
Utilities for authoring SageMath packages

Release 0.0.7

Nicolas M. Thiéry et al.

Feb 10, 2019

Contents

1	setuptools utilities for SageMath packages	1
2	A Sphinx extension for writing SageMath-style documentation	3
3	Sphinx extension demo	5
4	Live SageMath examples in static HTML pages with Thebe and binder	7
4.1	In a nutshell	7
4.2	Components	7
4.3	For authors	8
4.4	Thebe versus SageMath Cell	9
5	Indices and tables	11
	Python Module Index	13

setuptools utilities for SageMath packages

setuptools utilities for SageMath packages

```
class sage_package.setuptools.SageTest (dist, **kw)  
    Bases: setuptools.command.test.test  
    A TestCommand for running doctests with sage -t  
    run_tests ()
```

A Sphinx extension for writing SageMath-style documentation

A Sphinx extension for writing SageMath-style documentation

This extension sets up: - the styling theme - intersphinx: enables crosslinks to the Python and SageMath documentation
- extlinks: short hand roles for wikipedia, trac, arxiv, ...

```
sage_package.sphinx.setup(app)
```

Initialize this Sphinx extension

```
sage_package.sphinx.themes_path()
```

Retrieve the location of the themes directory from the location of this package

This is taken from Sphinx's theme documentation

Sphinx extension demo

This page illustrates some of the features provided by the Sphinx extension for writing documentation in the Sage's official documentation style:

- Cross-references to Sage's official documentation:
 - [Welcome to the Sage Thematic Tutorials!](#)
 - [Introduction to combinatorics in Sage](#)
 - `sage.functions.other.binomial()`
- Cross-references to Python's official documentation:
 - [Command Reference](#)
 - `os`
 - `slice()`
- Code blocks:

Assuming internet access, the page can be activated, making the following code blocks into live cells that can be modified and evaluated:

```
sage: 1 + 1
2
sage: y = 1

sage: z = 2
sage: y + z
3
```

LaTeX output:

```
sage: %display latex
sage: factor(x^100 - 1)
(x - 1)*(x + 1)*(x^2 + 1)*(x^4 - x^3 + x^2 - x + 1)*(x^4 + x^3 + x^2 + x + 1)*(x^8 - x^6 + x^4 - x^2 + 1)*(x^20 - x^15 + x^10 - x^5 + 1)*(x^20 + x^15 + x^10 + x^5 + 1)*(x^40 - x^30 + x^20 - x^10 + 1) (continues on next page)
```

(continued from previous page)

Plots:

```
sage: plot(sin(x))
```

3D plots are functional with the `threejs` backend:

```
sage: x,y = SR.var('x,y')
sage: plot3d(x^2 + y^2, (x,-2,2), (y,-2,2), viewer='threejs', online=True)
```

but not yet with the default `jmol` backend:

```
sage: plot3d(x^2 + y^2, (x,-2,2), (y,-2,2))
```

Interacts (non functional yet):

```
sage: @interact
....: def f(n=31):
....:     return factor(n)
```

Plain code blocks (i.e. not containing a sage prompt) are not affected:

```
> sage --notebook jupyter
```

- TODO's:

Todo: Fix the failing cross-references above!

- Mathematics:

- $\frac{\pi}{1+10^4}$
- Usual Sage macros , , ,

- Miscellaneous roles:

- [Wikipedia article SageMath](#)
- [trac ticket #12214](#)
- [OEIS sequence A000108](#)
- `pari:factor`
- [Arxiv 1412.4765](#)
- [doi:10.1088/1742-6596/600/1/012002](#)

Live SageMath examples in static HTML pages with Thebe and binder

Web pages may include live SageMath examples that can be edited and executed online after clicking `Activate`. If you read this page, you may have followed the `About` link of some of them:

- The documentation of some SageMath package, typically using the `sage-package` tools. See e.g. [this demo page](#), and click `Activate`.
- [More SageMath tutorials](#)
- (in the work) The [SageMath documentation](#), either online or locally.

