

スライド生成キット V2 取扱説明書

adl-deck-kit-v2 の仕組み、使い方、手元での改良手順

2026年8月

ARTHUR  LITTLE

CONTENTS

1	このキットについて	3
2	仕組みと特徴	7
3	このキットで作れるもの	12
4	他のスライド生成手法との比較	16
5	キットの使い方	20
6	キットの改良と追加	24
7	注意点	28

1 このキットについて

テンプレ準拠のpptxを、依頼から点検まで一度で作り切るための道具一式です

スライド生成 キット v2 の位置づけ	解く問題	- テンプレの体裁を守りながら、納品できる密度の本文を作るには、これまで人の手が必要だった - 体裁を機械に任せると中身が薄くなり、中身を人が作るとテンプレが崩れる - 自動生成された資料は「テンプレっばい」止まりで、ロゴ位置・配色・余白のどれかが必ずずれていた - 似せて描く方式は、テンプレの資産を一度も通らないため、ずれても検出できない - 版面は26種類あるのに、実際には3〜5種類ばかりが繰り返され、資料が単調になっていた - 内容と版面を結びつける基準がなく、選択が書き手の勘に委ねられていた
	提供するもの	- テンプレ原本から71ページを抜いて作った台紙と、そこからスライドを生やす生成エンジン - ロゴ・配色・書体・余白・ページ番号は版面から継承されるため、書き手が触らない - 実際に納品されたスライドから起こした7つの本文の型と、合計40種類の版面 1ページ400〜900字、図表2〜5個という密度の基準を型として持たせている - 生成物を内容・レイアウト・抜け漏れの3観点で点検する仕組み - 実際に描画した画像から占有率を測るため、薄いページが数値で分かる - Claude Code に読ませる規範と、型・版面・planキーの手引き、密度の見本
	前提	- Python 3.12 系と、python-pptx・PyYAML・Pillow の3つのライブラリだけで動く - 実描画による点検だけは PowerPoint など外部のアプリを使う - 出力は通常の pptx で、PowerPoint と Keynote のどちらでも開いて手で直せる - 生成後に手で直すことを前提にしており、画像やHTMLでは出力しない - フォルダの中だけで完結しており、外部のサービスに接続しない

従来のやり方は、テンプレートへの準拠と中身の作り込みのどちらかを諦めていました

	人が手で作る場合	従来の自動生成	埋めるべき差
1 テンプレート準拠	原本を複製して使うため崩れないが、コピー元を間違えると気づけない	見た目を似せて図形を並べるため、ロゴ位置・余白・配色が少しずつずれる	a 準拠の判定が人の目に依存する b ずれても生成時には検出されない
2 本文の密度	1ページに分析を詰め込める。ただし1ページあたり数十分かかる	箇条書き4行程度で終わり、図版のないページが並ぶ	a 密度の基準が言語化されていない b 薄いページを検出する仕組みがない
3 版面の選択	内容に応じて版面を選び分けられる	同じ版面ばかりが繰り返され、26種類あるうち3〜5種類しか使われない	a 内容と版面を結びつける表がない
4 生成後の編集	そのまま編集できる	画像やHTMLで出るため、受け取った側が直せない	a 提出後の修正が作り手に戻ってくる

▶ このキットは、準拠を台紙の継承で担保し、密度と版面選択を型として明文化しています

フォルダは、台紙・エンジン・コマンド・見本と手引きの4系統で自己完結しています

	場所	中身	役割
台紙	assets/ADL_Base.pptx	テンプレ原本から71ページのスライドだけを抜いた台紙（0.59MB・26レイアウト）	すべてのスライドの生成元
	assets/ADL_Slide_User_Guide_ORIGINAL.pptx	テンプレ原本そのもの（48.87MB・71ページ）	台紙の作り直しと仕様の参照用に同梱
生成エンジン	adl_deck/builder.py, pages.py	台紙を開き、レイアウトからスライドを生やす層	書体とテーマ色の適用、表紙・目次・章扉
	adl_deck/pro.py, parts.py, chrome.py	納品品質の型・細かい部品・ヘッダーと脚注	本文の作り込みを担う中心部
	adl_deck/planner.py, validate.py	plan の解釈と、生成物の静的点検	型の割り当てと座標・書体・配色の検査
コマンド	bin/build.py	plan を読んで pptx を作り、静的点検まで流す	生成のたびに使う
	bin/audit.py	実際に描画して占有率とはみ出しを実測する	生成のたびに使う
	bin/storyline.py	構成の一覧を出す	承認をもらう前に使う
	bin/render.py, package.py	PDF・PNG への描画と、配布用一式の作成	必要なときだけ使う
見本と手引き	examples/	納品密度の見本、軽い型の見本、型カタログ	書き方を真似る出発点
	docs/	型・版面・plan キー・ブランド規範の手引き	迷ったときの参照先
	CLAUDE.md, SKILL.md	Claude Code に読ませる規範と、Skill としての入口	キットの挙動を決める本体

*1 コードは合計4,294行。外部サービスへの通信は行わない
 出所：キット実測（ファイルサイズ・行数）

2 仕組みと特徴

テンプレが崩れないのは、見た目を真似ず原本の台紙から生やしているためです

台紙の作り方

- テンプレ原本から71ページのスライドだけを抜き、資産はすべて残す

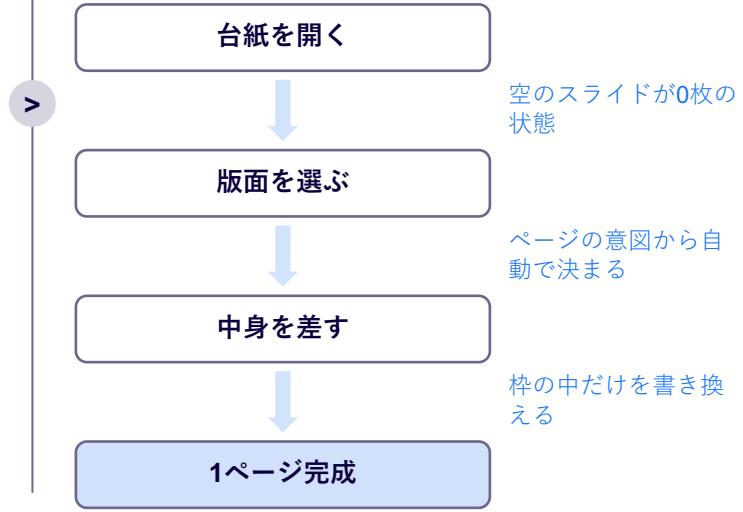
台紙に残っているもの

残した資産	内容
スライドマスター	版面全体の書式定義
レイアウト	表紙から付録まで26種類
テーマ	配色12色と書体の定義
ロゴ	位置とサイズを含む
ページ番号	差し込みの設定ごと
背景	ブリッジグラデーション

スライドの生やし方

- 図形を並べて似せるのではなく、26の版面から複製して中身だけを差し込む

1ページが作られる順序



得られる結果

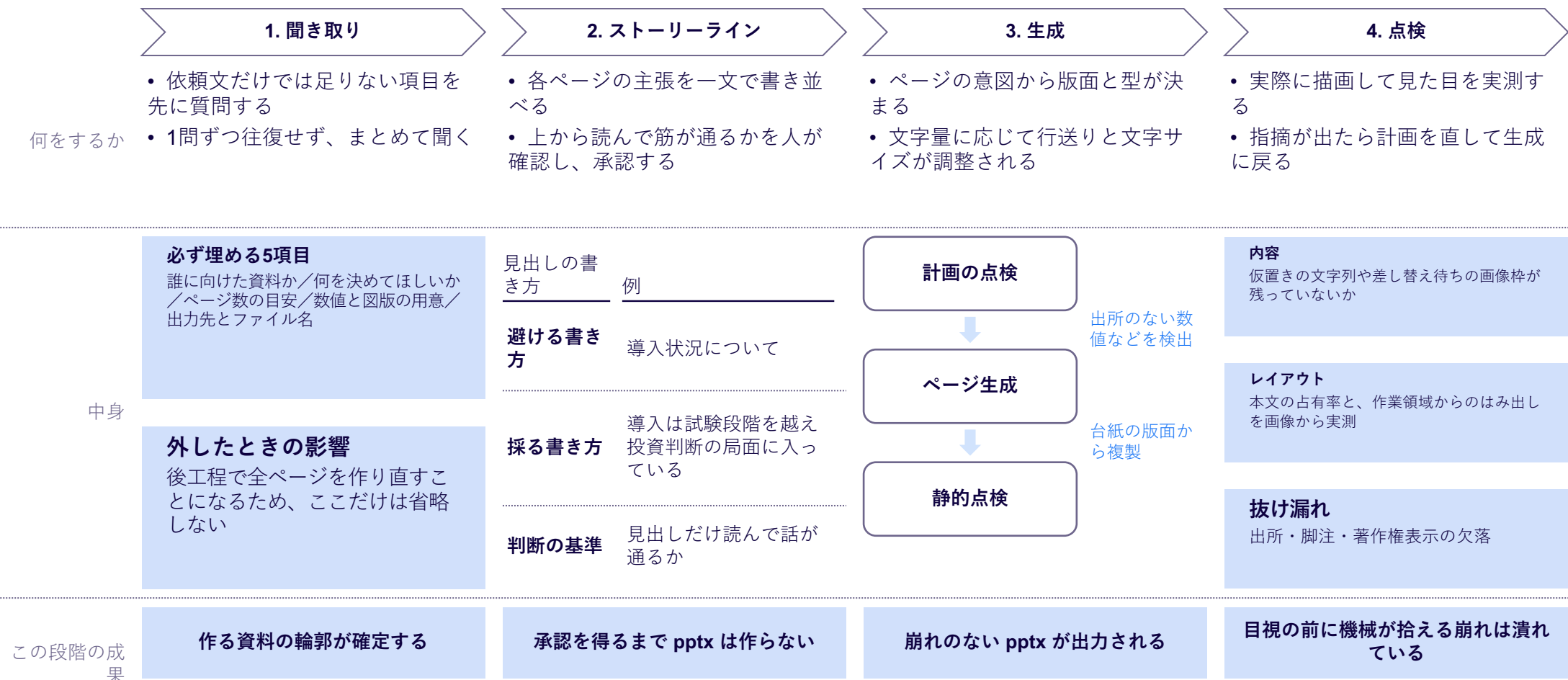
- ロゴ・配色・書体・余白・ページ番号が、書き手の操作なしで正しくなる

自動で正しくなる項目

項目	扱い
ロゴ	版面から継承
配色	テーマ12色に限定
欧文書体	Arial に固定
和文書体	日本語資料のみ併記
作業領域	横2.53～31.33cm 縦4.45～17.36cm
ページ番号	自動採番

▶ 前バージョンは図形を描いて変換していたため、テンプレの資産を一切通っていませんでした

処理は聞き取りから点検まで4つの段階を順に通る、生成の前に人の承認を1回挟みます



本文の密度は、実際に納品されたスライドの実測値を基準に決めています

	テンプレートどおりに作った場合	実際の納品物	キットの既定値
1 1ページの文字数	100～200字。1ページ1メッセージの発表資料になる	400～900字。1ページに分析を詰め込むコンサル様式	a 400～900字を目安とする b 下回るページは点検で検出する
2 1ページの図表	0～1個	2～5個。表・格子・マトリクス・フローを組み合わせる	a 本文ページは原則として図版を持つ b 箇条書きだけのページを並べない
3 本文の文字サイズ	16pt / 14pt	11pt / 10pt	a 実務型では11pt・10ptを既定とする
4 ヘッダーとフッター	ロゴとページ番号のみ	章バッジ・章名・論点ナビゲーション、出所・脚注・著作権表示	a 実務型では自動で付与する b 省略すると点検で指摘が出る

▶ 守るのはロゴ・配色・書体・ページ番号・表紙まわりで、本文の中身は伝えたい構造に合わせて組みます

生成物は、内容・レイアウト・抜け漏れの3観点で毎回自動点検されます

観点1 内容

- 仮置きのまま提出してしまう事故を、文字列の検出で防ぐ

検出する対象

検出するもの 扱い

仮置きの記号 要修正

差し替え待ちの画像枠 要修正

出所のない数値 計画段階で警告

空のページ 要修正

観点2 レイアウト

- 実際に描画した画像から占有率を測り、崩れを見つける

実測の手順

pptx を PDF にする

PowerPoint を優先

PDF を画像にする

ページごとに

占有率を測る

格子に割って判定

薄いページを列挙

観点3 抜け漏れ

- 体裁として必ず要る要素が落ちていないかを見る

必須の要素

ヘッダー

章バッジと章名。論点ナビゲーションは必要なページのみ

左下

出所の表記と脚注。9ptで揃える

右下

著作権表示

版面の偏り

同じ型が4枚続くと警告

▶ 機械が拾えるのは崩れまでで、読みやすさは書き出した画像を目で見て判断します

3 このキットで作れるもの

本文は、実際の納品物から起こした7つの型のどれかで組むのが基本です

	型の名前	どんな構造のときに使うか	中に置けるもの
要約と全体像	exec_summary	全体をひとつのくりにまとめ、その下に分類ごとの観察を並べる	階層ラベル2段と箇条書き
	framework	工程を列、観点を行にして格子のマスをすべて埋める	各マスに表・カード・フロー・グラフ
構造を見せる	logic_flow	論理が左から右へ流れる。列の間に接続の記号が入る	列ごとに見出し・結論・図版
	gap_analysis	二者を観点ごとに対比し、右端に差分を書く	番号バッジと差分のバッジ
	flow_vertical	主体や工程が上から下へ流れる	角丸の節と、矢印脇の注記
事実を並べる	pro_table	一覧を縦野のない表で見せる。行が少なくても間延びしない	2段のグループラベルと右端の注釈
	positioning	2軸で切って、該当するマスだけを塗る	軸のラベルと目盛

▶ 箇条書きだけの型は補助であり、本文の主力にはしません

型は、見た目の好みではなく、伝えたいことの構造から選びます

	要約したい	工程で見せたい	論理で運びたい	事実を並べたい
どんなとき	- 全体像を先に示し、その下に根拠を並べたい - 冒頭や章の入口に置く	- 時間や段階に沿って進む話を扱いたい - 各段階で何をするかを揃えて書く	- 前の列の結論が次の列の前提になる話を扱いたい - 列ごとに図版を持たせる	- 対比や一覧そのものが情報になる話を扱いたい - 解釈は下端の帯に置く
選ぶ型と分量	使う型 exec_summary (左に大分類、その右に中分類、さらに右に観察)	使う型 framework (列=工程、行=観点の格子) / flow_vertical (縦の流れ)	使う型 logic_flow (見出し・結論・図版を列ごとに積む)	使う型 gap_analysis (二者対比) / pro_table (一覧) / positioning (2軸)
	分量の目安 分類3~4本、各分類に箇条書き3~6行	分量の目安 列3~4本、行3本。最終行を結論の帯にする	分量の目安 列3本。各列に結論一文と図版1個	分量の目安 対比は観点4本、一覧は8行以上
読み手に起こること	読み手が全体を把握できる	段階ごとの過不足が見える	主張の導かれ方が追える	判断の材料がそろう

10ページでも140ページ超でも書き方は同じで、章を増やすだけで目次と章扉が付きます

小さい資料

- 章を1～2本にして、本文だけを並べる

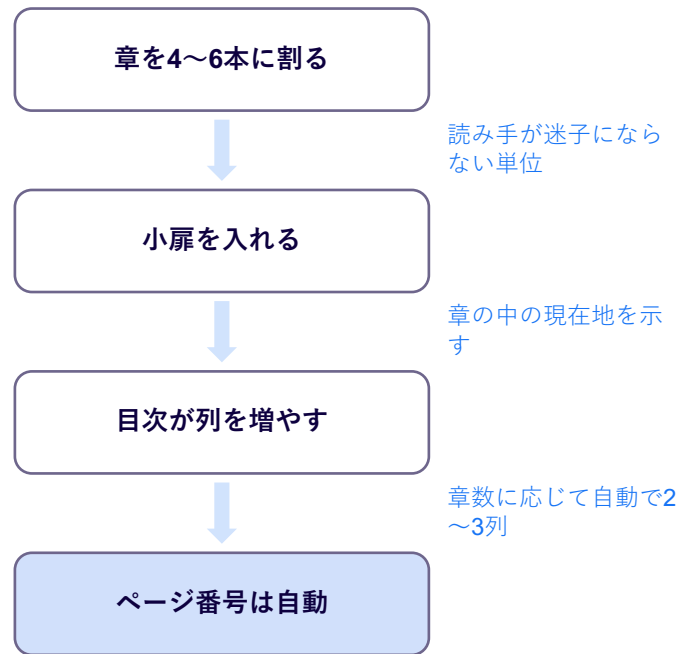
10ページ前後の構成

要素	扱い
表紙	自動
目次	省略できる
章扉	章ごとに自動
本文	書いた分だけ
裏表紙	自動

大きい資料

- 章を4～6本に割り、章の頭に小扉を入れて現在地を示す

章が増えたときの扱い



実測

- 143ページの生成に2.3秒。ページ数が増えても手順は変わらない

規模ごとの生成結果

総ページ	内訳	生成時間	要修正
10	本文5・章1本	1秒未満	0件
17	本文11・章3本	1秒未満	0件
29	本文22・章3本	1秒未満	0件
143	本文100・章40本	2.3秒	0件

4 他のスライド生成手法との比較

スライド生成には5つの方式があり、それぞれ得意な場面が違います

	方式	仕組み	テンプレ準拠	生成後の編集	向く用途
単独の方式	HTMLスライド	ブラウザで表示するスライドをコードで書く	別実装になる	HTMLを書ける人だけ	研修・登壇
	画像生成	1ページを画像として生成する	文字組みが崩れる	できない	表紙・イメージ図
	テンプレ流し込み	既存pptxの決まった枠に値を差し込む	崩れない	できる	定型の帳票
	AI丸投げ	生成サービスに任せ、独自形式で出力する	合わせられない	サービス上でのみ	社内のたたき台
組み合わせ	このキット	台紙の継承・AIによる構成と文章・Pythonによる図解を、役割で分けて組み合わせる	崩れない	pptxとして可能	顧客提出の調査資料



