

주피터, 스팅크스, 그리고 레이텍

HTML과 PDF를 동시에 만들기

이호재

PDF와 HTML

	PDF	HTML
페이지 크기	😊	😞
인쇄	😊	😞
검색	😊	😞
벡터 이미지	😊	😊
하이퍼링크	😊	😊
멀티미디어	😞	😊
자바스크립트	😞	😊

	PDF	HTML
만들기	😊	😞
검토하기	😊	😞
배포하기	😊	😞
훑어보기	😊	😞
열기	😞	😊

ePub은 모바일 기기를 벗어나는 순간 그 매력을 잃는다.

게다가 Mathjax를 지원하는 뷰어가 드물다.

HTML로 만들어야 하는 경우는

라따뚜이 조리법과 같이,
특정한 주제를 장황하지 않게 다룬,
종종 갱신되는 문서를,
사람들이 언제 어디에서나 접근할 수 있게 하려면?

HTML 문서를 만들려면?

- latex2html
- tex4ht
- plastex

사용자 정의 매크로를 처리할 수 있는가?

HTML 문서를 만들려면? (계속)

HTML 문서를 손수 작성할 수는 없고,
마이크로소프트 워드는 마뜩찮고,
XML은 너무 복잡하니 (응용 소프트웨어가 또한 너무 비싸고),
Markdown 같은 경량 마크업 언어가 해법이지 않을까?

마크업 언어를 사용하는

- Typora
- Multimarkdown
- Jupyter
- Sphinx
- AsciiDoc
- 3Rabbitz
- GitHub + Jekyll
 - <https://wisefour.github.io/gabe-manual/>
 - <https://github.com/jekyll/jekyll/wiki/Sites>

멀티마크다운에 대한 나의 목표는 80 퍼센트의 사람들이 작성하는 80 퍼센트의 문서들에 유용하게 쓰이는 것이다. ... 멀티마크다운 문서에 열 다섯 단어마다 토끼를 그리는 기능을 멀티마크다운이 갖고 있다면 자신의 삶이 완성될 수 있는 누군가가 저 어딘가에 있음을 나도 잘 안다. 그러나 그 기능은 지구 상에 다른 모든 이들에게 전혀 쓸모없을 것이다. 나는 토끼들을 그리도록 멀티마크다운을 프로그래밍하지 않을 것이다.

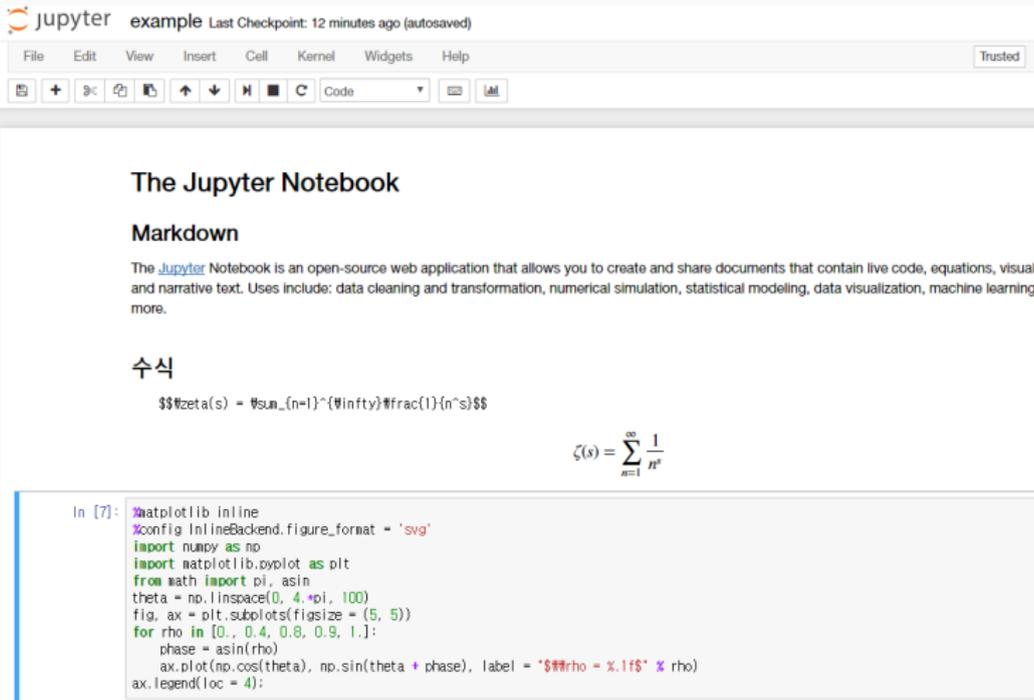
Fletcher Penney

왜 경량 마크업 언어를 사용하는가?