We provide here some background about the Jupyter-based service behind the scene, for readers and authors. A similar service is provided by [SageMathCell](#), which see for background. A brief comparison is provided below.

4.1 In a nutshell

Live code examples are handled by the [Thebe](#) javascript library, configured to run on [mybinder.org](#), using the latest version of [SageMath](#). See below for details on those free and open source components.

The `binder` service is meant for casual use. As for any cloud-based service, don't run calculations that could leak private information. See the [binder faq](#) for details.

Documentation pages produced with `sage-package` take further steps to reduce the dependence on network and cloud services. If the page is served by a Jupyter server, it will attempt to use it to fetch the Thebe javascript and run the computations.

4.2 Components

[SageMath](#) is a general purpose open source software for mathematical computations.

[Jupyter](#) is a project that fosters open-source software, open-standards, and services for interactive computing across dozens of programming languages, including SageMath.

[JupyterLab](#) is the next-generation web-based user interface for Project Jupyter, meant to phase out soon the classic Jupyter notebook.

[Thebe](#) is a small javascript library based on [JupyterLab](#) that enables embedding live code examples in an HTML page. Upon clicking *Run*, the code is sent to a Jupyter server for execution and the result displayed back.

[Binder](#) is a free and open source service for temporary computations within an arbitrary execution environment; [mybinder.org](#) is binder's original instance. The execution environment is described by metadata files hosted on a git repository.

For SageMath, the execution environment is described by default by this simple [Dockerfile](#), which uses the latest [SageMath Docker container](#).

4.3 For authors

Setting up Thebe+Binder+SageMath for you own HTML pages boils down to adding the following header to each page:

```
<!-- Thebe configuration !-->
<script type="text/x-thebe-config">
  thebeConfig = {
    binderOptions: {
      repo: "sagemath/sage-binder-env",
    },
    stripPrompts: {
      inPrompt: 'sage: ',
      continuationPrompt: '....: ',
      selector: 'pre:contains("sage: " )',
    },
    kernelOptions: {
      name: "sagemath",
    },
    requestKernel: true
  }
</script>
<!-- Load the Thebe library !-->
<script type="text/javascript" src="https://unpkg.com/thebelab@0.1.0" ></script>
<!-- Activate Thebe directly upon loading the page !-->
<script>window.onload = function() { thebelab.bootstrap(); };</script>
```

and writing the examples as:

```
<pre>
  sage: 1+1
  2
</pre>
```

For further customization, see also the source of the [Thebe examples](#).

For the Sphinx-generated documentation of a Sage based project, you can use the Sphinx extension provided by the [Sage package authoring utilities](#). It includes some additional styling and goodies, including the activate button and running computations locally when possible.

If you need to customize the computation environment, for example by installing additional packages, you can create your own Dockerfile, based on the above, and hosted in some github repository. See this [example](#).

4.4 Thebe versus SageMath Cell

Thebe is similar in principle to [SageMath Cell](#). It introduces additional flexibility by enabling the customization of the programming language (kernel), computing backend (e.g. a local Jupyter server, ...) and executable environment (e.g. via binder). It also targets a much broader community, with the potential to relieve the SageMath community from maintaining a custom solution. On the other hand it's still a relatively recent and quickly evolving technology with less settled sustainability. Also the SageMath Cell has been optimized to be more reactive on startup and reduce resource consumption. Those optimizations have not yet been ported to Thebe+binder.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`sage_package.setuptools`, 1
`sage_package.sphinx`, 3

R

`run_tests()` (`sage_package.setuptools.SageTest` method), 1

S

`sage_package.setuptools` (module), 1

`sage_package.sphinx` (module), 3

`SageTest` (class in `sage_package.setuptools`), 1

`setup()` (in module `sage_package.sphinx`), 3

T

`themes_path()` (in module `sage_package.sphinx`), 3