

Markdown でスライドを書いて Vivliostyle で組んでプレゼン

yamahige

v3 2025-08-02

v2 2025-04-03

v1 2025-03-17

1. はじめに

研究会発表のスライドを想定して、HTML+CSSでスライドを作る場合の嬉しいところをあげてみます。

- 学会の研究会での発表ならば、元になる論文／予稿があって、主張をテキストで表現できています
- ヘッダーやフッターに挿入する項目（「日付」、「研究会名」、など）や書式（「スライド番号/総スライド数」、など）が研究室などで指導されていたりします

2. テキストが図を回り込んでくれる

HTML+CSS では、テキストが図を回り込むのが普通で、特に工夫はいりません。

- なお、改行の位置で「あれ？」と思ったみなさん、自動的な改行の位置を調整できるのです。[別のスライド](#)で説明します。
- 色は匂へど散りぬるを我が世誰ぞ常ならむ有為の
奥山今日越えて浅き夢見し酔ひもせず。色は
匂へど散りぬるを我が世誰ぞ常ならむ有為の
奥山今日越えて浅き夢見し酔ひもせず
- 色は匂へど散りぬるを我が世誰ぞ常ならむ有為の
奥山今日越えて浅き夢見し酔ひもせず。色は
匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず
- 色は匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず。色は
匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず



3. 改行位置の調整

CSS では自動的に折り返される改行の位置を、`word-break` プロパティ^[1]を使って調整できます。

`word-break: normal;` という設定では、既定の規則で改行します:

- HTML+CSS では、テキストが図を回り込むのが普通です。
- 色は匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず。



`word-break: auto-phrase;` という設定では、日本語として、より自然な位置で改行します。これは好みで分かれるでしょう:

- HTML+CSS では、テキストが図を回り込むのが普通です。
- 色は匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず。



なお、このスライド全体には、試しに `word-break: auto-phrase;` と設定しています。

4. 文字数が均等になるように改行してくれる

- このスライドのタイトルは1行に収まらないくらい長いので自動的に改行しています。自然なところで改行してるように見えますし、ほぼ同じ長さの2行になっています。
- ただし、Markdownを見るとタイトルには改行が入っていません。

Markdownでスライドを書いてVivliostyleで組んでプレゼン



**Markdownでスライドを書いて
Vivliostyleで組んでプレゼン**

CSSで次の設定を使うと、これを実現できます。

- `word-break: auto-phrase;`で、ことばとして自然なところで改行します。
- `text-wrap: balance;`^[2]で、行の文字数が均等になるように改行します。

5. 約物の前後の空白の詰め

"「"や"（"といった約物が行頭・行末にきたり連続したりする場合の空白の詰めを、`text-spacing-trim` プロパティ^[3]で制御できます。

`text-spacing-trim: normal;`で、行頭の約物は詰めませんが、約物が連続するときに詰めます。これが既定値です。

- ・「色は匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず。」
- ・ヘッダーやフッターに挿入する項目（「日付」、「研究会名」、など）が研究室などで指導されていたりします

`text-spacing-trim: trim-both;`で、行頭行末や連続する約物の空白を詰めます。

- ・「色は匂へど散りぬるを我が世誰ぞ常ならむ有為の奥山今日越えて浅き夢見し酔ひもせず。」
- ・ヘッダーやフッターに挿入する項目（「日付」、「研究会名」、など）が研究室などで指導されていたりします

スライドはテキストが短く箇条書きも多いので、テキストのまとまりは行頭の揃えで見せるのがよいかもしれません。

このスライド全体には`text-spacing-trim: trim-both;`と設定されています

6. ヘッダーとフッター

スライド本文のテキストを抜き出して、ヘッダーやフッターに表示できます

- 発表のタイトル（h1要素）やセクションの見出し（##見出し、つまり h2 要素など）といった既定のタグの付いたテキスト
- 日付、発表者、研究会名など既定のタグが付いていないテキスト

総スライド数を数えてくれて、その値を自動生成するテキストに含められます

6.1 ヘッダー／フッターはCSSのマージン・ボックス

ヘッダーやフッターを表示するには、CSSのマージン・ボックスを利用します。マージン・ボックスは、@top-leftや@bottom-right-cornerなど、側面とコーナーの合計16個の場所が定義されています^{*1 *2 *3}。

top-left-corner	top-left	top-center	top-right	top-right-corner
left-top				right-top
left-middle				right-middle
left-bottom				right-bottom
bottom-left-corner	bottom-left	bottom-center	bottom-right	bottom-right-corner

*1 @page <https://developer.mozilla.org/ja/docs/Web/CSS/@page>

*2 CSS Paged Media Module Level 3 - 5. Page-Margin Boxes <https://www.w3.org/TR/css-page-3/#margin-boxes>

*3 CSS - @page - とほほのWWW入門 <https://www.tohoho-web.com/css/rule/page.htm>

6.2 スライド本文のテキストを抜き出して表示

スライド本文から所定のテキストを自動的に抜き出してヘッダーやフッターに表示できると、本文を修正したときの修正漏れを防げます。

これには、CSS の名前付き文字列(named string^{*4})という仕組みを使います。やり方は大まかに次の手順です：

1. 抜き出したいテキストに印を付ける。「印が付く」 = 「セレクトターで選べる」です
2. その印を手がかりとしてテキストに名前を付ける
3. その名前を使って、表示したい場所にテキストを生成する

次の順に説明します：

1. 発表のタイトルやセクションの見出し
2. 発表のタイトル、日付、発表者、研究会名など

^{*4} 1.1. Named strings - CSS Generated Content for Paged Media Module <https://www.w3.org/TR/css-gcpm-3/#named-strings>

6.3 発表のタイトルやセクションの見出し

発表タイトルには `h1`、セクションの見出しには `h2` という HTML 既定の印(タグ)を付けますね。
`h2` 見出しを、自動生成した番号付きで各スライドの `@top-right` マージンに表示するとします。

それには CSS の `string-set` プロパティを使って、生成した番号に、例えば `chapter-number` という名前を、テキストに `chapter` という名前を付けます。

```
h2 {  
  string-set: chapter-number content(before), chapter content();  
}
```

`content(before)` は `h2` の `::before` 疑似要素の内容を示します。 `content()` は `content(text)` という意味で、 `h2` のテキストを示します。

そして、 `@top-right` マージンの `content` プロパティの値で、 `string` 関数の中でこれらの名前を使ってテキストを参照します。 `first` によって、そのスライド中の最初の見出しを使います。

```
@page {  
  @top-right { content: string(chapter-number, first) " " string(chapter, first); }  
}
```

6.4 テキストの溢れを省略記号...で示す

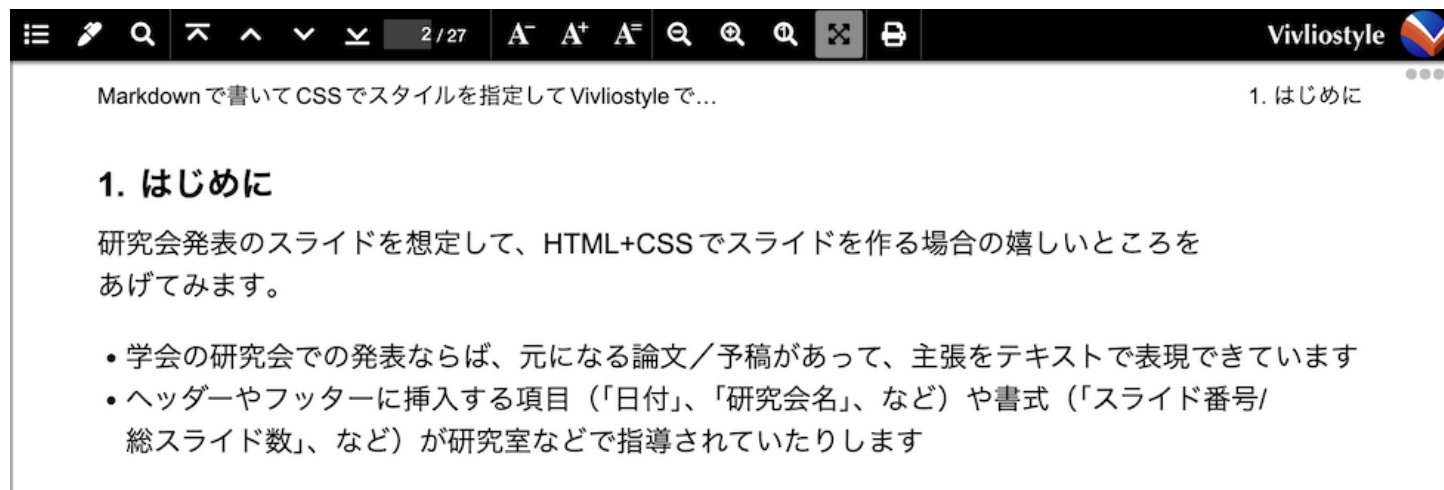
テキストが溢れて非表示になったとき、溢れて非表示になったテキストがあることを、`text-overflow: ellipsis;`と設定することで、省略記号...(U+2026 HORIZONTAL ELLIPSIS)で自動的に示せます。この機能をページ・マージンでも使えます(Vivliostyle.js v2.34.0、Vivliostyle CLI v9.5.0から使えるようになりました)。

次のように、`@top-left` マージンに表示するタイトルが長い場合...

```
# Markdownで書いてCSSでスタイルを指定してVivliostyleでスライドを組んでプレゼン #
```

溢れたテキストは折り返さず切り取る設定にしたうえで、「溢れて切り取られた場合は省略記号を表示する」設定にすると...

```
@top-left {  
  content: string(string-title);  
  max-width: 50%;  
  overflow: hidden; /* 溢れたテキストを切り取る */  
  white-space: nowrap; /* 折り返さない */  
  text-overflow: ellipsis; /* 溢れて切り取られたら省略記号を表示する */  
}
```



Markdown で書いて CSS でスタイルを指定して Vivliostyle で...

1. はじめに

研究会発表のスライドを想定して、HTML+CSSでスライドを作る場合の嬉しいところをあげてみます。

- 学会の研究会での発表ならば、元になる論文／予稿があって、主張をテキストで表現できています
- ヘッダーやフッターに挿入する項目（「日付」、「研究会名」、など）や書式（「スライド番号/総スライド数」、など）が研究室などで指導されていたりします

長いタイトルが「Vivliostyle で」で切れて、その後に...が表示されます:

Zoomや発表会場に途中から入った参加者にとって、今どの発表の最中なのかが分かるとありがたいですね。その場合、他の発表と区別できればよいので、このように発表タイトルが途中で切り取られていても大丈夫です。

6.5 日付、発表者、研究会名など - その1

「研究会名」といったHTML既定の印（タグ）はないので、印の工夫から始めます。

Vivliostyle用のMarkdownとして開発されている、VFM (Vivliostyle Flavored Markdown)^{*5}は、Markdownの見出しに応じてsection要素を生成して階層化してくれます^{*6}。これを利用します。

印を付ける

VFMで「研究会名」という見出しにconferenceクラスを設定すると、次のようなHTMLが生成されます:

```
### 研究会名{.conference}
```

第3回 プレゼン研究発表会



```
<section class="level3" aria-labelledby="研究会名">
  <h3 class="conference" id="研究会名">研究会名</h3>
  <p>第3回 プレゼン研究発表会</p>
</section>
```

"第3回 プレゼン研究発表会" に印が付きしました。:has(> .conference) > pというセレクトーで取り出せます。

*5 [Vivliostyle に特化した Markdown - VFM の使い方](#)

*6 [セクション分け - Sectionization](#)

名前を付ける

ここで次のようなCSSを適用すると、"第3回 プレゼン研究発表会"にstring-conferenceという名前が付きます。

```
h2.conference {  
    display: none;  
}  
:has(> .conference) > p {  
    string-set: string-conference content();  
}
```

フッターに生成する

@bottom-center マージンに研究会名名を表示します。

```
@page {  
    @bottom-center { content: string(string-conference); }  
}
```

6.6 日付、発表者、研究会名など - その2

「研究会名」を識別する印（タグ）を前提としないような、印の工夫から始めます。

印を付ける

VFMで次のように書いて、@bottom-center マージンに表示したい項目にbottom-centerクラスを設定すると、次のようなHTMLが生成されます（aria-labelledby属性などを省略してます）：

```
### 研究会名{.conference .bottom-center}
```

第3回 プレゼン研究発表会



```
<section>
  <h3 class="bottom-center">研究会名</h3>
  <p>第3回 プレゼン研究発表会</p>
</section>
```

"第3回 プレゼン研究発表会" に印が付きました。:has(> .bottom-center) > p というセレクターで取り出せます。

名前を付ける

ここで次のようなCSSを適用すると、"第3回 プレゼン研究発表会"にstring-bottom-centerという名前が付きます。

```
:has(> .bottom-center) > p:first-of-type {  
  string-set: bottom-center content();  
}
```

フッターに生成する

@bottom-center マージンに表示します。

```
@page {  
  @bottom-center { content: string(string-bottom-center); }  
}
```


6.7 日付、発表者、研究会名など - 検討

印と見出し

印を付けるために見出し##や###に`.conference`クラスを設定して利用しました。

- 印に見出しを利用すると、Markdownエディターのアウトライン表示に印が表示されます
- このことを覚えておいて、後で説明する[スライド区切りの設定](#)などに**必ず反映**します

Markdownエディターでのアウトライン表示にこだわらなければ、その1方式のMarkdownで次のように書いて印を付けられます。

```
これは<span class="conference">プレゼン学会 第4回研究発表会</span>で発表したものです。
```

方式の比較

その1方式とその2方式、どちらがよいかは意見が分かれるところでしょう。

その1方式

- Markdownを見ただけでは、フッター中央に何が表示されるか分かりません。何を表示するかはCSS側で決めます。
- CSSには`.conference`(を直下に持つ`section`の最初の`p`)をフッター中央に表示すると書いてあります。`.conference`はMarkdown（というかHTML）側が決めた印（クラス）ですが、CSS側はこれを前提にしています。
- このCSSをたまたま見つけてスタイルを気に入った発表者が、発表者名をフッター中央に表示したいと思ったら、**発表者名**に`.conference`クラスを設定するでしょう。

その2方式

- CSSを見ただけでは、フッター中央に何が表示されるか分かりません。何を表示するかはMarkdown側で`.bottom-center`クラスを指定して決めます。
- これはMarkdown側にスタイル情報を含めることを意味します。

6.8 スライド番号 / 総スライド数

各スライドに番号（ページ番号）があると、Q&Aタイムで各スライドにランダムアクセスしやすいです。また、総スライド数が表示されていると、発表者本人だけでなく座長や聴いてる人たちも安心ですね。

スライド番号（ページ番号）や総スライド数（総ページ数）は、それぞれ`page`と`pages`カウンターに設定されています。

そこで、CSSに次のように書くだけで、右下マージンに「スライド番号 / 総スライド数」が表示されます。

```
@page {  
  @bottom-right {  
    content: counter(page) " / " counter(pages);  
  }  
}
```

7. 参考文献を脚注や文末脚注として表示できる

参考文献を脚注で、各スライドの下部に表示することがあります。該当箇所に脚注参照(footnote call)を挿入して、脚注本体をスライド下部に表示する（脚注）か、最後のスライドにまとめて表示（文末脚注）します。

各スライドの下部に表示する例

Word では、脚注や文末脚注の参照を挿入する場所にカーソルを置いて「挿入」 - 「脚注...」を選びます。脚注や文末脚注は、後から相互に変換できます^{*7}。ちなみに、CSS (Cascading Style Sheets)でも脚注を実現できます^{*8}。

