
default-values

Release 0.5.3

**Sphinx extension to show default values in
documentation.**

Dominic Davis-Foster

Jul 21, 2023

Contents

1	Overview	1
2	Installation	3
2.1	from PyPI	3
2.2	from Anaconda	3
2.3	from GitHub	3
3	Configuration	5
3.1	default_description_format	5
3.2	Fields	5
4	Public API	7
4.1	default_regex	7
4.2	format_default_value	7
4.3	get_arguments	7
4.4	get_class_defaults	8
4.5	get_function_defaults	8
4.6	no_default_regex	8
4.7	process_default_format	8
4.8	process_docstring	8
4.9	setup	8
5	Demo	9
5.1	demo	9
6	Downloading source code	11
6.1	Building from source	12
	Python Module Index	13
	Index	15

Overview

This extension shows the default values in autodoc-formatted docstrings.

The default behaviour of `autodoc` is to turn this:

```
def namedlist(name: str = "NamedList") -> Callable:
    """
    A factory function to return a custom list subclass with a name.

    :param name: The name of the list.

    :return:
    """
```

into this:

```
domdf_python_tools.bases.namedlist (name='NamedList')
    A factory function to return a custom list subclass with a name.

Parameters name (str) – The name of the list.

Return type Callable
```

With `default_values` enabled, the documentation will now look like this:

```
domdf_python_tools.bases.namedlist (name='NamedList')
    A factory function to return a custom list subclass with a name.

Parameters name (str) – The name of the list. Default 'NamedList'.

Return type Callable
```

Default values are taken from the function/class signature. They can be overridden using the `default` option in the docstring. The default value can be suppressed using the `no-default` option.

No default value is shown if the argument does not have a default value.

The formatting of the default value can be customised using the `default_description_format` option in `conf.py`. By default this is 'Default %s'.

Installation

2.1 from PyPI

```
$ python3 -m pip install default_values --user
```

2.2 from Anaconda

First add the required channels

```
$ conda config --add channels https://conda.anaconda.org/conda-forge
$ conda config --add channels https://conda.anaconda.org/domdfcoding
```

Then install

```
$ conda install default_values
```

2.3 from GitHub

```
$ python3 -m pip install git+https://github.com/sphinx-toolbox/default_values@master --user
```

Enable `default_values` by adding the following to the `extensions` variable in your `conf.py`:

```
extensions = [
    ...
    'sphinx.ext.autodoc',
    'sphinxcontrib.default_values',
]
```

For more information see

<https://www.sphinx-doc.org/en/master/usage/extensions#third-party-extensions>.

Configuration

default_description_format

Type: `str`

Required: `False`

Default: `Default %s`

The format string for the default value.

3.2 Fields

These fields can be used in docstrings for classes, functions, methods etc., alongside other fields such as `:param:`.

:default <name>: value

Overrides the default value for <name>.

The value must be formatted how you would like it to be displayed in Sphinx. This can be useful when the default value in the signature is `None` and the true default value is assigned in the function body, such as for a mutable default argument.

Example:

```
:param name: The name of the list.  
:default name: :py:obj:`True`
```

`domdf_python_tools.bases.namedlist (name='NamedList')`

A factory function to return a custom list subclass with a name.

Parameters `name` (`str`) – The name of the list. Default `True`.

Return type `Callable`

:no-default <name>: value

Suppresses display of the default value for <name>.

This allows for default values to be suppressed on a per-argument basis.

Example:

```
:param name: The name of the list.  
:no-default name:
```

`domdf_python_tools.bases.namedlist (name='NamedList')`

A factory function to return a custom list subclass with a name.

Parameters `name` (`str`) – The name of the list.

Return type Callable

Public API

Sphinx extension to show default values in documentation.

Data:

<code>default_regex</code>	Regular expression to match default values declared in docstrings.
<code>no_default_regex</code>	Regular expression to match fields in docstrings to suppress default values.

Functions:

<code>format_default_value(value)</code>	Format the value as a string.
<code>get_arguments(obj)</code>	Returns a dictionary mapping argument names to parameters/arguments for a function.
<code>get_class_defaults(obj)</code>	Obtains the default values for the arguments of a class.
<code>get_function_defaults(obj)</code>	Obtains the default values for the arguments of a function.
<code>process_default_format(app)</code>	Prepare the formatting of the default value.
<code>process_docstring(app, what, name, obj, ...)</code>	Add default values to the docstring.
<code>setup(app)</code>	Setup <code>sphinxcontrib.default_values</code> .

`default_regex`

Type: `Pattern`

Regular expression to match default values declared in docstrings.

Changed in version 0.5.3: Change to be case insensitive.

