

# Quarto + reveal.js 活用メモ

## デモ研究室

2026-02-06



# 目次

- オリエンテーション
- セクションとサブスライド
  - サブスライドA
  - サブスライドB
- layout: true による共有背景
  - レイアウト継承スライド1
  - レイアウト継承スライド2
- 段階表示 (incremental)
- embed-resources: true の効果
- スピーカーノートの使い方
- layout + columns の例
- 図を右半分に配置する例
- 図を上半分に配置する例
- 図を右下1/4に固定する例
- PDF 化の選択肢
- まとめ

≡

# オリエンテーション

- Quarto = Markdownを多形式に変換する統合ツール
- reveal.js = ブラウザで動くスライドエンジン
- `.qmd` 1つでHTMLスライドとPDFを両立させる例を紹介



# セクションとサブスライド

- `##` 見出しごとに1枚のスライド
- 同じ `##` の直後に `###` を書くと垂直方向に重なるサブスライドになる

## サブスライドA

ここでは1ページ目。

## サブスライドB

下矢印で2枚目へ進む。



# layout: true による共有背景

このレイアウトスライドで設定した背景や要素は続くサブスライドに適用されます。

## レイアウト継承スライド1

- 見出し隣に記述した `{layout: true}` は reveal.js の「ここから先のスライドの基底テンプレート」にする仕組み
- 同じ背景色や配置を複数枚に渡って再利用できます

## レイアウト継承スライド2

- 研究の流れ図などを共通背景として差し込むケースに便利



# 段階表示 (incremental)

- YAMLで `format.revealjs.incremental: true` と書けば、箇条書きが段階的に現れる
- ここでは1クリックごとに行が表示される
- `:::{.incremental}` を本文側に書いて局所的にオンにすることも可能



## embed-resources: true の効果

- CSS/JS/画像を1つのHTMLにバンドル
- ネットワークに接続できない会場や、ファイル単体で配布する場合に便利
- ファイルサイズは大きくなるので、動画は外部ファイルのままが無難



# スピーカーノートの使い方

1. 本文の後に `... notes` ブロックを置く
2. 文章をMarkdownで書く（箇条書きも可）
3. reveal.js のプレゼンタービュー (`s` キー) や、chromeの印刷PDFで脚注扱いになる



# layout + columns の例

```
1 import numpy as np
2 np.random.seed(0)
3 print(np.random.rand)
```

- QuartoのコードブロックはKnitr/ Jupyter実行にも対応
- 今回は実行しない静的スライド



# 図を右半分に配置する例

- 2カラム構成にして、右側を100%幅で画像に割り当てる
- 左カラムに箇条書きや説明を置くと読みやすい
- `width` 属性でカラム比を微調整可能



Right Half Figure



# 図を上半分に配置する例



Top Half Figure

::: {.split-caption}

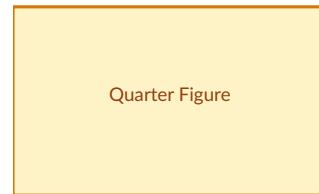
style="padding:1rem;"} - CSSグリッドで上下2段を作り、上段に図、下段に説明文 - `height:70vh` で画面高に対する比率を調整 - 背景を共有したい場合は `layout: true` と組み合わせても良い

:::



# 図を右下1/4に固定する例

- 通常の文章や箇条書きを書いたあと
- `position:absolute` で右下に小さな図を常時配置
- グラフの抜粋やロゴなど、視覚的な補助として便利



# PDF 化の選択肢

- Chrome印刷: `google-chrome --headless --print-to-pdf ...` でreveal.jsの `?print-pdf` 表示をそのままPDF化
- LaTeX: `quarto render slide_deck.qmd --to pdf` でBeamerベースのPDFを生成（要TeX環境）。デザインは異なる可能性あり。



# まとめ

- `.qmd` では YAML 設定・サブスライド・notesなど Markdown 以上の作法を使える
- reveal.js は ブラウザ登壇、Chrome で PDF 化すれば 配布資料も 確保
- 必要に応じて `format.pdf` を 有効にし、LaTeX ベース PDF を併産