最後のスライドにまとめて表示する例

Word では、脚注や文末脚注の参照を挿入する場所にカーソルを置いて「挿入」 - 「脚注...」を選びます。脚注や文末脚注は、後から相互に変換できます^[4]。ちなみに、CSS (Cascading Style Sheets)でも脚注を実現できます^[5]。

*7 [脚注と文末脚注を挿入する Microsoft サポート](#)

*8 CSS Generated Content for Paged Media Module 2. Footnotes <https://www.w3.org/TR/css-gcpm-3/#footnotes>

8. スライドに詰め込む

(was:文字サイズの調整)

「1行だけスライドから溢れてしまったけど、テキストを修正する余裕がない」ことってありますよね。

見出しより下の部分を `<div style="font-size: 90%;">...</div>` で囲んで文字サイズを調整します。

見出し

このスライドの文字を小さくしたい。

└ このスライドの文字を小さくしたい。

見出し

```
<div style="font-size: 60%;">
```

このスライドの文字を小さくしたい。

```
</div>
```

└ このスライドの文字を小さくしたい。

8.1 その他の方法

他の方法で調整できるかもしれません。前の例の `font-size: 70%;` 部分には、次のようなパターンが考えられます。ただし、読みにくくなって不評を買うかもしれません。

1. `font-size: 90%;`などで文字のサイズを少し小さくしてみます
2. `line-height: 1;`などで行の幅を指定します。1などの数字にフォントサイズを掛けたものになります。1.5や1.8になってることが多いので、少し狭くしてみます
3. `word-break: break-all;`で禁則なども無視して改行できるようにします。`word-break: auto-phrase;`は行数が増えがちです
4. `margin-inline-start: -2em;`、`margin-inline-end: -2em;`などで文字が進む方向の余白（マージン）の大きさを負にすると、行頭（-start）や行末（-end）で使える幅が広がって長い行が収まります
5. `margin-block-start: -1em;`や`margin-block-end: -1em;`などで行が進む方向の余白の大きさを負にすると、収まる行数が増えます...が、前後の行の文字と重なるかもしれません
6. `letter-spacing: -1px;`などで字間のスペースに負の値を指定します
7. `position: relative;` `left: -1em;` `top: -1em;`などとすると、サイズではなく位置（position）を上下左右にズラせます。この例だと左（left）に1文字分、上（top）に1文字分ズレ（relative）ます

9. スライドとしての基本的なスタイル設定とプレゼン操作

「A4 の論文の印刷」ではなく「プレゼンのスライド」であるために、以下のようなお膳立てが必要です。

9.1 用紙サイズを A5 横くらいに設定

A5 横を基本に 4:3 や 16:9 になるように調整すると、見出しなどの既定の文字サイズがほどよい大きさになると思います。

```
@page {  
  size: A5 landscape; /* 210mm 148mm */  
  /* size: 216mm 162mm; 4:3 */  
  /* size: 256mm 144mm; 16:9 */  
}
```

9.2 ##と###でスライドを区切る

- ##(HTMLのh2)に加えて###(HTMLのh3)くらいまでを既定のスライド区切りにしておきます
 - CSSとしては、**それらを直下に持つsection要素**を1枚のスライドに対応させます
 - ヘッダー／フッターに表示するため導入した`.conference`クラスなど、便利な印を設定したh2やh3をスライドの区切りから除外します
- 任意でスライドを区切るための`.break-before-page`クラスと、スライド区切りを止める`.break-before-auto`といったクラスも用意します。VFMが生成するsection要素に直接スタイル(`style`属性)を設定できないので、CSS側でこのようなお膳立てが必要です。

```
.break-before-page,
section:has(> h2), section:has(> h3:not(.conference, .bottom-center)) {
  break-before: page;
}
.break-before-auto,
section:has(> h2.break-before-auto, > h3.break-before-auto) {
  break-before: auto;
}
```