Pattern	<code>^: (default) []</code>
----------------	-------------------------------

`format_default_value(value)`

Format the value as a string.

New in version 0.5.3.

Parameters `value` (`Any`)

Return type `Optional[str]`

`get_arguments(obj)`

Returns a dictionary mapping argument names to parameters/arguments for a function.

Parameters `obj` (`Callable`) – A function (can be the `__init__` method of a class).

Return type `Mapping[str, Parameter]`

get_class_defaults (*obj*)

Obtains the default values for the arguments of a class.

Parameters *obj* (`Type`) – The class.

Return type `Iterator[Tuple[str, Any]]`

Returns An iterator of 2-element tuples comprising the argument name and its default value.

get_function_defaults (*obj*)

Obtains the default values for the arguments of a function.

Parameters *obj* (`Callable`) – The function.

Return type `Iterator[Tuple[str, Any]]`

Returns An iterator of 2-element tuples comprising the argument name and its default value.

no_default_regex

Type: `Pattern`

Regular expression to match fields in docstrings to suppress default values.

Changed in version 0.5.3: Change to be case insensitive.

Pattern	<code>^: (no [-_]default) []</code>
----------------	--------------------------------------

process_default_format (*app*)

Prepare the formatting of the default value.

Parameters *app* (`Sphinx`)

process_docstring (*app, what, name, obj, options, lines*)

Add default values to the docstring.

Parameters

- **app** (`Sphinx`) – The Sphinx app.
- **what** (`str`)
- **name** (`str`) – The name of the object being documented.
- **obj** (`Any`) – The object being documented.
- **options** (`Dict[str, Any]`) – Mapping of autodoc options to values.
- **lines** (`List[str]`) – List of strings representing the current contents of the docstring.

setup (*app*)

Setup `sphinxcontrib.default_values`.

Parameters *app* (`Sphinx`)

Return type `Dict[str, Any]`

Demo

demo (*a*, *b*=0.0, *c*="", *d*=' ', *e*='hello world', *f*=(), *g*=Decimal('12.34'), *h*=1234, *i*=None, *j*=None, *k*=None, *l*="", *m*='\t', *n*=...)

Parameters

- **a** (*Any*) – No default.
- **b** (*float*) – A float. Default 0.0.
- **c** (*str*) – An empty string. Default ''.
- **d** (*str*) – A space (or a smiley face?). Default ' '.
- **e** (*str*) – A string. Default 'hello world'.
- **f** (*Tuple*) – A Tuple. Default ().
- **g** (*Decimal*) – A Decimal. Default Decimal('12.34').
- **h** (*int*) – An int. Default 1234.
- **i** (*Optional[List[str]]*) – Default None. Default None.
- **j** (*Optional[List[str]]*) – Overridden default. Default [].
- **k** (*Optional[List[str]]*) – Suppressed default.
- **l** (*str*) – This is a really long description. It spans multiple lines. The quick brown fox jumps over the lazy dog. The default value should be added at the end regardless. Default ' '.
- **m** (*str*) – Tab. Default '\t'.
- **n** (*Any*) – This argument's default value is undefined.

The description for *d* lacked a fullstop at the end, but one was added automatically.

The default value of *n* was *Ellipsis*, but it wasn't shown.

The above example was created from the following Python code:

```
1 # noqa: D100
2
3 # stdlib
4 from decimal import Decimal # pragma: no cover
5 from typing import Any, List, Optional, Tuple # pragma: no cover
6
7 __all__ = ["demo"] # pragma: no cover
8
```

(continues on next page)

```

9
10 def demo(
11     a: Any,
12     b: float = 0.0,
13     c: str = '',
14     d: str = ' ',
15     e: str = "hello world",
16     f: Tuple = (),
17     g: Decimal = Decimal("12.34"),
18     h: int = 1234,
19     i: Optional[List[str]] = None,
20     j: Optional[List[str]] = None,
21     k: Optional[List[str]] = None,
22     l: str = '',
23     m: str = '\t',
24     n: Any = ...,
25 ): # pragma: no cover
26     """
27
28     :param a: No default.
29     :param b: A float.
30     :param c: An empty string.
31     :param d: A space (or a smiley face?)
32     :param e: A string.
33     :param f: A Tuple.
34     :param g: A Decimal.
35     :param h: An int.
36     :param i: Default None.
37     :param j: Overridden default.
38     :default j: ``[]``
39     :param k: Suppressed default.
40     :no-default k:
41     :param l: This is a really long description.
42         It spans multiple lines.
43         The quick brown fox jumps over the lazy dog.
44         The default value should be added at the end regardless.
45     :param m: Tab.
46     :param n: This argument's default value is undefined.
47
48     The description for ``d`` lacked a fullstop at the end, but one was added_
49     ↪automatically.
50
51     The default value of ``n`` was :py:obj:`Ellipsis`, but it wasn't shown.
52     """

```

The **PEP 484** type hints were added by `sphinx-autodoc-typehints`.

