monodocs サンプル

目次

概要 4
ビルド / プレビュー 4
このサイトで確認できる機能 4
markdown 5
GFM サンプル 5
見出し（ATX） 5
レベル 2 6
見出し（Setext） 6
小見出し（Setext H2） 6
段落と改行 6
水平線 7
引用 7
エスケープと実体参照 8
テキスト書式 9
強調 9
リンク 9
オートリンク（GFM 拡張） 10
画像 10
リスト 12
箇条書き（順序なし） 12
番号付き（順序あり） 12
タスクリスト（GFM 拡張） 13
ゆるいリストと詰まったリスト 14
項目に複数ブロックを含める 14
コードと表 16
フェンスドコードブロック 16
言語指定なし / インデントコード 17
表（GFM 拡張） 17
Mermaid 18
脚注 19
Footnotes 19
アラート（GFM alerts） 20
5 種類のアラート 20
複数段落・他の記法を含む 21
幅の広いコンテンツ 22
幅の広い表 22
行の長いコードブロック 22
大きな画像 23

Mermaid（横長の図） 23
数式 24
文中の数式 24
別行立ての数式 24
行列と揃えた式 25
advanced 26
応用（ネスト階層） 26
深いフォルダでもリンクは追従する 26
深い見出しと目次 27
レベル 2：概要 27
レベル 2：まとめ 27
asciidoc 28
AsciiDoc サンプル 28
セクション見出し 28
段落と改行 28
テキスト書式 29
置換（テキスト置換） 30
注釈（コメント） 30
リスト 31
箇条書き（順序なし） 31
番号付き（順序あり） 31
チェックリスト 32
説明リスト（description list） 32
用語と入れ子 33
リスト項目に複数ブロック 34
ブロック 35
ソースブロック（構文ハイライト） 35
リスティングブロック / リテラルブロック 36
例ブロック 36
サイドバーブロック 37
引用ブロック / 詩ブロック 37
Admonition（注意書き） 38
Mermaid（[source,mermaid]） 40
パススルー（生 HTML） 40
表 42
基本の表 42
セル内の書式と複数行 43
CSV 形式 43
リンクと画像 45
リンク 45
属性（attribute） 45
内部アンカーと相互参照 46

画像 46
UI マクロ 47
脚注 47
幅の広いコンテンツ 48
幅の広い表 48
行の長いリスティング 48
大きな画像 49
mixed 50
混在（同一フォルダ） 50
アンカー先の見出し 50
相棒（AsciiDoc） 51
アンカーの例 51
PDF 出力 52
PDF に含まれるもの 52
設定（monodocs.config.yml） 53
実行環境の要件 53
検索 54
表記のゆれの扱い 54
複数キーワードでの絞り込み 54
見出しに当たった結果 55
キーボード操作 55
本文のハイライト 55
緩く当てる語 55

概要

🌐 English · **日本語**

`monodocs` の対応記法と機能を **1 つのサイト**でまとめて確認できるサンプルです。 左サイドバーのフォルダから各カテゴリを開けます。

各記法ページは **ソース（生の Markdown / AsciiDoc）** を先に示し、その下に **表示（HTML 変換結果）** を並べる対比形式で、記述と描画の対応がひと目で分かります。

- **Markdown (GFM)**: Markdown サンプル
- **AsciiDoc**: AsciiDoc サンプル
- **混在（同一フォルダ）**: Markdown / AsciiDoc 混在

2 つ目は Markdown から AsciiDoc ページへのリンクです。形式をまたいだリンクも 単一 HTML 内の hash route に変換されます（混在対応）。

ビルド / プレビュー

```
monodocs build examples/ja -o dist/sample.html   # 単一 HTML
monodocs build examples/ja --format pdf -o dist/sample.pdf   # PDF
monodocs serve examples/ja            # http://127.0.0.1:4173/
```

単一 HTML を経由した **PDF 出力**（しおり・本文リンク・フォント要件など）は PDF 出力を参照してください。

検索には、このサンプルに対して実際に入力できるクエリをまとめてあります。 表記のゆれの扱い、キーボード操作、本文のハイライトをその場で確かめられます。

このサイトで確認できる機能

- 全文検索 / ページ内目次（スクロール連動）/ 前後ページナビ
- ダークモード / サイドバー折りたたみ・自動展開
- コードハイライト（shiki）/ Mermaid 図 / 数式（MathML）
- 画像の data URI 埋め込み / ファイル間リンク・AsciiDoc xref / 脚注
- 本文幅より広い表・コード・画像・図の表示（Markdown / AsciiDoc）
- 印刷時の全ページ縦展開 / PDF 出力（しおり・クリック可能なリンク付き）

GitHub Flavored Markdown サンプル

このサンプルは GitHub Flavored Markdown（CommonMark + GFM 拡張）の 主な記法を網羅し、`monodocs` での表示を確認するためのものです。

各項目は **ソース（生の Markdown）** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

ビルド例:

```
monodocs build examples/ja -o dist/sample.html
```

各カテゴリは左サイドバーから開けます: テキスト書式 / リスト / コードと表 / 脚注とリンク参照 / アラート / 幅の広いコンテンツ / 数式 / 応用（ネスト階層）。

見出し（ATX）

`#` を 1〜6 個でレベルを表します。

ソース:

```
## レベル 2

### レベル 3

#### レベル 4

##### レベル 5

###### レベル 6
```

表示:

レベル 2

レベル 3

レベル 4

レベル 5

レベル 6

見出し（Setext）

`===` / `---` の下線でも見出しになります（H1 / H2 のみ）。

ソース:

```
大見出し（Setext H1）
==========

小見出し（Setext H2）
----------
```

表示:

大見出し（Setext H1）

小見出し（Setext H2）

段落と改行

空行で段落を区切ります。強制改行は行末のバックスラッシュ `\`（または目に見えない空白 2 つ）で行います。 ここでは可視で分かりやすいバックスラッシュの例を示します。

ソース:

```
これは段落です。空行で段落を区切ります。

バックスラッシュで強制改行できます。\
このとおり次の行になりました。
```

表示:

これは段落です。空行で段落を区切ります。

バックスラッシュで強制改行できます。
このとおり次の行になりました。

水平線

`---` / `***` / `___` のいずれでも水平線になります。

ソース:

```
セクションの区切り。

---
```

表示:

セクションの区切り。

引用

`>` で引用、ネストや引用内のリストも書けます。

ソース:

```
> 引用です。
>
> > ネストした引用。
>
> - 引用内のリスト
> - 2 つ目
```

表示:

> 引用です。
>
> > ネストした引用。
> >
> > - 引用内のリスト
> > - 2 つ目

エスケープと実体参照

バックスラッシュで記号をエスケープし、HTML 実体参照も使えます。

ソース:

```
バックスラッシュで記号をエスケープ: \*not italic\* / \`not code\` / \# not heading

実体参照: &copy; &amp; &lt; &gt; &#169; &hearts;
```

表示:

バックスラッシュで記号をエスケープ: *not italic* / `not code` / # not heading

実体参照: © & < > © ♥

テキスト書式・リンク・画像

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

強調

イタリック・太字・取り消し線・インラインコードなどの装飾です。

ソース:

```
- イタリック: *アスタリスク* / _アンダースコア_
- 太字: **アスタリスク 2 つ** / __アンダースコア 2 つ__
- 太字イタリック: ***3 つ*** / **_混在_**
- 取り消し線（GFM）: ~~取り消し~~
- インラインコード: `const x = 1;` / バッククォートを含む場合: `` `code` ``
```

表示:

- イタリック: *アスタリスク* / *アンダースコア*
- 太字: **アスタリスク 2 つ** / **アンダースコア 2 つ**
- 太字イタリック: ***3 つ*** / ***混在***
- 取り消し線（GFM）: ~~取り消し~~
- インラインコード: `const x = 1;` / バッククォートを含む場合: `` `code` ``

リンク

インライン・タイトル付き・参照スタイル・同一サイト内リンクです。

ソース:

```
- インライン: [monodocs](https://example.com/monodocs)
- タイトル付き: [ホバーで表示](https://example.com "タイトルです")
- 参照リンク: [参照スタイル][ref] と [ラベル省略][]
- 同一サイト内の別ページ: [リストのページ](lists.md)

[ref]: https://example.com/reference
[ラベル省略]: https://example.com/collapsed
```

表示:

- インライン: monodocs
- タイトル付き: ホバーで表示
- 参照リンク: 参照スタイル と ラベル省略
- 同一サイト内の別ページ: リストのページ

オートリンク（GFM 拡張）

URL やメールアドレスを自動でリンクにします。

ソース:

```
- URL をそのまま: https://github.github.com/gfm/
- 山括弧で: <https://example.com>
- `www.` 始まり: www.example.com
- メール: <docs@example.com> / contact@example.com
```

表示:

- URL をそのまま: https://github.github.com/gfm/
- 山括弧で: https://example.com
- `www.` 始まり: www.example.com
- メール: docs@example.com / contact@example.com

画像

インライン画像と参照スタイルの画像です（画像は単一 HTML に data URI で埋め込まれます）。

ソース:

```
![monodocs ロゴ](images/logo.svg "ロゴ")

参照スタイルの画像:

![参照画像][logo]

[logo]: images/logo.svg "参照ロゴ"
```

表示:



参照スタイルの画像:



リスト

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

箇条書き（順序なし）

`-` / `*` / `+` のいずれもマーカーに使えます。インデントでネストします。

ソース:

```
- 項目 A
- 項目 B
  - ネスト B-1
  - ネスト B-2
    - さらにネスト
- 項目 C
```

表示:

- 項目 A
- 項目 B
  - ネスト B-1
  - ネスト B-2
    - さらにネスト
- 項目 C

番号付き（順序あり）

開始番号も指定できます。番号リスト同士は間に段落を挟むと別リストになり、2 つ目は指定した番号から始まります（空行だけでは同じリストの続きになります）。

ソース:

```
1. 最初
2. 次
3. その次
   1. ネスト 3-1
   2. ネスト 3-2

開始番号を指定（間に段落を挟んで別リストにする）：

5. 5 から
6. 6
7. 7
```

表示:

1. 最初
2. 次
3. その次
    1. ネスト 3-1
    2. ネスト 3-2

開始番号を指定（間に段落を挟んで別リストにする）：

5. 5 から
6. 6
7. 7

タスクリスト（GFM 拡張）

`- [ ]` / `- [x]` でチェックボックスになります。

ソース:

```
- [x] 完了したタスク
- [ ] 未完了のタスク
  - [x] 子タスク（完了）
  - [ ] 子タスク（未完了）
```

表示:

- ☑ 完了したタスク
- ☐ 未完了のタスク
    - ☑ 子タスク（完了）

- ☐ 子タスク（未完了）

ゆるいリストと詰まったリスト

項目間に空行があると「ゆるい（loose）」リストになり、各項目が段落として描画されます。

ソース:

```
詰まった（tight）リスト:

- 行間が詰まる
- 2 行目

ゆるい（loose）リスト（項目間に空行）:

- 段落として描画される

- 項目の間に空行があると緩くなる
```

