

Vivliostyle Viewer

Vivliostyle Viewer

Vivliostyle Viewer は、HTML+CSS 文書を組版・表示する Web アプリケーションです。

使い方

Vivliostyle Viewer をローカル環境で利用するには

Vivliostyle Viewer をローカル環境で利用するには、Vivliostyle CLI の preview コマンドを使うのが便利です。Vivliostyle CLI のインストール方法と使い方については、[Vivliostyle CLI ユーザーガイド](#)、preview コマンドについては[組版結果のプレビュー](#)をご覧ください。

それ以外の方法で Vivliostyle Viewer をローカル環境で利用する場合は、配布パッケージに同梱の README (<https://github.com/vivliostyle/vivliostyle.js/blob/master/packages/viewer/README.ja.md>) に記載の「使い方」の手順にしたがって Web サーバーを起動してください。

オンライン公開 Vivliostyle Viewer

オンラインで公開されている vivliostyle.org/viewer/ (<https://vivliostyle.org/viewer/>) も利用でき、これは常に最新のリリース版に更新されています。いち早く最新のバージョンを試したい場合は、vivliostyle.vercel.app (<https://vivliostyle.vercel.app>) を利用してください。

Vivliostyle Viewer を自分のサイトに配置

Vivliostyle Viewer の配布パッケージ (<https://vivliostyle.org/ja/download/> (<https://vivliostyle.org/ja/download/>) からダウンロード) を解凍したものを自分のサイトに配置すれば、自分のサイトで Vivliostyle Viewer で閲覧する出版物を公開することができます。

Node.js のアプリケーションで利用するには、npm パッケージ版の [@vivliostyle/viewer](https://www.npmjs.com/package/@vivliostyle/viewer) (<https://www.npmjs.com/package/@vivliostyle/viewer>) を使用してください。

Vivliostyle Viewer のスタートページ

パラメータを指定しないで [Vivliostyle Viewer](https://vivliostyle.org/viewer/) (<https://vivliostyle.org/viewer/>) を開くと、文書 URL の入力欄 (“Input a document URL”)、**Book Mode** と **Render All Pages** のチェックボックス、および使い方のヘルプが表示されます。

試してみる: <https://vivliostyle.org/viewer/> (<https://vivliostyle.org/viewer/>)

設定パネル

Vivliostyle Viewer の画面右上の設定ボタン



をクリック（またはキーボードの S を押下）すると、設定パネルが表示されます。

この設定パネルから各種パラメーターを設定し、“Apply” ボタンを押すことで、組版表示を変更することができます（通常は “Apply” の押下後パネルは閉じますが、設定ボタンを 2 度押しして開いた場合、押下後もパネルを開いたままにできます）。

設定パネルで設定した内容は、URL パラメータに反映されます。そうした設定付き URL を、ブックマークしたり公開することもできます。例：『Cascading Style Sheets Level 2 Revision 1 (CSS 2.1 Rec Specification』 (https://vivliostyle.org/viewer/#src=https://www.w3.org/TR/2011/REC-CSS2-20110607&style=data:/*%3Cview er%3E*/%0A@page%20%7B%20size:%20a5;%20margin:%2010%25;%20marks:%20crop%20cross;%20bleed:%203mm;%20%7D%0A:root%20%7B%20line-height:%201.8;%20%7D%0A/*%3C/viewer%3E*/%0A@page%20:first%20%7B%0A%20%20%20@top-left%20%7B%0A%20%20%20%20content:%20none;%0A%20%20%7D%0A%20%20@top-right%20%7B%0A%20%20%20%20content:%20none;%0A%20%20%7D%0A%20%20@bottom-center%20%7B%0A%20%20%20%20content:%20none;%0A%20%20%7D%0A%20%20@page%20:left%20%7B%0A%20%20%20font-size:%200.8rem;%0A%20%20@top-left%20%7B%0A%20%20%20%20content:%20env(pub-title);%0A%20%20%7D%0A%20%20@page%20:right%20%7B%0A%20%20%20font-size:%200.8rem;%0A%20%20@top-right%20%7B%0A%20%20%20%20content:%20env(doc-title);%0A%20%20%7D%0A%20%20@bottom-center%20%7B%0A%20%20%20%20content:%20counter(page);%0A%20%20%7D%0A%20%20%20%20spread=true) （設定パネルから “Crop marks” にチェックを入れるなどした例。“Edit CSS” をクリックすると、設定内容の詳細が確認できます）。

