org から LaTeX や HTML へのエクスポート自分用メモ

sugayu

2024 年 9 月 20 日

1 org-mode における LaTeX 文章の書き方

org-export を使うことで org 文章を LaTeX 文章へ変換することができる。パッケージ名は `ox-latex` 。

Emacs 上における LaTeX クラスは `init.el` に自分で調整して指定済み。状況に応じて以下の LaTeX クラスを使いわける。デフォルトのクラスは 1 段組みの `jsarticle` になっている。

```
#+LATEX_CLASS: jsarticle2  # 2 段組み。長い文章のとき。
#+LATEX_CLASS: jsarticle  # 短い文章や 1 行が長いとき。
#+LATEX_CLASS: plain-article  # 英文 1 段組み
```

クラスのオプションを指定したい場合は以下のようにしてオプションを変更できる。

```
#+LATEX_CLASS_OPTIONS: [oneside,twocolumn]
```

デフォルトのコンパイラは `ptex2pdf -u -l` を使って `uplatex` を起動させている (はず)。英文のクラスにする際にはコンパイラを変更する。

```
#+LATEX_COMPILER: pdflatex
```

ヘッダーキーワードは以下のように追加できる。ここでコマンドを定義すれば `\NaID` のようにコマンドを打ったときに NaID だと LaTeX が認識できる。ただし HTML は認識できないので別の方策を考える必要がある。

```
#+LATEX_HEADER: \input{template_sub}
#+LATEX_HEADER: \subtitle{2024/09/02--2024/09/08}
```

これらのヘッダー情報は、他の書類にも共通する内容であれば `init.el` に書きこんだり `\input{}` で挿入したりして共通化するのが良い。目次 (Table of Contents) を表示したくない場合には以下を打つ。

```
#+OPTIONS: toc:nil
```

2 要素の挿入

2.1 数式

`org-mode` は LaTeX の数式の仕様を導入してくれているので、そのまま使うことができる。文章中に数式を挿入すると $\Sigma_{n=0}^{\infty}(1/x)^n < 1$ $(x = 2)$ のようになる。数式環境で挿入すると

$$F = \int_{-\infty}^{\infty} f_\nu d\nu \tag{1}$$

のようになる。式番号も式 1 のように参照することができる。このとき `#+NAME` を使ってラベルをつける必要があることに注意する。

2.2 表

`org-table` が使える。

Name	[OIII]	[NII]	[CII]
M82	1×10^{-22}	3×10^{-23}	1×10^{-23}
M101	1×10^{-22}	4×10^{-23}	1×10^{-23}

2.3 画像

画像は `org-mode` と同じ要領で入れる。`#+ATTR_HTML: :width 30%` を使えば HTML 上で各画像のサイズを変更できる。

図 1　ここにキャプションが入る。M83 は南の回転花火銀河とも呼ばれているらしい (Credit: CTIO/NOIRLab/DOE/NSF/AURA)。

ただし LaTeX では挿入した位置にのまま画像が入るとは限らない。また、画像と本文の間に 1 行空けないと LaTeX 変換時に画像をうまく認識してくれないので、画像の次の行は別段落として扱われる。引用は「画像 1」のようにやる。

2.4 コード

`org-babel` を使って Python コードが挿入できる。デフォルトでは LaTeX に出力したときソースコードは全て `verbatim` で囲まれている。そのためソースコードに装飾を施すためには工夫が必要になる。

- Exporting Code Blocks (The Org Manual)
- Results of Evaluation (The Org Manual)

コードを書いたときのアウトプットは `:exports` 設定に依る。デフォルトの `:exports` は `code` なので、Python コードは `python` 環境のときはコードがそのまま表示される。

```
import numpy as np
a = np.arange(10)
print(a)
```

`ob-ipython` でも同様。

```
import numpy as np
np.arange(10)
```

このとき ob-ipython を使って計算結果を走らせていたとしても、 :results に drawer が入っていると表示されないので注意する。

```
import numpy as np
np.arange(10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

出力パラメータが :exports results の場合にはコードが走った結果のみが表示される。:exports both の場合にはコードと出力が表示される。:results raw で org 上で出力していると、出力先には 2 重で表示されるので注意する。また、あたりまえだが、LaTeX や HTML に出力するたびにコードが走るので動作が非常に遅くなって効率が悪い。度々走らせる必要があるコード以外は先に出力しておいてそれを表示するだけの方が良い。

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```