py$_toolsDocumentation$

Release 0.1.0

Thorsten Beier

Jul 30, 2019

CONTENTS:

CHAPTER

ONE

PY_TOOLS

1.1 Features

Current features include:

- conda ready
- pytest unit test
- continous integration
- coverall code coverage
- documentation with sphinx
- documentation on readthedocs

CHAPTER TWO

INSTALLATION

todo

CHAPTER THREE

USAGE

To use py_tools in a project:

```
import py_tools
```

CHAPTER

FOUR

EXAMPLES

Note: Click *here* to download the full example code

4.1 This is my example script

This example doesn't do much, it just makes a simple plot

Out:

```
my variable is 2
my variable plus 2 is 4
```

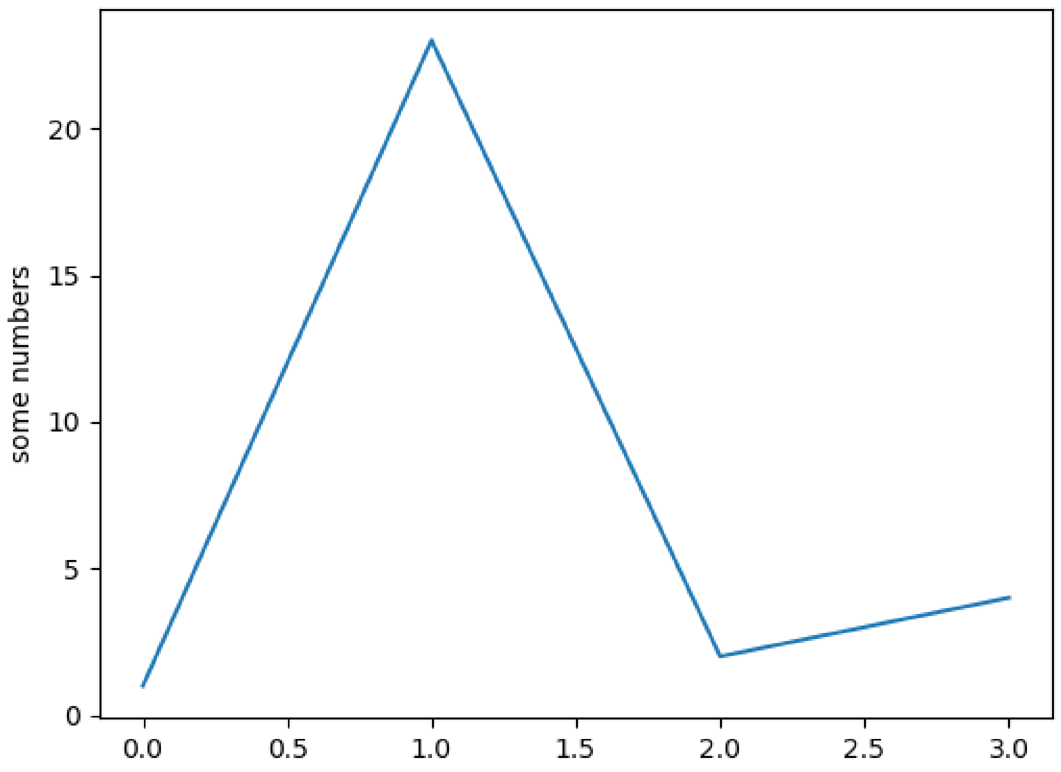
```
#%%
# This is a section header
# ------------------------
# This is the first section!
# The `#%%` signifies to Sphinx-Gallery that this text should be rendered as
# rST and if using one of the above IDE/plugin's, also signifies the start of a
# 'code block'.

import pytools

# This line won't be rendered as rST because there's a space after the last block.
myvariable = 2
print("my variable is {}".format(myvariable))
# This is the end of the 'code block' (if using an above IDE). All code within
# this block can be easily executed all at once.

#%%
# This is another section header
# ------------------------------
#
# In the built documentation, it will be rendered as rST after the code above!
# This is also another code block.

print('my variable plus 2 is {}'.format(myvariable + 2))
```

Total running time of the script: (0 minutes 0.017 seconds)

Note: Click *here* to download the full example code

4.2 This is my example script

This example doesn't do much, it just makes a simple plot



Out:

```
my variable is 2
my variable plus 2 is 4
```

```
#%%
# This is a section header
# ------------------------
# This is the first section!
```

(continues on next page)

(continued from previous page)

```
# The `#%%` signifies to Sphinx-Gallery that this text should be rendered as
# rST and if using one of the above IDE/plugin's, also signifies the start of a
# 'code block'.

import py_tools

# This line won't be rendered as rST because there's a space after the last block.
myvariable = 2
print("my variable is {}".format(myvariable))
# This is the end of the 'code block' (if using an above IDE). All code within
# this block can be easily executed all at once.

#%%
# This is another section header
# ------------------------------
#
# In the built documentation, it will be rendered as rST after the code above!
# This is also another code block.

print('my variable plus 2 is {}'.format(myvariable + 2))





#%%
# This is another section header
# ------------------------------
#
# Plots look nice in examples
import matplotlib.pyplot as plt

plt.plot([1,23,2,4])
plt.ylabel('some numbers')

plt.show()
```

Total running time of the script: (0 minutes 0.105 seconds)

Note: Click *here* to download the full example code

CHAPTER

FIVE

THIS IS MY EXAMPLE SCRIPT

This example doesn't do much, it just makes a simple plot

Out:

```
my variable is 2
my variable plus 2 is 4
```

```python
#%%
# This is a section header
# ------------------------
# This is the first section!
# The `#%%` signifies to Sphinx-Gallery that this text should be rendered as
# rST and if using one of the above IDE/plugin's, also signifies the start of a
# 'code block'.

import pytools

# This line won't be rendered as rST because there's a space after the last block.
myvariable = 2
print("my variable is {}".format(myvariable))
# This is the end of the 'code block' (if using an above IDE). All code within
# this block can be easily executed all at once.

#%%
# This is another section header
# ------------------------------
#
# In the built documentation, it will be rendered as rST after the code above!
# This is also another code block.

print('my variable plus 2 is {}'.format(myvariable + 2))
```

Total running time of the script: (0 minutes 0.018 seconds)

CHAPTER SIX

API

6.1 py_tools

6.1.1 py_tools package

Subpackages

py_tools.cli package

Submodules

py_tools.cli.main module

Module contents

Submodules

py_tools.version module

Module contents

`py_tools.`**`pure_python`**`()`

hello

CHAPTER

SEVEN

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

7.1 Types of Contributions

7.1.1 Report Bugs

Report bugs at https://github.com/DerThorsten/py_tools/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

7.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

7.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

7.1.4 Write Documentation

py_tools could always use more documentation, whether as part of the official py_tools docs, in docstrings, or even on the web in blog posts, articles, and such.

7.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/DerThorsten/py_tools/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

7.2 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function / classes with a proper documentation, and add the feature to the list in README.rst.

CHAPTER

EIGHT

CREDITS

8.1 Development Lead

- Thorsten Beier <derthorstenbeier@gmail.com>

8.2 Contributors

None yet. Why not be the first?

CHAPTER

NINE

HISTORY

9.1 0.1.0 (2019-07-30)

- First release on PyPI.

CHAPTER TEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX