メインタイトル

サブタイトル

バージョン　：0.0.0

改定日　　：2020/3/1

XXXX 株式会社
YYYY 事業部
ZZZZ 課

ほげほげ　太郎

確認	検認	承認

目次

1. ASCIIDOCって？

まだ使いこなせていませんが、Markdownで不便だったところが解消出来ます。

- 番号付の段落で階層表現が出来る
- ことと、テーブル内での改行や、セルの結合が出来る
- ドキュメントを分割して好きな所でインポート出来る（これはMarkdownでも出来る？）
- Markdownは方言がおおくて、変換される見た目が大きく違ったが、方言が少ない（ASCIIDOCも見た目が違うところがありますが方言は少ないです）

2. 環境構築手順

2.1. 一括インストール

1. initialScriptsフォルダ内のsetupDev.batを実行してください

2.2. 個別インストール

過去の記述を残しているだけですので、一括インストールを使ってインストールしてください
インストールを行うとVSCodeでのAsciiDocプレビュー時に利用するKrokiを実行するタスクがタスクマネージャに登録されます

2.2.1. Rubyのインストール

https://rubyinstaller.org/downloads
ここからRuby（2.2以降）をダウンロードする
このメモを書いた時には2.7.0のWITHOUT DEVKITをインストールしました

2.2.2. PLANTUML関連のインストール

ASCIIDOCの場合インストールしなくてもいいみたい +
あとでgemでasciidoctor-diagramを入れるのでそこで解決されるようです。 +
うまくいかない場合はここにある物をインストールしてパスを通せば動くと思います。

1. Java8をインストールする
   https://www.java.com/ja/download/manual.jsp
   ここからインストーラーをダウンロードしてインストールしてください
2. PLANTUMLをインストールする
   http://plantuml.com/download
   ここからインストーラーをダウンロードしてインストールしてください
3. Graphvizをインストールする
   http://www.graphviz.org/Download_windows.php
   ここからインストーラーをダウンロードしてインストールしてください

2.2.3. ASCIIDOCTOR関連のインストール

1. Rubyのgemを使ってインストールするためRubyにProxyの設定を追加します
   ★Proxyを使わない環境の場合は必要ありません
   c:\\users\\[ユーザーID]\\.gemrc を以下のように編集する

```
gem: --no-ri --no-rdoc
http_proxy: http://[ユーザーID]:[パスワード]@proxyserver:8080
https_proxy: http://[ユーザーID]:[パスワード]@proxyserver:8080
```

```
環境変数に登録しても可
```

```
set http_proxy=http://[ユーザーID]:[パスワード]@proxyserver:8080
set https_proxy=http://[ユーザーID]:[パスワード]@proxyserver:8080
```

2. コマンドプロンプトを以下のコマンドを実行してASCIIDOC関連のモジュールをインストールする

```
: asciidoctorのインストール
gem install asciidoctor
: asciidoctor-pdfのインストール
gem install --pre asciidoctor-pdf
: コードのシンタックスハイライト用
gem install coderay
: PDF変換のレイアウト崩れ対応
gem install asciidoctor-pdf-cjk
: PlantUMLなどの図を使用
gem install asciidoctor-diagram
```

3. 変換方法

convert.bat を実行するとHTMLとPDF変換に変換します JenkinsのJOBでもconvert.batを呼び出してHTML,PDF変換をするようにしました。

変換バッチファイル convert.bat

```
@echo off

setlocal

ECHO ASCII DOC CONVERTER

cd %~d0%~p0
echo %~d0%~p0


REM PlantUMLのJARファイルのパスを指定
set PLANTUML_JAR="%~d0%~p0lib\plantuml-1.2025.4.jar"

del /f /q /s "%~d0%~p0HTML"
del /f /q /s "%~d0%~p0PDF"

echo START COVER Convert
powershell.exe  -NoProfile -ExecutionPolicy RemoteSigned '"%~d0%~p0convertpdf.ps1"'
echo FINISH COVER Convert

echo START HTML Convert
powershell -Command "$env:DIAGRAM_PLANTUML_CLASSPATH='%PLANTUML_JAR%'; Start-Process
-NoNewWindow -Wait asciidoctor -ArgumentList '%~d0%~p0AsciiDocSample.adoc', '-o',
'%~d0%~p0HTML\\AsciiDocSample.html', '-r', 'asciidoctor-diagram'"

echo FINISH HTML Convert

echo START PDF Convert
powershell -Command "$env:DIAGRAM_PLANTUML_CLASSPATH='%PLANTUML_JAR%'; Start-Process
-NoNewWindow -Wait asciidoctor-pdf -args '"%~d0%~p0AsciiDocSample.adoc" "-o"
"%~d0%~p0PDF\AsciiDocSample.pdf" "-r" "asciidoctor-diagram" "-a" "pdf-theme=conf/theme.yml" "-a"
"pdf-fontsdir=fonts" "-a" "converttype=pdf"'
echo FINISH PDF Convert

endlocal
```

4. 表紙

ASCIIDOCの表紙機能は貧弱で、たいていのプロジェクトは表紙だけは凝っていることが多いので、表紙についてはWORDで編集した物をPDF変換して結合するようにしました。 +
conver\cover_page.docx
を各プロジェクト等の体裁に合わせて編集してください。
convert.batの中でWORDをPDFに変換してPDFへ結合するようにしてあります。

5. リファレンスサンプル

このドキュメントのソースコードを参照して下さい

5.1. ドキュメント情報

表紙以外のASCIIDOC本文部分で使用する情報は、ADOCのヘッダー部分に定義します。
以下のように定義してください。下記のサンプルでは改行していますが、改行には対応していないようです。

```
:author: 会社名 +
部署名 +
課名 +
TEL：000-0000-0000
:revnumber: v0.0.0
:revdate: 2020/03/01
:copyright: Copyright &#169;2020 SAMPLE COMPANY CORPORATION All right reserved.
```

タイトルとサブタイトルはレベル0のタイトルが使われるようです
PDF変換時はタイトルを出力しない代わりにcover_page.docで置き換わりますので[表紙]のタイトルとサブタイトルと同じ文字にしてください。

```
= ASCIIDOCの使い方 : サブタイトル
```

5.2. アンカーの確認

サブドキュメントにアンカーを定義すると最後にincludeされたドキュメントのアンカーに飛びます
差し込み文書A
普通のアンカー 表を埋め込む

5.3. シーケンス図を埋め込むには？

以下のようなコードをadocファイルに埋め込んでおくと、UMLの画像に変換して取り込んでくれます。

- UMLサンプルコード

```
[plantuml,generated-image-format="svg"]
----
@startuml
title シーケンス図のサンプル
hide footbox

actor ユーザー as user
participant 制御 as control <<Control>>
participant "<u>共通データ</u>" as model <<Model>>
participant 画面 <<View>> #98FB98
```

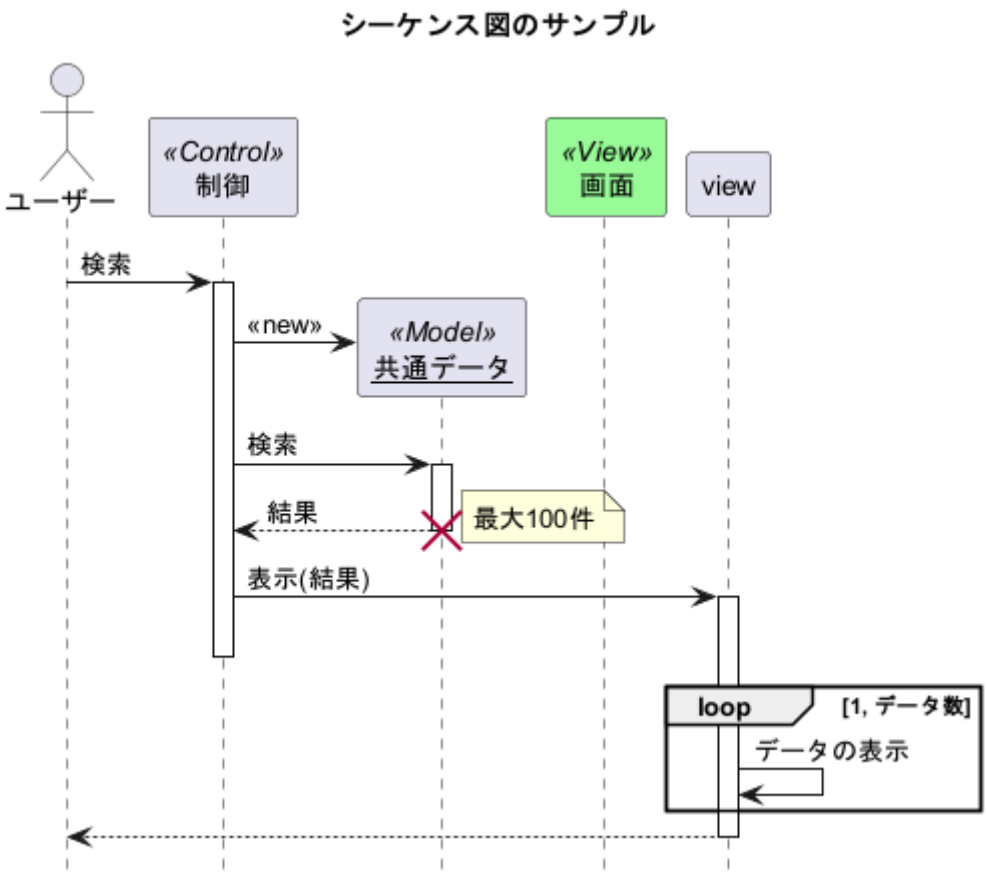
```
user -> control : 検索
activate control
create model
control -> model : << new >>
control -> model : 検索
activate model
control <-- model : 結果
note right : 最大100件
deactivate model
destroy model

control -> view : 表示(結果)
activate view
deactivate control
loop 1, データ数
 view -> view : データの表示
end
view --> user
deactivate view

@enduml
----
```

