目次

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を書くのが得意
- 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に変換できる

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. chcolatey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

以下を参考にインストールします
https://chocolatey.org/install https://qiita.com/konta220/items/95b40b4647a737cb51aa
https://qiita.com/NaoyaOura/items/1081884068fe3ea79570

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"
```

必ず公式サイトからコマンドをコピペして実行すること
インストールが成功すれば以下のコマンドで確認することができる
`choco -v`

*chocolatey*とは？

- windows上で動作するソフトウェアをコマンドラインからインストールすることができるパッケージマネージャー
- chocolateyのリポジトリに登録されているパッケージをコマンドラインから一発でインストールできる

メリット

- 使いたいソフトのインストーラをダウンロードして毎回YESをクリックする 手間が省ける
- 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 -y ruby ①
cinst -y graphviz ②
cinst -y jdk8 ③
cinst -y vscode ④
```

① ruby

② Graphviz

③ Java

④ Visual Studio Code(テキストエディタ、以下`VScode`とする)

cinst = choco install の略コマンド
インストールが成功すれば以下のコマンドでverを確認できる
`ruby -v`
`gem -v`
GraphvizとJavaはPlantUML等で図のプレビューに必要となる
VScodeの拡張機能でasiidocをプレビューしながら編集が可能

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

2.4. 作業ディレクトリを作成する

ドキュメントを作成するための作業ディレクトリを作っておく

```
|-test          // ドキュメントを格納するフォルダ(任意の名前でOK)
   |-dist       // 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
```

以下のファイルを作業ディレクトリに格納する

- htmlのスタイルファイル
  windowsの場合は以下に入っているので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
  window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
```

public_style.ymlとリネームし、下記サイトを参考に編集
https://qiita.com/tamikura@github/items/5d3f62dae55617ee42bb

- 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
      |-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
   |-images
   |-style
      |-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:
//章見出しのChapte.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
//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
//アイコンフォントを利用するフラグ
:icons: font
//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:
----
```

公式リファレンス
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

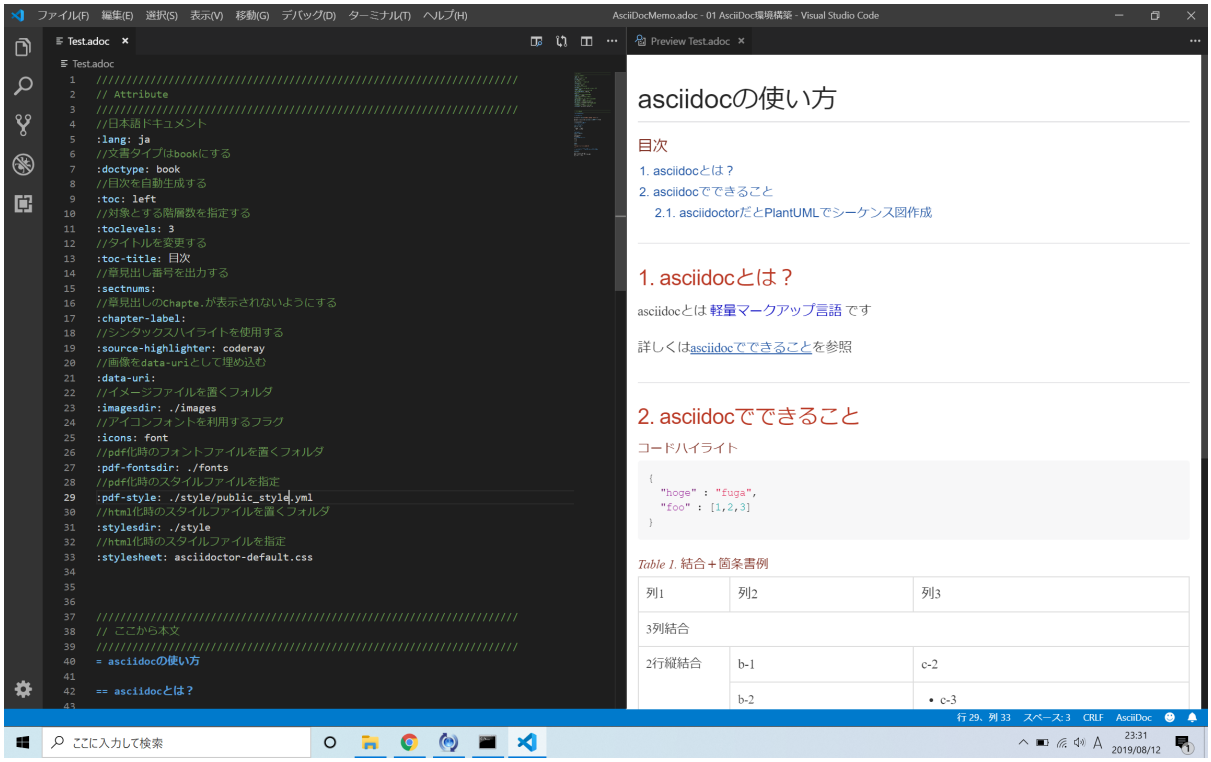
VScodeの設定を行うことでプレビュー(ショートカット `Ctrl+K→V`)が可能

拡張機能をインストールします
https://qiita.com/o_sol06/items/a07ebcb0b48295a4c3b3
画面上部の[表示]→[拡張機能]から `AsciiDoc` を検索しインストール

asciidoctorの設定を変更します
https://qiita.com/hyt126/items/fdeff36f09bb221dfac0
画面上部の[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 -r asciidoctor-diagram
asciidoctorpdf_command : asciidoctor-pdf -r asciidoctor-diagram -r
asciidoctor-pdf-cjk
use_asciidoctor_js  : false(チェックを外す)
```

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



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

- htmlファイルに変換

```
asciidoctor -o dist/*.html *.adoc
```

- pdfファイルに変換

```
asciidoctor-pdf -o dist/*.pdf *.adoc
```

```
actor ユーザー as user
user -> ログイン : ログインする
ログイン --> user:
```