

---

# **pyshields Documentation**

*Release unknown*

**ShixiangWang**

**Dec 24, 2019**



---

# Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	pyshields . . . . .	3
1.2	License . . . . .	3
1.3	Contributors . . . . .	4
1.4	Changelog . . . . .	4
1.5	pyshields . . . . .	4
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



This is the documentation of **pysields**.

---

**Note:** This is the main page of your project's [Sphinx](#) documentation. It is formatted in [reStructuredText](#). Add additional pages by creating `rst`-files in `docs` and adding them to the `toc` tree below. Use then [references](#) in order to link them from this page, e.g. [Contributors](#) and [Changelog](#).

It is also possible to refer to the documentation of other Python packages with the [Python domain syntax](#). By default you can reference the documentation of [Sphinx](#), [Python](#), [NumPy](#), [SciPy](#), [matplotlib](#), [Pandas](#), [Scikit-Learn](#). You can add more by extending the `intersphinx_mapping` in your `Sphinx`'s `conf.py`.

The pretty useful extension [autodoc](#) is activated by default and lets you include documentation from docstrings. Docstrings can be written in [Google style](#) (recommended!), [NumPy style](#) and [classical style](#).

---



## 1.1 pyshields

pyshields is a project inspired by <https://shields.io/> and badger.

### 1.1.1 Description

A longer description of your project goes here...

Related projects:

- <https://github.com/google/pybadges>
- <https://github.com/andrewp-as-is/shields.py>

### 1.1.2 Note

This project has been set up using PyScaffold 3.2.2. For details and usage information on PyScaffold see <https://pyscaffold.org/>.

## 1.2 License

The MIT License (MIT)

Copyright (c) 2019 ShixiangWang

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.3 Contributors

- ShixiangWang <wangshx@shanghaitech.edu.cn>

## 1.4 Changelog

### 1.4.1 Version 0.1

- Feature A added
- FIX: nasty bug #1729 fixed
- add your changes here!

## 1.5 pyshields

### 1.5.1 pyshields package

#### Submodules

#### pyshields.skeleton module

This is a skeleton file that can serve as a starting point for a Python console script. To run this script uncomment the following lines in the [options.entry\_points] section in setup.cfg:

```
console_scripts = fibonacci = pyshields.skeleton:run
```

Then run *python setup.py install* which will install the command *fibonacci* inside your current environment. Besides console scripts, the header (i.e. until `_logger...`) of this file can also be used as template for Python modules.

Note: This skeleton file can be safely removed if not needed!

```
pyshields.skeleton.fib(n)
```

Fibonacci example function

**Parameters** `n` (*int*) – integer

**Returns** n-th Fibonacci number

**Return type** `int`

```
pyshields.skeleton.main(args)
```

Main entry point allowing external calls



**Parameters** `args` (*[str]*) – command line parameter list

`pyshields.skeleton.parse_args` (*args*)

Parse command line parameters

**Parameters** `args` (*[str]*) – command line parameters as list of strings

**Returns** command line parameters namespace

**Return type** `argparse.Namespace`

`pyshields.skeleton.run` ()

Entry point for console\_scripts

`pyshields.skeleton.setup_logging` (*loglevel*)

Setup basic logging

**Parameters** `loglevel` (*int*) – minimum loglevel for emitting messages

## Module contents



## CHAPTER 2

---

### Indices and tables

---

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



**p**

`pyshields`, 5

`pyshields.skeleton`, 4



## F

`fib()` (*in module `pyshields.skeleton`*), 4

## M

`main()` (*in module `pyshields.skeleton`*), 4

## P

`parse_args()` (*in module `pyshields.skeleton`*), 5

`pyshields` (*module*), 5

`pyshields.skeleton` (*module*), 4

## R

`run()` (*in module `pyshields.skeleton`*), 5

## S

`setup_logging()` (*in module `pyshields.skeleton`*), 5