

Programavimo kalba **Python**

penktoji paskaita

Marius Gedminas
<mgedmin@b4net.lt>

<http://mg.b4net.lt/python/>





Dokumentaciniai testai (doctests)

```
def factorial(n):
```

```
    """Compute the factorial of n.
```

```
    >>> factorial(5)
```

```
    120
```

```
    """
```



Interaktyvios Python sesijos išrašas dokumentacinėje eilutėje

Kodo naudojimo pavyzdys kurio veikimą galima patikrinti



Kaip patikrinti?

```
def factorial(...):
```

```
    ...
```

```
if __name__ == '__main__':
```

```
    import doctest
```

```
    doctest.testmod()
```



arba

...

```
if __name__ == '__main__':  
    if '--test' in sys.argv:  
        import doctest  
        doctest.testmod()  
        sys.exit()  
    else:  
        main()
```

arba galima juos įtraukti į
testų rinkinį

```
import unittest
```

```
import doctest
```

```
...
```

```
def test_suite():
```

```
    suite = unittest.TestSuite()
```

```
    suite.addTest(doctest.DocTestSuite(
        'mypackage.mymodule'))
```

```
    return suite
```

```
if __name__ == '__main__':
```

```
    unittest.main(defaultTest='test_suite')
```



Tik nereikia persistengti!

Dokumentacijos eilutės neturėtų
užgožti paties kodo

Alternatyvos:
atskiri testų moduliai
tekstiniai failai



Dokumentacinių testų modulis

```
def doctest_factorial():
```

```
    """
```

```
    ...
```

```
    """
```

```
def doctest_integral():
```

```
    """
```

```
    ...
```

```
    """
```




Dokumentacinių testų failas

Matematinės funkcijos

Galite skaičiuoti faktorialus:

```
>>> from mymath import factorial
```

```
>>> factorial(4)
```

```
24
```

O galite ir neskaičiuoti...



reStructured Text



O prasmė?

Du zuikiai vienu šūviu: dokumentacija ir testai

Trys zuikiai:
dokumentacija, testai
ir dokumentacijos testai

Trys dok. testų rūšys:

1. pavyzdžiai

2. dokumentacija

3. testai

Pavyzdžiai:
trumpa funkcijos naudojimo
instrukcija
(dok. eilutėje)

Dokumentacija:
detali paketo/modulio instrukcija
(tekstiniam faile)

Testai:

rinkinys testų

(Python modulyje)



Testai: o prasmė?



unittest ar doctest?

```
# test.py
```

```
class TestSolve(unittest.TestCase):
```

```
    def test_no_solutions(self):
```

```
        #  $x^{**2} + 4 = 0$ 
```

```
        self.assertEqual(
            solve(1, 0, 4), [])
```

```
def doctest_no_solutions(self):  
    """Test for solve().
```

$x^{**2} + 4 = 0$ has no real solutions

```
>>> solve(1, 0, 4)
```

```
[]
```

```
"""
```

Rašant dok. testus norisi pridėti
paaiškinimų



Todėl juos lengviau skaityti



Dok. testuose galima naudoti 'print'

The list of calendars is now as follows

```
>>> for cal in person.visible_calendars:  
...     print '%s (%s)' % (cal.title,  
...                        cal.colour)
```

Site-wide calendar (green)

My own calendar (blue)

My dog's calendar (brown)

We can register a callback

```
>>> def callback_fn(arg):  
...     print "callback:", arg  
>>> someobj.add_callback(callback_fn)
```

which gets called during processing

```
>>> someobj.process()  
callback: 42
```

Skirtumą tarp gauto ir norimo
rezultato
galima pamatyti diff formatu

Failed example:

```
for cal in person.visible_calendars:  
    print '%s (%s)' % (cal.title,  
                      cal.colour)
```

Differences (ndiff with -expected +actual):

Site-wide calendar (green)

-My own calendar (blue)

+My calendar (blue)

My dog's calendar (brown)



Bet tai reikia įjungti

...

```
suite.addTest(doctest.DocTestSuite(  
    'mypackage.mymodule',  
    optionflags=  
        doctest.REPORT_NDIFF))
```

...

Dok. testai leidžia atsakymą lyginti
ne griežtai
(daugtaškiai)

The user name is included in the output:

```
>>> print_report()
```

```
Fiscal Report
```

```
-----
```

```
...
```

```
Report generated for mgedmin
```

```
...
```

Daugtaškis priima bet kokį teksto
gabalą



Bet jį reikia įjungti

```
...  
suite.addTest(doctest.DocTestSuite(  
    'mypackage.mymodule',  
    optionflags=doctest.ELLIPSIS))  
...
```



Dokumentacinių testų parametrai

ELLIPSIS

NORMALIZE_WHITESPACE

REPORT_NDIFF

REPORT_ONLY_FIRST_FAILURE



Juos galima įjungti testų viduje

The user name is included in the output:

```
>>> print_report()
```

```
... # doctest: +ELLIPSIS
```

```
Fiscal Report
```

```
-----
```

```
...
```

```
Report generated for mgedmin
```

```
...
```




Arba globaliai

...

```
suite.addTest(doctest.DocTestSuite(  
    'mypackage.mymodule',  
    optionflags=  
        doctest.ELLIPSIS |  
        doctest.REPORT_NDIFF))
```

...

Testai kaip projektavimo įrankis („mokslinė fantastika“)



viskas