

AsciiDoc環境構築 覚え書き

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. Attributeを書く	6
3.2. テストサンプルを書く	6
3.3. プレビューを行う	8
3.4. HTMLやPDFに変換する	8

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への変換もできる

2. 環境構築手順

1. パッケージマネージャーの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 (PlantUML等で記述した図の表示に必要)
- ④ Visual Studio Code (AsciiDocをプレビュー可能な編集エディタ)
- 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、一括アップデートする



```
choco upgrade all -y
```

※初めから最新verをインストールしない理由は以下を参照

<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>

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  
| -dist          // HTMLやPDFの出力先  
| -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

windowsの場合は以下に入っている所以cssファイルをコピーして格納

```
// 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

windowsの場合は以下に入っているのでyamlファイルをコピーして格納

```
// 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
```



設定参考 (public_style.ymlにリネームして使用) <https://qiita.com/tamikura@github/items/5d3f62dae55617ee42bb>

PDF化時に文字の色が変わるようにする 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

Windowsの場合は以下に入っているので中身を全てコピーして格納

```
// 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\*
```

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して格納

拡張子 : .adoc
文字コード : UTF-8

作業フォルダ内はこんな感じになる

```
| -test
|   |-*.adoc
|   |-dist
|   |-fonts
|       |-IPA_Font_License_Agreement_v1.0.txt
|       |-ipagp.ttf
|       |-LICENSE-mplus-testflight-58
|       |-LICENSE-noto-2015-06-05
|       |-mplus1mn-bold_italic-ascii.ttf
|       |-mplus1mn-bold-ascii.ttf
|       |-mplus1mn-italic-ascii.ttf
|       |-mplus1mn-regular-ascii-conums.ttf
|       |-mplus1p-regular-fallback.ttf
|       |-notoserif-bold_italic-subset.ttf
|       |-notoserif-bold-subset.ttf
|       |-notoserif-italic-subset.ttf
|       |-notoserif-regular-subset.ttf
|       |-Readme_IPAfont00303.txt
|   |-images
|   |-style
|       |-asciidoctor-default.css
|       |-default-theme.yml
|       |-public_style.yml
```

3. 実際にAsciiDocを書いてみる

VScodeで*.adocファイルを開く

3.1. Attributeを書く

とりあえず最低限の指定を行う

```
//日本語ドキュメント
:lang: ja
//文書タイプはbookにする
:doctype: book
//目次を自動生成する
:toc: left
//対象とする階層数を指定する
:toclevels: 3
//タイトルを変更する
:toc-title: 目次
//章見出し番号を出力する
:sectnums:
//PDF化時の章見出しのChapter.が表示されないようにする
:chapter-label:
//シンタックスハイライトを使用する
:source-highlighter: coderay
//アイコンフォントを利用するフラグ
:icons: font
//マクロを使用する（ショートカットキーとか）
:experimental:
//画像をdata-uriとして埋め込む
:data-uri:
//イメージファイルを置くフォルダ
:imagesdir: ./images
//PDF化時のフォントファイルを置くフォルダ
:pdf-fontsdir: ./fonts
//PDF化時のスタイルファイルを指定
:pdf-style: ./style/public_style.yml
//HTML化時のスタイルファイルを置くフォルダ
:stylesdir: ./style
//HTML化時のスタイルファイルを指定
:stylesheet: asciidoctor-default.css
```

3.2. テストサンプルを書く

Attributeに続けて下記のテストサンプルを書く

= asciidocの使い方

== asciidocとは？

asciidocとは [blue]#軽量マークアップ言語# です

詳しくは<<can_asciidoc,asciidocでできること>>を参照

[[can_asciidoc]]

== asciidocでできること

.コードハイライト

[source, json]

```
{
  "hoge" : "fuga",
  "foo" : [1,2,3]
}
```

.結合+箇条書例

[cols="1,2a,3a"]

|====

|列1|列2|列3

3+|3列結合

.2+|2行縦結合|b-1|c-2

|b-2|

* c-3

* c-4

|====

[NOTE]

====

* format="csv"ではできません

====

=== asciidoctorだとPlantUMLでシーケンス図作成

[plantuml]

actor ユーザー as user

user -> ログイン : ログインする

ログイン --> user:



文法リファレンス(日本語)

https://takumon.github.io/asciidoc-syntax-quick-reference-japanese-translation/#_%E8%84%9A%E6%B3%A8

3.3. 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

asciidoctorの設定を変更

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

```
asciidoctor_command : asciidoctor -n -r asciidoctor-diagram -o-
asciidoctorpdf_command : asciidoctor-pdf -n -r asciidoctor-diagram -r
asciidoctor-pdf-cjk -o-
use_asciidoctor_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「3.2.テストサンプル」のプレビュー結果を以下に示す

The screenshot displays the VS Code editor with the AsciiDoc extension. The left pane shows the source file with settings like `lang: ja`, `doctype: book`, and `toc: left`. The right pane shows the rendered HTML output, including a table of contents and a table with 3 columns and 3 rows.

列1	列2	列3
3列結合		
2行縦結合	b-1	c-2
	b-2	c-3

3.4. 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 dist/*.html *.adoc
```

- PDFファイルに変換

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

PlantUMLの内容編集の度に画像ファイルが増えるのを防ぐため
[plantuml,画像ファイル名]としてファイル名を指定する

```
[plantuml,test]  
----  
class Animal {  
    run()  
}  
  
class Cat extends Animal {  
}  
----
```



↓ プレビュー結果