5.3.1. シーケンス図サンプル

- UMLサンプルイメージ



5.4. アイコンを指定する

```
[NOTE]
====
ノートを書きます。
====

[CAUTION]
====
注意を書きます
====

[TIP]
====
Tipを書きます
====

NOTE: NOTE

TIP: TIP

IMPORTANT: IMPORTANT

WARNING: WARNING

CAUTION: CAUTION
```


ノートを書きます。


注意を書きます


Tipを書きます


NOTE


TIP


IMPORTANT


WARNING


CAUTION

5.5. Twitterのアイコンなど

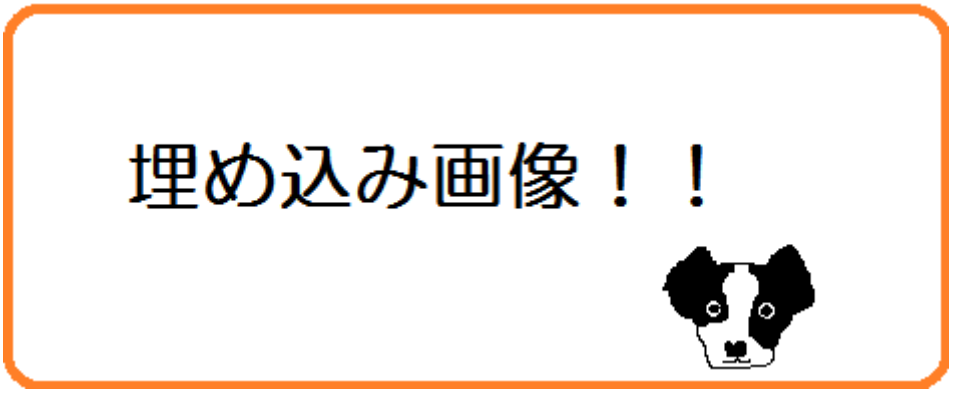
```
icon:font[]
icon:fire[]
icon:hand-stop-o[]
icon:amazon[]
[aqua]#icon:twitter[]#
```

5.6. 画像の埋め込み

画像を埋め込む場合は、ヘッダーでイメージ格納フォルダを指定して
HTML内にデータを埋め込む指定をしておく

```
// 画像フォルダの指定
:imagesdir: ./images
// イメージをHTML内に埋め込む指定
:data-uri:

// イメージを差し込む個所のサンプル
image::imagesample.png[]
```



↓ここにドキュメントが挿入されるはず

5.7. 差し込み文書A

↓で確認しているように、ドキュメントのフォルダを変えてしまうと、サブドキュメントを編集する際にimagesフォルダが見えなくなってしまうので、全てルートで管理した方が良いと思います。