表示:

詰まった（tight）リスト:

- 行間が詰まる
- 2 行目

ゆるい（loose）リスト（項目間に空行）:

- 段落として描画される

- 項目の間に空行があると緩くなる

項目に複数ブロックを含める

リスト項目にインデントを揃えて段落や引用を続けられます。

ソース:

```
- 項目に複数段落を含む例。

  これは同じ項目の 2 段落目です（インデント）。

  > 項目内の引用。
```

表示:

- 項目に複数段落を含む例。

  これは同じ項目の 2 段落目です（インデント）。

  > 項目内の引用。

コードブロックと表

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

フェンスドコードブロック

言語を指定すると shiki で構文ハイライトされます。

ソース:

````
```js
// JavaScript
function greet(name) {
  return `Hello, ${name}!`;
}
console.log(greet("monodocs"));
```
````

表示:

```js
// JavaScript
function greet(name) {
  return `Hello, ${name}!`;
}
console.log(greet("monodocs"));
```

チルダ（`~~~`）のフェンスも使えます。

ソース:

````
~~~yaml
title: tilde fence
nested: true
~~~
````

表示:

```
title: tilde fence
nested: true
```

言語指定なし / インデントコード

言語指定の無いフェンス、または半角スペース 4 つのインデントでもコードブロックになります（ハイライトなし）。

ソース:

````
```
言語指定なしのコードブロック（ハイライトなし）
```

    indented code block
    second line
````

表示:

```
言語指定なしのコードブロック（ハイライトなし）
```

```
indented code block
second line
```

表（GFM 拡張）

`:---` で左寄せ、`:---:` で中央、`---:` で右寄せ。セル内で `**強調**` や `` `code` `` も使えます。

ソース:

```
| 機能             | Markdown | AsciiDoc |
| ---------------- | :------: | -------: |
| 見出し           |    ✅    |       ✅ |
| 表               |    ✅    |       ✅ |
| 脚注             |    ✅    |       ✅ |
| コードハイライト |  shiki   |    shiki |
```

表示:

機能	Markdown	AsciiDoc
見出し	✅	✅
表	✅	✅
脚注	✅	✅
コードハイライト	shiki	shiki

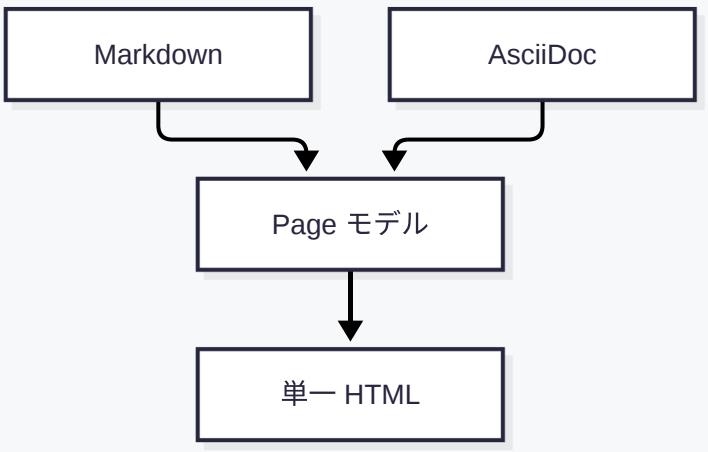
Mermaid

```mermaid``` フェンスは図として描画されます。

ソース:

````
```mermaid
graph TD
  A[Markdown] --> B[Page モデル]
  C[AsciiDoc] --> B
  B --> D[単一 HTML]
```
````

表示:


````

脚注（GFM 拡張）

`[^id]` で脚注参照、`[^id]: ...` で本文を書きます。脚注の ID は単一 HTML 内でページごとに一意化され、 ページをまたいでも衝突しません。参照や戻りリンクをクリックすると、ページを切り替えずに該当箇所へスクロールします。

ソース:

```
本文中に脚注参照を置けます[^1]。複数置くこともできます[^note]。
同じ脚注を再利用することもできます[^1]。長い注釈も書けます[^long]。

[^1]: これが 1 つ目の脚注です。
[^note]: 名前付きラベルの脚注。**強調**や `code`、[リンク](https://example.com)も含められます。
[^long]:
    複数段落の脚注。

    インデントを続けると 2 段落目になります。
```

表示:

本文中に脚注参照を置けます[1]。複数置くこともできます[2]。 同じ脚注を再利用することもできます[1]。長い注釈も書けます[3]。

Footnotes

1. これが 1 つ目の脚注です。 ↩ ↩[2]

2. 名前付きラベルの脚注。**強調**や `code`、リンクも含められます。 ↩

3. 複数段落の脚注。

   インデントを続けると 2 段落目になります。 ↩

アラート（GitHub alerts）

引用記法の先頭に `[!NOTE]` などのマーカーを置くと、注意書き（alert）になります。 種別は **NOTE / TIP / IMPORTANT / WARNING / CAUTION** の 5 つで、これは AsciiDoc の admonition と同じ構造・配色で表示されます（AsciiDoc の Admonition と対比できます）。

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

5 種類のアラート

マーカーは引用ブロックの先頭行に単独で置きます（同じ行に本文を続けると通常の引用になります）。

ソース:

```
> [!NOTE]
> 補足情報。読み手が知っておくと役立つ内容。

> [!TIP]
> ちょっとした助言・近道。

> [!IMPORTANT]
> 目的の達成に欠かせない重要事項。

> [!WARNING]
> 注意が必要な内容。見落とすと問題になる可能性。

> [!CAUTION]
> 望ましくない結果やリスクを伴う操作への警告。
```

