
dummyspackage Documentation

Release 0.9.1

Dummy Dumson

Feb 25, 2019

CONTENTS:

1	dummyspackage package	1
1.1	Subpackages	1
1.1.1	dummyspackage.nested package	1
1.1.1.1	Submodules	1
1.1.1.2	dummyspackage.nested.baz_module module	1
1.1.1.3	Module contents	2
1.2	Submodules	2
1.3	dummyspackage.bar_module module	2
1.4	dummyspackage.foo_module module	2
1.5	Module contents	2
1.5.1	Section about sections:	2
1.5.1.1	Subsection:	3
1.5.2	Other structures:	3
2	Indices and tables	4
	Python Module Index	5
	Index	6

DUMMYPACKAGE PACKAGE

1.1 Subpackages

1.1.1 dummypackage.nested package

1.1.1.1 Submodules

1.1.1.2 dummypackage.nested.baz_module module

Module with dummy functionality that isn't tested, to show a gap in the code coverage.

```
class dummypackage.nested.baz_module.Baz
```

```
    Bases: object
```

```
    A class with many lines that don't get tested
```

```
    a = 123
```

```
    b = 123
```

```
    c = 123
```

```
    d = 123
```

```
    e = 123
```

```
    f = 123
```

```
    g = 123
```

```
    h = 123
```

```
    i = 123
```

```
    j = 123
```

```
    k = 123
```

```
    m = 123
```

```
    n = 123
```

```
    o = 123
```

```
    p = 123
```

```
    q = 123
```

```
    r = 123
```

1.1.1.3 Module contents

1.2 Submodules

1.3 dummyspackage.bar_module module

Module mimicking foo with more expensive memory and runtime requirements.

class dummyspackage.bar_module.**Bar** (*size=1000000*)

Bases: *dummyspackage.foo_module.Foo*

Similar to Foo, with higher memory and runtime requirements.

1.4 dummyspackage.foo_module module

Module containing a simple class with low memory and runtime requirements.

class dummyspackage.foo_module.**Foo** (*size=1000000*)

Bases: object

A simple class with low memory and runtime requirements.

get_result ()

Basic getter. :returns: the field stored in *self._result*

loop (*times*)

Restart result and run computation a number of times. :param times: The number of times, expected non-negative int.

1.5 Module contents

Main init file docstring. It exemplifies the usage of *restructured text*,

like:

- *Italics*
- **Bold**
- **Numbered and nested lists:**
 1. This is a numbered list
 2. Nested lists have at least three characters indentation
- `Inline literals`
- Parameter fields: see class and method docstrings.

1.5.1 Section about sections:

- Surrounding chars have to be at least as long as the title
- No explicit hierarchy, but this recommended: #, *, =, -, ^, " (the first two with overline).

1.5.1.1 Subsection:

To exemplify the usage of \LaTeX and nested quotes, nothing best that the words of Isaac Newton himself:

“If I have seen further it is by standing on the shoulders of Giants.”

Or, in other words:

$$\sum_{k=1}^{\infty} k = -\frac{1}{12}$$

1.5.2 Other structures:

Field lists:

Author Homer J. Simpson

Email hjs@compuglobalhypermega.net

Literal blocks, preceded by double colon:

```
This is a literal block

Markups are **not** rendered here.
```

Doctest blocks can be tested by the doc tool:

```
>>> [factorial(n) for n in range(6)]
[1, 1, 2, 6, 24, 120]
>>> [factorial(long(n)) for n in range(6)]
[1, 1, 2, 6, 24, 120]
```

Grid tables must be indented:

Header 1	Header 2	Header 3
body row 1	column 2	column 3
body row 2	Cells may span columns.	
body row 3	Cells may span rows.	<ul style="list-style-type: none"> Cells contain blocks.
body row 4		

Simple table:

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

INDICES AND TABLES

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

PYTHON MODULE INDEX

d

- `dummyspackage`, [2](#)
- `dummyspackage.bar_module`, [2](#)
- `dummyspackage.foo_module`, [2](#)
- `dummyspackage.nested`, [2](#)
- `dummyspackage.nested.baz_module`, [1](#)

INDEX

A

a (*dummyspackage.nested.baz_module.Baz attribute*), 1

B

b (*dummyspackage.nested.baz_module.Baz attribute*), 1

Bar (*class in dummyspackage.bar_module*), 2

Baz (*class in dummyspackage.nested.baz_module*), 1

C

c (*dummyspackage.nested.baz_module.Baz attribute*), 1

D

d (*dummyspackage.nested.baz_module.Baz attribute*), 1

dummyspackage (*module*), 2

dummyspackage.bar_module (*module*), 2

dummyspackage.foo_module (*module*), 2

dummyspackage.nested (*module*), 2

dummyspackage.nested.baz_module (*module*), 1

E

e (*dummyspackage.nested.baz_module.Baz attribute*), 1

F

f (*dummyspackage.nested.baz_module.Baz attribute*), 1

Foo (*class in dummyspackage.foo_module*), 2

G

g (*dummyspackage.nested.baz_module.Baz attribute*), 1

get_result () (*dummyspackage.foo_module.Foo method*), 2

H

h (*dummyspackage.nested.baz_module.Baz attribute*), 1

I

i (*dummyspackage.nested.baz_module.Baz attribute*), 1

J

j (*dummyspackage.nested.baz_module.Baz attribute*), 1

K

k (*dummyspackage.nested.baz_module.Baz attribute*), 1

L

loop () (*dummyspackage.foo_module.Foo method*), 2

M

m (*dummyspackage.nested.baz_module.Baz attribute*), 1

N

n (*dummyspackage.nested.baz_module.Baz attribute*), 1

O

o (*dummyspackage.nested.baz_module.Baz attribute*), 1

P

p (*dummyspackage.nested.baz_module.Baz attribute*), 1

Q

q (*dummyspackage.nested.baz_module.Baz attribute*), 1

R

r (*dummyspackage.nested.baz_module.Baz attribute*), 1