
astatine

Release 0.3.3

Some handy helper functions for Python's AST module.

Dominic Davis-Foster

May 15, 2024

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from Anaconda	1
1.3	from GitHub	1
2	astatine	3
2.1	get_attribute_name	3
2.2	get_constants	3
2.3	get_contextmanagers	4
2.4	get_docstring_lineno	4
2.5	get_toplevel_comments	4
2.6	is_type_checking	4
2.7	kwargs_from_node	4
2.8	mark_text_ranges	5
3	Downloading source code	7
3.1	Building from source	8
4	License	9
	Python Module Index	11
	Index	13

Installation

1.1 from PyPI

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

1.2 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 astatine
```

1.3 from GitHub

```
$ python3 -m pip install git+https://github.com/python-formate/astatine@master --user
```


astatine

Some handy helper functions for Python's AST module.

Functions:

<code>get_attribute_name(node)</code>	Returns the elements of the dotted attribute name for the given AST node.
<code>get_constants(module)</code>	Returns a <code>name: value</code> mapping of constants in the given module.
<code>get_contextmanagers(with_node)</code>	For the given <code>with</code> block, returns a mapping of the contextmanager names to the individual nodes.
<code>get_docstring_lineno(node)</code>	Returns the line number of the start of the docstring for <code>node</code> .
<code>get_toplevel_comments(source)</code>	Returns a list of comment lines from <code>source</code> which occur before the first line of source code (including before module-level docstrings).
<code>is_type_checking(node)</code>	Returns whether the given <code>if</code> block is <code>if typing.TYPE_CHECKING</code> or equivalent.
<code>kwargs_from_node(node, posarg_names)</code>	Returns a mapping of argument names to the AST nodes representing their values, for the given function call.
<code>mark_text_ranges(node, source)</code>	Recursively add the <code>end_lineno</code> and <code>end_col_offset</code> attributes to each child of <code>node</code> which already has the attributes <code>lineno</code> and <code>col_offset</code> .

`get_attribute_name (node)`

Returns the elements of the dotted attribute name for the given AST node.

New in version 0.3.1.

Parameters `node` (AST)

Raises `NotImplementedError` – if the name contains an unknown node (i.e. not `ast.Name`, `ast.Attribute`, or `ast.Call`)

Return type `Iterable[str]`

`get_constants (module)`

Returns a `name: value` mapping of constants in the given module.

New in version 0.3.1.

Parameters `module` (`Module`)

Return type `Dict[str, Any]`

get_contextmanagers (*with_node*)

For the given `with` block, returns a mapping of the contextmanager names to the individual nodes.

New in version 0.3.1.

Parameters `with_node` (`With`)

Return type `Dict[Tuple[str, ...], withitem]`

get_docstring_lineno (*node*)

Returns the line number of the start of the docstring for `node`.

Parameters `node` (`Union[FunctionDef, ClassDef, Module]`)

Warning: On CPython 3.6 and 3.7 the line number may not be correct, due to <https://bugs.python.org/issue16806>.

CPython 3.8 and above are unaffected, as are PyPy 3.6 and 3.7

Accurate line numbers on CPython 3.6 and 3.7 may be obtained by using https://github.com/domdfcoding/typed_ast, which contains the backported fix from Python 3.8.

Return type `Optional[int]`

get_toplevel_comments (*source*)

Returns a list of comment lines from `source` which occur before the first line of source code (including before module-level docstrings).

Parameters `source` (`str`)

Return type `StringList`

is_type_checking (*node*)

Returns whether the given `if` block is `if typing.TYPE_CHECKING` or equivalent.

Parameters `node` (`AST`)

Return type `bool`

kwargs_from_node (*node*, *posarg_names*)

Returns a mapping of argument names to the AST nodes representing their values, for the given function call.

New in version 0.3.1.

Parameters

- **node** (`Call`)
- **posarg_names** (`Union[Iterable[str], Callable]`) – Either a list of positional argument names for the function, or the function object.

Return type `Dict[str, AST]`

mark_text_ranges (*node*, *source*)

Recursively add the `end_lineno` and `end_col_offset` attributes to each child of `node` which already has the attributes `lineno` and `col_offset`.

Parameters

- **node** (`AST`) – An AST node created with `ast.parse()`.
- **source** (`str`) – The corresponding source code for the node.

Downloading source code

The `astatine` source code is available on GitHub, and can be accessed from the following URL: <https://github.com/python-formate/astatine>

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