表示:

> **Note**
> 補足情報。読み手が知っておくと役立つ内容。

> **Tip**
> ちょっとした助言・近道。

> **Important**
> 目的の達成に欠かせない重要事項。

> **Warning**
> 注意が必要な内容。見落とすと問題になる可能性。

> **Caution**
> 望ましくない結果やリスクを伴う操作への警告。

複数段落・他の記法を含む

マーカーの行のあとに、段落・リスト・コードなどをそのまま続けられます。

ソース:

```
> [!NOTE]
> 1 つ目の段落。
>
> 2 つ目の段落では **強調** や `コード`、リストも書けます。
>
> - 箇条書き 1
> - 箇条書き 2
```

表示:

> **Note**
> 1 つ目の段落。
>
> 2 つ目の段落では **強調** や `コード`、リストも書けます。
>
> - 箇条書き 1
> - 箇条書き 2

幅の広いコンテンツ

本文は読みやすい幅（既定で最大 860px）に保たれます。設定ファイルの `html.contentWidth` で固定幅を変えたり、`full` で横幅いっぱいまで広げたりできます。 読者に幅を切り替えさせない場合は `html.contentWidthToggle` を `false` にします。 読者が選択するまでワイド幅で開くには `html.contentWidthDefault` を `wide` にします。 それより幅の広い表・コードブロック・画像・図がどう表示されるかを確認するサンプルです。

> **Note**
> 既定テーマでは本文の幅は `html.contentWidth` で調整できます。本文幅に収まらない表やコードは、 **その要素の中で横スクロール**して全体を閲覧できます。画像や図は**本文幅に合わせて縮小表示** されます（ページ全体には横スクロールが出ません）。

幅の広い表

列数の多い表は本文幅に収まりません。表全体がスクロール領域になり、**表の上で横スクロール**して 残りの列を見られます。

指標	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
ページビュー	1200	1840	2010	1750	2230	2680	3120	2990	2640	2880	3310	3720
ユニーク	820	1190	1320	1180	1460	1710	1980	1920	1700	1840	2100	2350
直帰率 (%)	58	55	53	54	51	49	47	48	50	49	46	44

行の長いコードブロック

長い行を含むコードは、**コードブロックの中で横スクロール**します。短いコードはそのまま表示されます。

長い行:

```
docker run --rm -it --name monodocs-dev -v "$(pwd)":/work -w /work/app -e
NODE_ENV=development -p 4173:4173 monodocs-dev node packages/cli/dist/index.js serve
../examples/ja --host 0.0.0.0 --port 4173
```

短い行:

```
const width = 860;
```

大きな画像

intrinsic 幅が本文幅より大きい画像は、**本文幅に合わせて縮小表示**されます（`max-width: 100%`）。



Mermaid（横長の図）

横に長いフロー図も、本文幅に合わせて表示されます。



数式

`$...$` で文中に、 `$$...$$` または `math` のコードブロックで別行に数式を書けます。 数式はビルド時に MathML へ変換されるため、出力にスクリプトは要りません。

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

> **Note**
> 数式はブラウザが OpenType MATH フォントで描きます。PDF では、どのマシンでも monodocs が持っている New Computer Modern Math で組みます。HTML では、読者のブラウザが読者のフォントを使います。Windows なら Cambria Math、Debian / Ubuntu なら Latin Modern Math（`fonts-lmodern`）です。無いと括弧や 根号が伸びません。

文中の数式

ソース:

```
質量とエネルギーは $E = mc^2$ の関係にあり、ギリシャ文字 $\alpha + \beta = \gamma$ も文中に置けます。
```

表示:

質量とエネルギーは $E = mc^2$ の関係にあり、ギリシャ文字 $\alpha + \beta = \gamma$ も文中に置けます。

別行立ての数式

ソース:

````
$$
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
$$

```math
\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}
```
````

表示:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

行列と揃えた式

ソース:

````
```math
A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \det A = ad - bc
```


```math
\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
\nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial
\mathbf{E}}{\partial t}
\end{aligned}
```
````

表示:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \det A = ad - bc$$

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
\nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}
\end{aligned}$$

応用：3 階層目のサンプル

このフォルダ（`markdown/advanced/`）は **3 階層目** に置かれたページ群です。 入力を深くフォルダ分割しても、サイドバーはフォルダ構造をそのままツリーにします。

- 階層が深くなってもサイドバーで畳む／開くで辿れます（`sidebar.collapseDepth` で 「この階層より深いものは既定で畳む」設定にできます。**隠す**のではなく**畳む**だけなので、 どのページにも常に到達できます）。
- 同じ階層のページへもリンクできます: 深い見出しのページ

深いフォルダでもリンクは追従する

3 階層目から 1 階層目へ戻るリンクも、単一 HTML 内の hash route に変換されます。

- 1 つ上のフォルダ: Markdown サンプル
- ルート: 概要

深い見出しと目次

ページ内目次（右側）に出す見出しの深さは `toc.maxLevel` で調整できます。 既定では **h2〜h3** を目次に出します（h1 はページタイトル相当のため常に除外）。 `toc.maxLevel: 4` にすると h4 まで、 `toc.maxLevel: 2` にすると h2 だけが目次に並びます。

見出しは目次に出るかどうかに関わらず本文には常に表示されるため、目次の深さを浅くしても 情報が消えるわけではありません（目次が長くなりすぎるのを防ぐための設定です）。