9.3 アウトライン番号

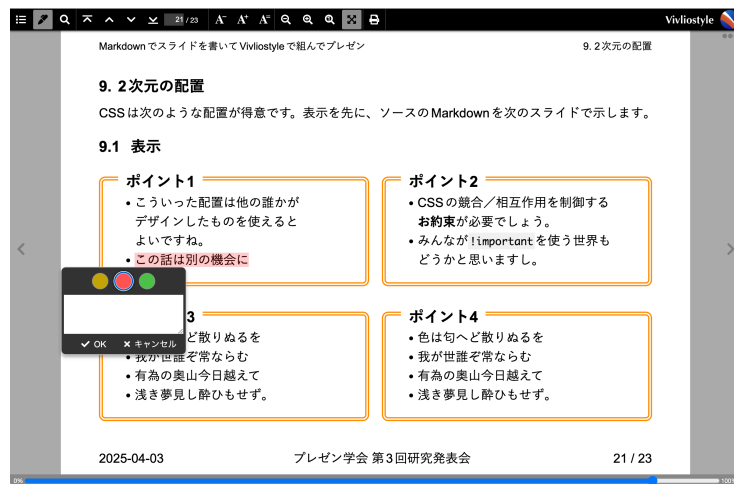
アウトライン(section構造)に、「1.」、「1.1」などと番号が振ってあると、プレゼンのときに今どこの話をしてるのか、聞き手が理解する助けになります

9.4 プレゼンはVivliostyleやPDFで

Vivliostyle Viewerではズーム(拡大／縮小)
したり文字サイズを
変更したりハイライトしたりできます

- ズームではスライド全体が拡大・縮小します
- 文字サイズの変更では、そのサイズで再レイアウトされ、スライド数が増えたり減ったりすることがあります
- 図は、文字サイズに基づいて`block-size: 2em;`などと指定していると、一緒にサイズが変わります。`block-size: 320px;`などの指定では、サイズは変わりません。

PDFで保存してPDFでプレゼンすることもできます



10. 2次元の配置

CSS で次のような配置ができます。先に表示結果を、次にソースの Markdown を示します。

10.1 表示

ポイント1

- こういった配置は他の誰かがデザインしたものを使えるとよいですね。
- この話は別の機会に

ポイント2

- CSS の競合／相互作用を制御する**お約束**が必要でしょう。
- みんなが `!important` を使う世界もどうかと思いますし。

ポイント3

- 色は匂へど散りぬるを
- 我が世誰ぞ常ならむ
- 有為の奥山今日越えて
- 浅き夢見し酔ひもせず。

ポイント4

- 色は匂へど散りぬるを
- 我が世誰ぞ常ならむ
- 有為の奥山今日越えて
- 浅き夢見し酔ひもせず。

10.2 ソースの Markdown

ポイント1

- こういった配置は他の誰かがデザインしたものを使えるとよいですね。
- この話は別の機会に

ポイント2

- CSSの競合／相互作用を制御する**お約束**が必要でしょう。
- みんなが`!important`を使う世界もどうかと思いますし。

ポイント3

- 色は匂へど散りぬるを
- 我が世誰ぞ常ならむ
- 有為の奥山今日越えて
- 浅き夢見し酔ひもせず。

ポイント4

- 色は匂へど散りぬるを
- 我が世誰ぞ常ならむ
- 有為の奥山今日越えて
- 浅き夢見し酔ひもせず。

参考文献

1. word-break <https://developer.mozilla.org/ja/docs/Web/CSS/word-break>↵
2. text-wrap <https://developer.mozilla.org/ja/docs/Web/CSS/text-wrap>↵
3. text-spacing-trim <https://developer.mozilla.org/ja/docs/Web/CSS/text-spacing-trim>↵
4. [脚注と文末脚注を挿入する Microsoft サポート](#)↵
5. CSS Generated Content for Paged Media Module 2. Footnotes <https://www.w3.org/TR/css-gcpm-3/#footnotes>↵