

DUALIS かるた ワークスペースガイド

本ドキュメントは、このワークスペースを初めて開く方や共同で運用する方向けに、プロジェクトの目的、構成、セットアップ、CSVの編集方法、PDF生成、Pythonスクリプトの内部構造、カスタマイズやトラブルシュートまでを一通り解説します。

概要と目的

- 目的: 日本十進分類法（NDC）を楽しく学べる「かるた」を、A4用紙に面付けして印刷できるPDFとして生成します。
- 対象: 学生団体 DUALIS メンバー、および関連する学生・団体。
- 出力: 名刺サイズ (91mm x 55mm) の札を A4 に 2 列 5 行 (10枚/ページ) で面付け。両面印刷（長辺綴じ）に対応。
- PDFページ構成（この順で出力）：
 - 絵札A（表）：分類番号
 - 絵札B（裏）：名称（行ごとに左右反転配置）
 - 読み札A（表）：「番号 + 名称」
 - 読み札B（裏）：ボナスゲーム名（必要な札のみ、行ごとに左右反転配置）

リポジトリ構成（主要）

```
DUALISかるた/  
├ generate_pdf.py          # PDF生成スクリプト（面付け、表裏反転、日本語フォ  
└   ト・CJK折返し等）  
├ JDC_karuta.csv          # かるたデータ（番号・名称・色・フォント等）  
├ requirements.txt        # 依存（reportlab, Pillow）  
├ .vscode/  
│   └ tasks.json          # VS Codeタスク（build: pdf-generator）  
├ fonts/  
└   等）                  # 日本語フォント(TTF/OTF)を配置する場所（HaranoAji  
├ output/                 # 生成PDFの出力先  
├ Docs/  
│   └ WORKSPACE_GUIDE.md  # 本ドキュメント  
└   README.md             # クイックスタート
```

セットアップ手順

- 推奨: VS Code のビルドタスク（Ctrl+Shift+B）を使えば仮想環境の Python で自動実行されます。
- 手動セットアップ（必要な場合）：
 - 仮想環境作成: `python -m venv .venv`
 - 有効化: (OSに応じて)
 - Linux/macOS: `source .venv/bin/activate`
 - Windows PowerShell: `.venv\Scripts\Activate.ps1`
 - 依存インストール: `pip install -r requirements.txt`

使い方（PDF生成）

- CSVを編集: **JDC_karuta.csv** をUTF-8で編集し、番号・名称・色・フォント等を更新
- 生成: VS Codeで Ctrl+Shift+B (タスク: pdf-generator)
- 出力先: **output/DUALIS_karuta_print_<範囲>_<YYYYMMDD>.pdf**
 - 例: **output/DUALIS_karuta_print_000-009_20250910.pdf**
 - 範囲はCSV内の **number** から数字を抽出して最小-最大を付与（見つからなければ **ALL**）
- 印刷設定: 両面印刷（長辺綴じ）を選択。裏面は行ごとに左右反転配置済みで、表裏がぴったり合う設計です。

CSV仕様（JDC_karuta.csv）

- 文字コード: UTF-8（Excelで保存する場合もUTF-8を選択）
- 1行=1札。ヘッダーは以下の通り:

```
number,name,color_code,font,point_size,is_bonus_game,bonus_game_name
```

各列の意味:

- **number**: 分類番号（例: **000**, **007** など）。数字以外が含まれても動作しますが、範囲計算は数字のみ抽出します。
- **name**: 名称（日本語OK）。日本語フォントで自動表示されます。
- **color_code**: **#RRGGBB** 形式のカラーコード。未指定なら **number** の3桁目で自動着色（簡易マップ）
- **font**: フォント名。存在しない場合は自動フォールバック
 - 日本語を含むテキストは **fonts/** 配下の TTF/OTF（HaranoAji等）があればそれを優先
 - ない場合、ReportLab内蔵CIDフォント **HeiseiMin-W3** を使用
- **point_size**: 文字サイズ（数値）。推奨 20~36
- **is_bonus_game**: ボーナス札フラグ（**true/1/yes/y** のいずれかで真）
- **bonus_game_name**: ボーナスゲーム名。長い場合はCJK対応の自動改行で折り返します

サンプル:

```
number,name,color_code,font,point_size,is_bonus_game,bonus_game_name
000,総記,#8E8E93,Helvetica,28,false,
001,知識・学問・学術,#007AFF,Helvetica,28,false,
009,書誌学,#FF9F0A,Helvetica,28,true,図書館の自由に関する宣言を読み上げる
```

