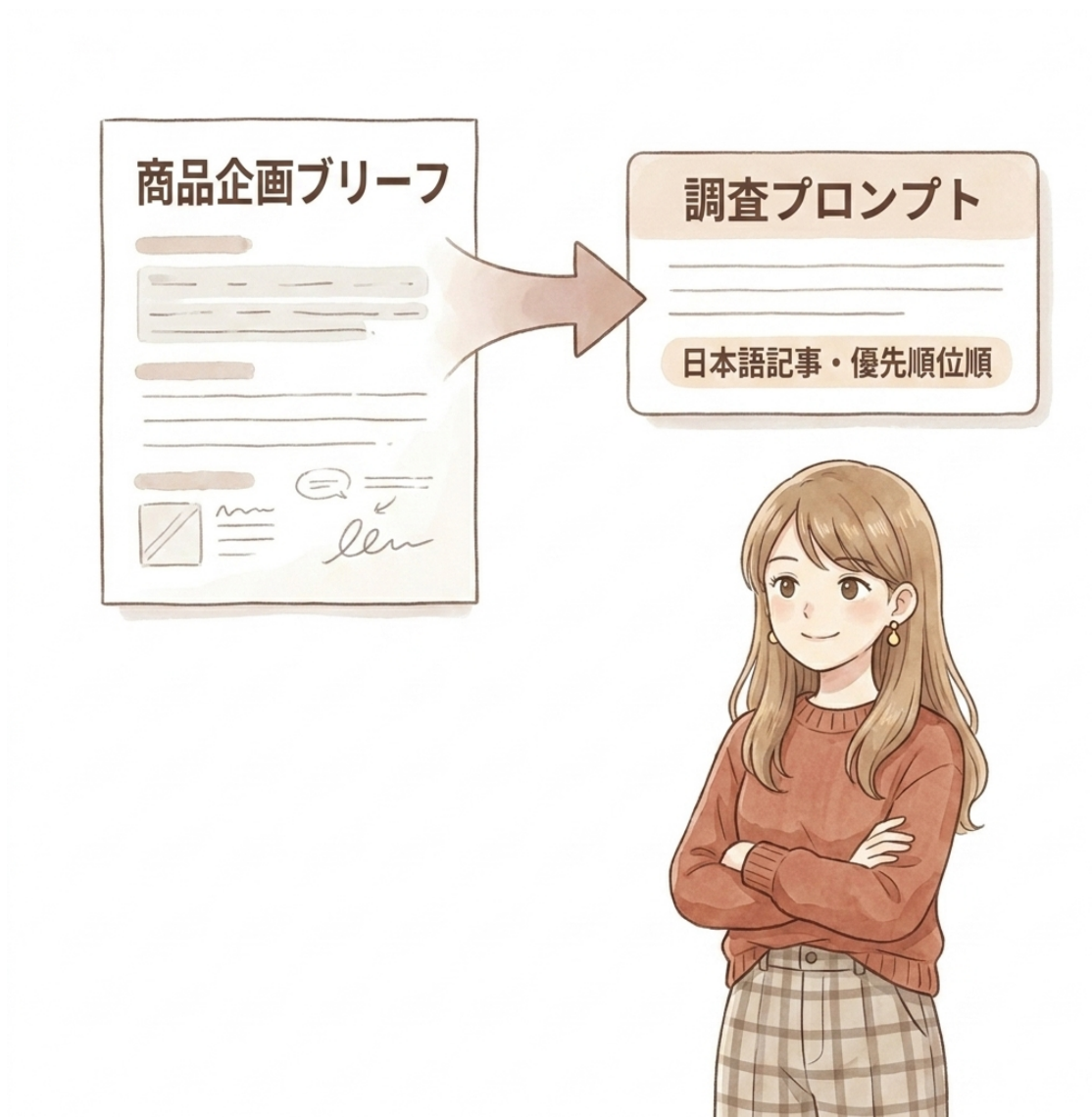
いきなり 100 本で重い場合は、まず 20～30 本で試しても大丈夫です。

方向性が合っていれば、そこから 50 本、100 本へ広げていきます。

逆に、最初の 20 本の時点でズレているなら、100 本集めてもズレた素材が増えるだけです。

X や Grok で使う場合は、「日本語記事」を「日本語の長文ポスト、スレッド、対象アカウントの投稿」に置き換えてください。

これだけで、動線 1(Google 検索と記事リサーチ)と動線 3(X と Grok)の両方で、同じテンプレートが使えます。



3 パターンの記入例で、感覚を掴む

実際に、3つのテーマで埋めてみます。

【記入例 1: AI 活用】

棚の段: AI 活用シリーズ

読者の現在地: AI を仕事で少し触り始めた会社員

欲しい変化: 日々の業務を効率化して、残業や精神的な負担を減らしたい

テーマ: 会社員の業務効率化に使える AI 活用

【記入例 2: 英語学習】

棚の段: 英語学習シリーズ

読者の現在地: これから独学で英語を始めたい

欲しい変化: 半年後にビジネスで使えるレベルに近づきたい

テーマ: 30 代社会人の独学英語

【記入例 3: 副業】

棚の段: 副業シリーズ

読者の現在地: 会社員で、副業をこれから始めたい

欲しい変化: 月 3 万円の副収入を半年で作りたい

テーマ: Kindle 以外の現実的な副業

3 パターン並べてみると、テンプレートの汎用性が見えてきます。

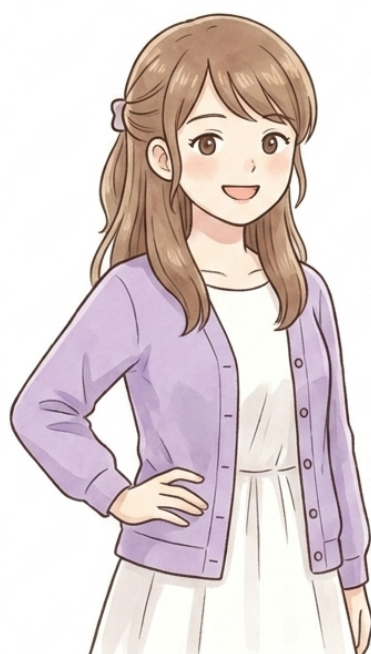
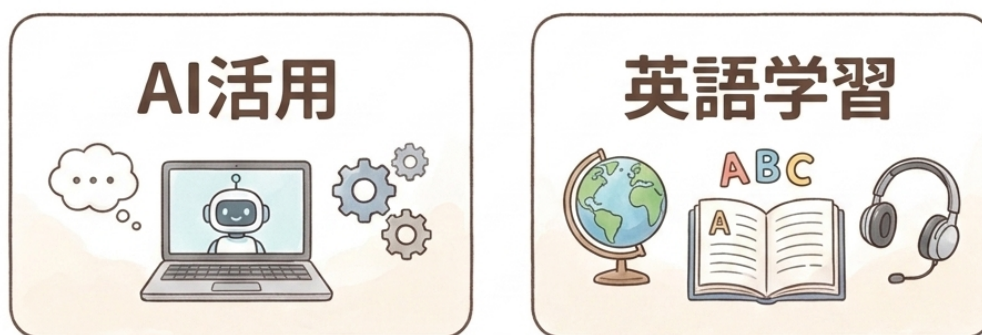
テーマが変わっても、() の埋め方は同じ構造です。

読者の現在地と欲しい変化、これさえブリーフから持ってこれれば、AI 活用でも英語学習でも副業でも、プロンプトはそのまま動きます。

この依頼を調査部のデジタル社員に渡すと、整理された 100 本のリストが返ってきます。

100 本といっても、すべてが本に使える素材になるわけではありません。

ざっと眺めて、「これは使える」「これは違う」を仕分ける作業が、このあとに残ります。



同じテンプレで動く

集めた 100 本をどう絞り込むかは、4-5 で扱います

ここまでで、動線 1 (Google 検索と記事リサーチ) と動線 3 (X と Grok) の「集める」プロンプトの作り方は、ひと通り渡しました。

