

# Markdown Tips

mebiusbox software

2018 年 12 月 10 日

## 1. はじめに

---

個人的にドキュメントやメモを書くときのテキストフォーマットに、テキストファイル、Html、Textile、TeX、ReStructuredText とか試してきて、今は Markdown 形式に落ち着いています。メモやプレゼンの資料を作成するときなど、ほとんどが Markdown で書ける環境になりました。記述するときのフォーマットは Markdown ですが、使用するソフトウェアは用途によって異なります。この記事では、用途に応じた Markdown ファイルの活用方法について紹介したいと思います。なお、私の主な PC 環境は Windows ですので、Windows で動作確認しているものを紹介しますが、ほとんどがマルチプラットフォームで動作すると思います。

## 2. Atom

---

Atom - <https://atom.io/>

Markdown ファイルを書くときには Atom を使っています。デフォルトの Markdown Preview は使っておらず、**Markdown Preview Enhanced** パッケージを使っています。他にも Markdown 形式で記述するときに便利なパッケージがありますので、重宝しています。

## 3. Visual Studio Code

---

Visual Studio Code - <https://code.visualstudio.com/>

Visual Studio Code も Markdown を書くのに快適な環境になっています。**Markdown Preview Enhanced** も Visual Studio Code でも動作する拡張機能があります。Visual Studio Code は Atom よりもファイルを開くのが速い感じがします。以前は Atom よりも機能が劣っていたのですが、今は改善されて十分使えるエディタになっています。

## 4. Typora

---

Typora - <https://typora.io/>

こちらもエディタではありますが、エディタとしての機能はイマイチで、主にビューアとして使っています。とはいえ、簡単な文書を書く程度ならこれで十分かもしれません。PDF のエクスポートが比較的良好なので、Markdown 形式を PDF に変換したいときなどに便利です。epub 形式などもサポートしています。スタイルシートで見た目のカスタマイズ出来るのもいいところです。

## 5. Marp

---

Marp - <https://yhatt.github.io/marp/>

Markdown 形式で記述できるプレゼン資料作成ソフトです。かなり捗るので、作成するプレゼンのスライド枚数がいつの間にかすごい量になってたりします。個人的に PowerPoint など特に不便を感じているのが数式の記述なのですが、Mathjaxのおかげで TeX 形式で書けるので大変便利です。スタイルシートで見た目のカスタマイズ出来るのもいいところです。

## 6. Boostnote

Boostnote - <https://boostnote.io/>

Markdown 形式で記述できるメモソフトです。ファイル構成もシンプルで同期しやすく、複数 PC でも快適です。メモソフトとして、個人的に必須な数式の記述も出来るし、とても使いやすいです。

## 7. 他の人に渡すときは

たとえ相手がエンジニアでも Markdown ファイルを直接渡すことはしていません。PDF にするか、HTML 形式に変換して渡しています。HTML 形式の場合は Atom の Markdown Preview Enhanced を使ってエクスポートしています。体裁を調整する場合は、エクスポートした HTML ファイルを直接修正してスタイルシートをいじっています。PDF や印刷する場合は、エクスポートした HTML ファイルを Chrome で開いて、印刷したり PDF で保存しています。また、Typora でエクスポートすれば様々な形式で出力することが出来ます。

## 8. Markdeep

Markdeep - <https://casual-effects.com/markdeep/>

Markdeep は Markdown 形式で書かれたファイルを論文のような体裁で表示してくれる Javascript です。使い方は簡単で、Markdown ファイルをそのまま拡張子 `.html` にして保存し、特定のコードを入れるだけで綺麗な文書を作成出来ます。公式では PDF に変換するときは `'wkhtmltopdf'` を使う方法が紹介されていましたが、どうも上手くいかなかったのが、Chrome から PDF にしています。ここでは、私が使っている設定などを解説します。

多言語をサポートしています。ファイルのヘッダにメタ属性を埋め込みます。

```
<meta lang="ja" charset="utf-8">
```

日本語のフォント（NotoSans）を使うために、以下の指定をします。

```
<link rel="stylesheet" href="https://fonts.googleapis.com/earlyaccess/notosansjp.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:600,800">
```

フォントの指定など体裁を少しカスタマイズします。 `font-size` は印刷時などで調整します。

```
body { font-size: 10pt; font-family: "Noto Sans JP"; max-width: 1200px; },
.md code { font-family: Consolas; }
.md pre.backtick {
  border-top: 1px solid #CCC;
  border-bottom: 1px solid #CCC;
  padding: 5px 0 5px 20px;
  margin: 0 0 0 0;
  background: #FCFCFC;
  page-break-inside: avoid;
}
```

数式に自動で番号が割り振られてしまうと困るときがあるので、以下で回避できます。

```
<script type="text/x-mathjax-config">
MathJax.Hub.Config({
  TeX: {
    equationNumbers: {
      autoNumber: false
    }
  }
});
</script>
```

これを指定しても `align` で指定しているものは番号が割り振られるので、そのときは `align*` を使います。  
これらを設定をファイルの先頭に、そしてファイルの末尾に以下を入れます。

```
<!-- Markdeep: --><style class="fallback">body{visibility:hidden;white-space:pre;font-family:,monospace}<
/style><script src="https://casual-effects.com/markdeep/latest/markdeep.min.js?"></script><script>window.
alreadyProcessedMarkdeep||(document.body.style.visibility="visible")</script>
```

