

# AsciiDoc環境構築 覚え書き

K0.1, 2019/08/24

# 目次

1. AsciiDocについて .....	1
2. 環境構築手順 .....	2
2.1. Chocolateyをインストール .....	2
2.2. Ruby等のパッケージをインストール .....	2
2.3. AsciiDoc関連ツールをインストール .....	3
2.4. 作業ディレクトリを作成する .....	3
3. 実際にAsciiDocを書いてみる .....	6
3.1. テストサンプルを書く .....	6
3.2. プレビューを行う .....	10
3.3. HTMLやPDFに変換する .....	11

# 1. AsciiDocについて

## AsciiDocとは？

- Markdownのような軽量マークアップ言語の一つ
- プレーンテキストで体裁が整った文章が書ける
- 可読性が高くMarkdownよりも表現力が高い

## メリット

- テキスト形式なのでGitで管理しやすい
- インクルード機能により外部ファイルの読み込みができる
- HTML、PDFへの変換ができる



AsciiDoc公式サイト  
<http://asciidoc.org/>

## 2. 環境構築手順

1. パッケージマネージャのChocolateyをインストール
2. Chocolateyのリポジトリに登録されているRuby等のパッケージをインストール
3. Ruby製のAsciiDoc関連ツールをインストール
4. 作業ディレクトリを作成

### 2.1. Chocolateyをインストール

コマンドプロンプト(管理者権限)で以下を実行

```
@'%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe' -NoProfile  
-InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object  
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" &&  
SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```



公式サイト

<https://chocolatey.org/install#installing-chocolatey>

インストール手順解説(日本語)

<https://qiita.com/konta220/items/95b40b4647a737cb51aa>

#### Chocolateyとは?

- Windows上で動作するソフトウェアをコマンドラインでパッケージ管理可能なツール

#### メリット

- Chocolateyのリポジトリに登録されているパッケージを**一発でインストール**できる
- Chocolateyでインストールしたソフトは**一括でアップデート**できる

### 2.2. Ruby等のパッケージをインストール

コマンドプロンプト(管理者権限)で以下を実行

```
cinst ruby -y ①  
cinst graphviz -y ②  
cinst jdk8 -y ③  
cinst vscode -y ④  
cinst sourcetree --version 2.5.5 -y ⑤
```

- ① Ruby (AsciiDoc関連ツールを利用するのに必要)
- ② Graphviz (PlantUML等で記述した図の表示に必要)
- ③ Java (同上)
- ④ Visual Studio Code (AsciiDocをプレビュー可能なテキストエディタ)
- ⑤ SourceTree (GitのGUIツール)

Atlassianアカウントを作成してSourceTreeのサインインに成功したら  
コマンドプロンプト(管理者権限)で以下を実行しアップデートする



```
choco upgrade all -y
```

※初めから最新verをインストールしないのはBitbucketに登録せずに利用するため  
<https://hepokon365.hatenablog.com/entry/2019/03/25/222814>

## 2.3. AsciiDoc関連ツールをインストール

コマンドプロンプトで以下を実行

```
gem install asciidoctor ①  
gem install --pre asciidoctor-pdf ②  
gem install asciidoctor-pdf-cjk ③  
gem install asciidoctor-diagram ④  
gem install coderay ⑤
```

- ① AsciiDoc→HTMLに変換用
- ② AsciiDoc→PDFに変換用
- ③ PDF変換のレイアウト崩れ対応用
- ④ PlantUML等の図の記述用
- ⑤ コードのシンタックスハイライト用

社内のネットワークから実施する場合はgemにproxyを指定する



```
gem install xxxx -p proxy http://アドレス:ポート
```

proxyの確認手順  
<https://pasokatu.hateblo.jp/entry/2017/07/04/111147>

asciidoctor公式サイト  
<https://asciidoctor.org/>

## 2.4. 作業ディレクトリを作成する

ドキュメント作成のための作業ディレクトリを用意

```
└─ test          // ドキュメント(*.adoc)を格納するフォルダ  
  └─ csv         // CSVファイルを格納  
  └─ fonts       // フォントファイルを格納  
  └─ images      // イメージファイルを格納  
    └─ style     // スタイルファイルを格納
```

### HTMLのスタイルファイル

asciidoctorの配布ファイルがWindowsの場合は以下にあるのでコピペして利用

