

Claude Code でスライドを 作るテンプレート

IR を書けば、検査・図・アニメーション・公開まで揃う

はじめに

AI に直接スライドを書かせると、毎回ぶれる

AI に直接スライドを書かせると、毎回ぶれる

レイアウトや配色がデッキごとに違う

AI に直接スライドを書かせると、毎回ぶれる

レイアウトや配色がデッキごとに違う

アニメーションの指定が場当たりのになる

AI に直接スライドを書かせると、毎回ぶれる

レイアウトや配色がデッキごとに違う

アニメーションの指定が場当たりのになる

構成図を手で描き直すことになる

AI に直接スライドを書かせると、毎回ぶれる

レイアウトや配色がデッキごとに違う

アニメーションの指定が場当たりのになる

構成図を手で描き直すことになる

何を直せば良いか機械的に分らない

AI に直接スライドを書かせると、毎回ぶれる

レイアウトや配色がデッキごとに違う

アニメーションの指定が場当たりのになる

構成図を手で描き直すことになる

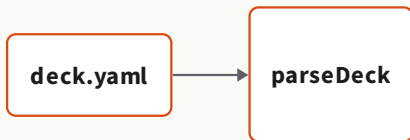
何を直せば良いか機械的に分らない

| 正は YAML、スライドは生成物

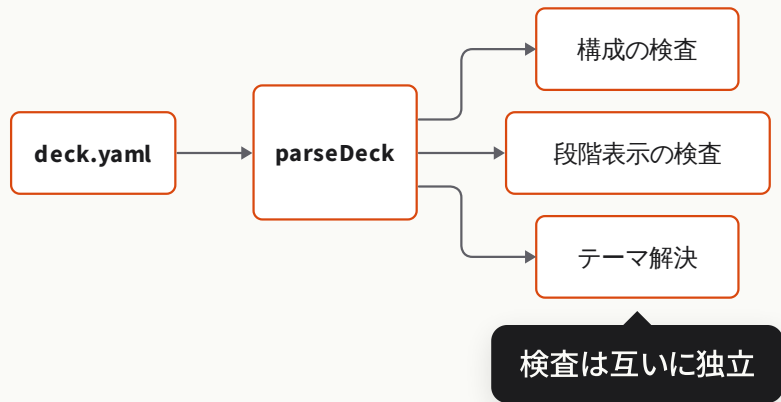
純粋関数のパイプラインで slides.md を作る

`deck.yaml`

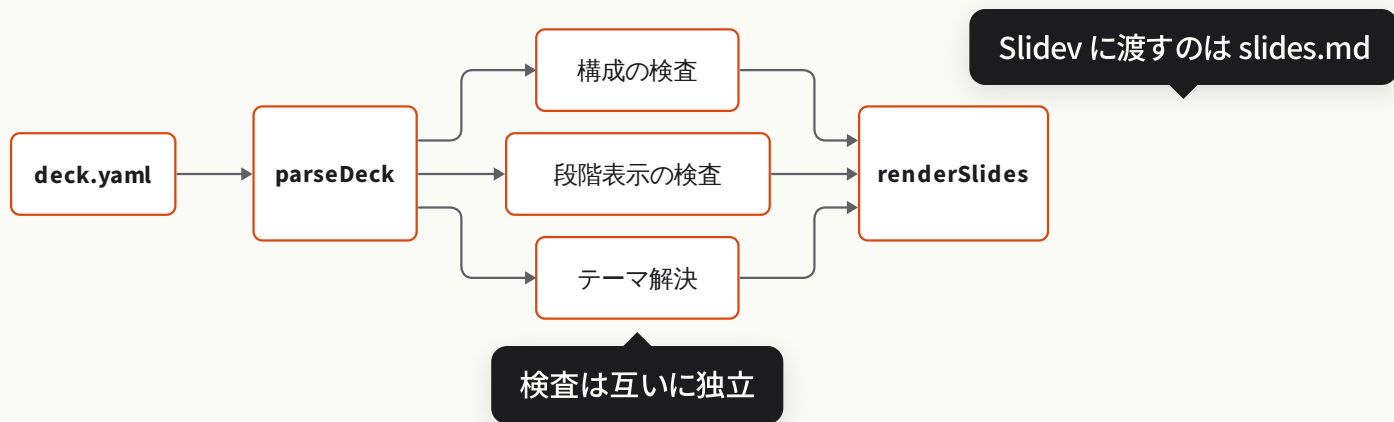
純粋関数のパイプラインで slides.md を作る



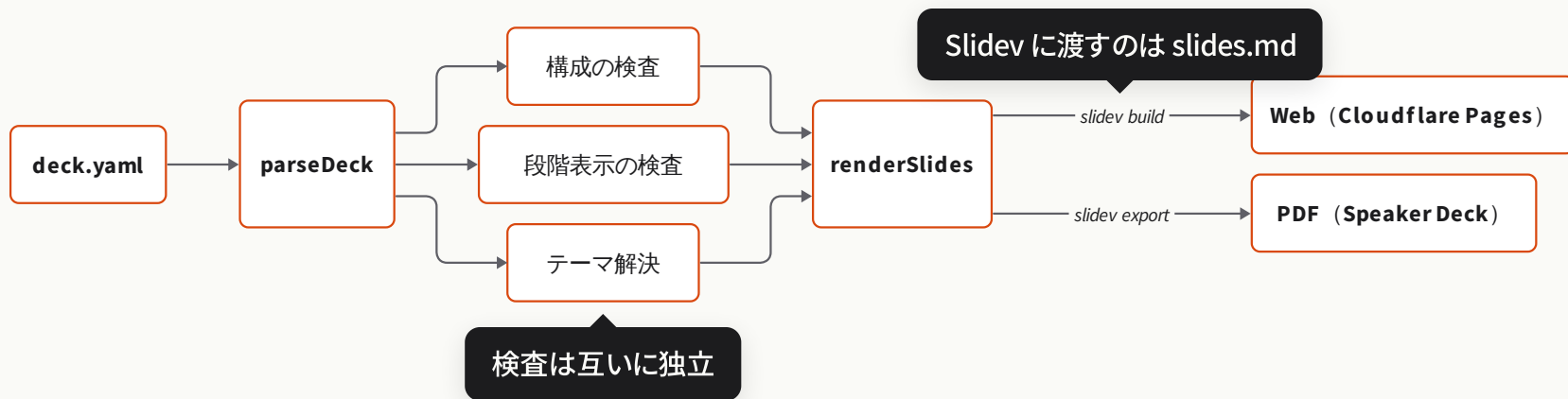
純粋関数のパイプラインで slides.md を作る



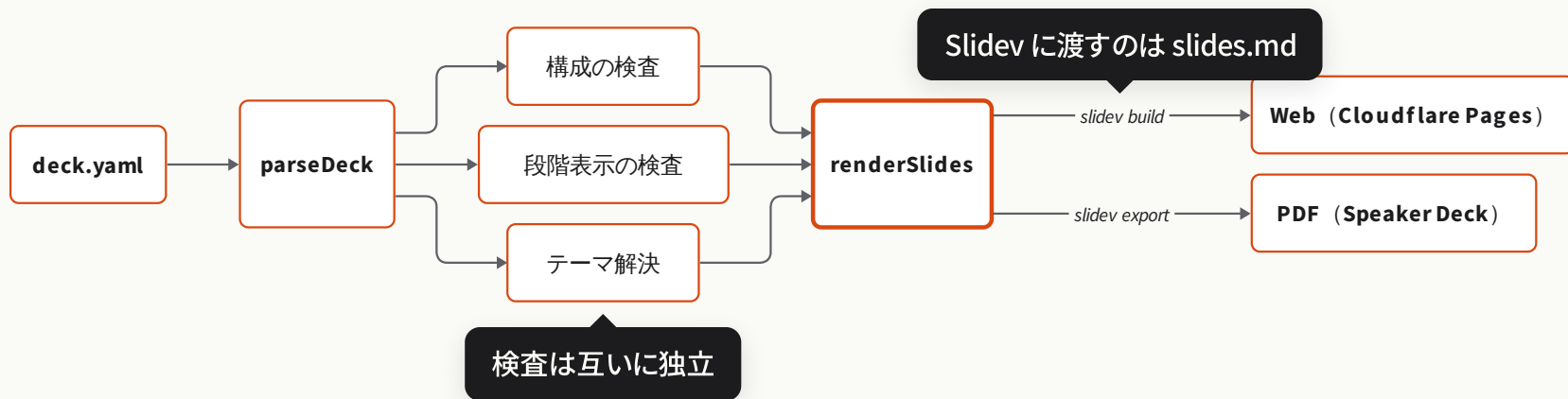
純粋関数のパイプラインで slides.md を作る



純粋関数のパイプラインで slides.md を作る



純粋関数のパイプラインで slides.md を作る



Claude には「何を言うか」だけを書かせる

Claude が書く

テンプレートが決める

1スライド1メッセージの主張

段階表示の順序

図のノードとエッジ

Claude には「何を言うか」だけを書かせる

Claude が書く

1スライド1メッセージの主張

段階表示の順序

図のノードとエッジ

テンプレートが決める

レイアウトと配色

クリックの実装

図の配置と描画

1枚のスライドはこれだけ

```
- id: problem
  layout: bullets
  message: AI に直接スライドを書かせると、毎回ぶれる
  items:
    - { id: layout, text: レイアウトや配色がデッキごとに違う }
    - { id: review, text: 何を直せば良いか機械的に分からない }
  steps: [layout, review, { highlight: review }]
```

壊れた参照はビルド前に止まる

```
$ pnpm deck check example
```

```
error    sections.1.slides.1.steps.2  要素 zz がこのスライドに無い (unknown-element)
```

```
warning sections.0.slides.1.message  message は 40 文字以内が目安 (message-too-long)
```

**YAML を書けば、検査済みの
スライドが Web と PDF が出る**