画像のサイズを全体的に調整したい場合は、次のようにします。

```
<!-- Markdeep: --><style class="fallback">body{visibility:hidden;white-space:pre;font-family:,monospace}<
/style><script src="https://casual-effects.com/markdeep/latest/markdeep.min.js?"></script><script>window.
alreadyProcessedMarkdeep||(document.body.style.visibility="visible")</script>
<style>
.md img { width: 30%; };
</style>
```

また、デフォルトでは目次が自動で作られて表示されますが、これを無効にしたい場合は次のようにします。

```
<script>window.markdeepOptions = {tocStyle:'none'}</script><!-- Markdeep: --><style class="fallback">body
{visibility:hidden;white-space:pre;font-family:,monospace}</style><script src="https://casual-effects.com
/markdeep/latest/markdeep.min.js?"></script><script>window.alreadyProcessedMarkdeep||(document.body.style
.visibility="visible")</script>
```

スタイルシートを使ってマルチカラム形式で表示することが出来ます。次のような感じで表示されます。

互いに識別できる「もの」の集まりを「集合」といいます。たとえば、1から5までの自然数1,2,3,4,5の集まりは集合です。集合を $A$ とすると $A = \{1, 2, 3, 4, 5\}$ のように記述します（列記法）。または、 $A = \{x   1 \leq x \leq 5, x \text{ は整数}\}$ と書きま	す（条件法）。「もの」の1つ1つを「要素」といいます。 $a$ がある集合 $A$ の要素であることを $a \in A$ と表します。 $b$ がある集合 $A$ の要素でないことを $b \notin A$ と表します。
互いに識別できる「もの」の集まりを「集合」といいます。たとえば、1から5までの自然数1,2,3,4,5の集まりは集合です。集合を $A$ とすると	$A = \{1, 2, 3, 4, 5\}$ のように記述します（列記法）。または、 $A = \{x   1 \leq x \leq 5, x \text{ は整数}\}$ と書きま
互いに識別できる「もの」の集まりを「集合」といいます。たとえば、1から5までの自然数1,2,3,4,5の集	まりは集合です。集合を $A$ とすると $A = \{1, 2, 3, 4, 5\}$ のように記述します（列記法）。また は、
	$A = \{x   1 \leq x \leq 5, x \text{ は整数}\}$ と書きま
	す（条件法）。「もの」の1つ1つを「要素」といいます。 $a$ がある
	集合 $A$ の要素であることを $a \in A$ と表します。 $b$ がある集合 $A$ の要素でないことを $b \notin A$ と表しま

Fig.1: マルチカラム

これをするために、スタイルシートに以下を追加します。

```
.mcol2 {
columns:2;-webkit-columns:2;-moz-columns:2;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
.mcol3 {
columns:3;-webkit-columns:3;-moz-columns:3;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
.mcol4 {
columns:4;-webkit-columns:4;-moz-columns:4;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
```

たとえば、2カラムの場合

```
<div class="mcol2">
  2カラムで表示されます
</div>

<div class="mcol3">
  3カラムで表示されます
</div>

<div class="mcol4">
  4カラムで表示されます
</div>
```

自動で整列されるので、スタイルシートを使って適当な余白を入れたり、++++++ でページ送りが出来ます。

Markdeep は見出しに自動的に番号を割り振ってくれます。細かいフォーマットを指定できますので、詳細は公式サイトを参照してください。たとえば、この番号を非表示にしたいとき、スタイルシートに次を追加します。

```
.md h3:before, .md h4:before { content:none }
```

最後に個人的に使っている基本テンプレートです。まずファイルの先頭には

```
<meta lang="ja" charset="utf-8">
<link rel="stylesheet" href="https://fonts.googleapis.com/earlyaccess/notosansjp.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:600,800">
<style>
body { font-family: "Noto Sans JP"; max-width: 1200px; },
.md code { font-family: Consolas; }
.md pre.backtick {
border-top: 1px solid #CCC;
border-bottom: 1px solid #CCC;
padding: 5px 0 5px 20px;
margin: 0 0 0 0;
background: #FCFCFC;
page-break-inside: avoid;
}
.mcol2 {
columns:2;-webkit-columns:2;-moz-columns:2;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
```

```

.mcol3 {
columns:3;-webkit-columns:3;-moz-columns:3;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
.mcol4 {
columns:4;-webkit-columns:4;-moz-columns:4;column-gap:3em;-webkit-column-gap:3em;-moz-column-gap:3em;
}
</style>
<script type="text/x-mathjax-config">
MathJax.Hub.Config({
  TeX: {
    equationNumbers: {
      autoNumber: false
    }
  }
});
</script>

```

そして、ファイルの末尾には

```

<!-- Markdeep: --><style class="fallback">body{visibility:hidden;white-space:pre;font-family:,monospace}<
/style><script src="https://casual-effects.com/markdeep/latest/markdeep.min.js?"></script><script>window.
alreadyProcessedMarkdeep||(document.body.style.visibility="visible")</script>

```

## 9. さいごに

実際に書いてまとめてみると、あんまり使いこなしていない感じがしてきましたが…本記事が参考になれば幸いです。