集めた 100 本を、どう「これ使えるな」に絞り込んで、判断材料として整理するか。

これは、4-5 節「集めた情報を、整理された材料に変える」で扱います。

ただし、その前に、もう1つ寄り道があります。

書く前のインプットでは、YouTube も重要な動線でした。

動線 4 で軽く触れた「YouTube を文字起こしして読む」を、次の 4-4 節でもう少し具体的に開いていきます。

Tactiq という無料サイトを使う方法と、上級者向けに自動化する方法、2 つを並べます。

4-4 で YouTube の文字起こし方法を渡し終えてから、4-5 で集めた素材の整理に入ります。



4-4. YouTube 文字起こしは、まず Tactiq で十分

4-2 の動線 4 で軽く触れた「YouTube を文字起こしして読む」を、ここで具体的に開きます。

ここで覚えていただきたいのは、1 つだけ。

動画を見るより、文字起こしを読むほうが、圧倒的に効率的、ということです。

動画を見るより、文字起こしを読むほうが速い

なぜ文字起こしのほうが良いのか。

動画は、時間がかかります。

1.5 倍速で見ても、20 分の動画は 13 分かかります。

20 本見ようとする、それだけで 4 時間以上溶けます。

書く前のインプットに、毎回これだけの時間を使うのは現実的ではありません。

そして動画は、流れていきます。

「あ、この観点」と思っても、戻して、見直して、メモを取って、を繰り返すのが面倒です。

文字起こしなら、20 本分でも、見出しや気になる言葉だけなら、かなり短時間でざっと見渡せます。

検索もかけられます。

「この発信者は、ここでこんな言葉を使っている」と気になった瞬間に、コピーして自分のメモに貼れます。

前作の Brain の第 3 章追記で、NotebookLM に YouTube の URL を入れて要約してもらう手法を紹介しました。

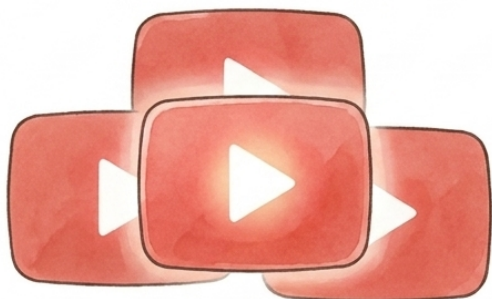
あれは骨子作成には便利な方法ですが、Input フェーズで使うには、要約されすぎるのが弱点です。

その領域で語られている具体的な事例や切り口は、要約には乗ってこないこともあります。

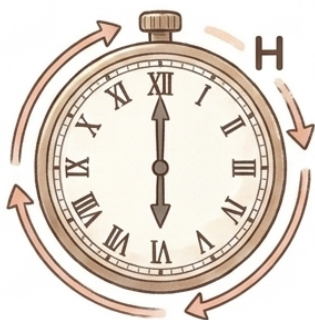
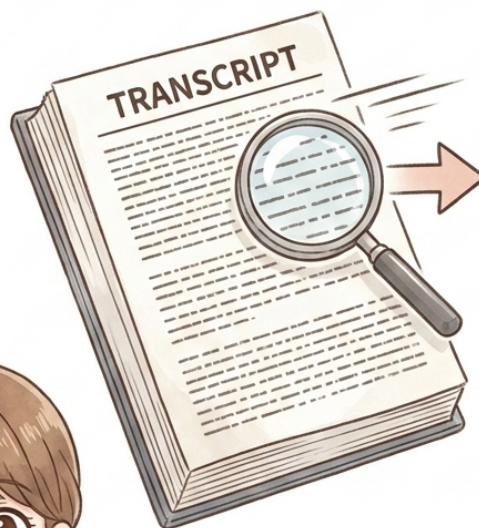
発信者の生の言葉を取りこぼさずに自分の中にインプットして、自分の言葉で語れるようにする。

