sphinxcontrib-katex

Release 0.4.1

Hagen Wierstorf

Jan 08, 2019

Contents

CHAPTER 1

Usage

Installation:

```
pip install sphinxcontrib-katex
```

In `conf.py` of your sphinx project, add the extension with:

```
extensions = ['sphinxcontrib.katex']
```

CHAPTER 2

Configuration

The behavior of `sphinxcontrib.katex` can be changed by configuration entries in `conf.py` of your documentation project. In the following all configuration entries are listed and their default values are shown.

```
katex_css_path = \
    'https://cdn.jsdelivr.net/npm/katex@0.10/dist/katex.min.css'
katex_js_path = \
    'https://cdn.jsdelivr.net/npm/katex@0.10/dist/katex.min.js'
katex_autorender_path = \
    'https://cdn.jsdelivr.net/npm/katex@0.10/contrib/auto-render.min.js'
katex_inline = [r'\(', r'\)']
katex_display = [r'\[', r'\]']
katex_options = ''
```

The specific delimiters written to HTML when math mode is encountered are controlled by the two lists `katex_inline` and `katex_display`.

The string variable `katex_options` allows you to change all available official KaTeX rendering options, e.g.

```
katex_options = r'''{
    displayMode: true,
    macros: {
        "\\RR": "\\mathbb{R}"
    }
}'''
```

You can also add KaTeX auto-rendering options to `katex_options`, but be aware that the `delimiters` entry should contain the entries of `katex_inline` and `katex_display`.

CHAPTER 3

LaTeX Macros

Most probably you want to add some of your LaTeX math commands for the rendering. In KaTeX this is supported by LaTeX macros (`\def`). You can use the `katex_options` configuration setting to add those:

```
katex_options = r'''macros: {
        "\\i": "\\mathrm{i}",
        "\\e": "\\mathrm{e}^{#1}",
        "\\vec": "\\mathbf{#1}",
        "\\x": "\\vec{x}",
        "\\d": "\\operatorname{d}\\!{}",
        "\\dirac": "\\operatorname{\\delta}\\left(#1\\right)",
        "\\scalarprod": "\\left\\langle#1,#2\\right\\rangle",
    }'''
```

The disadvantage of this option is that those macros will be only available in the HTML based Sphinx builders. If you want to use them in the LaTeX based builders as well you have to add them as the `latex_macros` setting in your `conf.py` and specify them using proper LaTeX syntax. Afterwards you can include them via the `sphinxcontrib.katex.latex_defs_to_katex_macros` function into `katex_options` and add them to the LaTeX preamble:

```
import sphinxcontrib.katex as katex

latex_macros = r"""
    \def \i                 {\mathrm{i}}
    \def \e               #1{\mathrm{e}^{#1}}
    \def \vec             #1{\mathbf{#1}}
    \def \x                 {\vec{x}}
    \def \d                 {\operatorname{d}\!}
    \def \dirac           #1{\operatorname{\delta}\left(#1\right)}
    \def \scalarprod    #1#2{\left\langle#1,#2\right\rangle}
"""

# Translate LaTeX macros to KaTeX and add to options for HTML builder
katex_macros = katex.latex_defs_to_katex_macros(latex_macros)
katex_options = 'macros: {' + katex_macros + '}'

# Add LaTeX macros for LATEX builder
latex_elements = {'preamble': latex_macros}
```

CHAPTER 4

Math Rendering Examples

The examples start always with a code box showing the commands, which is followed by the resulting Sphinx output.

4.1 Inline math

```
Some inline math :math:`x_1 + x_2 + ... + x_n, n \in \mathbb{Z}`,
followed by text.
```

Some inline math $x_1 + x_2 + ... + x_n, n \in \mathbb{Z}$, followed by text.

4.2 Macros

You can define macros directly in your math directive.

```
.. math::

    \def \x {\mathbf{x}}
    \def \w {\omega}
    \def \d {\operatorname{d}\!}

    P(\x,\w) = \oint_{\partial V} D(\x_0,\w) G(\x-\x_0,\w) \d A(\x_0)
```

$$P(\mathbf{x},\omega) = \oint_{\partial V} D(\mathbf{x}_0,\omega) G(\mathbf{x}-\mathbf{x}_0,\omega) \operatorname{d}\!A(\mathbf{x}_0)$$

If you want to use them in the whole document, the best is to define them in `conf.py` as part of the `katex_options`, see *LaTeX Macros*. Afterwards, you can use them in every math directive.