Downloading source code

The `default_values` source code is available on GitHub, and can be accessed from the following URL: https://github.com/sphinx-toolbox/default_values

If you have `git` installed, you can clone the repository with the following command:

```
$ git clone https://github.com/sphinx-toolbox/default_values
```

```
Cloning into 'default_values'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a ‘zip’ file by clicking:

Clone or download → Download Zip

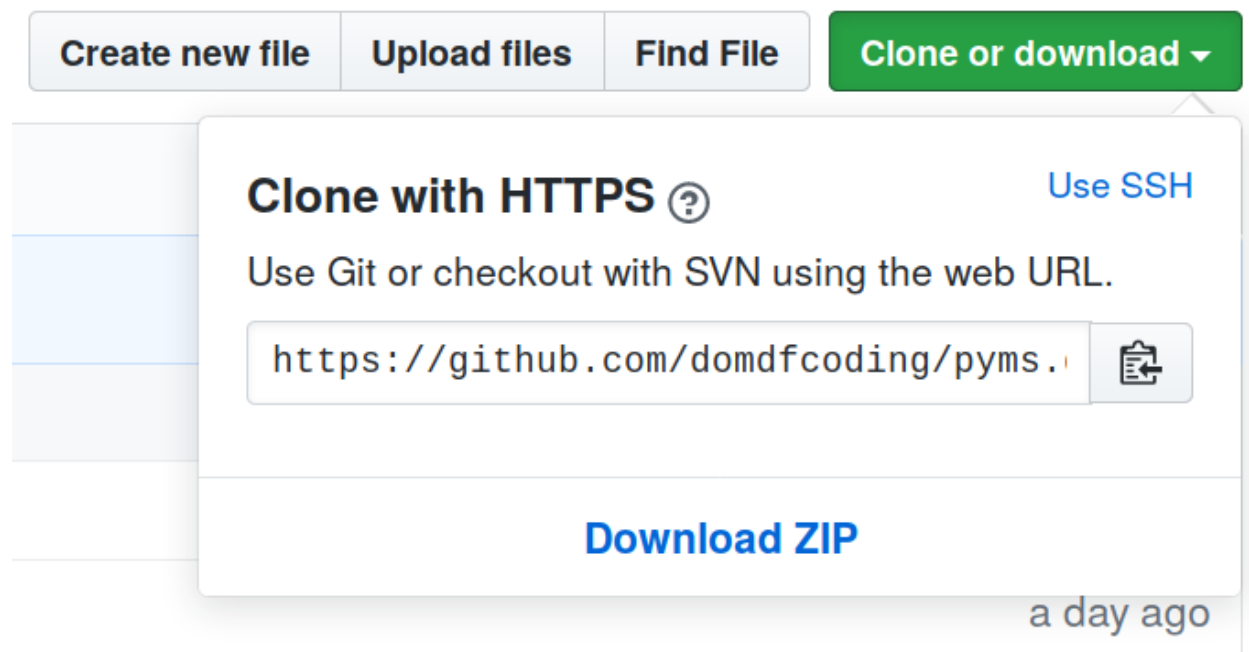


Fig. 1: Downloading a ‘zip’ file of the source code

6.1 Building from source

The recommended way to build `default_values` is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

Python Module Index

S

`sphinxcontrib.default_values`, [7](#)

`sphinxcontrib.default_values.demo`, [9](#)

Symbols

:default <name>: (*field*), 5
 :no-default <name>: (*field*), 5

D

default_description_format (*configuration value*), 5
 default_regex (*in module sphinxcontrib.default_values*), 7
 demo () (*in module sphinxcontrib.default_values.demo*), 9

F

format_default_value () (*in module sphinxcontrib.default_values*), 7

G

get_arguments () (*in module sphinxcontrib.default_values*), 7
 get_class_defaults () (*in module sphinxcontrib.default_values*), 8
 get_function_defaults () (*in module sphinxcontrib.default_values*), 8

M

module
 sphinxcontrib.default_values, 7
 sphinxcontrib.default_values.demo, 9

N

no_default_regex (*in module sphinxcontrib.default_values*), 8

P

process_default_format () (*in module sphinxcontrib.default_values*), 8
 process_docstring () (*in module sphinxcontrib.default_values*), 8
 Python Enhancement Proposals
 PEP 484, 10
 PEP 517, 12

S

setup () (*in module sphinxcontrib.default_values*), 8

sphinxcontrib.default_values
 module, 7
 sphinxcontrib.default_values.demo
 module, 9