これが、Input フェーズでフル文字起こしを取り出す目的です。

**動画を見る
(時間がかかる)**



**文字起こしを読む
(密度が高い)**



標準ルートは Tactiq、URL を貼り付けるだけで終わります

私が普段使っているシンプルな方法の 1 つが、Tactiq の YouTube 文字起こしツールです。

執筆時点では、YouTube URL を貼るだけで字幕テキストを取り出せる無料ツールとして使えます。

ただし、URL や無料範囲、仕様は変わる可能性があります。

使う前に『Tactiq YouTube transcript』などで検索して、最新のページを確認してください。

字幕が取得できる動画であれば、URL を貼るだけでテキストを取り出せます。

1 本ずつなら、これが最速です。

使い方は、こうです。

マーケ部 2-5 節で見当をつけたチャンネルを、Input フェーズでテーマワードで拾い直します。

たとえば英語学習の本を書こうとしているなら、「ビジネス英語 独学 30 代」のような検索ワードで YouTube を叩いて、目当ての発信者の動画を 5~10 本拾います。

その URL を、1 本ずつ Tactiq に貼り付ける。

字幕テキストが出てきたら、コピーして自分のメモ帳に貼る。

これだけです。

地味な作業ですが、地味なだけに、再現性は高いです。

読者の方は、まずここから始めるのがおすすめです。



動画を見るより、文字起こしを読む

上級者向けに、Claude Code+Whisper の一気通貫もあります

Tactiq で十分な方は、ここから先は読み飛ばして大丈夫です。

(Kindle 出版の成果を出すために必須の工程ではありません。)

「20 本まとめて文字起こしをかけたい」「50 本を一気に処理したい」みたいな場面が出てくると、1 本ずつ Tactiq に貼り付けるのが、少ししんどくなってきます。

そのときは、Claude Code で自動化する選択肢があります。

上級者であれば、Claude Code に相談しながら、yt-dlp や Whisper を組み合わせた文字起こし環境を作ることもできます。

ただし、これは PC 環境、Python のバージョン、YouTube 側の仕様、各ライブラリの状態によって挙動が変わります。

必ず同じように動くものではありません。

なので、本編では標準ルートを Tactiq とし、自動化は『環境構築に慣れている方の選択肢』として扱います。

具体的なプロンプトと手順は、4-7 節の末尾に上級者向け補足として置いておきます。

夜寝る前に走らせて、朝に文字起こしが揃っている運用

Claude Code + Whisper の良いところは、夜間運用ができることです。

50 本まとめて文字起こしをかけると、字幕 API で取れる動画はサクッと終わりますが、Whisper で音声解析する動画が混ざると、30 分～1 時間以上かかることもあります。

本業の合間や、寝る前の時間を、ここに使ってもらいます。

私の運用は、こうです。

夜、ClaudeCode にプロンプトを投げて、『あとは朝までに進めておいて』という感覚で寝ます。

朝、文字起こしが揃っているのを見ると、かなり助かります。

こういう処理を寝ている間に任せられるのは、AI 活用の大きな利点です。

コーヒーを淹れながら、ざっと眺める。

「あ、この発信者のこの観点、強いな」と思ったら、自分のメモに移す。

これは、テーマを選びません。

AI 活用でも、英語学習でも、副業でも、夜間に走らせて朝に拾う、という運用の形は同じです。

ただし、健康・医療・法律・投資など、読者の判断に大きな影響を与えるテーマでは、YouTube や SNS の情報だけで判断せず、公式情報や専門家情報を必ず確認してください。