単独の方式はどれかを必ず諦めており、両立させるには役割を分けるほかありません

テンプレ準拠と生成後の編集の2軸で見ると、両立できている方式は限られます

2軸で見る

- 縦に生成後の編集可否、横にテンプレ準拠を取って方式を配置する

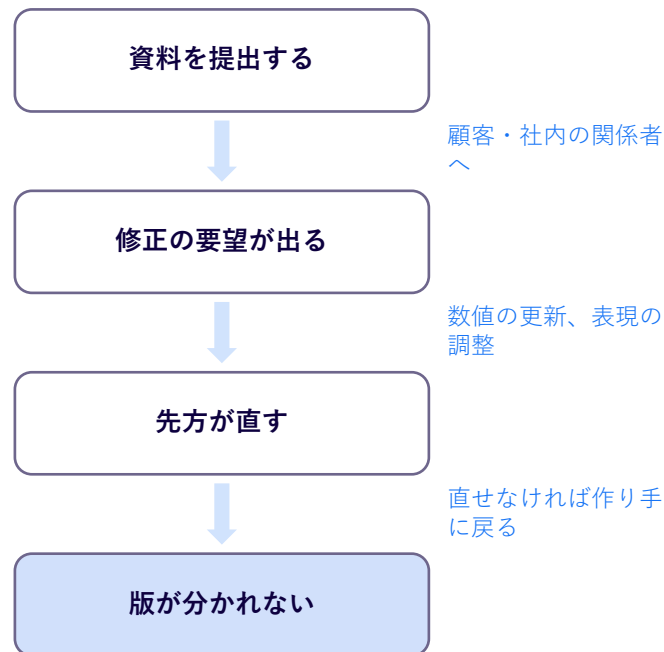
方式の位置づけ

生成後の編集	pptxで直せる	AI丸投げ	テンプレ流し込み このキット
	直せず	画像生成 HTMLスライド	該当なし
		合わない	崩れない
		テンプレ準拠	

両立が要る理由

- 提出した資料は先方の手で直されるため、pptxとして開けることが条件になる

提出後に起きること



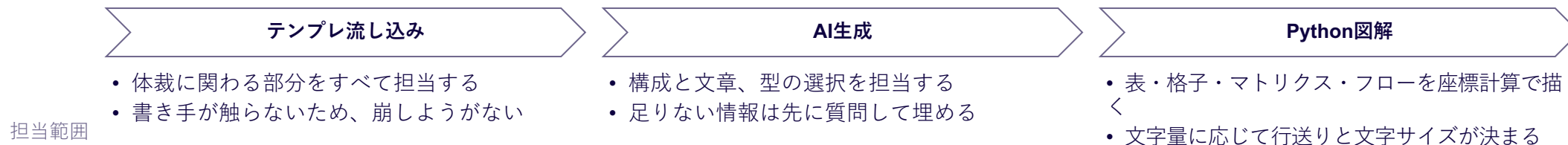
このキットの位置

- 台紙の継承で準拠を担保し、出力が通常のpptxなので編集も残る

満たしている条件

条件	このキットでの扱い
テンプレ準拠	台紙の版面を継承
図解の作り込み	Pythonで描画
構成と文章	AIが担当
生成後の編集	通常のpptx
体裁の検査	生成のたびに自動

テンプレ流し込み・AI生成・Python図解を、役割で分けて組み合わせています



具体

担当する要素	出どころ
ロゴ	版面
配色	テーマ12色
書体	テーマ定義
余白	版面
ページ番号	台紙
表紙・目次・章扉	版面

担当する範囲

全体の構成、各ページの主張、本文の文章、ページごとの型の選択

描けるもの

縦罫のない表、行×列の格子、2軸マトリクス、縦横のフロー、棒・折れ線グラフ、番号バッジや矢印などの部品

担当しない範囲

座標の計算と体裁の適用。ここは機械が持つため、AIの出力ぶれが体裁に伝播しない

効く場面

与えられた高さを使い切るため、行数が少なくても間延びしない

得られるもの

体裁が自動で正しくなる

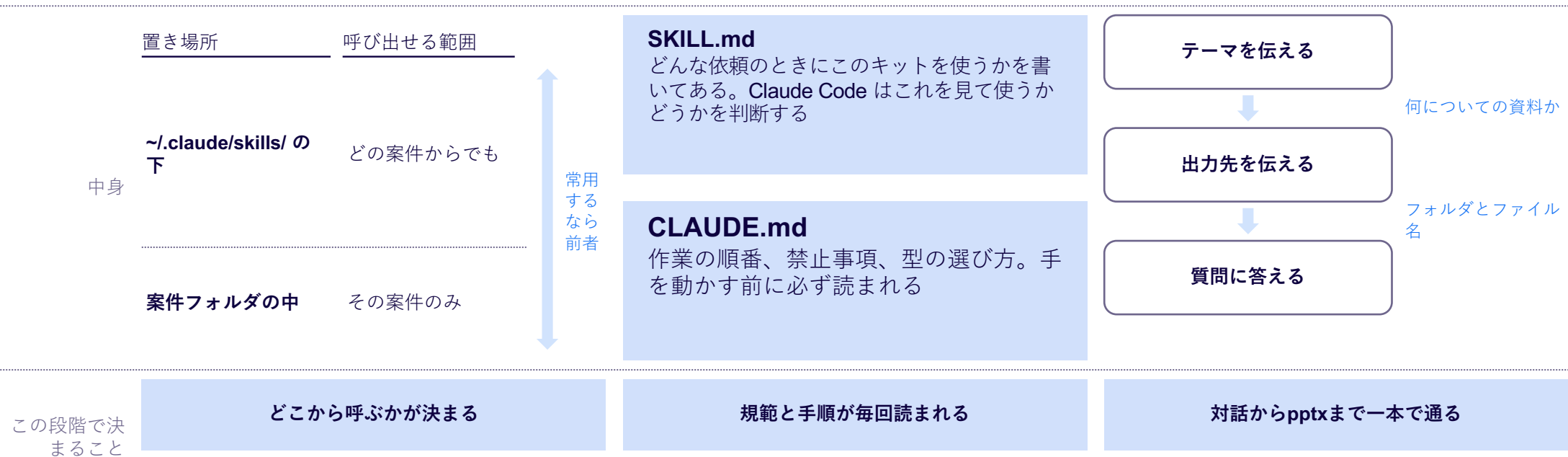
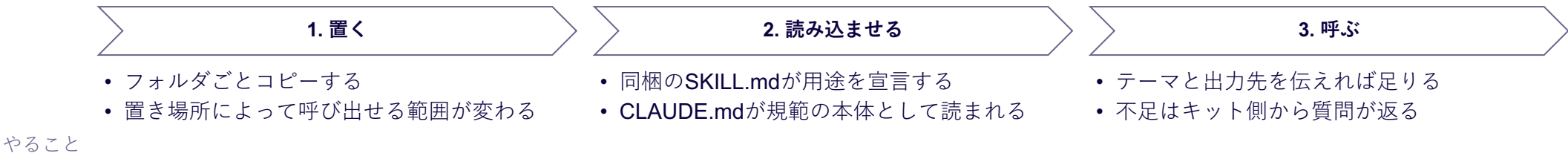
中身が案件に合わせて作られる