```
// ruby2.6でasciidoctorのverが2.0.10の場合  
C:\tools\ruby26\lib\ruby\gems\2.6.0\gems\asciidoctor-  
2.0.10\data\stylesheets\asciidoctor-default.css
```

## PDFのスタイルファイル

asciidoctor-pdfの配布ファイルがWindowsの場合は以下にあるのでコピーして利用

```
// ruby2.6でasciidoctor-pdfのverが1.5.0.beta.2の場合  
C:\tools\ruby26\lib\ruby\gems\2.6.0\gems\asciidoctor-pdf-  
1.5.0.beta.2\data\themes\default-theme.yml
```



デフォルトのスタイルファイル  
css : そのままでも十分使えそう  
yaml : いまいちなのでHTML化時のスタイルに寄せた設定に修正してみた (public\_style.ymlとして利用)

公式サイト

<https://github.com/asciidoctor/asciidoctor-pdf/blob/master/docs/theming-guide.adoc>

色表現

[https://www.lab-nemoto.jp/www/leaflet\\_edu/ColorMaker.html](https://www.lab-nemoto.jp/www/leaflet_edu/ColorMaker.html)

PDF化時に文字の色が変わるようにする

[https://blog.siwa32.com/asciidoctor\\_pdf\\_color/](https://blog.siwa32.com/asciidoctor_pdf_color/)

→「2.2 asciidoctor-pdfのソースを修正する」

## フォントファイル

asciidoctor-pdfの配布ファイルがWindowsの場合は以下にあるのでコピーして利用

```
// ruby2.6でasciidoctor-pdfのverが1.5.0.beta.2の場合  
C:\tools\ruby26\lib\ruby\gems\2.6.0\gems\asciidoctor-pdf-  
1.5.0.beta.2\data\fonts\*.ttf
```



参考サイト

<https://ryuta46.com/267>

<https://qiita.com/kuboaki/items/67774c5ebd41467b83e2>

## ドキュメントファイル

適当にメモ帳で以下の設定で作成する

```
拡張子 : .adoc  
文字コード : UTF-8
```

## 格納後の作業フォルダ内はこんな感じになる

```
└ test
  ├── csv
  ├── fonts
  │   ├── *.ttf
  │   └ ...
  ├── images
  ├── style
  │   ├── asciidoctor-default.css
  │   ├── default-theme.yml
  │   └ public_style.yml
  └ *.adoc
```

## 3. 実際にAsciiDocを書いてみる

VScodeで\*.adocファイルを開く

### 3.1. テストサンプルを書く

AsciiDocの文章の設定を行うAttributeと表紙の設定に続いて本文という構成にしてみた

```
////////////////////////////////////
// Attribute
////////////////////////////////////

//ドキュメント種類
:doctype: book
//ドキュメント言語
:lang: ja
//目次生成
:toc: left
//目次生成階層数
:toclevels: 2
//目次タイトル
:toc-title: 目次
//章番号生成
:sectnums:
//章番号生成階層数
:sectlevels:
//PDF化時の章タイトル
:chapter-label:
//シンタックスハイライト
:source-highlighter: coderay
//アイコンフォント
:icons: font
//UIマクロ
:experimental:
//HTML化時の画像のdata-uri要素化
:data-uri:
//画像ファイル格納先
:imagesdir: ./images
//HTML化時のスタイルファイル格納先
:stylesdir: ./style
//HTML化時のスタイルファイル
:stylesheet: asciidoctor-default.css
//PDF化時のスタイルファイル
:pdf-style: ./style/public_style.yml
//PDF化時のフォントファイル格納先
:pdf-fontsdir: ./fonts

////////////////////////////////////
// 表紙
////////////////////////////////////
```



```
//ドキュメントタイトル、表紙に入る
= テストサンプル
//ドキュメントタイトル、ヘッダーに入る
:docname: テストサンプル
//著者
:author:
//改定番号
:revnumber: K0.1
//改定日
:revdate: 2019/08/24
//改定番号のラベル
:version-label:
//ロゴ画像
//:title-logo-image:
//表紙背景画像
//:title-page-background-image:

////////////////////////////////////
// 本文
////////////////////////////////////

== よく使う文法の紹介

いくつかの文法の利用にはAttribute `:XXX:` の指定が必要です +

=== リスト

先頭に `*` を付けるとリストになる

* level 1
* level 1
** level 2
*** level 3
* level 1

ラベル名に続けて `::` を付けるとラベル付きリストになる

CPU:: コンピューターの中心的な処理装置
RAM:: 読み書き可能な主記憶装置
SSD:: フラッシュメモリを使用した補助記憶装置
キーボード:: キーを押すことで信号を送信する入力装置
マウス:: コンピューターのポインティングデバイス
モニター:: 映像を表示する出力装置

=== 段落

* 通常の改行は無視
* 空行で別段落
* `+` で改行できる
```

.例)

ただ改行しただけだと  
文章はつながったままです

空行を設けると別段落扱いになります

改行させたいところで `+` をつけば +  
改行できます

### === セクション

\* `=` でタイトルを示す

\* Level0 (`=` が1個) は文章中でひとつしか使えない (ドキュメントタイトル扱い)

\* Level1以上 (`=` が2個以上) で自動的にナンバリングされる

### === ブロック

\* `----` や `====` など囲ってブロックを指定する

\* ブロックのヘッダーに `.XXX` を付けるとタイトルが指定できる

.例)

```
----  
y = a × b + c  
----
```

### === コードブロック

\* ソースコードをハイライト表示できる

\* ブロックのヘッダーに `[source, 使用する言語]` の形で指定します

.例) C言語のサンプルコード

```
[source, C]  
----  
#include <stdio.h>  
#include <stdlib.h>  
  
int main(void)  
{  
    puts("Hello World!");  
    return EXIT_SUCCESS;  
}  
----
```

### === 脚注

\* `NOTE` , `TIP` , `IMPORTANT` , `CAUTION` , `WARNING` の5種類

\* ブロックのヘッダーに `[NOTE]` の形で指定します

## [NOTE]

====

ブロックの中に内容を書きます

====

## === テキストフォーマット

- \* 太字: 文字を `\*` で囲う
- \* モノスペース: 文字を `` で囲う

.例)

\*太字の語句\* と \*\*太\*\*字の文\*\*字\*\* +  
`モノスペースの語句` と ``モノ``スペースの文``字``

## === マーカー

- \* 蛍光ペン: 文字を `#` で囲う
- \* アンダーライン: 文字を `#` で囲い、頭に `[.underline]` を付ける
- \* 取り消し線: 文字を `#` で囲い、頭に `[.line-through]` を付ける
- \* 文字拡張: 文字を `#` で囲い、`[.big]` or `[.small]` を付ける
- \* 文字色: 文字を `#` で囲い、`[color]` を付ける

.例)

#冬# よりかは [.underline]#夏# の方が [.line-through]#嫌い# [red]#好き# [.big]#  
だ#

## [NOTE]

====

PDF化時に文字の色が変わるようにする +  
[https://blog.siwa32.com/asciidoctor\\_pdf\\_color/](https://blog.siwa32.com/asciidoctor_pdf_color/) +  
→「2.2 asciidoctor-pdfのソースを修正する」

====

## === URL

- \* httpなどを自動で判定してリンクを生成してくれます +  
<https://ja.wikipedia.org/wiki/AsciiDoc>
- \* 別名を指定する際は後ろに `[xxx]` を付与します +  
<https://ja.wikipedia.org/wiki/AsciiDoc>[ここをクリック]

## === コメント

- \* `//` でコメントアウト
- \* `//` で囲うと複数行にわたってコメントアウト

// 一文コメントアウトになる

