

# claude code と marp で作る 対話型スライド作成

AIとの対話で効率的にプレゼンテーションを作成する方法

2025年1月



## 目次

1. Claude Codeとは
2. 4段階ワークフローの概要
3. フォルダ構造とCLAUDE.md
4. 対話でスライドを作る流れ
5. 実際の対話例（初回作成）
6. XX\_Resourceフォルダでフィードバック
7. 問題発見から修正までの流れ
8. 効率的な指示のコツ



## Claude Codeとは

### Anthropic公式のAI開発ツール

#### 特徴:

**対話型開発:** 自然言語での指示で開発

**ファイル操作:** 読み書き・編集が可能

**コマンド実行:** bashコマンドの実行

**マルチファイル対応:** 複数ファイルの同時編集

#### メリット:

プログラミング知識不要でスライド作成

反復作業の自動化

リアルタイムでのフィードバック反映



## 4段階ワークフローの概要

### スライド作成の体系的アプローチ

00\_アイデア方針（何を作るか）



01\_ドラフト（内容を書く）



02\_Marp（スライド形式に変換）



03\_Export（PPTXに出力）

#### 各段階の役割:

**00:** 方針決定とルール設定

**01:** 通常のMarkdownで内容作成

**02:** Marp形式に自動変換

**03:** 最終成果物の出力



## フォルダ構造とCLAUDE.md

### プロジェクトの基本構造

```
プロジェクト/  
├─ CLAUDE.md          # AIへの指示書  
├─ 00_アイデア方針/  
│   └─ 方針.md        # 作成方針  
├─ 01_ドラフト/  
│   └─ スライド.md    # 内容  
├─ 02_Marp/  
│   └─ スライド.md    # Marp形式  
├─ 03_Export/  
│   └─ PPTX/  
│       └─ スライド.pptx  
└─ XX_Resource/      # フィードバック用
```



## CLAUDE.mdの重要性

### プロジェクトのルールブック

#### CLAUDE.mdとは:

プロジェクトのルールブック

AIとの約束事を記載

継続的に更新・改善

#### 記載内容の例:

基本原則（日本語対話、削除時確認など）

プロジェクト固有ルール

ワークフローの定義

頻出する修正パターン



# 対話でスライドを作る流れ

## 基本的な作成プロセス

### 1. 初回の指示

「生成AIツールの紹介スライドを作ってください」

### 2. Claudeの応答

フォルダ構造の作成

方針の確認

ドラフト作成

### 3. 確認と修正

「もっとビジネス向けの内容にして」

### 4. 自動的な処理



## 実際の対話例（初回作成）

### リアルな会話の様子

ユーザー:

「Marpでスライドを作成する手順を教えるプレゼンを作って」

Claude Code:

了解しました。以下の手順で作成します：

1. フォルダ構造を作成
2. 00\_アイデア方針に方針.mdを作成
3. 01\_ドラフトに内容を作成
4. 02\_Marpに変換しますか？

ユーザー: 「はい、お願いします」

Claude Code:

02\_Marpに変換しました。  
エクスポートしますか？





# XX\_Resourceフォルダでフィードバック

## 視覚的な問題の共有方法

### 1. 問題の発見

PPTXを開いて確認

文字切れや表示崩れを発見

### 2. スクリーンショット保存

```
XX_Resource/  
└─ スクリーンショット.png
```

### 3. Claudeへの報告

「XX\_Resourceフォルダを作りました。  
スクリーンショットを見てください。  
文字が切れています」



# 問題発見から修正までの流れ

## 実際の修正プロセス

### 1. 問題の報告

ユーザー：「コマンドが途切れて見えません」

### 2. Claudeの分析

Claude：「スクリーンショットを確認しました。  
以下を修正します：

- フォントサイズを16pxに縮小
- 長いコマンドに改行を追加

### 3. 自動修正

02\_Marpのスタイル調整

コンテンツの改行追加

再エクスポート



## 効率的な指示のコツ

### Claude Codeを最大限活用する方法

#### 良い指示の例:

「ビジネス向けのスライドを20枚程度で」

「各スライドに具体例を含めて」

「文字が切れているので修正して」

#### 避けるべき指示:

「いい感じにして」(曖昧すぎる)

「全部やり直して」(非効率)

#### プロジェクト管理のコツ:

1. **CLAUDE.mdを活用**: ルールを明文化

2. 段階的に進める: ②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳㉑㉒㉓㉔㉕㉖㉗㉘㉙㉚㉛㉜㉝㉞㉟㊱㊲㊳㊴㊵㊶㊷㊸㊹㊺㊻㊼㊽㊾㊿



## ワークフローの実例

### 今日の作業の流れ

#### 1. 最初の指示

「スライド作成手順のプレゼンを作って」

#### 2. 途中の修正

「AIツール紹介じゃなくて手順説明にして」

「もっとMarp自体の説明を減らして」

#### 3. 視覚的フィードバック

スクリーンショットで問題共有

「文字が切れている」と報告

#### 4. 最終調整

「Claude Codeとの対話方法に焦点を当てて」



## CLAUDE.mdの活用

### プロジェクトの一貫性を保つ

#### 記載すべき内容:

# CLAUDE.md

#### ## 基本原則

- 日本語で対話
- ファイル削除時は確認
- 4段階ワークフローを厳守

#### ## プロジェクト固有ルール

- PPTXのみエクスポート
- フォントサイズは16px以上
- 長いコマンドは改行



## CLAUDE.mdの更新タイミング

### ルールを育てる

#### 更新すべきとき:

新しいルールが必要になったとき

繰り返し同じ修正をするとき

プロジェクトの方針が変わったとき

#### 例:

「毎回フォントサイズの修正をしている」

↓

CLAUDE.mdに追記:

「コードブロックは16px以上」



## よくある質問と対処法

**Q: スライドのデザインを変えたい**

**A:** 以下のように指示：

「背景を青系にして、  
フォントを大きくしてください」

**Q: 内容を大幅に変更したい**

**A:** 00\_アイデア方針から修正：

「方針を変更したいので、  
00フォルダから修正してください」

# まとめ

---

Claude CodeとMarpの組み合わせの強み





# 効率化のポイント

## 成功の4つの要素

### 1. 対話型開発

自然な言葉で指示

### 2. 自動化

4段階の変換を自動実行

### 3. 即座の修正

フィードバックを即反映

### 4. 学習機能

CLAUDE.mdでルール蓄積



## 成功の秘訣

### 効果的な使い方

#### 重要なポイント:

明確な指示を心がける

フィードバックループを活用

CLAUDE.mdを育てる

段階的に改善

#### 覚えておくこと:

完璧を求めず、徐々に改善

視覚的なフィードバックは強力

ルールの蓄積が効率化の鍵

# 次のステップ

---

実際に試してみよう



# 始め方

## 4つのステップ

### 1. Claude Codeをインストール

公式サイトからダウンロード

### 2. 簡単なスライドから開始

5枚程度の短いプレゼンで練習

### 3. XX\_Resourceでフィードバック

スクリーンショットを活用

### 4. CLAUDE.mdにルールを追加

学んだことを記録



## 参考リソース

### さらに学ぶために

#### Claude Code公式ドキュメント

インストール方法

基本的な使い方

#### Marp公式サイト

Markdown記法

スタイリングガイド

#### 本プロジェクトのCLAUDE.md

実践的なルール集

バースト・プロセッシング

# ありがとうございました

AIとの対話で、スライド作成を効率化しましょう！

質問・ご相談はお気軽に