図解の品質が生成ごとにぶれない

▶ どれか1つを外すと、体裁・中身・図解のいずれかが必ず落ちます

5 キットの使い方

使うときは、キットをスキルとして置き、テーマと出力先を伝えるだけです



*1 Python 3.12 系と、python-pptx・PyYAML・Pillowが入っていること。requirements.txtに記載
 出所：キット仕様（SKILL.md / CLAUDE.md）

依頼から納品まで、人が判断するのは承認と目視の2回だけです

依頼と質問

- テーマと出力先を渡すと、足りない項目がまとめて質問される

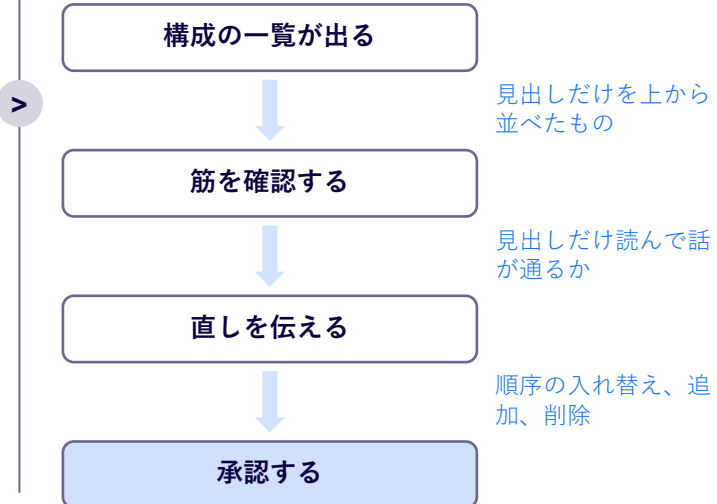
聞かれる項目

項目	答えないと起きること
対象読者	語り口がぶれる
決めてほしいこと	結論が定まらない
ページ数の目安	分量が合わない
数値の用意	出所なしの数値が入る
出力先	保存場所が決まらない

構成の承認

- 各ページの主張を一文で並べた一覧が出るので、そこで筋を確認する

承認までの流れ



生成と点検

- 承認後は自動で進み、最後に書き出した画像を目で見て終える

承認後の処理

処理	人の関与
計画の点検	なし
pptx の生成	なし
静的な点検	なし
実描画と実測	なし
画像の目視	ここだけ人が見る

▶ 承認前にpptxは作られません。作り直しの手戻りを構成の段階で止めるためです

コマンドは5本ありますが、普段使うのは生成と点検の2本だけです

	コマンド	何をするか	いつ使うか
普段使う	python3 bin/build.py plan.yaml	plan を読んで pptx を作り、静的点検まで一度に流す	生成のたび
	python3 bin/audit.py 出力.pptx	実際に描画して占有率・はみ出し・抜け漏れを測る	生成のたび
必要なときだけ	python3 bin/storyline.py plan.yaml	各ページの主張を一覧にして出す	承認をもらう前
	python3 bin/render.py 出力.pptx --sheet	PDF と PNG を書き出し、一覧シートも作る	ページ数が多く全体を見たいとき
	python3 bin/package.py 出力先 --zip	中間生成物を落として配布用の一式を作る	キットを人に渡すとき

