
pguard Documentation

Release 0.3.1

Kouhei Maeda

Nov 03, 2018

Contents

1	pguard	3
1.1	Status	3
1.2	Requirements	3
1.3	Features	3
1.4	Setup	3
1.5	Simple usage	4
2	References & sample usage	5
3	ChangeLog	9
3.1	0.3.1 (2018-11-03)	9
3.2	0.3.0 (2016-04-30)	9
3.3	0.2.1 (2016-04-29)	9
3.4	0.2.0 (2016-04-27)	9
3.5	0.1.4 (2016-04-24)	9
3.6	0.1.3 (2016-04-23)	10
3.7	0.1.2 (2016-04-23)	10
3.8	0.1.1 (2016-04-23)	10
3.9	0.1.0 (2016-04-23)	10
4	Indices and tables	11
	Python Module Index	13

Contents:

Guard like Haskell for Python.

1.1 Status

A horizontal bar with a dark grey background. It is divided into four segments: 'build' (dark grey), 'passing' (green), 'coverage' (dark grey), and '100%' (green).

1.2 Requirements

- Python 2.7 or Python 3.5 over or PyPy 2.4.0 over

1.3 Features

- guard with lambda
- Enable to use function or method as `guard_cl` statement argument.

1.4 Setup

```
$ pip install --user pguard  
or  
(venv)$ pip install pguard
```

1.5 Simple usage

```
>>> from pguard import guard
>>> from pguard import guard_cl as g
>>> (lambda n: guard(
... g(-1, n < 0),
... g(0, n == 0),
... g(1)  ## otherwise
... ))(0)
0
```

References & sample usage

Guard like Haskell for Python.

`pguard.guard(*guard_clauses)`
Guard function.

Parameters `*guard_clauses` (*tuple*) – guard clauses.

```
>>> g = guard_cl
>>> (lambda n: guard(
... g(-1, n < 0),
... g(0, n == 0),
... g(1)  ## otherwise
... ))(0)
0
```

```
>>> s = lambda n: guard(
... g(-1, n < 0),
... g(0, n == 0),
... g(1)  ## otherwise
... )
>>> [s(i) for i in [-10, 3, 0, 1, -12.0, 2]]
[-1, 1, 0, 1, -1, 1]
>>> s2 = lambda n: guard(
... g(-1, n < 0),
... g(0, n == 0),
... )
>>> [s2(i) for i in [-10, 3, 0, 1, -12.0, 2]]
[-1, False, 0, False, -1, False]
```

```
>>> def fibo(n):
...     if n < 0 : return -1
...     if n == 0 or n == 1: return 1
...     return fibo(n - 1) + fibo(n - 2)
>>> f = lambda n: guard(
```

(continues on next page)

(continued from previous page)

```

... g(Exception('out of range'), n < 0),
... g(1, n < 2),
... g(fibo(n - 1) + fibo(n - 2))
... )
>>> [f(i) for i in range(-1, 10)]
[Exception('out of range'), 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
>>> # Python3.7
>>> # [Exception('out of range'), 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

```

```

>>> b = lambda w, h: guard(
... g('Very severely underweight', w / h ** 2 < 15.0),
... g('Severely underweight', w / h ** 2 < 16.0),
... g('Underweight', w / h ** 2 < 18.5),
... g('Normal (healthy weight)', w / h ** 2 < 25.0),
... g('Overweight', w / h ** 2 < 30.0),
... g('Obese Class I (Moderately obese)', w / h ** 2 < 35.0),
... g('Obese Class II (Severely obese)', w / h ** 2 < 40.0),
... g('Obese Class III (Very severely obese)')
... )
>>> b(67.3, 1.68)
'Normal (healthy weight)'

```

```

>>> def foo(x):
...     return x * 2

```

```

>>> def bar():
...     return 'negative'

```

```

>>> l = lambda n: guard(
... g(bar, n < 0),
... g(foo, n == 0, (n,)),
... g(foo, n == 1, (n + 1,)),
... g(foo, params=(n + 2,))
... )
>>> [l(i) for i in range(-1, 4)]
['negative', 0, 4, 8, 10]

```

```

>>> guard(g(foo))
Traceback (most recent call last):
...
ValueError: Invalid parameters.

```

```

>>> class Hoge(object):
...     def hoge(self, a):
...         return a * 2
...     def moge(self, a, b):
...         return a + b * 2

```

```

>>> h = Hoge()
>>> (lambda n: guard(
... g(h.hoge, n > 0, (n,))
... ))(10)
20

```

pguard.guards.guard(*statement*, *condition=None*, *params=None*)

Guard clause.

Returns (statement, params) or False

Parameters

- **statement** – expression statement.
- **condition** – condition statement.
- **params** (*tuple*) – expression statement parameters.

```
>>> (lambda n: (  
... guard_cl(-1, n < 0),  
... guard_cl(0, n == 0),  
... guard_cl(1))) (0)  
(False, (0, None), (1, None))
```


3.1 0.3.1 (2018-11-03)

- Fixes pylint violations.

3.2 0.3.0 (2016-04-30)

- Fixes lazy evaluation.
- Changes `guard_cl()` respond values.

3.3 0.2.1 (2016-04-29)

- Fixes Python 2.7 classmethod behavior bugs.

3.4 0.2.0 (2016-04-27)

- Supports “function” and “method” as `guard_cl` statement argument.

3.5 0.1.4 (2016-04-24)

- Fixes `setup.py` url.

3.6 0.1.3 (2016-04-23)

- Fixes README.
- Adds check long-description to pre-commit hook.

3.7 0.1.2 (2016-04-23)

- Fixes setup.py version.

3.8 0.1.1 (2016-04-23)

- completed coverage 100%.
- Adds sample usages.

3.9 0.1.0 (2016-04-23)

- Initial release.

CHAPTER 4

Indices and tables

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

p

pguard, 5

G

`guard()` (in module `pguard`), 5

`guard_cl()` (in module `pguard`), 6

P

`pguard` (module), 5