- 凡例：●ラジオボタン、■チェックボックス、▶折りたたまれた設定項目

• ►Page View Mode

◦ ①Page Spread View : ページ表示

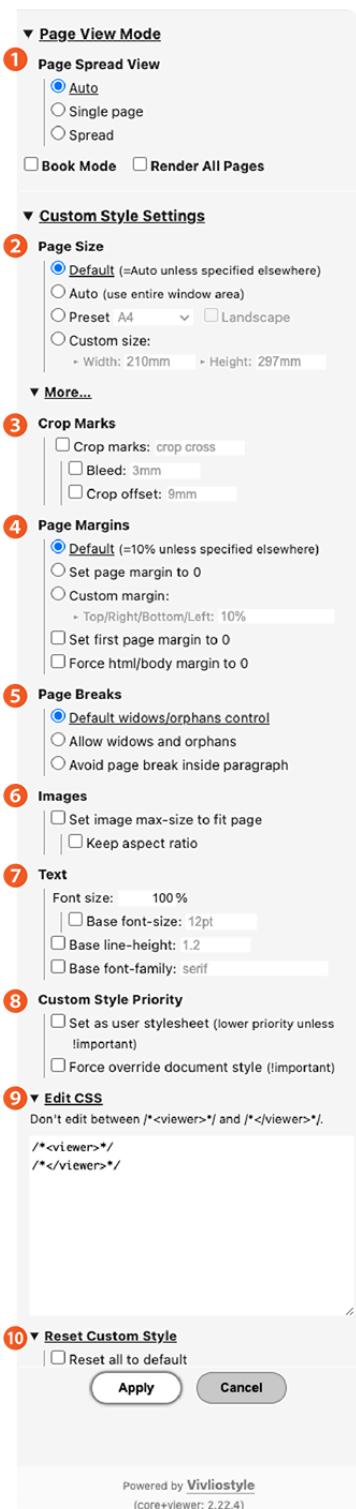
- ◎Auto : ウィンドウが見開きで表示可能な縦横比であれば、自動的に見開きにします

- ◎Single page : 常に単一ページで表示します

- ◎Spread : 常に見開きページで表示します

◦ ■Book Mode : チェックを入れるとブックモードになります

- ■Render All Pages : チェックを入れると、常に全ページをレンダリングします。正確なページ番号が表示される反面、表示まで時間がかかります。詳細は「全ページをレンダリング (Render All Pages) On/Off 設定」 (https://vivliostyle.github.io/vivliostyle_doc/ja/vivliostyle-user-guide-vol1/shinyu/index.html#%E5%85%A8%E3%83%9A%E3%83%BC%E3%82%B8%E3%82%92%E3%83%AC%E3%83%B3%E3%83%80%E3%83%AA%E3%83%B3%E3%82%80%EF%BC%88render-all-pages%EF%BC%89onoff%E8%A8%AD%E5%AE%9A) を参照



• ►Custom Style Settings

◦ ②Page Size (ページサイズ)

- ◎Default : 文書でページサイズ指定されていればそのサイズが、指定されていない場合はその時開かれているウィンドウのサイズが適用されます
- ◎Auto : ページサイズ指定の有無にかかわらず、その時開かれているウィンドウのサイズが適用されます
- ◎Preset : 次の値の中から選択できます。letter、legal、Ledger、A10～A0、B10 (ISO)～B0 (ISO)、JIS-B10～JIS-B0、C10～C0 (なお、日本で使われるB判は「JIS-B」であることにご注意ください)
- ■Landscape : Presetで選択したサイズの横置きにします
- ◎Custom size : 任意の幅と高さを指定できます。単位は CSS の単位 (mm, cm, in, pt, px など) が使えます

• ►More...