```
$ git clone https://github.com/python-formate/astatine
```

```
Cloning into 'astatine'...
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

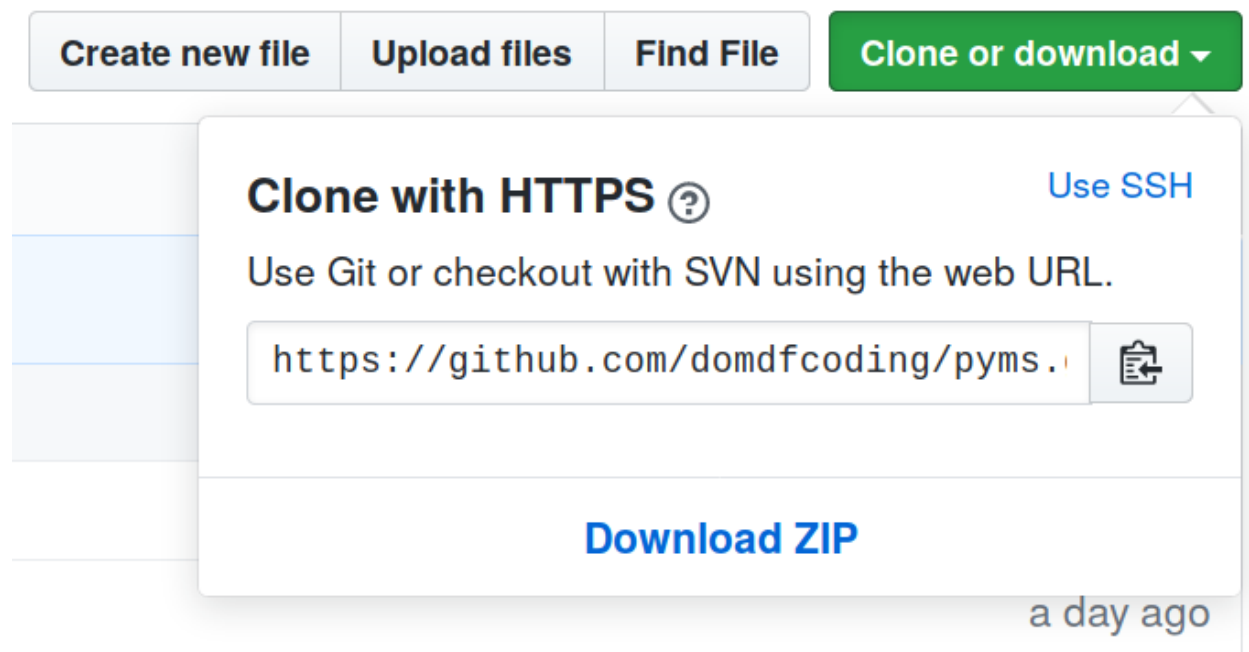


Fig. 1: Downloading a ‘zip’ file of the source code

3.1 Building from source

The recommended way to build `astatine` 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.

License

astatine is licensed under the [MIT License](#)

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.

Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](#) ⇒

```
Copyright (c) 2021-2022 Dominic Davis-Foster
```

```
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.
```


Python Module Index

a

astatine, [3](#)

A

astatine
 module, 3

G

get_attribute_name() (*in module astatine*), 3
get_constants() (*in module astatine*), 3
get_contextmanagers() (*in module astatine*), 4
get_docstring_lineno() (*in module astatine*), 4
get_toplevel_comments() (*in module astatine*),
 4

I

is_type_checking() (*in module astatine*), 4

K

kwargs_from_node() (*in module astatine*), 4

M

mark_text_ranges() (*in module astatine*), 5
MIT License, 9
module
 astatine, 3

P

Python Enhancement Proposals
 PEP 517, 8