

# *Narration&Reference* WSLへのPandocのインストール

Base File Name: NarrationReference\_WSL\_Ubuntu\_Pandoc\_ja

2018.11.8

## *WSLへのPandocのインストール*

by *Shuichi Ohtsu*

---

### 動作環境

このビデオでは、*WSL(Windows Subsystem for Linux)*にインストールされたUbuntuにPandocをインストールする方法をご紹介します。

Pandocは、あるマークアップ形式で書かれた文書を他の形式へ変換するためのツールです。

カリフォルニア大学バークレー校のジョン・マクファーレーン教授(John MacFarlane)がHaskellにより開発したツールです。

対応している形式は大変多く、Markdown、HTML、LaTeX、EPUB、PDFなどにも対応しています。

ここでは、Markdown形式のファイルをPDF形式に変換することを想定して、インストールを行っていくことにします。

Pandocの優れていることは、Markdown形式のファイルのなかに、LaTeXの構文を直接書き込みができる点です。

もちろんこれを実現するためには、LaTeXもインストールしておく必要がありますので、この方法もご紹介します。

また、参考文献や相互参照の機能も追加のツール、Pandoc-citeprocやPandoc-crossrefのインストールも行います。

なお、本ビデオは、日本語のPDFファイルを作成すること想定していますので、他の言語のPDFファイルを作成する場合は、言語パッケージが必要ない場合や別の言語パッケージを組み入れる必要がありますので、その点予めご了承ください。

またPandocの概要や文法については、*Narration&Reference*ファイルのReferenceをご覧ください。

---

### Pandocのインストール

まず予め、`sudo apt update`、`sudo apt upgrade`を実行し、システムを更新しておいてください。

そして、`sudo apt install pandoc`と入力して、Pandocをインストールします。

---

## Pandoc-Citeprocのインストール

次にPandoc-citeprocをインストールします。

`sudo apt install pandoc-citeproc`と入力します。

---

## TeXのインストール

次にTeXをインストールします。

`sudo apt install texlive-luatex`と入力します。

---

## TeXの言語パッケージなどのインストール

次に各言語に関連したパッケージをインストールします。ここでは、日本語を前提としてますが、他の言語の場合は、必要に応じて、該当の言語パッケージをインストールする、ないしこのプロセスをスキップしてください。

`sudo apt install texlive-xetex texlive-lang-japanese`と入力します。

---

## Haskellのインストール

次に、Haskellをインストールします。これは、Pandoc自体がHaskellによって開発されたことによります。

`sudo apt install haskell-platform`と入力します。

---

## Pandoc-Crossrefのインストール

次に、Pandoc-crossrefをインストールするツールCabalをインストールします。

`sudo apt install cabal-install`と入力します。

---

次に、cabal関連の更新情報を取得します。

`sudo cabal update`と入力します。

---

次に、cabalを使って、pandoc-crossrefをインストールします。

`sudo cabal install --global pandoc-crossref`と入力します。

ここで`--global`を追加していることに注意してください。

`pandoc-crossref`は、フィルタとして使用しますので、この指定が必要になります。

なお、このプロセスにはかなりの時間を要します。

---

## Visual Studio Codeの起動

次に、Pandoc用のフォルダを作成することにします。

最初にコマンドラインから `cd /mnt/c` と打ち込んでWindows Cドライブに移動します。

次に、`ls` と打ち込んでCドライブの内容を表示してみます。

OKです。

---

次に、特定のディレクトリにPandoc用のディレクトリを作成します。

ここでは `__myprg` ディレクトリに `Pandoc` という名前のディレクトリを作成しました。

---

このPandocディレクトリに移動し、`code .` と入力してVisual Studio Codeを起動します。

---

## 文献リスト・スタイルのダウンロード

今回は、参考文献リストまで出力しますので、そのためのスタイルを予めダウンロードしておく必要があります。

まず、ブラウザでGitHubのスタイルサイトを開きます。

そのURLは、<https://github.com/citation-style-language/styles/> です。

この中の `chicago-author-date.csl` をクリックして、そのデータを表示し、全行をコピーします。

---

VS Codeに戻り、`chicago-author-date.csl` ファイルを新規作成し、その中にペーストします。

このファイルを保存します。

---

## Bibファイルの作成

次に参考文献用の `bib` ファイルを作成します。

ここでは、`myref01.bib` ファイルを新規作成しました。

ここに文献情報を書き込んでいきます。

その記法については、*Reference*を参照してください。

---

## サンプル用Markdownファイルの作成

次にテスト用のサンプルデータをMarkdown形式で作成します。

ここでは、`sample01.md`ファイルとして作成しました。

---

## Node.jsファイルの作成

次に、Pandocによってコンパイルをすることになりますが、たいへん多くのパラメータを指定する必要がありますので、Node.jsのスクリプトを作成することにしました。

