PYTHON

PYTHON

Testowanie

Pawel Dettlaff

Maciej Ogrodniczek

Maciej Chmiel

Nokia Siemens Networks

Agenda

- Testowanie - przypomnienie
- doctest
- unittest

Testowanie - TDD

Add a test

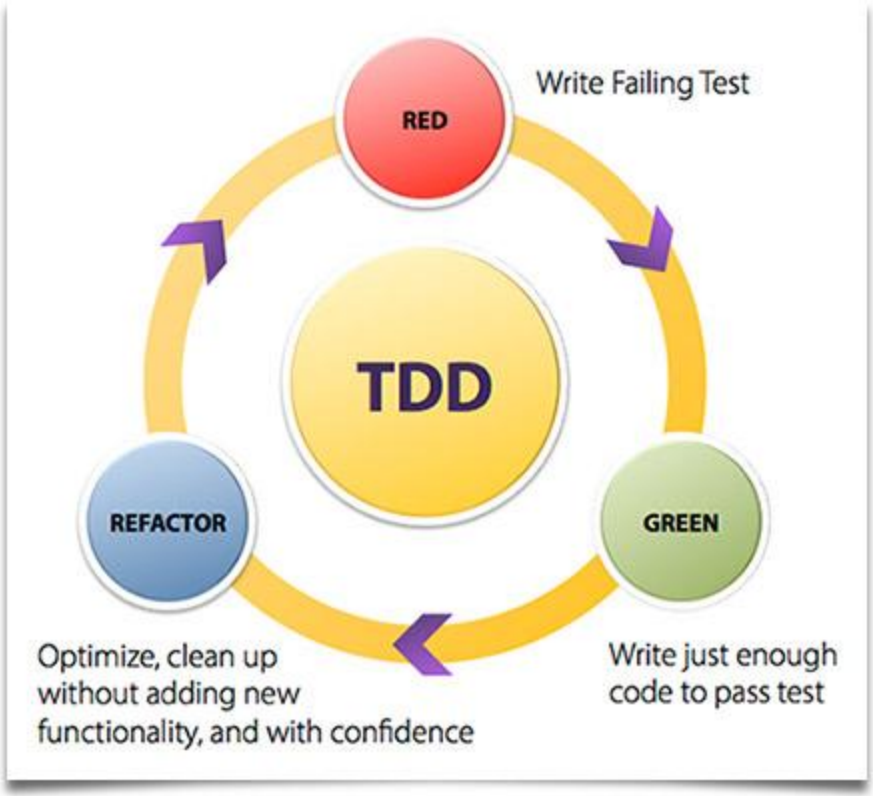
Run all tests and see if the new one fails

Write some code

Run tests

Refactor code

Repeat



doctest

- Pozwala na zawarcie prostych testów w docstrigu
- Łatwa składnia testów – imitujemy tryb interaktywny Pythona
- Testy mogą służyć również jako przykłady użycia

```
def name_generator(name1, name2):
    """
    >>> name_generator('Sally', 'Tommy')
    'Salmmy'
    >>> name_generator('Macius', 'Lucja')
    'Maccja'
    >>> name_generator()
    Traceback (most recent call last):
    ...
    TypeError: name_generator() takes exactly 2 arguments (0 given)
    """
    return name1[:3] + name2[-3:].lower()
```


Nokia Siemens Networks

doctest

- Sposób użycia:

  `python -m doctest <nazwa_modułu>`

```
def name_generator(name1, name2):
    """
    >>> name_generator('Sally', 'Tommy')
    'Salmmy'
    >>> name_generator('Macius', 'Lucja')
    'Maccja'
    >>> name_generator()
    Traceback (most recent call last):
    ...
    TypeError: name_generator() takes exactly 2 arguments (0 given)
    """
    return name1[:3] + name2[-3:].lower()
```


Nokia Siemens Networks

Zadanie 1

Napisz doctesty dla kodu.

unittest

```
def pole_kwadratu(a):
    """
    Zwraca pole kwadratu.
    """
    return a**2
```

```
import unittest
import pola

class PolaTest(unittest.TestCase):
    def test_proste_pole_kwadratu(self):
        wynik = pola.pole_kwadratu(2)
        oczekiwany_wynik = 4
        self.assertEqual(wynik, oczekiwany_wynik)


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

Zadanie 2

Napisz unittesty dla kodu.

Nokia Siemens Networks

Zadanie 3

Napisz kod na podstawie testów