版面の密度

この文書は同じ原稿から 4 回組まれています。`relaxed` ・ `normal` ・ `compact` ・ `tight` の 4 つで、 用紙も余白も同じです。4 つの PDF で違うのは `pdf.density` だけなので、ここで見えている違いは すべてその設定によるものです。

どこを見るか

密度が動かす 4 つの値は、文書が何枚になるかを決めている 4 つです。どれもこのページに出ています。

- **文字の大きさ。** 本文をどれだけの大きさで組むか。誰でも最初に気づく値であり、密度が最後に 手を出すべき値でもあります。小さくすると 1 行が長くなるからです。
- **行送り。** ある行のベースラインから次の行のベースラインまでの距離。画面向けのスタイルシート はここに余裕を持たせていて、紙の上ではその余裕こそが枚数の大半を払っています。
- **見出し上の空き。** 節と節を隔てる空間。画面ではスクロールで通り過ぎますが、紙では枚数で 買うことになります。
- **表のセル余白。** 表が少ないうちは目に入らず、表が主体の文書では、その表が 1 ページに収まるか 2 ページになるかを決めます。

枚数はどこから出てくるか

`relaxed` と `normal` のあいだで、文字の大きさは変わりません。どちらも本文は 16px です。変わるの は行送りと見出し上の空き、そして表のセル余白だけです。それでも文書は短くなります。画面向けに 組まれたページは「紙向けのページを大きくしたもの」ではなく、その差の大半は文字ではなく空きだ、 ということです。

`normal` より下では文字も小さくなり、それに連れてもう 1 つのことが起こります。本文の段の幅は `pdf.margin` が残した幅そのもので、密度はそれを狭めません。つまり文字を 1 段小さくするたびに、 1 行に入る文字数は増えます。A4 の既定余白では、16px でおよそ 42 字、12px では 56 字ほどになり ます。これは実際の代償です。詰まったプリセットを既定にせず、選んで使うものにしてある理由でも あります。

行を長くせずに文字だけ小さくしたいときは、同じ変更で `pdf.margin` も広げてください。この 2 つを 別の設定にしてあるのは、1 行の適切な長さが文書の書かれている文字体系によって変わり、プリセット が代わりに決められるものではないからです。

プリセット

プリセット	文字	行送り	見出し上	セル余白	どう読めるか
`relaxed`	16px	1.7	1.8em	0.5rem 0.8rem	画面で見ているままの文書
`normal`	16px	1.45	0.9em	0.35rem 0.6rem	既定。画面と同じ文字で、紙向けに組む
`compact`	14px	1.35	0.8em	0.3rem 0.5rem	配って回す報告書
`tight`	12px	1.3	0.6em	0.2rem 0.35rem	通しで読まれない付録

`relaxed` は画面の設定に名前を付けたものです。スタイルシートが既にしていることそのものなので、 指定しても印刷用の規則は 1 つも書き出されません。この HTML を自分のブラウザから印刷する人に 文字サイズを固定してしまう規則も、書かれません。

指定のしかた

たいていは名前で足ります。

```yaml
pdf:
  pageSize: A4
  margin: { top: 20mm, right: 15mm, bottom: 20mm, left: 15mm }
  density: compact
```

プリセットが惜しいところまで来ていて合わない場合は、名前の代わりにオブジェクトを書きます。 `base` が土台にするプリセットを指し、オブジェクトは名指しした値だけを差し替えます。1 つ変える ために残り 3 つを書き写す必要はありません。

```yaml
pdf:
  density:
    base: compact
    fontSize: 13px
    lineHeight: 1.4
```

セル余白のための表

セル余白はいちばん見落としやすく、文書が表ばかりになった途端にいちばん無視できなくなる値です。 下の行はそれを目に見えるようにするために置いてあります。

作業	担当	期限	状態	備考
元原稿を集める	ドキュメント	第 1 週	完了	2 本はまだ wiki の中にあった
節の順序を決める	ドキュメント	第 1 週	完了	ファイル名ではなくサイドバーで決める
表を作り直す	開発	第 2 週	完了	横に広いものは列の切り方から見直した
図を描く	開発	第 2 週	進行中	Mermaid。PDF にランタイムを要らなくするため事前描画
ビルド機のフォントを確認する	開発	第 2 週	完了	`fontCheck` が成果物を作る前に不足を報告する
版面の密度を決める	ドキュメント	第 3 週	進行中	この文書がその比較
余白を決める	ドキュメント	第 3 週	未着手	上の密度次第
紙で通して読む	レビュー	第 3 週	未着手	実際に印刷するプリンタで
印刷で出た問題を直す	ドキュメント	第 4 週	未着手	必ず何かある
フッタにページ番号を入れる	開発	第 4 週	完了	既定のフッタ。設定は要らない
レイアウトを承認する	レビュー	第 4 週	未着手	署名は 4 つではなく 1 つ
PDF を引き渡す	ドキュメント	第 4 週	未着手	組み直せるよう原稿も一緒に

行送りのための箇条書き

番号付きの箇条書きは本文と同じ行送りで組まれるので、同じ違いが出ます。

1. 同じ原稿から、密度ごとに文書を組む。
2. 画面ではなく、実際に読まれる紙に 1 枚ずつ印刷する。
3. まず本文を読み、次に表、最後に見出しを見る。
4. 決められた枚数に収まる範囲で、いちばん緩いものを選ぶ。
5. 選んだ値を `monodocs.config.yml` に書く。次のビルドが同じ文書を作るように。

飛ばされるのは最後の 1 つです。一度選んで設定に残した密度は、形を保つ文書になります。プリンタの 前で選んだ密度は、誰かが組み直すたびに形の変わる文書になります。

Generated by monodocs v0.19.0