

Tartalomjegyzék

- [1 megoldás](#)
- [2 Keresés](#)
- [3 Kivételek](#)
- [4 Unittest](#)
- [5 Mentés, visszatöltés](#)
- [6 Menü](#)

megoldás

A unittest-es részeket nem tartalmazza, az csak kitér?ként volt bemutatva...

[itt látható](#)

Keresés

Egészítsd ki a jegyzetömböt úgy hogy a jegyzetek tartalmára keresni lehessen.

Kivételek

Egészítsük ki a jegyzetömb implementációinkat, hogy ha a jegyzetömb konstruktora nem megfelel? típust kap akkor TypeError adjon vissza. Írjunk saját kivételt ami leszármazik TypeError-tól és ha a jegyzet üresen szeretnék létrehozni akkor dobjunk ilyen kivételt.

Unittest

írj testeket a jegyzetömbre. pl.:

- jól működik-e az add?
- a konstruktor dobja-e a kivételt ha rosszul hívják
- stb.

unittest használata [manual](#):

```
import random
import unittest

class TestSequenceFunctions(unittest.TestCase):

    def setUp(self):
        self.seq = range(10)

    def tearDown(self):
        pass

    # def runTest(self):
    #     pass

    def test_shuffle(self):
        # make sure the shuffled sequence does not lose any elements
        random.shuffle(self.seq)
        self.seq.sort()
        self.assertEqual(self.seq, range(10))
```

```
# should raise an exception for an immutable sequence
self.assertRaises(TypeError, random.shuffle, (1,2,3))

def test_choice(self):
    element = random.choice(self.seq)
    self.assertTrue(element in self.seq)

def test_sample(self):
    with self.assertRaises(ValueError):
        random.sample(self.seq, 20)
    for element in random.sample(self.seq, 5):
        self.assertTrue(element in self.seq)

# command line test run
if __name__ == '__main__':
    unittest.main()

suite = unittest.TestLoader().loadTestsFromTestCase(TestSequenceFunctions)
unittest.TextTestRunner(verbosity=2).run(suite)

Method Checks that New in
assertEqual(a, b) a == b
assertNotEqual(a, b) !=ab
assertTrue(x)bool(x) is True
assertFalse(x)bool(x) is False
assertIs(a, b) as b 2.7
assertIsNot(a, b) is not b 2.7
assertIsNone(x) x is None 2.7
assertIsNotNone(x) is not None 2.7
assertIn(a, b) in b 2.7
assertNotIn(a, b) not in b 2.7
assertIsInstance(a, b) isinstance(a, b) 2.7
assertNotIsInstance(a, b) not isinstance(a, b) 2.7
assertRaises(exc, fun, *args, **kwds) fun(*args, **kwds) raises exc
assertRaisesRegexp(exc, re, fun, *args, **kwds) fun(*args, **kwds) raises exc and the message matches re
assertAlmostEqual(a, b)round(a-b, 7) == 0
assertNotAlmostEqual(a, b)round(a-b, 7) != 0
assertGreater(a, b) > ba 2.7
assertGreaterEqual(a, b) >= b 2.7
assertLess(a, b) < b 2.7
assertLessEqual(a, b) <= a b 2.7
assertRegexpMatches(s, re) re.search(s) 2.7
assertNotRegexpMatches(s, re)not regex.search(s) 2.7
assertItemsEqual(a, b)sorted(a) == sorted(b) and works with unhashable objs 2.7
assertDictContainsSubset(a, b)all the key/value pairs in a exist in b 2.7
assertMultiLineEqual(a, b) strings 2.7
assertSequenceEqual(a, b) sequences 2.7
assertListEqual(a, b) lists 2.7
assertTupleEqual(a, b) tuples 2.7
assertSetEqual(a, b)sets or frozensets 2.7
assertDictEqual(a, b) dicts 2.7
```

Mentés, visszatöltés

jegyzetomb kiegészítése mentés funkcióval, hogy ha később indításkor a meglévő jegyzeteket vissza tudjuk olvasni.

Menü

Csináljunk kezdetleges menüt a jegyzetek, jegyzettömbök kezelésére. A programot próbáljuk osztályokba szervezni!