レベル 2：概要

ここは h2。既定の目次（h2〜h3）に出ます。

レベル 3：詳細

ここは h3。既定の目次に出ます。

レベル 4：補足

ここは h4。既定では目次に出ませんが、 `toc.maxLevel: 4` で出るようになります。

レベル 5：細目

ここは h5。 `toc.maxLevel: 5` 以上で目次に出ます。

レベル 6：脚注的メモ

ここは h6。Markdown の見出しの最深レベルです。

レベル 2：まとめ

深い見出しを多用するページでは、目次の深さを浅く保つと一覧性が上がります。

AsciiDoc サンプル

このサンプルは AsciiDoc Syntax Quick Reference の主な記法を網羅し、`monodocs` での表示を確認するためのものです。

各項目は **ソース（生の AsciiDoc）** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

ビルド例:

```
monodocs build examples/ja -o dist/sample.html
```

各カテゴリは左サイドバーから開けます: リスト / ブロック / 表 / リンクと画像 / 幅の広いコンテンツ。

セクション見出し

`==` の数でレベルが決まります（`=` は文書タイトル）。

ソース:

```
=== レベル 2

==== レベル 3

===== レベル 4
```

表示:

レベル 2

レベル 3

レベル 4

段落と改行

空行で段落を区切り、行末の `+` で強制改行します。`[.lead]` ロールでリード段落になります。

ソース:

```
これは段落です。空行で段落を区切ります。

行末に `+` を置くと +
強制改行になります。

[.lead]
リード段落（`[.lead]` ロール）。
```

表示:

これは段落です。空行で段落を区切ります。

行末に `+` を置くと
強制改行になります。

リード段落（`[.lead]` ロール）。

テキスト書式

太字・イタリック・等幅・ハイライト・上付き/下付きなどの装飾です。

ソース:

```
* 太字: *bold* と **un**constrained
* イタリック: _italic_ と __un__constrained
* 等幅: `monospace`
* 太字イタリック等幅: `*_重ね掛け_*`
* ハイライト: #highlighted#
* 上付き/下付き: E = mc^2^ / H~2~O
* 引用符: "`曲がった二重引用符`" と '`単一引用符`'
```

表示:

- 太字: **bold** と **un**constrained
- イタリック: *italic* と *un*constrained
- 等幅: `monospace`
- 太字イタリック等幅: ***重ね掛け***
- ハイライト: highlighted

- 上付き/下付き: $E = mc^2$ / H_2O
- 引用符: “曲がった二重引用符” と ‘単一引用符’

置換（テキスト置換）

記号や矢印は自動で置換されます。

ソース:

```
(C) (R) (TM) は記号に、矢印 -> => <- <= はそのまま、`...` は省略記号になります。
```

表示:

© ® ™ は記号に、矢印 → ⇒ ← ⇐ はそのまま、`…` は省略記号になります。

注釈（コメント）

`//` 始まりの行はコメントで、出力されません。

ソース:

```
// これはコメントで、出力されません。

NOTE: `//` 始まりの行コメントは出力されません。
```

表示:

> **Note**
> `//` 始まりの行コメントは出力されません。

リスト

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

箇条書き（順序なし）

`*` の数でネストします。

ソース:

```
* 項目 A
* 項目 B
** ネスト B-1
** ネスト B-2
*** さらにネスト
* 項目 C
```

表示:

- 項目 A
- 項目 B
    - ネスト B-1
    - ネスト B-2
        - さらにネスト
- 項目 C

番号付き（順序あり）

`.` で番号付き、`..` でネストします。

ソース:

```
. 最初
. 次
. その次
.. ネスト 2-1
.. ネスト 2-2
```

表示:

1. 最初
2. 次
3. その次
    a. ネスト 2-1
    b. ネスト 2-2

チェックリスト

`[x]` / `[ ]` でチェックボックスになります。

ソース:

```
* [x] 完了したタスク
* [ ] 未完了のタスク
* [*] 完了（別記法）
```

表示:

- ☑ 完了したタスク
- ☐ 未完了のタスク
- ☑ 完了（別記法）

説明リスト（description list）

`用語

説明` で用語と説明の組を作ります。

ソース:

```
CPU:: 中央処理装置。
メモリ:: 主記憶。
ストレージ::
  補助記憶。複数行の説明も書けます。
```

表示:

CPU

中央処理装置。

メモリ

主記憶。

ストレージ

補助記憶。複数行の説明も書けます。

用語と入れ子

`::` / `:::` でレベルを変えて入れ子にできます。

ソース:

```
オペレーティングシステム::
  Linux:::
    オープンソースの OS。
  macOS:::
    Apple の OS。
```

表示:

オペレーティングシステム

Linux

オープンソースの OS。

macOS

Apple の OS。

リスト項目に複数ブロック

継続行（`+`）で同じ項目に段落やブロックをつなげられます。

ソース:

```
* 項目に複数のブロックを含む例。
+
これは継続行（`+`）でつないだ同じ項目の段落です。
+
[source,bash]
----
echo "リスト項目内のコードブロック"
----
```

表示:

- 項目に複数のブロックを含む例。

  これは継続行（`+`）でつないだ同じ項目の段落です。

  ```
  echo "リスト項目内のコードブロック"
  ```

ブロック

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

ソースブロック（構文ハイライト）

`[source,言語]` を付けた `----` ブロックは shiki でハイライトされます。

ソース:

```
[source,python]
----
def fib(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a + b
    return a
----
```

表示:

```python
def fib(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a + b
    return a
```

`// <1>` のようなコールアウトで注釈を付けられます。

ソース:

```
[source,js]
----
function greet(name) { // <1>
  return `Hello, ${name}!`; // <2>
}
----
<1> 関数を定義する。
<2> テンプレートリテラルで返す。
```