- ③■Crop Marks : トンボを入れる場合はここにチェックを入れます。Bleedにも自動的にチェックが入ります。コーナートンボだけを入れたい場合は crop 、センタートンボだけを入れたい場合は cross と記述します。デフォルトは crop cross 、つまり両方入ります

- ■Bleed:コーナートンボにおける裁ち切り線（外側の線）と仕上がり線（内側の線）の間隔を単位付きの数値で指定します（通常は 3mm）
- ■Crop offset:仕上がり線（内側の線）から用紙端までの余白サイズを指定したい場合は単位付きの数値で指定します。単位はCSSの単位（mm, cm, in, pt, px など）が使えます
- ④**Page Margins** : ページ余白
 - ●Default (=10% unless specified elsewhere) : 文書でページ余白が指定されていればそれに従い、指定されていない場合は上下左右の余白をページサイズ（高さ／幅）の10%ずつにします
 - ●Set page margin to 0 : ページ余白を0にします。固定レイアウトのEPUBなど、画面いっぱいに表示するのに適したコンテンツを閲覧するのに向いています
 - ●Custom margin : 任意のページ余白を指定できます。上右左下の順でそれぞれを半角スペースで区切って記述します。また、 20mm 15mm 10mm などと3つの値を記述すると、上の余白が 20mm 、右と左の余白がともに 15mm 、下の余白が 10mm になります。
 - ■Set first page margin to 0 : 最初のページ余白を0にします。最初のページがカバー画像で余白は不要の場合は、ここにチェックを入れます
 - ■Force html/body margin to 0 : html/body マージンを0にします。もともとHTML文書で指定されているスタイルシートで body 要素にマージンが設定されている場合、Vivliostyleで設定されたページ余白が加わると余白が異常に大きくなってしまいます。そのような場合ここにチェックを入れると回避できます
- ⑤**Page Breaks** : 改ページ
 - ●Default widows/orphans control : デフォルトとして、段落を分割する際にページの先頭／末尾に2行以上が残るように制御します
 - ●Allow widows and orphans : ページの先頭／末尾に1行だけ孤立する状態を許容します
 - ●Avoid page break inside paragraph : 段落の途中で改ページしてほしくない場合に指定します。例えば音声読み上げでテキストを読む場合に、段落の途中でページが変わるとそこで読み上げが中断してしまう問題を解決します
- ⑥**Images** : 画像
 - ■Set image max-size to fit page : 画像の最大サイズをページに収まるよう調整します
 - ■Keep aspect ratio : 画像サイズを変更する場合、画像の縦横比が変わらないよう調整して変更します
- ⑦**Text** : テキスト
 - Font size : 文字サイズをパーセント値で設定します。100%はブラウザのデフォルトと同様、16px=12ptです。この設定はツールバーにあるA- (Text: Smaller)、A+ (Text: Larger) ボタンで変更できる文字サイズと連動します
 - ■Base font-size : ルート要素での font-size プロパティの値を設定します。実際に表示される文字サイズは **Font size** のパーセント値と掛け合わされたものになります

- Base line-height : ルート要素での `line-height` プロパティの値を設定します。単位をつけない数値で文字サイズを基準にした行の高さを指定します。ここで指定しない場合、ブラウザと同じ `line-height: normal` が設定され、使われるフォントに依存します（おおむね1.2）
- Base font-family : ルート要素における `font-family` プロパティの値を設定します
- ⑧Custom Style Priority
 - Set as user stylesheet (lower priority unless !important) : 設定パネルでの設定内容を、ユーザースタイルシートとして設定します。デフォルトは作成者スタイルシートです。これらの違いは [CSS カスケード入門](https://developer.mozilla.org/ja/docs/Web/CSS/Cascade#E3%82%AA%E3%83%AA%E3%82%B8%E3%83%B3%E3%81%AE%E7%A8%AE%E9%A1%9E) をご参照ください
 - Force override document style (!important) : 設定パネルでの設定内容により、スタイルシートを強制的に上書きします（スタイルシートに `!important` が付く）
- ⑨►Edit CSS : クリックすると入力フィールドがあらわれ、それまでの設定した内容がCSSコードで確認できます。また、ここからユーザーがCSSコードを直接書くことができます。その場合、自動生成されたコメント `/*<viewer>*/` と `/*</viewer>*/` で囲まれた部分はそのままにして、その前か後にコードを追加するようにしてください
- ⑩►Reset Custom Style
 - Reset all to default : それまでの設定内容を初期化し、デフォルトに戻します