PDFレイアウトと表裏設計

- サイズ: A4（210x297mm）
- 札サイズ: 91mm x 55mm（名刺）
- 面付け: 2列 x 5行（10枚/ページ）
- 余白: ページ中央で均等化
- 絵札A（番号）/絵札B（名称）/読み札A（番号+名称）/読み札B（ボーナス）の順でページを生成
- 裏面（B面）は行単位で左右反転配置（長辺綴じで表裏が一致）

フォント（日本語フォールバック）

- 推奨: 原ノ味フォント (HaranoAji) を `fonts/` に配置
 - 例: `fonts/HaranoAjiMincho-Regular.ttf` (明朝)、`fonts/HaranoAjiGothic-Medium.ttf` (ゴシック)
- 未配置でも `HeiseiMin-W3` (内蔵CIDフォント) へ自動フォールバック
- CSVに `font` を指定しても、日本語を含むテキストは日本語フォントに自動切替

ボーナスゲームの割当ロジック

- CSVで `is_bonus_game = true` の行はそのまま尊重
- `bonus_game_name` が空ならデフォルトプールから自動補完
- 全体として概ね「20枚に1枚」になるよう、不足分は20枚ごとの位置 (例: 20, 40, ...のカード) に自動追加 (決定論的)

CJK対応の自動改行 (はみ出し対策)

- 日本語を含む文は「文字単位」の貪欲法で改行し、枠幅に収まるように調整
- 英数字中心 (スペース含む) の文は単語ベースの改行 (ReportLab の `simpleSplit`)
- それでも収まらない場合は、CSVで `point_size` を下げるか、文言を短縮してください

出力ファイル名のルール

- 形式: `DUALIS_karuta_print_<範囲>_<YYYYMMDD>.pdf`
 - 範囲: CSVの `number` から数字だけを抽出して最小-最大 (ゼロ埋め)。数字が見つからない場合 `ALL`
 - 日付: 生成日を `YYYYMMDD` で付与

スクリプトの内部構造 (generate_pdf.py)

主な定数:

- `PAGE_SIZE = A4`, `CARD_W_MM = 91`, `CARD_H_MM = 55`, `COLS = 2`, `ROWS = 5`

主要関数:

- `load_cards(csv_path)`: CSVを読み込み、各行を辞書化 (色・フォント・サイズの補正含む)
- `assign_bonus(cards)`: ボーナス札の自動割当と名前の補完
- `layout_positions(page_w, page_h)`: 2x5 面付けの座標を計算
- `draw_card_border(...)`: 枠線の描画 (カラーコード適用)
- `draw_centered_text(...)`: テキスト描画 (日本語フォント切替、CJK折返し、中央寄せ)
- `draw_picture_front/back(...)`: 絵札A/B のページ描画 (裏面は行ごとに左右反転)
- `draw_reading_front/back(...)`: 読み札A/B のページ描画 (裏面は行ごとに左右反転)
- `wrap_text(...)`: CJK対応の改行 (文字単位の貪欲折返し)
- `generate(pdf_path, csv_path)`: 全ページ生成のオーケストレーション
- `__main__`: 出力範囲と日付からファイル名を組み立て、`output/` へ保存

カスタマイズのポイント

- 札のサイズ: `CARD_W_MM`, `CARD_H_MM`
- 行列数: `COLS`, `ROWS` (面付け枚数を増減)

- 余白/枠線太さ: `layout_positions` と `draw_card_border` (`setLineWidth`)
- 文字サイズ/行間: CSVの `point_size`、`draw_centered_text` の `leading`
- 色の既定マップ: `DIGIT_COLOR`
- ボーナスプール: `BONUS_POOL`
- フォント: `fonts/` に任意フォントを追加、内部登録名はファイル名から決まります

VS Codeタスク

- `.vscode/tasks.json` に `pdf-generator` が定義され、`"${command:python.interpreterPath}"` を使用
- クロスプラットフォームで仮想環境内の Python が呼ばれます
- 実行: Ctrl+Shift+B (ビルドタスク)

トラブルシューティング

- `ModuleNotFoundError: reportlab` など
 - 仮想環境が有効か確認 → 依存を再インストール: `pip install -r requirements.txt`
- 日本語が豆腐 (□) になる
 - `fonts/` に HaranoAji などの日本語フォントを配置
 - それでも出ない場合はフォントファイルの破損や権限を確認
- 両面印刷の表裏がずれる
 - プリンタ設定を「長辺綴じ」にする
 - 用紙の向き・余白がプリンタ側で自動調整されないよう注意
- CSVが文字化け
 - UTF-8 で保存。Excelからエクスポート時の文字コードに注意

ライセンス

- 本リポジトリのコード: MIT (必要に応じて `LICENSE` を追加してください)
- フォントのライセンスは各フォント配布元の規約に従います (HaranoAji など)

このガイドに追記が必要な場合は、`Docs/WORKSPACE_GUIDE.md` を直接編集して Pull Request を作成してください。