50 本の動画を、起きている時間で 1 本ずつ見ようとしたら、それだけで 2 日溶けます。

寝てる間に文字起こしが揃っているなら、その 2 日は別のことに使えます。

Input フェーズの「集める」を、ここまで効率化できます。

集める部分は、ここまでです。

次の 4-5 節では、ここまでで集めた 100 本の記事と、20~50 本の YouTube 文字起こしを、どう判断材料として整理するかを開きます。

夜、走らせる

朝、そろっている



4-5. 集めた情報を、整理された材料に変える

4-3 で集めた記事 100 本、4-4 で集めた YouTube 文字起こし 20～50 本。

手元にはかなりの素材があります。

ただ、ここで集まったものは、まだ「材料」ではありません。

素材です。

これを材料に変える作業が、4-5 の話です。

もちろん、最初からここまで集めなくても大丈夫です。

まずは記事 20 本、YouTube 文字起こし 5 本でも十分です。

小さく集めて、小さく整理する。

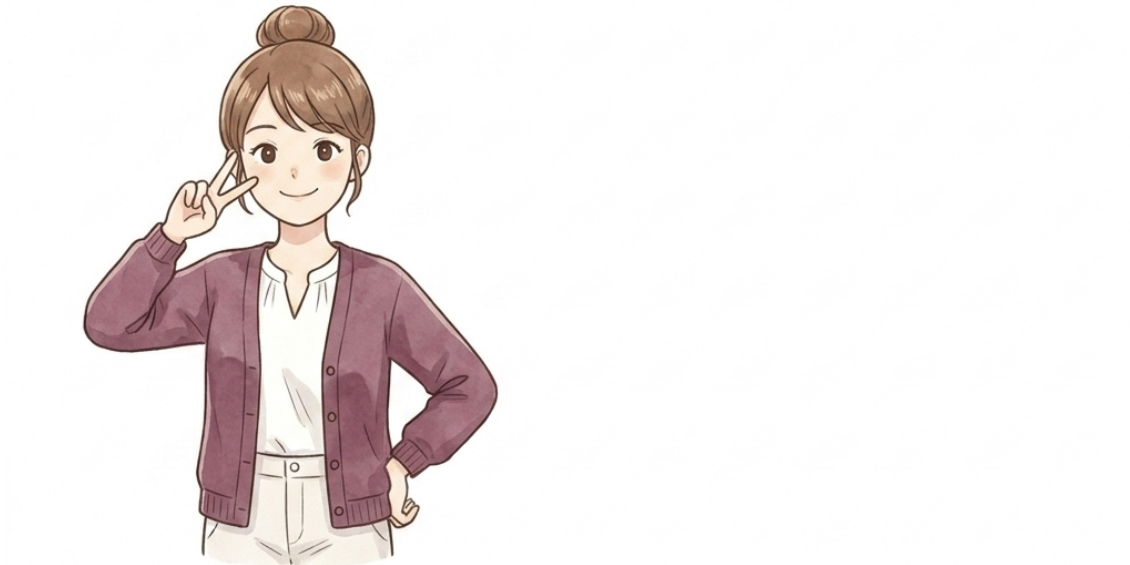
慣れてきたら、50 本、100 本へ広げていく。

この順番で構いません。

素材



整理された材料



AI は判断者ではなく、整理係

理想を言えば、調査部の Claude に集めた素材をぜんぶ渡して、「この読者マスの悩み 3 つ、検索される言葉、競合の弱点を、スパッと出してください」と頼めれば、編集長としての判断材料は一瞬で揃います。

ただ残念ながら、実際は、そんなにスパッととは出てきません。

私自身、何度も試してきました。

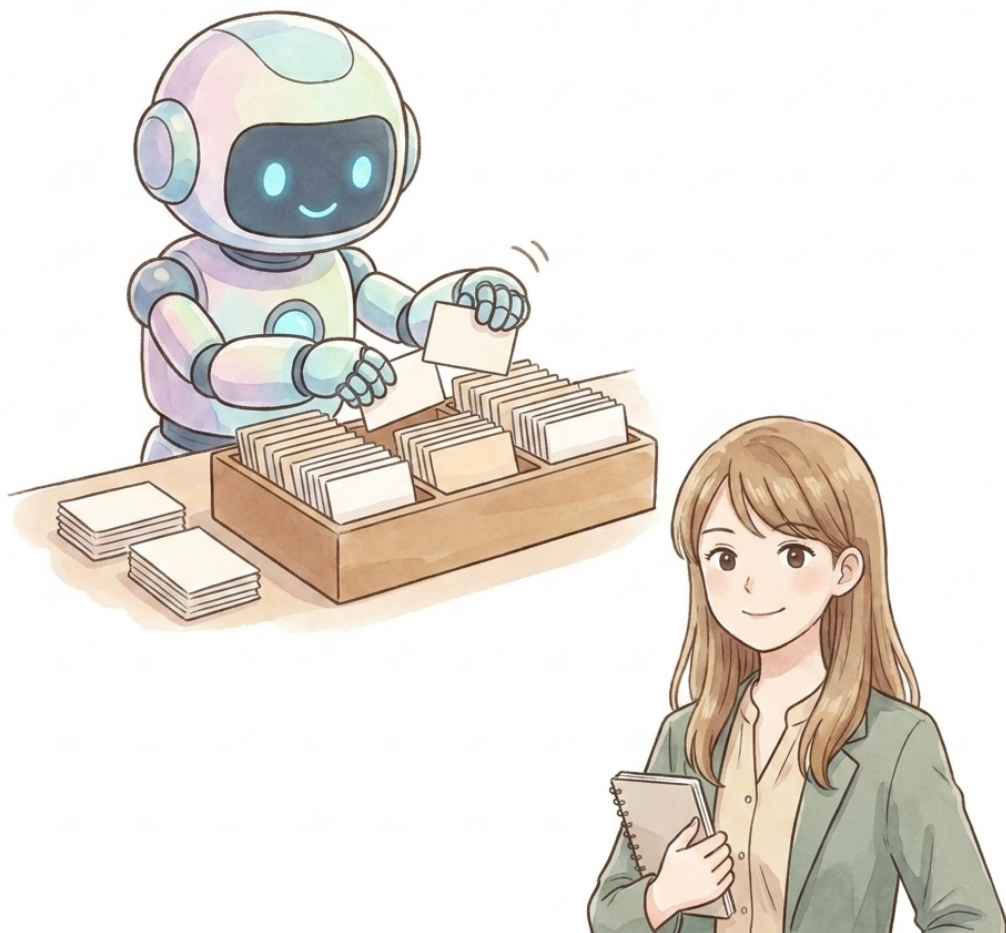
プロンプトを丁寧に書いても、素材を整えて渡しても、Claude が「これが悩み 3 つです、これが検索される言葉です」と気持ちよく出してくれることは、まだそんなに多くないです。