サポートされている文書の種類

- HTML文書+ページメディア用のCSS
- 本のような出版物（目次付き）（Book Mode: オン）
 - Web出版物（複数のHTML文書からなるコレクション）：最初のHTMLまたはマニフェストファイルのURLを指定します。
 - ZIP解凍済みのEPUB: OPFファイルのURLまたは解凍されたEPUBファイルのトップディレクトリを指定します。

メモ

- GitHubとGistのURLを直接指定することができます。そのようなURLが指定された場合、Vivliostyleはgithub/gistのrawコンテンツをロードします。
- ⚠ Mixedコンテンツ（“http:”のURLが“https:”のVivliostyle Viewerに指定された場合）は通常ブラウザによってブロックされます。
- ⚠ Cross-Origin（異なるドメインへのリクエスト）は、サーバーがCross-Origin Resource Sharing (CORS)を許可するように設定されていない限り、通常はブラウザによってブロックされます。

URLパラメータのオプション

- `#src=<document URL>`
- `&bookMode=[true | false]` (Book Mode)
 - **true** (デフォルト): 本のような出版物 (目次付き) 用
 - HTML文書の URL が指定された場合、その出版物マニフェストまたは目次 (`<nav role="doc-toc">` などでマークアップ) からリンクされた一連の HTML 文書が自動的に読み込まれます。
 - **false**: 単体の HTML 文書用
- `&renderAllPages=[true | false]` (Render All Pages)
 - **true** (デフォルト): 印刷用 (すべてのページが印刷可能で、ページ番号は期待されるとおりに機能します)
 - **false**: 閲覧用 (おおまかなページ番号を使って、クイックロード)
- `&spread=[true | false | auto]` (Page Spread View)
 - **true**: 見開き表示
 - **false**: 単一ページ表示
 - **auto** (デフォルト): 自動見開き表示
- `&style=<追加の (カスタム) スタイルシートの URL>`
- `&userStyle=<ユーザー・スタイルシートの URL>`

オプションは設定パネル (Settings) でも設定できます。

NOTE: Book Mode のデフォルトは以前はオフでしたが、Vivliostyle Viewer v2.18.0 からオンに変更されました。デフォルトで有効なのでパラメータ `&bookMode=true` の指定は省略することができます。HTML ファイル内の目次のリンク先の HTML 文書を連結しないで最初の HTML ファイルだけを表示するには、`&bookMode=false` を指定してください。

表示する HTML 文書の指定

HTML ファイルを Vivliostyle Viewer で表示するには、Web サーバーが読み込む場所 (上記手順にしたがって Web サーバーを起動している場合は、配布パッケージを解凍してできたフォルダ内) にそのファイル (およびそのファイルから読み込まれる CSS や画像ファイル) を置いた上で、次のようにパラメータを附加した URL をブラウザで開きます:

(Vivliostyle ViewerのURL)`#src=(表示するファイルのURL (Vivliostyle Viewerからの相対))`

注: Vivliostyle Viewer 本体とは別ドメインの文書を読み込もうとする場合、文書がある Web サーバーの設定によって、文書が読み込めない場合があります。文書を読み込ませるためには、サーバー側で CORS (Cross-Origin Resource Sharing) の設定が必要です。