5.7.1. 段落

1. あ
    a. ああ
2. い
    a. いい
        i. いいい
3. う
    a. ううサブ文書内の画像参照

- ↓と指定している場合、imagesを起点にした相対パス

```
:imagesdir: ./images
```

- ↓と指定している場合、かimagesdirが指定されていない場合はメインのadocを起点にした相対パス

```
:imagesdir: ./
```

imagesフォルダに全ての画像を集約する方向で使用することとします。
images配下にドキュメントと同じ構成のサブフォルダを作ってもいいかもしれないが、基本的にはadocのファイル名_画像名.pngのようにした方がいいと思います。
パスの指定の仕方で画像のリンクが切れてしまい埋め込まれない状態が発生するので気を付けてください

- adocと画像の配置イメージ

```
main.adoc
subdoc/subdoca.adoc
subdoc/image_b.png
subdoc/images/image_c.png
images/image_a.png
images/subdoc/image_d.png
```

- 以下の指定をする()

```
image::image_a.png[]
image::images/image_a.png[]
image::../images/image_a.png[]
image::image_b.png[]
image::../image_b.png[]
image::images/image_c.png[]
image::../images/image_c.png[]
image::image_c.png[]
image::subdoc/image_d.png[]
```

- 出力イメージ

[image a] | images/image_a.png

[image a] | ./images/image_a.png

[image b] | image_b.png

[image b] | ./image_b.png

[image c] | images/image_c.png

[image c] | ./images/image_c.png

[image c] | image_c.png

5.8. 表を埋め込む

Ascii Docでは表のセル結合やセル内での改行が出来るので、複雑な表現を実現出来ます。
また、単純な表の場合はCSVを読み込んで表にするなど選択の幅もあります。

参考) https://qiita.com/hotteam/items/80dc15012bde4d35de24

表 1. テーブル内タグ

No	項目
1	対応ブラウザ • Firefox • Chrome • edge
2	対応OS 1. Windows 2. Linux 3. Mac

```
[cols="<1,^2,>2", options="header,autowidth"]
|===========================
|左寄せ|中央寄せ|右寄せ

|左寄せ +
改行出来る
|中央寄せ +
改行出来る
|右寄せ +
改行出来る

|左寄せ +
改行出来る
2.+^.|中央寄せ +
結合も改行も出来る
* aaa
* bbb
|===========================
```

左寄せ	中央寄せ	右寄せ
左寄せ 改行出来る	中央寄せ 改行出来る	右寄せ 改行出来る

左寄せ	中央寄せ	右寄せ
左寄せ 改行出来る	中央寄せ 結合も改行も出来る aaa bbb	

A	B	C
A • A-1 • A-2 • A-3	1. aaa 2. bbb	C 改行出来る
1セルだけ ヘッダー指定	列結合	

↓ここにも同じドキュメントが挿入されるはず

5.9. 差し込み文書A

↓で確認しているように、ドキュメントのフォルダを変えてしまうと、サブドキュメントを編集する際にimagesフォルダが見えなくなってしまうので、全てルートで管理した方が良いと思います。

5.9.1. 段落

1. あ
    a. ああ
2. い
    a. いい
        i. いいい
3. う
    a. ううサブ文書内の画像参照

- ↓と指定している場合、imagesを起点にした相対パス

```
:imagesdir: ./images
```

- ↓と指定している場合、かimagesdirが指定されていない場合はメインのadocを起点にした相対パス

```
:imagesdir: ./
```

imagesフォルダに全ての画像を集約する方向で使用することとします。
images配下にドキュメントと同じ構成のサブフォルダを作ってもいいかもしれないが、基本的にはadocのファイル名_画像名.pngのようにした方がいいと思います。
パスの指定の仕方で画像のリンクが切れてしまい埋め込まれない状態が発生するので気を付けてください

- adocと画像の配置イメージ

```
main.adoc
subdoc/subdoca.adoc
subdoc/image_b.png
subdoc/images/image_c.png
images/image_a.png
images/subdoc/image_d.png
```

- 以下の指定をする()

```
image::image_a.png[]
image::images/image_a.png[]
image::../images/image_a.png[]
image::image_b.png[]
image::../image_b.png[]
image::images/image_c.png[]
image::../images/image_c.png[]
image::image_c.png[]
image::subdoc/image_d.png[]
```

- 出力イメージ

[image a] | images/image_a.png

[image a] | ./images/image_a.png

[image b] | image_b.png

[image b] | ./image_b.png

[image c] | images/image_c.png

[image c] | ./images/image_c.png

[image c] | image_c.png

5.10. アンカーの確認

サブドキュメントにアンカーを定義すると最後にincludeされたドキュメントのアンカーに飛びます

差し込み文書A

普通のアンカー 表を埋め込む