「AI に丸投げで判断してもらえる」ところまでは、書く前の素材整理のフェーズでは、まだ来ていないというのが、私の体感です。

ここでの考え方を、こう置いてください。

AI は判断者ではなく、整理係です。

AIは判断者ではなく整理係



「これ使える」を選ぶのは、結局、自分です

整理結果が手元に来て、自分の本に何を入れるかを決めるのは、編集長です。

Claude が「この観点は強そうです」と並べてくれた中から、「これは自分の本に入れる」「これは別の角度から書く」「これは触れない」を、1 つずつ仕分けます。

ここは、AI にお願ひできない部分が、まだまだあります。

自分の本の旗、面、軸、マス、文体、過去の本との並びを、いちばん理解しているのは、結局、自分です。

Claude はそれを完全には知らないのです、最後の「これ使える」の判定は、編集長としてやるしかありません。

整理されたリストを、上からざっと眺めていきます。

「ここの観点、自分の体験と重ねられるな」

「この事例は前作で似たことを書いたから、別の切り口で」

と、1 つずつ印を付けていく。

地道な作業ですが、ここを人任せにすると、本の輪郭がブレます。

整理されたものを見たとき、私は次の 3 つを重視して見えています。

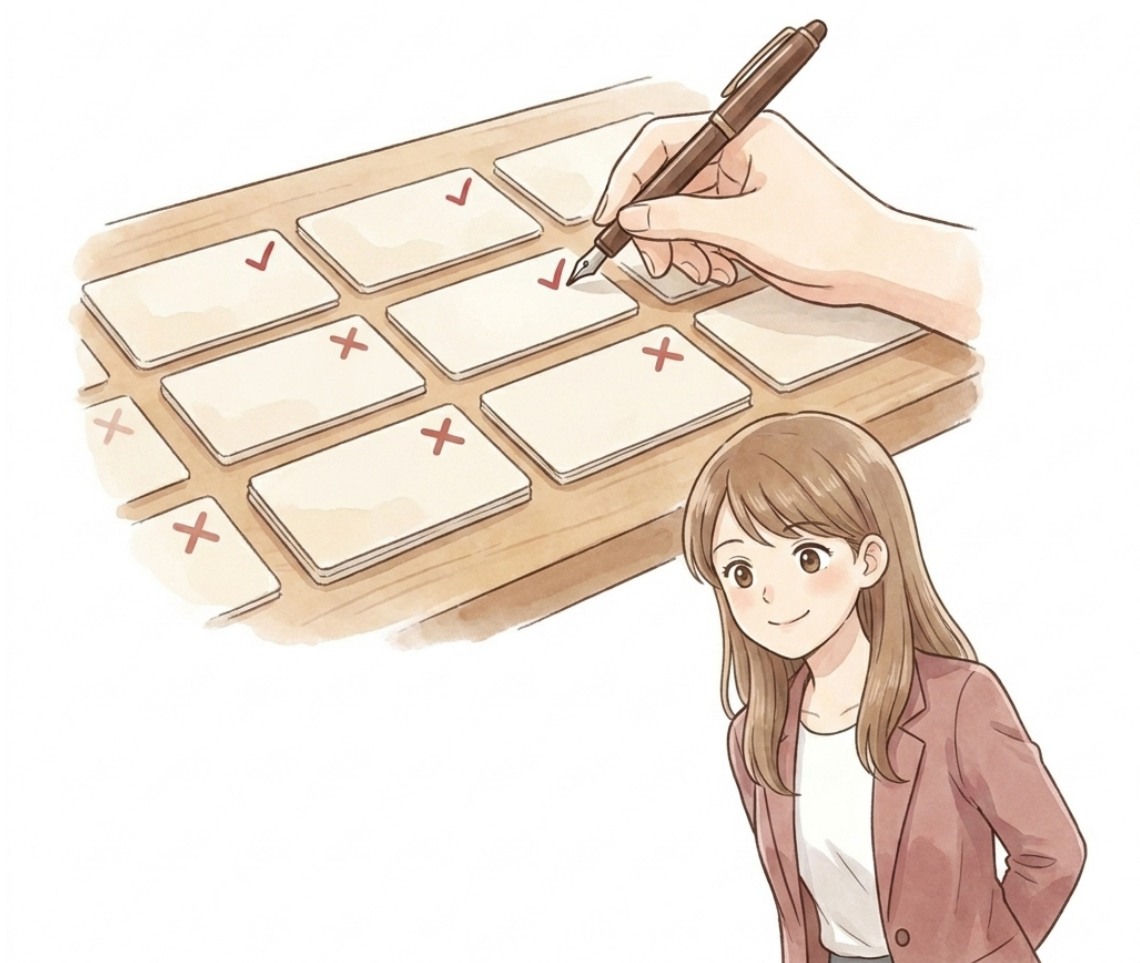
1 つ目、何度も出てくる悩み。

2 つ目、自分がまだ知らなかった用途。

3 つ目、自分の本で、自分の言葉で深掘りできそうな観点。

この 3 つに反応したものを、次の第 5 章「Value Proposition」に持っていきます。

これ使える、を選ぶのは自分



大画面モニターは、本気で投資する価値があります

集めた素材を整理する段階で、地味に効いてくるのが作業環境です。

特に、大画面モニターは本気で投資する価値があります。

素材整理では、複数の Claude 画面、ブラウザ、メモ、KDP ダッシュボードなどを並べて見る場面が増えます。

画面が狭いと、切り替えだけで集中力が削られます。

私は、常に 3〜4 つの部署のチャット画面を、モニターに立ち上げて並行作業しています。

マーケ部の Claude、調査部の Claude、執筆部の Claude、画像生成の AI。

それぞれの処理待ちの時間に、別の部署のブラウザを触って、進められる作業を並行して進めます。

たとえば、調査部の Claude が 100 本の素材を整理している間に、別画面で執筆部の Claude に note の原稿を頼みつつ、もう 1 画面で画像生成の AI に挿絵を作ってもらいながら、ついでに X のリプライを返したりしています。