*1 build.py に --lint-only を付けると、生成せず計画の点検だけを行う
出所：キット仕様（bin/ 配下）

6 キットの改良と追加

改良は、内容・型・部品の3階層に分かれており、触る場所で影響範囲が決まります

	第1層 内容を変える	第2層 型を足す	第3層 部品を足す
何をするか	- plan.yaml だけを直す - コードには触らない	- 新しいページの型を追加する - 既存の型の見た目を変えるのもここ	- 型の中で使う細かい図形を追加する - バッジ、矢印、囲みなど
中身	できること 文章の書き換え、ページの追加と削除、型の変更、章構成の組み替え	できること 納品物で見かけた新しい図解の型を、以後どの資料からも呼べるようにする	できること 既存の型に無い表現を作る。複数の型から共通で呼べる
	影響範囲 その資料だけ。ほかの案件には波及しない	影響範囲 キット全体。既存の資料を作り直すと見た目が変わる場合がある	影響範囲 その部品を使う型すべて
	必要な知識 YAML の書き方のみ。docs/PLAN_REFERENCE.md に全キーがある	必要な知識 Python と、cm 単位の座標計算	必要な知識 python-pptx の図形操作。既存の部品を写して直すのが早い
位置づけ	案件ごとの調整はここで足りる	案件をまたいで再利用できる	表現の幅が広がる

▶ 迷ったら第1層から試します。多くの要望は plan.yaml の書き換えで済みます

新しい型を足すときは、4つのファイルに順に手を入れます

描く関数を書く

- 与えられた枠の中に図を描く関数を1本追加する

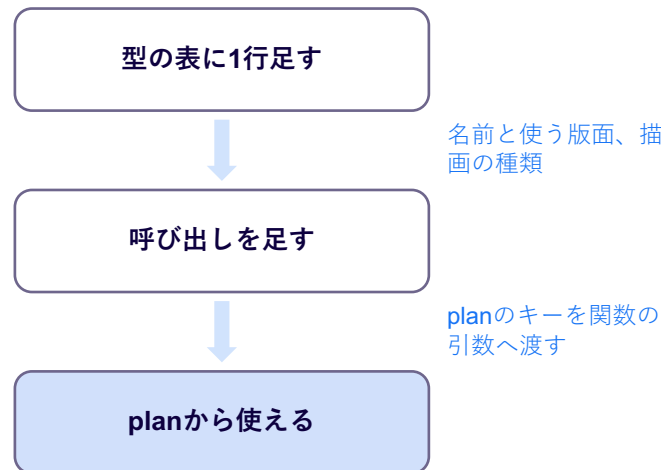
関数の作り方

決めること	書き方
置き場所	adl_deck/pro.py
受け取るもの	ページとデータと枠
枠の使い方	高さを使い切る
色の指定	テーマ色から選ぶ
文字サイズ	既定の定数を使う
参考にする関数	近い形の既存の型