表示:

```
function greet(name) { // (1)
  return `Hello, ${name}!`; // (2)
}
```

1. 関数を定義する。
2. テンプレートリテラルで返す。

リスティングブロック / リテラルブロック

`----` は等幅でそのまま、`....` はリテラル（空白も保持）。

ソース:

```
----
リスティングブロック（等幅・そのまま表示）
----

....
リテラルブロック
  空白も保持されます。
....
```

表示:

```
リスティングブロック（等幅・そのまま表示）
```

```
リテラルブロック
  空白も保持されます。
```

例ブロック

`====` で本文や他のブロックをまとめます。

ソース:

```
====
例ブロック。本文や他のブロックをまとめます。
====
```

表示:

例ブロック。本文や他のブロックをまとめます。

サイドバーブロック

`**` は補足情報の囲みです。

ソース:

```
****
サイドバーブロック。補足情報の囲み。
****
```

表示:

サイドバーブロック。補足情報の囲み。

引用ブロック / 詩ブロック

`[quote]` は引用、`[verse]` は改行・字下げを保持する詩ブロックです。

ソース:

```
[quote, 山田太郎, 単一ファイル配布の心得]
____
ドキュメントは分割して書き、配布は 1 ファイルに。
____

[verse, 作者名]
____
改行や
  字下げが
保持される詩ブロック。
____
```

表示:

> ドキュメントは分割して書き、配布は 1 ファイルに。
>
> — 山田太郎
> *単一ファイル配布の心得*

```
改行や
  字下げが
保持される詩ブロック。
```

— 作者名

Admonition（注意書き）

`NOTE:` などのラベルで注意書きになります。

ソース:

```
NOTE: 単純な NOTE。

TIP: 役立つヒント。

IMPORTANT: 重要な注意。

WARNING: 警告。

CAUTION: 注意。
```

表示:

> **Note**
> 単純な NOTE。

> **Tip**
> 役立つヒント。

> **Important**
> 重要な注意。

> **Warning**
> 警告。

> **Caution**
> 注意。

`[NOTE]` + `====` のブロック形式なら複数段落や他のブロックを含められます。

ソース:

```
[NOTE]
====
ブロック形式の NOTE。複数段落や他のブロックを含められます。

* リストも
* 入れられます
====
```

表示:

> **Note**
> ブロック形式の NOTE。複数段落や他のブロックを含められます。
>
> - リストも
> - 入れられます

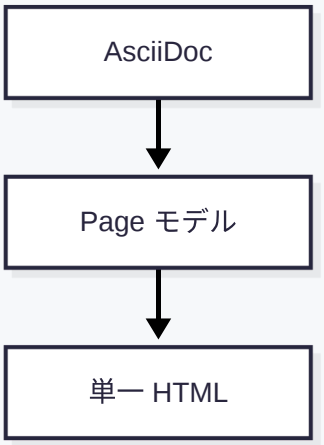
Mermaid（[source,mermaid]）

`[source,mermaid]` ブロックは図として描画されます。

ソース:

```
[source,mermaid]
----
graph TD
  A[AsciiDoc] --> B[Page モデル]
  B --> C[単一 HTML]
----
```

表示:



パススルー（生 HTML）

`++++` で囲むと生 HTML をそのまま埋め込みます（信頼できる入力前提）。

ソース:

```
++++
<div style="padding:0.5rem;border:1px dashed #888;border-radius:6px;">
  パススルーで埋め込んだ生 HTML（信頼できる入力前提）。
</div>
++++
```

表示:

パススルーで埋め込んだ生 HTML（信頼できる入力前提）。

表

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

基本の表

`cols` で列幅と寄せ（`^` 中央 / `>` 右寄せ）、`options="header"` で 1 行目をヘッダーに、`.タイトル` でキャプションを付けます。

ソース:

```
.対応状況
[cols="1,^1,>1", options="header"]
|===
| 機能 | Markdown | AsciiDoc

| 見出し
| ✅
| ✅

| 表
| ✅
| ✅

| コードハイライト
| shiki
| shiki
|===
```

表示:

Table 1. 対応状況

機能	Markdown	AsciiDoc
見出し	✅	✅
表	✅	✅
コードハイライト	shiki	shiki

セル内の書式と複数行

セル内で `太字` や `code` を使えます。`a|` プレフィックスでセルに AsciiDoc ブロック（リストなど）を入れられます。

ソース:

```
[cols="1,2", options="header"]
|===
| 項目 | 説明

| 強調
| セル内で *太字* や `code`、_イタリック_ が使えます。

| 箇条書き
a|
セルに AsciiDoc ブロック（リストなど）を入れるには `a` プレフィックスを使います:

* 項目 1
* 項目 2
|===
```

表示:

項目	説明
強調	セル内で **太字** や `code`、*イタリック* が使えます。
箇条書き	セルに AsciiDoc ブロック（リストなど）を入れるには `a` プレフィックスを使います: • 項目 1 • 項目 2

CSV 形式

`format=csv` でカンマ区切りのデータから表を作れます。

ソース:

```
[%header, format=csv]
|===
名前,役割
core,変換処理の中核
cli,コマンドライン
|===
```

表示:

名前	役割
core	変換処理の中核
cli	コマンドライン

リンクと画像

各項目は **ソース** を先に示し、その下に **表示（HTML 変換結果）** を並べます。

リンク

URL マクロ・メール・別ファイルへの相互参照（xref）です。

ソース:

```
* URL マクロ: {url-repo}[monodocs リポジトリ]
* そのまま: https://docs.asciidoctor.org/
* メール: mailto:docs@example.com[ドキュメント担当]
* 別ファイルへの相互参照（xref）: xref:index.adoc[トップに戻る] /
  xref:blocks.adoc[ブロックのページ]
```