4.3 Aligned environment

```
.. math::

    \begin{aligned}
        \dot{x} & = \sigma(y-x) \\
        \dot{y} & = \rho x - y - xz \\
        \dot{z} & = -\beta z + xy
    \end{aligned}
```

$$\begin{aligned}\dot{x} &= \sigma(y-x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

4.4 Array environment

```
.. math::

    \begin{array}{c:c:c:c:c:c}
        \Gamma & \Delta & \Theta & \Lambda & \Xi & \Pi \\ \hdashline
        \gamma & \delta & \theta & \lambda & \xi & \pi
    \end{array}
```

$$\begin{array}{c:c:c:c:c:c}\Gamma & \Delta & \Theta & \Lambda & \Xi & \Pi \\ \hdashline \gamma & \delta & \theta & \lambda & \xi & \pi\end{array}$$

4.5 Case definitions

```
.. math::

    f(n) = \begin{cases}
        \frac{n}{2}, & \text{if } n\text{ is even} \\
        3n+1,        & \text{if } n\text{ is odd}
    \end{cases}
```

$$f(n) = \begin{cases}\frac{n}{2}, & \text{if } n\text{ is even} \\ 3n+1, & \text{if } n\text{ is odd}\end{cases}$$

4.6 Matrices

A simple matrix defined with the `pmatrix` environment:

```
.. math::

    \begin{pmatrix}
        a_{11} & a_{12} & a_{13}\\
        a_{21} & a_{22} & a_{23}\\
        a_{31} & a_{32} & a_{33}
    \end{pmatrix}
```

$$\begin{pmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{pmatrix}$$

The `pmatrix*` environment is not available, but you can use the `array` environment for more complex matrices:

```
.. math::

    \def \msum {-\textstyle\sum}
    \def \psum {\phantom{-}\textstyle\sum}
    I_{ik} = \left(
    \begin{array}{lll}
        \psum m (y^2+z^2) & \msum m x y       & \msum m x z          \\
        \msum m y x       & \psum m (x^2+z^2) & \msum m y z          \\
        \msum m z x       & \msum m z y       & \psum m (x^2 + y^2)
    \end{array}
    \right)
```

$$I_{ik} = \left(\begin{array}{lll} \sum m(y^2+z^2) & -\sum mxy & -\sum mxz \\ -\sum myx & \sum m(x^2+z^2) & -\sum myz \\ -\sum mzx & -\sum mzy & \sum m(x^2+y^2) \end{array} \right)$$

CHAPTER 5

Contributing

If you find errors, omissions, inconsistencies or other things that need improvement, please create an issue or a pull request at https://github.com/hagenw/sphinxcontrib-katex/. Contributions are always welcome!

5.1 Development Installation

Instead of pip-installing the latest release from PyPI, you should get the newest development version from Github:

```
git clone https://github.com/hagenw/sphinxcontrib-katex.git
cd sphinxcontrib-katex
python setup.py develop --user
```

This way, your installation always stays up-to-date, even if you pull new changes from the Github repository.

If you prefer, you can also replace the last command with:

```
pip install --user -e .
```

... where `-e` stands for `--editable`.

5.2 Building the Documentation

If you make changes to the documentation, you can re-create the HTML pages using Sphinx. You can install it and a few other necessary packages with:

```
pip install -r doc/requirements.txt --user
```

To create the HTML pages, use:

```
python setup.py build_sphinx
```

The generated files will be available in the directory `build/sphinx/html/`.

It is also possible to automatically check if all links are still valid:

```
python setup.py build_sphinx -b linkcheck
```

5.3 Running the Tests

You'll need pytest for that. It can be installed with:

```
pip install -r tests/requirements.txt --user
```

To execute the tests, simply run:

```
python -m pytest
```

5.4 Creating a New Release

New releases are made using the following steps:

1. Bump version number in `sphinxcontrib/katex.py`
2. Update `NEWS.rst`
3. Commit those changes as "Release x.y.z"
4. Create an (annotated) tag with `git tag -a x.y.z`
5. Clear the `dist/` directory
6. Create a source distribution with `python setup.py sdist`
7. Create a wheel distribution with `python setup.py bdist_wheel --universal`
8. Check that both files have the correct content
9. Upload them to PyPI with twine: `twine upload dist/*`
10. Push the commit and the tag to Github and add release notes containing a link to PyPI and the bullet points from `NEWS.rst`
11. Check that the new release was built correctly on RTD, delete the "stable" version and select the new release as default version

CHAPTER 6

Version History

Version 0.4.1 (2019-01-08):

- Fix macros example in documentation

Version 0.4.0 (2018-12-14):

- KaTeX version 0.10.0
- Remove configuration option katex_version
- Add Sphinx documentation and setup RTD page
- Add Travis-CI tests
- Make compatible with Sphinx>=1.6

Version 0.3.1 (2018-10-08):

- Fix incompatibility with sphinx>=1.8 (#8)

Version 0.3.0 (2018-09-06):

- Allow for user defined autorendering delimiters (#7)
- Fix bug if katex_options was blank (#5)

Version 0.2.0 (2018-06-22):

- Remove katex_macros option
- Document all configuration settings
- Automatic setting of delimiters for KaTeX auto-renderer

Version 0.1.6 (2018-04-12):

- Equation numbering across pages with sphinx>=1.7
- KaTeX version 0.9.0

Version 0.1.5 (2017-12-19):

- Improvement of code readability
- Fix mouse over for equation numbers in Firefox
- Add helper function to convert LaTeX defs to KaTeX macros

Version 0.1.4 (2017-11-27):

- Move equation numbers to the right and center vertically

Version 0.1 (2017-11-24):

- Initial release