型として登録する

- 名前と描画関数を結びつけ、planから呼べる状態にする

手を入れる2か所



手引きに残す

- 書き方を残さない型は、次に使われずに消える

残す内容

型の手引き

どんな構造のときに使う型か、planでの書き方の例を追記する

見本

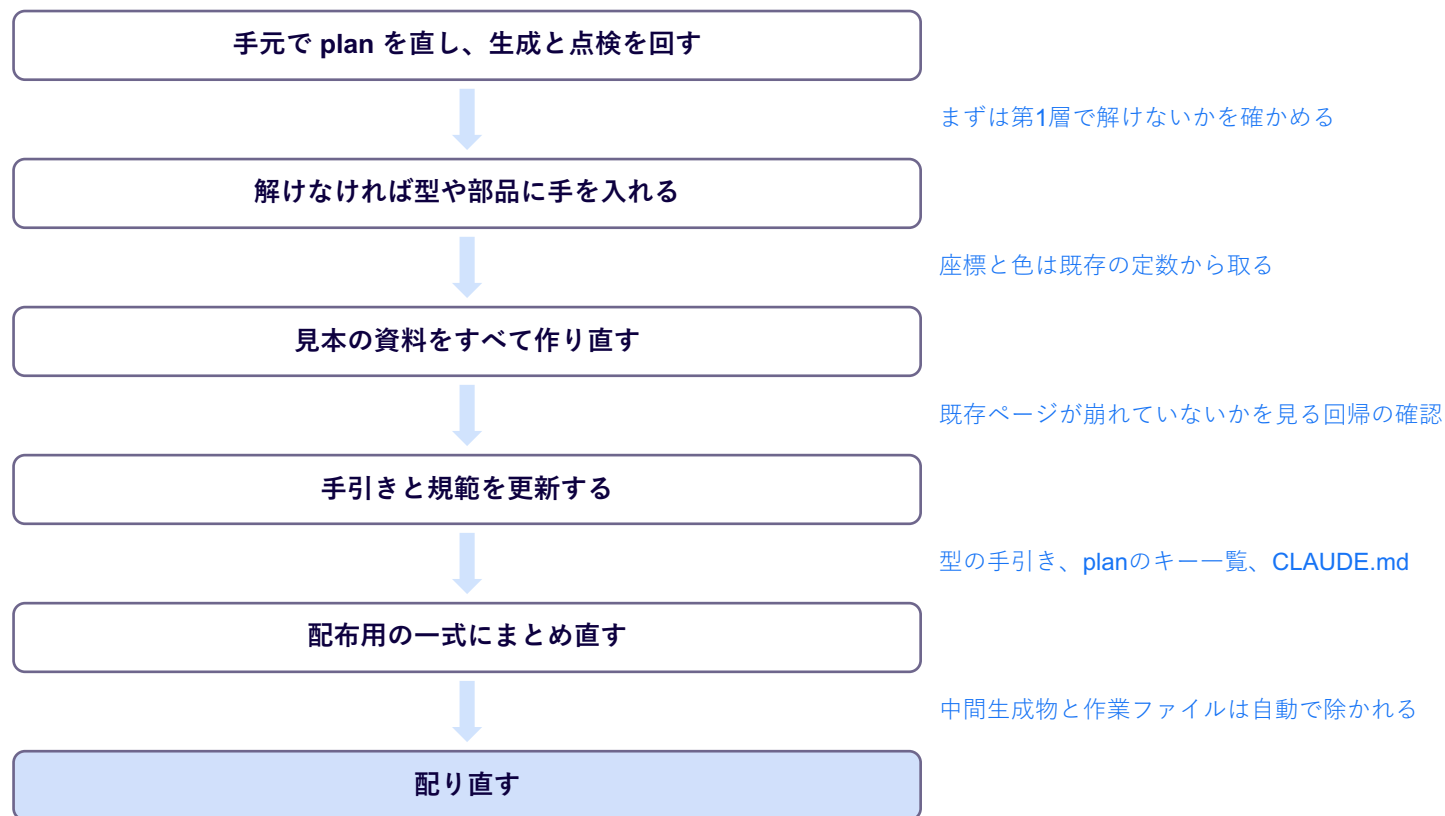
見本の資料に1ページ足して、実際に生成できることを示す

回帰の確認

見本をすべて作り直し、既存のページが崩れていないことを確認する

*1 描画関数・呼び出し・型の表・手引きの4か所。どれかを飛ばすと、動くが誰も使えない型になる
出所：キット構成（adl_deck/ 配下）

手元での改良は、見本での回帰確認を通してから配布用に固め直します



▶ キットは配って終わりではなく、案件で足りなかった型を足して育てる前提で作っています

7 注意点

出所のない数値を書かないことと、承認前に生成しないことが最も重要な制約です

	制約	なぜそうするか	破ったときに起きること
中身	出所のない数値を書かない	生成された数値は根拠がなく、提出後に確認を求められる	資料全体の信頼が落ち、全ページの数値を洗い直すことになる
	架空の固有名詞を実在扱いしない	例示のつもりの社名が実在することがある	先方に誤解が生まれ、訂正の連絡が必要になる
	箇条書きだけのページを並べない	納品物の密度に届かず、資料としての価値が下がる	点検で薄いページとして検出され、作り直しになる
見た目	テーマにない色を使わない	テンプレの配色定義から外れる	ブランドの体裁が崩れ、準拠と言えなくなる
	欧文書体を指定しない	テンプレは Arial を定義しており、上書きすると混在する	書き出した資料で書体が揃わない
	作業領域の外に本文を置かない	テンプレが明記している版面の制約	印刷と投影で文字が切れる
進め方	構成の承認前に生成しない	構成の誤りは全ページの作り直しになる	手戻りが最も大きい段階でやり直すことになる
	点検を通さずに完了としない	目で見えない崩れが残る	提出後に崩れが見つかる
	台紙を差し替えない	台紙はテンプレ原本から作った唯一の準拠の根拠	準拠が保証できなくなる

環境まわりで詰まる箇所は決まっており、事前に避けられます

	症状	原因	対処
書き出し	PDF への書き出しが応答しなくなる	出力先が隠しフォルダ（先頭がドットのフォルダ）の下にあり、PowerPoint と Keynote が開けない	出力先を案件フォルダなど通常のフォルダに変える
	書き出しが失敗して止まる	PowerPoint が異常終了したときの一時ファイルが残っている	生成時に自動で外れる。残る場合は出力先の ~\$ で始まるファイルを削除する
生成	画像への変換ができない	PDF を画像にする外部コマンドが入っていない	poppler を導入する（brew install poppler）
	同じ型が続くと警告が出る	版面が偏っており、読み手に単調な印象を与える	本文6枚につき1種類は違う型を混ぜる
	ページが薄いと指摘される	箇条書きだけで図版がない	図版を足すか、前後のページと統合する
	画像の枠が空のまま残る	画像を用意していない	plan に指示を書いて場所取りし、後から差し替える

▶ 出力先を通常のフォルダにしておくだけで、詰まりの大半は起きません

ARTHUR  LITTLE

THE DIFFERENCE