表示:

- URL マクロ: monodocs リポジトリ
- そのまま: https://docs.asciidoctor.org/
- メール: ドキュメント担当
- 別ファイルへの相互参照（xref）: トップに戻る / ブロックのページ

属性（attribute）

ヘッダーで定義した属性 `https://example.com/monodocs` を本文で展開できます。

ソース:

```
ヘッダーで定義した属性 `{url-repo}` を本文で展開できます: {url-repo}
```

表示:

ヘッダーで定義した属性 `https://example.com/monodocs` を本文で展開できます: https://example.com/monodocs

内部アンカーと相互参照

`[#id]` で見出しにアンカーを付け、`テキスト` で参照します。クリックするとページを切り替えずに該当箇所へスクロールします。

ソース:

```
[#install-section]
=== インストール手順（アンカー付き）

この見出しには `[#install-section]` でアンカー（ID）を付けています。
別の場所から <<install-section,インストール手順へ>> のように参照できます。
```

表示:

インストール手順（アンカー付き）

この見出しには `[#install-section]` でアンカー（ID）を付けています。 別の場所から インストール手順へ のように参照できます。

画像

`image::` はブロック画像（キャプション付き）、`image:` はインライン画像です（画像は単一 HTML に data URI で埋め込まれます）。

ソース:

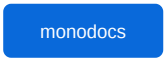
```
.monodocs ロゴ
image::images/logo.svg[ロゴ, 120]

インライン画像 image:images/logo.svg[ロゴ, 80] を文中に置くこともできます。
```

表示:



Figure 1. monodocs ロゴ

インライン画像 

を文中に置くこともできます。

UI マクロ

キー入力・ボタン・メニューを表すマクロです。

ソース:

```
* キー入力: kbd:[Ctrl+S] で保存、kbd:[Ctrl+Shift+P]
* ボタン: btn:[OK] をクリック
* メニュー: menu:File[New > Project] を選択
```

表示:

- キー入力: kbd:[Ctrl+S] で保存、kbd:[Ctrl+Shift+P]
- ボタン: btn:[OK] をクリック
- メニュー: menu:File[New > Project] を選択

脚注

`[1]` で脚注、`[2]` で再利用できる名前付き脚注になります。

ソース:

```
本文に脚注を付けられますfootnote:[これは脚注です。単一 HTML 内で一意な ID になります。]。
名前付き脚注footnote:ref1[再利用できる脚注。]も使え、同じものを再度参照できます
footnote:ref1[]。
```

表示:

本文に脚注を付けられます[3]。 名前付き脚注[2]も使え、同じものを再度参照できます[2]。

---

1. …
2. …
3. これは脚注です。単一 HTML 内で一意な ID になります。

幅の広いコンテンツ

本文は読みやすい幅（既定で最大 860px）に保たれます。設定ファイルの `html.contentWidth` で固定幅を変えたり、`full` で横幅いっぱいまで広げたりできます。読者に幅を切り替えさせない場合は `html.contentWidthToggle` を `false` にします。読者が選択するまでワイド幅で開くには `html.contentWidthDefault` を `wide` にします。それより幅の広い表・リスティング・画像が AsciiDoc でどう表示されるかを確認するサンプルです。本文幅に収まらない 要素はその要素の中で横スクロールし、画像は本文幅に合わせて縮小表示されます。

幅の広い表

列数の多い表は本文幅に収まらないため、表の中で横スクロールして残りの列を見られます。

指標	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
ページビュー	1200	1840	2010	1750	2230	2680	3120	2990	2640	2880	3310	3720
ユニーク	820	1190	1320	1180	1460	1710	1980	1920	1700	1840	2100	2350
直帰率 (%)	58	55	53	54	51	49	47	48	50	49	46	44

行の長いリスティング

長い行を含むリスティングブロックは、ブロックの中で横スクロールします。短いコードはそのまま表示されます。

```
docker run --rm -it --name monodocs-dev -v "$(pwd)":/work -w /work/app -e
NODE_ENV=development -p 4173:4173 monodocs-dev node packages/cli/dist/index.js serve
../examples/ja --host 0.0.0.0 --port 4173
```

```
const width = 860;
```

大きな画像

intrinsic 幅が本文幅より大きい画像は、本文幅に合わせて縮小表示されます。



Markdown / AsciiDoc 混在

同じフォルダ内に Markdown と AsciiDoc を混在させられます。形式をまたいだリンクも 単一 HTML 内の hash route に変換されます。

- このページ（Markdown）から AsciiDoc の相棒ページ へ。
- 見出しを指すリンクは、その見出しの位置へ直接ジャンプします: 相棒ページの「アンカーの例」。

アンカー先の見出し

相棒ページから、この見出しを指すリンクが張られています。

相棒（AsciiDoc）

同じフォルダの Markdown ページへ戻る: 混在トップ。

> **Note**
> AsciiDoc 側の admonition もそのまま使えます。

アンカーの例

このセクションには `[#anchor-demo]` で ID を付けてあり、Markdown 側から `companion.adoc#anchor-demo` で直接ジャンプできます。

逆向きも同じです: 混在トップの「アンカー先の見出し」。

PDF 出力

`monodocs` は、生成した **単一 HTML を経由して PDF** も出力できます。入力は分割管理したまま、 配布物を 1 ファイル（HTML でも PDF でも）にまとめられます。

```
monodocs build examples/ja --format pdf  -o dist/sample.pdf   # PDF のみ
monodocs build examples/ja --format both -o dist/sample       # HTML と PDF の両方
```