*HTML*과 *CSS*를 모르더라도 *HTML* 문서를 만들고,
레이텍을 모르더라도 *PDF* 문서를 만들겠다는
Minimalism?

연비가 좋으면서 빠른 차를 기대하는 것은 기술적 문제라기보다 논리적 모순!

주피터란?



Jupyter example Last Checkpoint: 12 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted

+

The Jupyter Notebook

Markdown

The [Jupyter](#) Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visual and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning more.

수식

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

```
In [7]: %matplotlib inline
%config InlineBackend.figure_format = 'svg'
import numpy as np
import matplotlib.pyplot as plt
from math import pi, asin
theta = np.linspace(0, 4.*pi, 100)
fig, ax = plt.subplots(figsize = (5, 5))
for rho in [0., 0.4, 0.8, 0.9, 1.]:
    phase = asin(rho)
    ax.plot(np.cos(theta), np.sin(theta + phase), label = '$\rho = %.1f$' % rho)
ax.legend(loc = 4):
```

- Markdown
- 파이썬 코드와 출력
- reST 같은 다른 언어로 작성하는 것을 허용

File > Download as > HTML

File > Download as > PDF via LaTeX

포맷 변환의 관점에서 레이텍은 PDF를 만들어내는 거의 유일한 실질적 표준이다.

주피터 노트북을 PDF로 만들 때 발생하는 문제들

- SVG: Inkscape (MSI 버전을 설치해야)

- 한글

```
.../site-packages/nbconvert/templates/latex/base.tplx  
\usepackage{kotex} % <- 추가  
\usepackage[T1]{fontenc}
```

- 표

- 상호참조: 표, 그림, 수식

Jinja: 템플릿 엔진

```
((*- if nb.metadata.latex.language -*))  
\documentclass[10pt, language=((( nb.metadata.latex.language )))]{hzguide}  
((*- else -*))  
\documentclass[10pt, language=korean]{hzguide}  
((*- endif -*))  
\usepackage{myjupyter}  
  
((* block title *))  
((*- if nb.metadata.latex.title -*))  
\title{((( nb.metadata.latex.title )))}  
((*- else -*))  
\title{((( resources.metadata.name | ascii_only | escape_latex )))}  
((*- endif -*))  
((* endblock title *))
```

주피터를 위한 레이텍 스타일

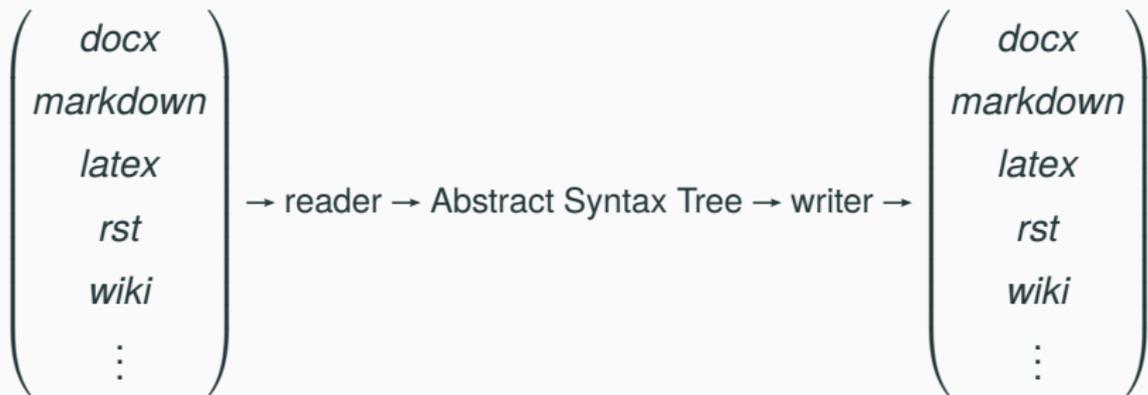
```
\DeclareDocumentCommand \includegraphics { o m }
{
  \tl_set:Nn \l_tmpa_tl {#2}
  \hbox_set:Nn \l_image_box{
    \OrgIncludegraphics[scale=\ImageScale]{\l_tmpa_tl}
  }
  \dim_set_towidth:Nn \l_tmpa_dim { \box_use:N \l_image_box }
  \dim_compare:nTF { \l_tmpa_dim > \linewidth }
  {
    \OrgIncludegraphics[width=\linewidth]{\l_tmpa_tl}
  }{
    \object_locate:N \l_tmpa_dim
    \box_use:N \l_image_box
  }
}
```

주피터 한번 해보자!