注: 数式の表示 (MathML および TeX 形式に対応) にインターネット上の JavaScript ライブラリー (MathJax (<https://www.mathjax.org>)) を使用するため、文書に数式が含まれる場合はインターネット接続が必要です。

例: HTML ファイル https://vivliostyle.github.io/vivliostyle_doc/samples/gon/index.html (https://vivliostyle.github.io/vivliostyle_doc/samples/gon/index.html) を表示する場合:

https://vivliostyle.org/viewer/#src=https://vivliostyle.github.io/vivliostyle_doc/samples/gon/index.html (https://vivliostyle.org/viewer/#src=https://vivliostyle.github.io/vivliostyle_doc/samples/gon/index.html)

EPUB

Vivliostyle Viewer では ZIP 解凍済みの EPUB ファイルを表示することができます。この場合、次のようにパラメータを指定します:

```
#src=(表示する解凍済みEPUBフォルダのURL)&bookMode=true
```

GitHub 上に公開されている ZIP 解凍済みの EPUB ファイルを表示する例:

- IDPF/epub3-samples (<https://github.com/IDPF/epub3-samples/>) の 『Accessible EPUB 3 (https://github.com/IDPF/epub3-samples/tree/master/30/accessible_epub_3/)』

https://vivliostyle.org/viewer/#src=https://github.com/IDPF/epub3-samples/tree/master/30/accessible_epub_3/&bookMode=true (https://vivliostyle.org/viewer/#src=https://github.com/IDPF/epub3-samples/tree/master/30/accessible_epub_3/&bookMode=true)

Web 出版物 (複数 HTML 文書)

Vivliostyle Viewer では Web 出版物 (複数の HTML 文書からなるコレクション) を表示することができます。この場合、次のようにパラメータを指定します:

```
#src=(最初のHTML文書またはマニフェストファイルのURL)&bookMode=true
```

Web出版物のマニフェストの形式については、W3C ドラフトの Publication Manifest (<https://www.w3.org/TR/pub-manifest/>) および Radium Web Publication Manifest (<https://github.com/readium/webpub-manifest/>) をサポートしています。

HTMLの目次

Web出版物のマニフェストが存在しなくても、指定したHTML文書内の目次要素内に他のHTML文書へのリンクがある場合は、それらの文書が自動的にロードされます。VivliostyleはHTML文書内の次のCSSセレクタにマッチする要素を目次要素として扱います: `[role=doc-toc], [role=directory], nav li, .toc, #toc`

Web上に公開されている複数のHTML文書からなる出版物を表示する例:

- Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification (<https://www.w3.org/TR/CSS2/>)

<https://vivliostyle.org/viewer/#src=https://www.w3.org/TR/CSS2/&bookMode=true> (<https://vivliostyle.org/viewer/#src=https://www.w3.org/TR/CSS2/&bookMode=true>)

目次パネル

`&bookMode=true` (あるいは **Book Mode** チェックボックス On) を指定すると、単一のHTMLファイルでもWeb出版物と同様に、HTML内に目次要素があれば目次パネル (画面左上の目次アイコン  をクリックして開く) が有効になります。

見開きビューモード

Vivliostyle Viewer は、表示領域の横幅が大きいとき (高さの1.45倍以上)、自動的に見開きページ表示にします。これを変えるには次の指定をURLに追加します:

`&spread=true` (常に見開きページ表示)

`&spread=false` (常に単ページ表示)

`&spread=auto` (自動切り替え=デフォルト)

Vivliostyle Viewer の設定パネル (画面右上のアイコン  をクリックして開く) でも、ページ表示モードの変更ができます。

スタイルシートの追加（カスタム・スタイルシート）

HTMLファイルに指定されているスタイルシートに加えて、追加のスタイルシート（CSSファイル）を指定することができます。これにより文書のスタイルをカスタマイズするためのカスタム・スタイルシートを指定することができます。追加のスタイルシート（カスタム・スタイルシート）を使うには、次の指定をURLに追加します:

&style=(スタイルシートのURL (Vivlio style Viewer からの相対))