- `--format both` は `-o` を **ディレクトリ**として扱い、その中に `docs.html` と `docs.pdf` を出力します。
- 内部ではヘッドレス Chromium で単一 HTML を開き、印刷用レイアウト（全ページを縦に展開）で PDF 化します。**このページ自体も PDF に含められます。**

PDF に含まれるもの

- **しおり（アウトライン）**: HTML の左サイドバーと同じ **フォルダ → ページ** の階層。 ビューアのしおりパネルから各ページへジャンプできます。
- **本文中のリンク**: ページ間リンク・脚注・見出しアンカーは、PDF 内でもクリックして移動 できます。
- **画像**: data URI として埋め込むため、PDF 単体で完結します（外部ファイルを参照しません）。
- **コードハイライト / 図 / 表 / アラート**: HTML と同じ体裁で描画されます。
- **改ページの配慮**: アラートや図・コードブロックはできる限りページ途中で分割しません。

> **Note**
> ダーク/ライトのトグルは HTML 用の機能です。PDF は印刷用のライト配色で出力されます。

設定（`monodocs.config.yml`）

```yaml
pdf:
  pageSize: "A4" # A4 / Letter など
  margin:
    top: "20mm"
    right: "15mm"
    bottom: "20mm"
    left: "15mm"
  printBackground: true # 背景色・背景画像を印刷する
  bookmarks: true # しおり（サイドバー構造）を付与する
```

実行環境の要件

> **Important**
>
> PDF 出力はヘッドレス Chromium を使うため、実行環境に **Chromium** が必要です。また PDF は **実行環境のシステムフォント**で描画されるため、本文に出す文字種（日本語・絵文字など）の フォントが無いと豆腐（白い四角や×印の付いた四角）になります。開発用イメージには日本語（CJK）と絵文字フォントを 同梱しています。

> **Warning**
>
> バンドル版 CLI（単一 `.cjs` / 単一実行ファイル）では PDF 出力は利用できません（`puppeteer-core` を同梱しないため）。PDF を出す場合はパッケージインストール版を使ってください。

検索

生成された単一 HTML には全文検索が組み込まれています。 辞書も検索ランタイムも同梱しないため、この機能のためにファイルが大きくなることはほとんどありません。

サイドバーの検索欄に語を入れると、タイトル、見出し、本文を対象に候補が絞り込まれます。 以下に挙げるクエリは、すべてこのサンプルに実在する語に当たります。 実際に入力しながら読んでください。

表記のゆれの扱い

日本語の文書では、同じ語でも書き手によって表記が揺れます。 入力した語と本文の表記が一字でも違えば当たらない、という状態にならないよう、次の違いは同じものとして扱います。

検索欄に入れる語	当たる本文	吸収している違い
`はいらいと`	ハイライト	カタカナとひらがな
`ｲﾝｽﾄｰﾙ`	インストール	半角カタカナと全角カタカナ
`コ―ド`	コード	長音記号とダッシュ類
`h2～h3`	h2〜h3	全角チルダと波ダッシュ
`ＰＤＦ`	PDF	全角英数と半角英数

`コ―ド` の 2 文字目は長音記号ではなくダッシュです。 打ち間違えたまま入力しても、意図した語に当たります。

結果にはこのページ自身も出ます。 両方の表記をこの表に並べているためで、確かめたいのは他のページのほうです。 たとえば `はいらいと` で出てきたページを開くと、ひらがなで入力した語がカタカナの本文に印として付きます。

複数キーワードでの絞り込み

空白で区切ると、そのすべてを含むページだけが残ります。

まず `画像` と入力してください。 画像に触れているページが並びます。 続けて `埋め込み` を足すと、両方を含むページだけが残ります。

見出しに当たった結果

結果はタイトル、見出し、本文の順に重み付けされます。 見出しに当たった結果を開くと、ページの先頭ではなくその見出しまで移動します。

`インストール` と入力してください。 AsciiDoc サンプルの見出しに当たり、開くとその見出しの位置から表示されます。

キーボード操作

検索欄から手を離さずに結果を辿れます。

- `↓` と `↑`：結果を移動します。両端で折り返します。
- `Enter`：選んでいる結果を開きます。まだ動かしていなければ先頭の結果を開きます。
- `Escape`：入力を消します。

検索欄の右端の × ボタンでも入力を消せます。 結果と本文の印も消え、カーソルは検索欄に残ります。

移動しているあいだもカーソルは検索欄に残るため、そのまま入力を続けて絞り込めます。 日本語入力の変換中はこれらのキーを IME に譲るので、変換候補の選択が妨げられることはありません。

本文のハイライト

結果を開くと、開いた先の本文でも一致した語に印が付きます。 印は検索を開いているあいだ残るので、前後ページへ移動しても、どこに語があるかを見失いません。 入力を書き換えるか消すと、本文は元の状態に戻ります。

緩く当てる語

表記のゆれのうち、送り仮名と英語の語形変化は、本文ではなく入力した語のほうを緩めて照合します。 辞書は使いません。

- **送り仮名**：漢字と漢字の間の仮名は、2 字まで、あってもなくても当たります。`埋込み` は `埋め込み` に当たります。
- **英語の語形変化**：語の頭からの語幹で当てます。 `installing` は `install` としか書いていないページにも当たります。語幹が 4 字に満たない語は緩めません。

緩く当たったページは、そのまま一致したページより後ろに並びます。 本文の印は、入力した表記ではなく本文に書かれた表記に付きます。

Generated by monodocs v0.19.0