```
////  
囲うと複数行の  
コメントアウトもできる  
////
```

...

## 3.2. プレビューを行う

VScodeの設定を行うことでプレビュー(ショートカット `Ctrl+K → V`)が可能

### 拡張機能をインストール

[表示]→[拡張機能]から **AsciiDoc** を検索しインストール



参考サイト

[https://qiita.com/o\\_sol06/items/a07ebcb0b48295a4c3b3](https://qiita.com/o_sol06/items/a07ebcb0b48295a4c3b3)

VScodeを日本語表示にする拡張機能

<https://marketplace.visualstudio.com/items?itemName=MS-CEINTL.vscode-language-pack-ja>

### asciidocotorの設定を変更

[ファイル]→[基本設定]→[設定]から **asciidocotor** を検索し、以下の設定を行う

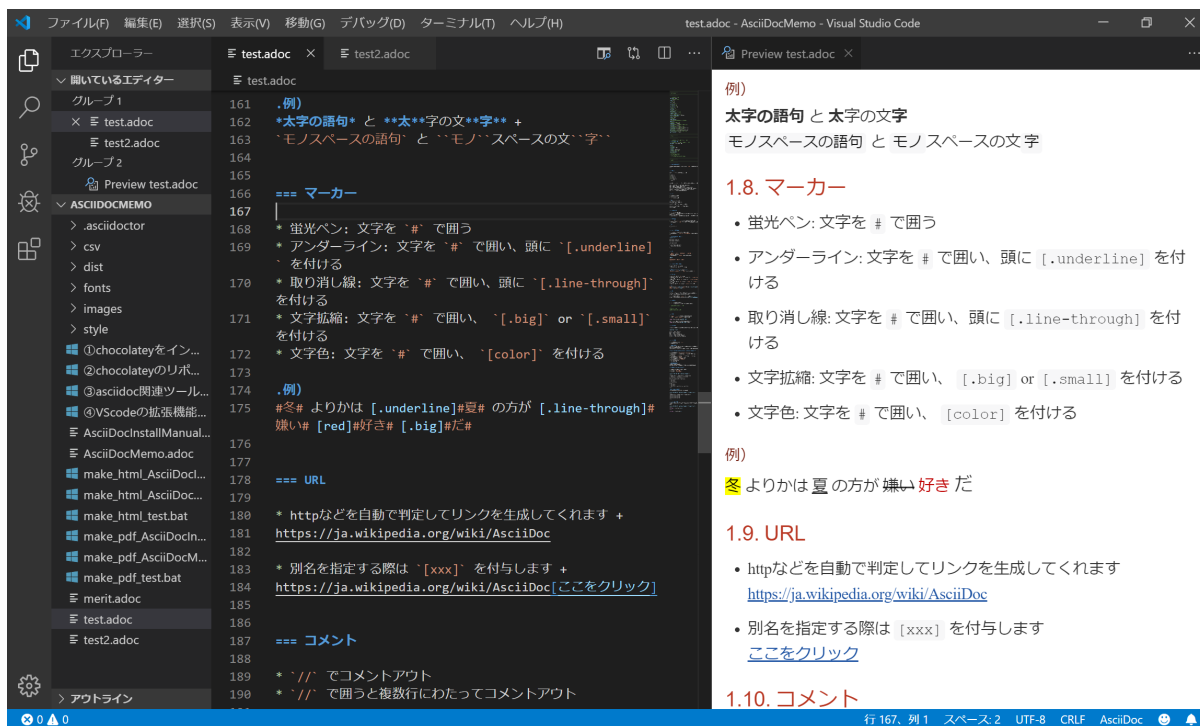
```
asciidocotor_command : asciidocotor -n -r asciidocotor-diagram -o-  
asciidocotorpdf_command : asciidocotor-pdf -n -r asciidocotor-diagram -r  
asciidocotor-pdf-cjk -o-  
use_asciidocotor_js : false(チェックを外す)
```



参考サイト

<https://qiita.com/hyt126/items/fdeff36f09bb221dfac0>

参考までに「3.1.テストサンプル」のプレビュー結果を以下に示す



### 3.3. HTMLやPDFに変換する

コマンドプロンプトで以下を実行(\*にファイル名を指定)

- HTMLファイルに変換

```
asciidoctor -r asciidoctor-diagram -o *.html *.adoc
```

- PDFファイルに変換

```
asciidoctor-pdf -r asciidoctor-diagram -r asciidoctor-pdf-cjk -o *.pdf *.adoc
```