この方法で指定したスタイルシートは、HTMLファイルで指定されているスタイルシートと同様（制作者スタイルシート）の扱いで、よりあとに指定されたことになるので、CSSのカスケーディング規則により、HTMLファイルからのスタイルの指定を上書きすることになります。

ユーザー・スタイルシート

これに対して、次のようにしてユーザー・スタイルシート（スタイル指定に `!important` を付けないかぎり、制作者スタイルシートのスタイル指定を上書きしない）の指定もできます：

&userStyle=(ユーザー・スタイルシートのURL (Vivliostyle Viewer からの相対))

複数個の `&style=` あるいは `&userStyle=` を使うことで、複数個のスタイルシートを指定できます。

データ URL も利用できます。例えば:

&style=data:,html{writing-mode:vertical-rl}

設定パネルでのカスタム・スタイル設定

カスタム・スタイルの設定は、設定パネルの **Custom Style Settings** からも行うことができます。設定内容の CSS は設定パネルの **Edit CSS** で確認することができます（CSS コードをそこで追加することもできます）。

Web上に公開されている文書に、設定パネルからカスタム・スタイルの設定を加えた例:

- Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification (<https://www.w3.org/TR/CSS2/>)

```
0%20@bottom-center%20%7B%0A%20%20%20%20content:%20none;%0A%20%20%7D%0A%7D%0A%0A@page%20:left%  
20%7B%0A%20%20font-size:%200.8rem;%0A%20%20@top-left%20%7B%0A%20%20%20%20content:%20env(pub-title);%0  
A%20%20%7D%0A%20%20@bottom-center%20%7B%0A%20%20%20%20content:%20counter(page);%0A%20%20%7D%0  
A%20%20%7D%0A%20%20@page%20:right%20%7B%0A%20%20font-size:%200.8rem;%0A%20%20@top-right%20%7B%0A%20%20%2  
0%20content:%20env(doc-title);%0A%20%20%7D%0A%20%20@bottom-center%20%7B%0A%20%20%20%20content:%20c  
ounter(page);%0A%20%20%7D%0A%7D)
```

```
/*<viewer>*/  
@page {  
    size: A4;  
}  
/*</viewer>*/  
  
@page :first {  
    @top-left {  
        content: none;  
    }  
    @top-right {  
        content: none;  
    }  
    @bottom-center {  
        content: none;  
    }  
}  
  
@page :left {  
    font-size: 0.8rem;  
    @top-left {  
        content: env(pub-title);  
    }  
    @bottom-center {  
        content: counter(page);  
    }  
}  
  
@page :right {  
    font-size: 0.8rem;  
    @top-right {  
        content: env(doc-title);  
    }  
    @bottom-center {  
        content: counter(page);  
    }  
}
```

この例のように、設定パネルからスタイルの設定を行うと、カスタム・スタイルのCSS内のコメント `/*<viewer>*/` と `/*</viewer>*/` で囲んだ部分に、設定パネルの項目から設定したスタイルのCSSコードが

生成され、設定パネルの **Edit CSS** に表示されます。ユーザー独自の CSS コードをそれに追加することができます。この例ではページヘッダーとページフッターを追加しています。

印刷／PDF として出力

ブラウザの印刷・PDF保存機能を利用して、レンダリングされたドキュメントを PDF に変換できます。

例えば、Google Chrome で PDF に出力するには、メニューから「印刷」を開き、出力先として「PDFに保存」、背景のグラフィック ON と指定して「保存」します。

注: 文書の全ページを印刷する場合、設定パネルで **Render All Pages** が On になっていることを確認してください。これが Off の場合、すでに表示されたページしか印刷できず、またページ番号が正しく出力されません。

Vivliostyle CLI

Vivliostyle CLI (<https://github.com/vivliostyle/vivliostyle-cli>) を使うとコマンドラインから PDF に出力できます。印刷に適した PDF/X-1a 形式への変換も可能です。

サポートされている CSS の機能

Vivliostyle がサポートする CSS の機能については [サポートする CSS 機能](#) を参照してください。