Anaconda

```
> pip install --upgrade jupyter
> pip install --upgrade nbconvert
> jupyter notebook
> jupyter nbconvert --to=latex --template=foo.tplx foo.ipynb
```

파이썬 배포판에 팬독(Pandoc)이 포함되어 있지 않다면, 팬독을 별도로 설치해야 한다.



책을 만들고 싶은데

그 형태가 PDF이든 ePub이든, 책을 만든다 함은

- 여러 주제의 글들을 조직화하는 것이다.
- 그래서 “차례”가 있어야 하고,
- 저자를 비롯한 출판 정보도 나타내야 하고,
- 필요하다면 색인이나 참고문헌도 넣어야 한다.

스핑크스: reStucturedText

Introduction

```
.. table:: Duis aute irure dolor in reprehenderit
   :name: tab-bb
```

```
=====
Lorem      Ipsum      Dolor
=====
Sit         Amet        Consectetur adipiscing elit, sed do
Quis       Nostrud     Exercitation ullamco laboris nisi
Ut         Aliquip     Ex ea commodo consequat.b
=====
```

See :numref:`tab-bb`

레이텍의 `\include{}` 명령에 해당하는

```
.. toctree::  
    :maxdepth: 3  
    :numbered:  
  
    foo.rst  
    foo.md  
    foo.ipynb  
    .  
    .  
    .  
    bibliography.rst
```

스핑크스로 HTML을 만들면

- 다양한 HTML 테마가 제공된다.
- 텍스트 검색이 가능하다.
- 색인도 만들 수 있다.
- 심지어 참고문헌도 만들 수 있다. (bib 파일을 지원한다.)
- 그리고 독스트링 (DocString)을 빼놓을 수 없다.

독스트링: foo.py

```
def example_generator(n):
    """Generators have a ``Yields`` section instead of a ``Returns`` section.

    Parameters
    -----
    n : int
        The upper limit of the range to generate, from 0 to `n` - 1.

    Yields
    -----
    int
        The next number in the range of 0 to `n` - 1.

    Examples
    -----
    Examples should be written in doctest format, and should illustrate how
    to use the function.

    >>> print([i for i in example_generator(4)])
    [0, 1, 2, 3]

    """
    for i in range(n):
        yield i
```

```
.. automodule:: foo
   :members:
   :undoc-members:
   :inherited-members:
   :show-inheritance:
```

스핑크스로 레이텍을 만들때

- 한글
- SVG
- 레이텍 컴파일은 알아서 하라?
 - > sphinx-build.exe -M html
 - > sphinx-build.exe -M latex

스핑크스를 위한 레이텍 템플릿

```
\documentclass[  
    language=<%= language if language is defined else 'korean' %>,  
    pairquote=false  
]{hzguide}  
\usepackage{mysphinx}
```

스핑크스를 위한 레이텍 스타일

```
\DeclareDocumentCommand \sphinxincludegraphics { o m }
{
  \group_begin:
  \tl_set:Nn \l_tmpa_tl {##2}
  \tl_replace_once:Nnn \l_tmpa_tl {.svg}{.pdf}
  \tl_if_in:NnTF \l_tmpa_tl {.pdf}
  {
    \par\image[scale=\g_image_scale_vector]{\l_tmpa_tl}
  }{
    \par\image[scale=\g_image_scale_bitmap]{\l_tmpa_tl}
  }
  \group_end:
}
```

레이텍 컴파일을 위한 래퍼 프로그램

```
>docbuild.exe latex -t foo.tex -j -f
```

```
if os.path.exists('images'):  
    cmd = 'svg2pdf.exe images/*.svg'  
    os.system(cmd)
```

```
if args.AutoJosa:  
    cmd = 'autojosa.exe ' + tex  
    os.system(cmd)
```

유념해야 할 것은

토끼 같은 것들에 목매지 마시라! 주피터를 (또는 주피터와 함께 스프링크스를) 사용할 때 토끼가 될 법한 것들:

- 표
- 표, 그림, 수식의 상호참조
- SVG