

---

**interdoc**  
*Release 0.1.0*

**Oct 03, 2019**



---

## Contents

---

<b>1 Overview</b>	<b>1</b>
1.1 Installation . . . . .	1
1.2 Documentation . . . . .	1
1.3 Development . . . . .	1
<b>2 Installation</b>	<b>3</b>
<b>3 Usage</b>	<b>5</b>
<b>4 Reference</b>	<b>7</b>
4.1 interdoc package . . . . .	7
<b>5 Contributing</b>	<b>9</b>
5.1 Bug reports . . . . .	9
5.2 Documentation improvements . . . . .	9
5.3 Feature requests and feedback . . . . .	9
5.4 Development . . . . .	10
<b>6 Authors</b>	<b>13</b>
<b>7 Changelog</b>	<b>15</b>
7.1 0.1.0 (2019-10-03) . . . . .	15
<b>8 Indices and tables</b>	<b>17</b>
<b>Python Module Index</b>	<b>19</b>
<b>Index</b>	<b>21</b>



# CHAPTER 1

---

## Overview

---

docs	
tests	
package	

Interactive docs

- Free software: MIT License

## 1.1 Installation

```
pip install interdoc
```

## 1.2 Documentation

<https://interdoc.readthedocs.io/>

## 1.3 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>

## CHAPTER 2

---

### Installation

---

At the command line:

```
pip install interdoc
```





## CHAPTER 3

---

### Usage

---

To use interdoc in a project:

```
import interdoc
```



## 4.1 interdoc package

### 4.1.1 Submodules

### 4.1.2 interdoc.app module

```
interdoc.app.main()
```

### 4.1.3 interdoc.cli module

Module that contains the command line app.

Why does this file exist, and why not put this in `__main__`?

You might be tempted to import things from `__main__` later, but that will cause problems: the code will get executed twice:

- When you run `python -minterdoc` python will execute `__main__.py` as a script. That means there won't be any `interdoc.__main__` in `sys.modules`.
- When you import `__main__` it will get executed again (as a module) because there's no `interdoc.__main__` in `sys.modules`.

Also see (1) from <http://click.pocoo.org/5/setuptools/#setuptools-integration>

### 4.1.4 Module contents



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

### 5.1 Bug reports

When [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.

### 5.2 Documentation improvements

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

### 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/metatooling/interdoc/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 code contributions are welcome :)

## 5.4 Development

To set up *interdoc* for local development:

1. Fork *interdoc* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/interdoc.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)<sup>1</sup>.
2. Update documentation when there’s new API, functionality etc.
3. Add a file in `changelog.d/` describing the changes. The filename should be `{id}.{type}.rst`, where `{id}` is the number of the GitHub issue or pull request and `{type}` is one of `breaking` (for breaking changes), `deprecation` (for deprecations), or `change` (for non-breaking changes). For example, to add a new feature requested in GitHub issue #1234, add a file called `changelog.d/1234.change.rst` describing the change.
4. Add yourself to `AUTHORS.rst`.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

<sup>1</sup> If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.

It will be slower though ...

detox





## CHAPTER 6

---

### Authors

---

- Interdoc contributors - <https://github.com/metatooling/interdoc>



### 7.1 0.1.0 (2019-10-03)

#### 7.1.1 Changes

- First release on PyPI.

—



## CHAPTER 8

---

### Indices and tables

---

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



**i**

interdoc, 7  
interdoc.app, 7  
interdoc.cli, 7





**I**

`interdoc` (*module*), 7

`interdoc.app` (*module*), 7

`interdoc.cli` (*module*), 7

**M**

`main()` (*in module interdoc.app*), 7