これ、大画面モニターだと、ブラウザを横並びで全部立ち上げられるので、現実的に回ります。

ノート PC の画面 1 枚で全部やろうとすると、切り替えだけで集中力が切れます。

Kindle 出版の運営において、いちばん費用対効果の高い投資は、大画面モニターかもしれない、と密かに思っています。

AI だけに目が行きがちですが、ハードの環境を整えることも、地味に重要だと思っています。

集めた素材を整理する作業は、画面が広いほど早く終わります。

早く終われば、編集長としての「これ使える」を選ぶ時間に、頭を回せます。

さて。

集めて、整理して、選び終わったら、Input フェーズの大半は終わりです。

ただし、具体事例が集まるほど、扱い方には注意が必要です。

他人の言葉や構成を、そのまま自分の本に移してはいけません。

Input は、盗むための作業ではありません。

自分の言葉と考察を育てるための材料集めです。

次の 4-6 節では、このインプットの倫理を整理します。

4-6. インプットの倫理:パクリは絶対にしない

4-5 の章末で軽く触れた倫理を、ここで整理します。

長くは扱いません。

著作権法や複製権・翻案権の細かい話、KDP 規約の詳細は、前作の Brain『25 万円の印税が毎月口座に振り込まれる方法を全部書きました』の第 8 章にまとめてあります。

本編では、編集長として持つべき姿勢に絞ります。

集めた素材を、丸ごと本に持ち込まない

4-3 で集めた記事、4-4 で集めた YouTube 文字起こし、これらは「材料」であって「コピー素材」ではありません。

YouTube 文字起こしを段落丸ごとコピーして本文にする。

Kindle 本の文章を、出典や引用範囲も曖昧なまま、自分の本文のように挟む。

X の長文ポストを、表現はそのままに本に転載する。

Brain の構成や事例を、丸ごと自分の本に持ち込む。

どれも NG です。

「参考にする」と「移す」は、別物です。

参考にした上で、自分の頭で考え直して、自分の言葉で書き直す。

ここまでやって、初めて自分の本の素材になります。

もう 1 つ、AI 時代にやりがちなのが、「AI にリライトさせたから大丈夫」と考えることです。

YouTube の文字起こしや他人の文章を AI に入れて、「この内容を自分の文体でリライトしてください」と頼むやり方です。

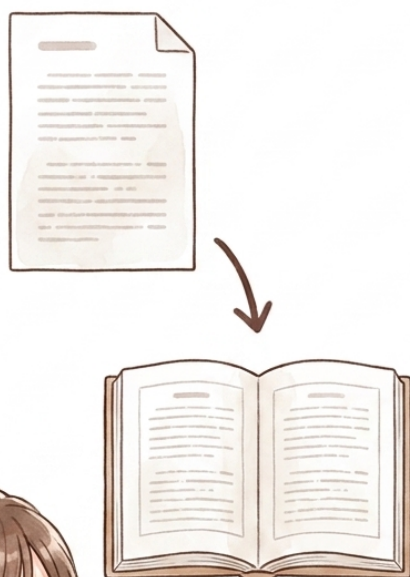
でも、元の構成、事例、主張の流れがそのまま残っているなら、言葉だけ変えても、自分の本にしたとは言えません。

AIで表現を変えることと、自分の頭で再構築することは、別物です。

ここを混同すると、無自覚にパクりに片足を突っ込みます。

AI時代の編集長として、自分でも気づきにくい落とし穴なので、頭に置いておいてください。

✕ 丸ごとコピー



○ 読んで、自分の言葉で書き直す



4つの理由で、パクリは選択肢にない

なぜ、パクリは編集長の選択肢にないのか。

理由は4つです。

1つ目、倫理的にダメ。

集めた素材の向こうには、書いた人、作った人がいます。

その人の労力と時間に、ただ乗りすることになります。

2つ目、法的なリスクがあります。

著作権法の複製権や翻案権に抵触する可能性があります。

細かい話は、前作の Brain の第8章を見てください。

3つ目、KDP のアカウントが止まるリスクがあります。

著作権侵害は、通報されると、Amazon は厳しく対応します。

1冊 BAN で済むか、アカウント全停止までいくかは、状況次第です。

4つ目、自分の本棚の信用を壊します。

これが、編集長としていちばん大きい理由です。



自分の本棚の信用を壊してしまうことについて

本棚は、長期的な信用の積み上げです。

1冊1冊が、読者からの信用を少しずつ貯めていく。

その積み上げが、月50万円というラインに、結果的につながっていきます。

ところが、1冊でも丸パクリの本が混ざると、本棚全体の信用が崩れます。

レビューで指摘されることがあります。

SNS で言及されることがあります。

検索で見つかることもあります。

隠せる時代ではありません。

一度信用を失うと、本棚そのものが崩壊します。

せっかく積み上げてきた他の本まで、影響を受けます。

だから編集長として、パクリは「やってはいけない」ではなく、最初から「選択肢にならない」です。

では、集めた素材はどう使えばいいのか

「NG ばかりだと、何もできなくなるのでは」と思った方もいるかもしれません。

そんなことはありません。

安全に使う順番があります。

私が意識しているのは、こういう順番です。

まず、集めた素材から要点だけを抜き出します。

長文をそのまま持つのではなく、観点や事例の核を短く書き残す。

次に、複数の素材に共通して出てくる悩みや用途を見ます。

1 つの素材に依存せず、複数の素材を横に並べた上で、共通する核を取り出す。

そのうえで、自分の体験、自分の読者マス、自分の言葉に引き寄せます。

集めた素材は、ここで自分のフィルターを通過します。

最後に、「自分はこの読者に、何を伝えたいのか」を考えてから本文に置きます。

素材の言葉ではなく、自分の言葉で書き直したものを、本に並べます。

この順番を通すと、他人の言葉を移すのではなく、自分の本として再構築する形になります。

必要以上に怖がる必要はありません。

ただし、素材をそのまま移すのではなく、自分の頭で消化し、自分の言葉に変換することが前提です。

この順番を守ることで、集めた素材を本づくりに活かしやすくなります。

個別の規約や法律は、前作の Brain にまとめてあります

著作権法、複製権、翻案権の細かい解説、KDP 規約の詳細、4 つの規制（景表法、薬機法、特商法、ステマ規制）の周辺の話は、前作の Brain『25 万円の印税が毎月口座に振り込まれる方法を全部書きました』の第 8 章で扱っています。

本編では、編集長として持つべき姿勢に絞りました。

詳細を確認したい方は、前作の Brain の該当章に戻ってください。

集めた素材は、自分の頭で消化してから本に置く。

集めた素材を、自分の言葉と考察に変換してから本に置く。

それが、編集長としての Input の着地点です。

ここまでで、第 4 章の本編は終わりです。

次の 4-7 では、上級者向けの補足として、YouTube 文字起こしを夜間に自動化する方法をお渡しします。

ここまでで標準ルートは渡し切ったので、上級者向けの実装に興味がない方は、4-7 は読み飛ばしていただいても大丈夫です。

× パクリはしない



インプットの倫理

4-7. 上級者向け:YouTube 文字起こしを、寝ている間に自動化する

ここまでの内容だけで、第 4 章の Input は十分に回せます。

Google 検索で記事を拾う。

Kindle Unlimited で関連本を流し読みする。

X や Grok で長文投稿を確認する。

YouTube は Tactiq で文字起こしを取る。

集めた素材を Claude に入れて、整理してもらう。

ここまでできていれば、十分です。

ここから先は、さらに大量の YouTube 動画を、寝ている間にまとめて文字起こししたい方向けの上級者向け補足です。

必要な方だけ、読んでください。

なお、この節で扱う方法は、公開されている動画の内容を、自分の学習・リサーチのために文字起こしして読むためのものです。

動画そのものの再配布、文字起こし全文の転載、他人の発言を自分の文章として使うことはしません。

また、YouTube や各ツールの利用規約、ライブラリの仕様は変わることがあります。

実行前には、ご自身の環境と最新の規約を確認してください。

不安がある方は、この節は使わず、Tactiq などでも 1 本ずつ確認する標準ルートで十分です。

私自身、1 本ずつ確認したい時は Tactiq を使います。

ここから先の自動化は、20 本、50 本単位で動画を集めたい時だけ使う、上級者向けの方法です。

Claude Code をご存知ない方へ

「Claude Code、聞いたことはあるけど触ったことはない」という方は、まずこちらの 2 冊を読んでいただくと、Claude Code の全体像と使い方が掴めます。

『ClaudeCode いちばんやさしい 教科書』

<https://www.amazon.co.jp/dp/B0GT4T4F2B>

『Claude Code でできること全部やってみた』

<https://www.amazon.co.jp/dp/B0GWRM54S5>

どちらも、Claude Code を初めて触る方が、最短で動かせる状態を作れる内容です。

4-7 の以下を読み進める前に、こちらに目を通していただくと、後半の話がスッと入ってきます。



Claude Code+yt-dlp+Whisper の一気通貫

1 点補足です。

PC 環境 (Mac/Windows、Python のバージョン、ライブラリの最新状況) によって、同じプロンプトを投げていても挙動が変わります。

これから貼り付けるプロンプト通りに動かないことが、現実には起こります。

そのときのいちばん早い対処は、Claude Code でエラーが出たら、その画面をスクリーンショットで撮って、Claude チャットに貼り付けて『このエラーの原因と直し方を教えてください』と聞くと、かなりの確率で解決の糸口が見つかります。

それでも解決しない場合は、無理に自動化にこだわらず、Tactiq の標準ルートに戻って大丈夫です。

これから渡すプロンプトは、「私が実際に Claude Code に投げた事例として、参考までに置いておきます」という位置付けで読んでください。

動かなかったら、スクショ + Claude チャットに質問で大体乗り切れます。

これから作るツールがやることは、こうです。

キーワード入力→YouTube 検索→低品質動画のフィルタリング→URL リスト生成→字幕 API で取得(取れない動画は Whisper でフォールバック)→全動画の文字起こしを 1 つの md ファイルに連結。

私が実際に Claude Code に依頼して構築した実物の手順を、そのまま渡します。

Claude Code に投げるプロンプトは、2 段階です。

プロンプト①、URL 自動収集と文字起こしの構築

最初に、Claude Code に以下をそのままコピーします。

以下の YouTube 動画の文字起こしツールを作してほしい。

Python で。

基本仕様

- 実行コマンド: `python3 transcribe.py --search "30代 独学英語" --max 50 --transcribe`
- yt-dlp の ytsearch 機能で検索結果を取得し、フィルタリング後 `urls.txt` に書き出す
- 検索キーワード・取得件数は引数で渡せるように(他テーマでも使い回したい)

- 既存の urls.txt は上書き(--append オプションで追記モードも選べると良い)
- 検索結果は日本語の動画のみを対象とする

低品質動画を除外するフィルタ条件

検索結果から以下を除外:

- 動画長: 60 秒以下(ショート)と、3 時間以上(ライブ配信アーカイブの可能性)を除外
- 再生数: 1000 回未満を除外
- タイトルに以下のキーワードが含まれるものを除外: 「shorts」「ショート」「広告」「PR」「sponsored」、過度な絵文字や全角記号の連続
- チャンネル重複: 同一チャンネルからは最大 3 本まで
- アップロード日: 6 ヶ月以内のみ

並び順

最終的な urls.txt の出力順は、アップロード日の新しい順にソート。

取得時の追加メタデータ

URL だけでなく、以下を urls.txt の各行のコメントとして付与(人間が目視で精査しやすくするため):

`https://www.youtube.com/watch?v=XXXXX # 【タイトル】チャンネル名 / 再生数: 12,345 / 投稿: 2025-12-01 / 長さ: 12:34`

