Testavimas pagal mus!

Albertas Agejevas <alga@pov.lt>
POV
http://www.pov.lt/

Pycon-LT 2009

Kalba eina apie web aplikacijas

Kas daro?

Pakelkit rankas.

Automatizuoti testai

Kam jų išvis reikia?

PyUnit

```
import unittest

class TestMax(unittest.TestCase):

    def test_int(self):
        self.assertEquals(max(1, 2), 2)
        self.assertEquals(max(1, -2), 1)
        self.assertEquals(max([3, 4]), 4)

    def test_str(self):
        self.assertEquals(max('aaa', 'AAA'), 'aaa')

if __name__ == '__main__':
    unittest.main()
```

doctest

```
# -*- coding: utf-8 -*-
import doctest

def fact(n):
    """Faktorialas tai yra funkcija, apibrėžta taip:
         n! := n * (n - 1) * ... * 2 * 1

       >>> fact(2)
       2
       >>> fact(4)
       24

    Mažesniems už vienetą skaičiams faktorialas neapibrėžtas,
    bet mes grąžinam nulį:

       >>> fact(0)
       1
       >>> fact(-20)
       1
```

doctest

```
    Lūžtam su stringais:
       >>> fact("of life")
       Traceback (most recent call last):
         ...
       TypeError: unsupported operand type(s) for -: 'str' and 'int'
    """
    if n <= 0:
        return 1
    return n * fact(n-1)

if __name__ == '__main__':
    doctest.testmod()
```

PyUnit vs Doctest

- atomiškumas
- refaktorabilumas
- skaitomumas

Kompromisas?

```
def doctest_fact():
    """Testai faktorialui

    Faktorialas tai yra funkcija, apibrėžta taip:
         n! := n * (n - 1) * ... * 2 * 1

       >>> fact(2)
       2
       >>> fact(4)
       24

    Mažesniems už vienetą skaičiams faktorialas neapibrėžtas,
    bet mes grąžinam nulį:

       >>> fact(0)
       1
       >>> fact(-20)
       1
    """
```

Test runner

- zope.testing
- schooltool
- py.test
- nose

Senovė

```
class TestEditPages(SurveyTestBase):

    def test_admin(self):
        self.assertRaises(Unauthorized, self.publish, '/admin.html')
        response = self.publish('/admin.html', basic='admin:pass')
        self.assertEquals(response.getStatus(), 200)

    def test_edit(self):
        response = self.publish('/edit.html', basic='admin:pass')
        self.assertEquals(response.getStatus(), 200)
        self.assert_("Edit mode" in response.getBody())
```

Viduramžiai

Užpildykim formą:

```
>>> print http(r"""
... POST /survey_2701/edit.html HTTP/1.1
... Authorization: Basic mgr:mgrpw
... Content-Type: application/x-www-form-urlencoded
...
... title=Hyper+survey%21&q1.question=Type+your+question+here&q2.question=Typ
HTTP/1.1 303 See Other
...
     <td><h3>A free text question</h3></td>
...
     <td><h3>Multiple choice question
    (attitude)</h3></td>
...
```

zope.testbrowser

Patikrinkim, ar Delfi rašo apie karą. Atsidarykim puslapį ir užpildykim paieškos formą:

```
>>> from zope.testbrowser import Browser
>>> browser = Browser('http://www.delfi.lt/')
>>> browser.getControl(name="q").value = "karas"
>>> browser.getControl("Ieškoti").click()
```

Dabar matome, kad rezultatuose yra karas:

```
>>> print browser.contents
<...karas...
```

Paspauskime nuorodą apie karą:

```
>>> print browser.getLink("karas").click()
```

Abstrakcijos lygiai

- Modulio testas (izoliuotas)
- Modulio testas su tikroviška aplinka
- Posistemės testas
- Funkcinis testas su testbrowser
- Funkcinis testas su Selenium

Abstrakcijos lygiai

Visi gali praverst.

Abstrakcijos lygiai

Reikia pasirinkti kelis.