`makepdf01.js`というファイルを新規作成し、ここに処理プロセスを記述します。

なお、このソースは、GitHubにアップロードしましたので、詳しくはそれをご覧ください。

---

## Node.jsファイルの起動

次にコマンド・ラインから、`makepdf01.js`を実行します。

`node makepdf01.js sample01`と入力します。

---

コンパイルが終了すると、新たに`sample01.pdf`ファイルが生成されます。

VS Code上で、ファイル・エクスプローラーを起動し、このPDFファイルを表示します。

---

## 生成されたPDFファイルの確認

PDFファイルが作成され、目次も作成されました。

各リンクも機能しています。

参考文献リストも生成されています。

---

## コマンドラインからのPandocの直接起動

ここまでは、Node.jsを利用したPandocによるコンパイルでしたが、Node.jsをインストールしていない場合でも、コマンド・ライン上にたくさんのパラメータを指定することにより、直接Pandocによりコンパイルすることもできます。

その場合には、

```
pandoc sample02.md -o sample02.pdf -V documentclass=ltjarticle --pdf-engine=lualatex --toc --bibliography=myref01.bib --cs1 chicago-author-date.csl --filter pandoc-crossref.
```

となります。

---

実際に入力してみます。

pdf-engine名が間違っていました。これを修正して再起動します。

今度は、sample02の拡張子が不足していました。これを修正して再起動します。

OKです。

sample02.pdfが生成されました。

---

## 生成されたPDFファイルの確認

このファイルを表示してみます。

OKです。

---

ご清聴ありがとうございました。

---

## Reference

- "Sample data and program(Ohtsu/pandoc-o2-makepdf)",  
<https://github.com/Ohtsu/pandoc-o2-makepdf>
- "citation-style-language/styles",  
<https://github.com/citation-style-language/styles/>
- "Pandoc ユーザーズガイド 日本語版",  
<http://sky-y.github.io/site-pandoc-jp/users-guide/>
- "Pandoc User's Guide",  
<https://pandoc.org/MANUAL.html>
- "WSL (Windows Subsystem on Linux) で pandoc メモ",  
<https://qiita.com/miyamiya/items/4d2e93ad7895e302c27e>
- "プログラミングPandoc",  
[https://www.amazon.co.jp/dp/4274067815/ref=sxbs\\_sxwds-stvp\\_1?pf\\_rd\\_m=AN1VRQENFRJN5&pf\\_rd\\_p=14895845-6b63-47e2-b967-96bf0ca66fcb&pd\\_rd\\_wg=ZJDGq&pf\\_rd\\_r=0649V2CWECEG2NZ0KXHKB&pf\\_rd\\_s=desktop\\_sx-bottom-](https://www.amazon.co.jp/dp/4274067815/ref=sxbs_sxwds-stvp_1?pf_rd_m=AN1VRQENFRJN5&pf_rd_p=14895845-6b63-47e2-b967-96bf0ca66fcb&pd_rd_wg=ZJDGq&pf_rd_r=0649V2CWECEG2NZ0KXHKB&pf_rd_s=desktop_sx-bottom-)

[slot&pf\\_rd\\_t=301&pd\\_rd\\_i=4274067815&pd\\_rd\\_w=zZKAt&pf\\_rd\\_i=Pandoc&pd\\_rd\\_r=7e5f7e6c7-4c7f-9e7e-a4ef2c2531f2&ie=UTF8&qid=1541321932&sr=1](https://pinkmagenta.hatenablog.jp/entry/2017/12/20/124911)

- "化学系だけ Markdown でレポートを書いて Pandoc を使った",  
<http://pinkmagenta.hatenablog.jp/entry/2017/12/20/124911>
- "Pandocで相互参照",  
<http://pinkmagenta.hatenablog.jp/entry/2017/12/20/124911>
- "MarkdownとPandocを使って論文っぽい文章を書く",  
<https://inody1991.tumblr.com/post/134742076815/markdown%E3%81%A8pandoc%E3%82%9C>
- "Windows 10 Installation Guide",  
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- "Windows 10でLinuxプログラムを利用可能にするWSLをインストールする(バージョン1803以降対応版)",  
<http://www.atmarkit.co.jp/ait/articles/1608/08/news039.html>
- "Windows Subsystem for Linuxをインストールしてみよう!",  
<https://qiita.com/AruneKo/items/c79810b0b015bebf30bb>
- "「Windows Subsystem for Linux(WSL)」セットアップガイド【スクリーンショットつき解説】",  
<https://linuxfan.info/wsl-setup-guide>
- "Angular5, Angular6, Angular7 Custom Library: Step-by-step guide",  
<https://www.udemy.com/angular5-custom-library-the-definitive-step-by-step-guide/>
- "Angular5, Angular6, Angular7用 カスタムライブラリの作成: 完全ステップ・バイ・ステップ・ガイド",  
<https://www.udemy.com/angular5-1/>