取得結果のサマリ表示

実行後、標準出力に以下を表示:

- 検索ヒット総数
- フィルタ通過数(フィルタごとの除外件数も内訳表示)

- 最終的に urls.txt に書き出した数
- 取得した動画の投稿日範囲(最古～最新)

6ヶ月以内のフィルタを通った動画が --max に達しない場合は、達した本数のみで urls.txt を出力。

50本に満たない旨は標準出力に明示。

文字起こし機能

- urls.txt に記載された URL リストを読み込み、各動画を文字起こし
- 字幕 API を優先(速い・無料)。

字幕がない動画だけ Whisper でフォールバック

- 全動画を1つの .md ファイルに連結(ザーツと読める形式)
- 各動画の前にメタデータブロックを付与:

=== タイトル:○○ URL:https://... チャンネル:○○ ===

検索→収集→文字起こしを一気通貫で走らせるオプション:

```
python3 transcribe.py --search "○○" --max 50 --transcribe
```

このオプションが付いたら、URL 収集後にそのまま文字起こしまで実行する。

「○○」のところに、自分のテーマのキーワードを入れます。

「ClaudeCowork」「ビジネス英語 独学」「会社員 副業」など、4-3 の調査プロンプトで指定したテーマと、同じ言葉を使います。

「日本語の動画のみを対象とする」を必ず入れることが、ポイントです。

これがないと、AI 関連のキーワードでは特に、英語圏の動画が大量に混ざります(4-3・4-4 でお伝えした、海外記事と同じ落とし穴です)。

プロンプト②、レート制限対策

プロンプト①が完了したら、続けてこれを投げます。

既存の `transcribe.py` の `process_one` 関数の直後に `time.sleep(0.5)` を組み込んでほしい。

50 本処理時の YouTube 字幕 API レート制限対策。

これだけです。

50 本以上を一気に処理する場合、字幕 API のレート制限に引っかからないよう、安全装置を入れておきます。

実行手順と、運用 Tips

ツールが完成したら、以下の手順で動かします。

プロジェクトフォルダに移動

```
cd youtube-transcribe
```

仮想環境を有効化

```
source .venv/bin/activate
```

Step 1: URL リスト生成(目視チェック用)

```
python3 transcribe.py --search "〇〇" --max 50
```

Step 2: urls.txt の中身を目視チェック(不要な動画の行を削除)

Step 3: 文字起こし実行

```
python3 transcribe.py
```

一気通貫で走らせる場合:

```
python3 transcribe.py --search "〇〇" --max 50 --transcribe
```

出力先: output/transcripts.md

運用上の注意点を、まとめておきます。

ClaudeCowork と同時に動かすと、CPU を奪い合います。

Whisper フォールバックが走ると CPU 負荷が高くなるので、重い処理が走る間は、Cowork を止めるのが安全です。

処理時間の目安は、字幕 API で大半が取れば 5～10 分。

Whisper フォールバックが多いと、30 分～1 時間以上かかることもあります。

だから、4-4 でお伝えした夜間運用(寝る前に走らせて、朝に文字起こしが揃っている)が、現実的な選択肢になります。

進捗確認は、`tail -50 output/run.log` で最後の 50 行を確認できます。

Python バージョン警告が出ても、処理が最後まで進む場合は、そのまま使えることがあります。

ただし、途中で止まる場合は、Python のバージョンやライブラリの対応状況を確認してください。

不安な方は、エラー内容を Claude に貼って原因を確認するのが安全です。

Mac の sudo パスワード入力時は、入力しても画面には何も表示されません。

何も表示されなくても、パスワードをそのまま打ち込んで Enter を押せば大丈夫です。

Claude Code に依頼すると、環境によっては以下のような構成でツールが作られます。

- yt-dlp: YouTube 検索・メタデータ取得。必要な場合のみ、各規約に配慮した範囲で音声取得
- youtube-transcript-api: 字幕 API での文字起こし(優先)
- faster-whisper(base モデル): 字幕がない動画の音声文字起こし(フォールバック)
- venv: Python 仮想環境(依存関係を分離)
- HTTP 403 対策として player_client フォールバックも組み込み済み

保管場所は、自分が分かりやすいフォルダで構いません。

私の場合は、テーマ別の作業フォルダの下に置いています。

キーワードを変えれば、他テーマでも使い回せる再利用資産になります。

ツールの動作環境や挙動は、ライブラリのアップデートや仕様変更で変わる可能性があります。

実行時にエラーが出たら、エラー文を Claude Code や Claude チャットに貼り付けて、「このエラーの原因と、安全な直し方を教えてください」と聞くのが早いです。

分からないままコマンドを実行し続けるより、原因を確認しながら進めてください。

第 4 章のまとめ

ここまでが、第 4 章「Input: 高速インプット」です。

書く前のインプットを、6つの動線で集める。

動線0は自分の本棚と読者の反応。

動線1から5は、Google 検索／Kindle Unlimited／X と Grok／YouTube 文字起こし／Brain や有料教材。

マスから逆算する調査プロンプトで、集める素材のズレを防ぐ。

Tactiq と手動で文字起こしを取り、AI を整理係として使い、人間の判断で「これ使える」を選び、倫理を守って自分の言葉に変換する。

これが、編集長としての Input フェーズの全体像です。

集めて、整理して、自分の言葉で再構築する素材が、ここまでで手元にできました。

次の第5章「Value Proposition: 自分の出せる価値」では、ここまでで集めた素材から、自分にしか書けない切り口を取り出す作業に入ります。

AI に聞けば誰でも同じ答えが出る時代に、自分の体験と考察を真ん中に置く設計を、実録会話ベースで開いていきます。

では、第5章に進みましょう。

6つの動線で集める

自分の言葉に変える

