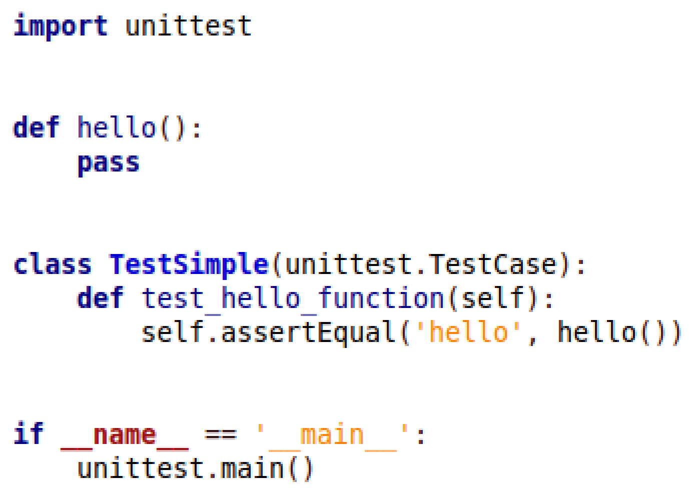
Example of
Simple Unit Testing
In
Python Programming Language
(unittest module)

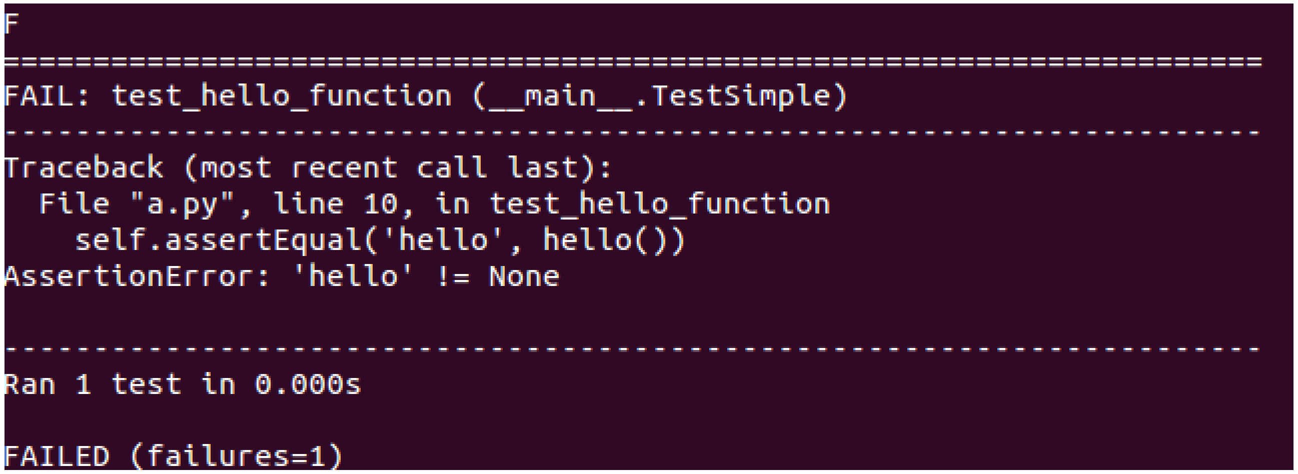
Case study: SQLiteBoy

Write a test case



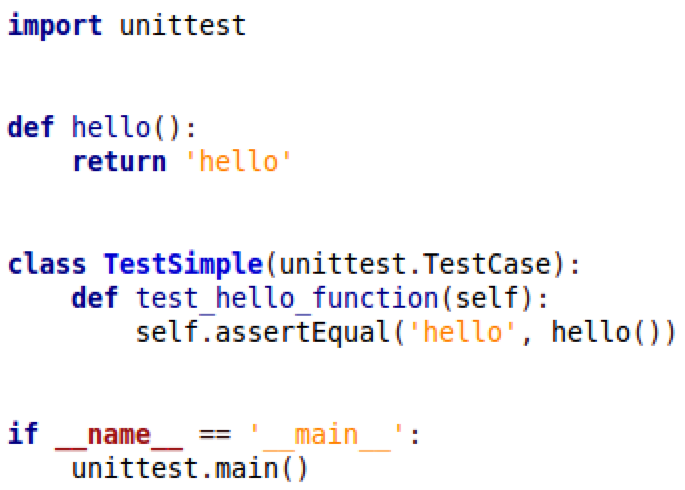
```python
import unittest


def hello():
    pass


class TestSimple(unittest.TestCase):
    def test_hello_function(self):
        self.assertEqual('hello', hello())


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

Run the test: failed



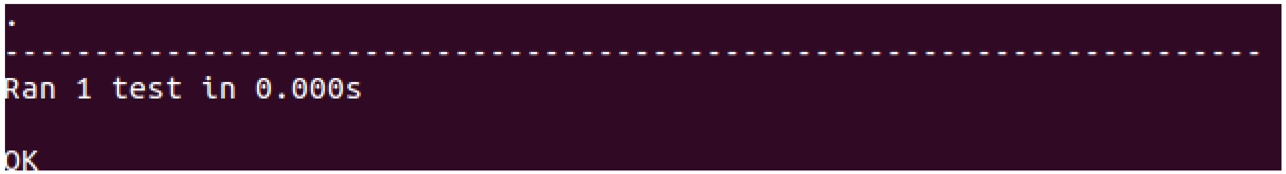
```
F
======================================================================
FAIL: test_hello_function (__main__.TestSimple)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "a.py", line 10, in test_hello_function
    self.assertEqual('hello', hello())
AssertionError: 'hello' != None


----------------------------------------------------------------------
Ran 1 test in 0.000s

FAILED (failures=1)
```

Write the codes



```python
import unittest


def hello():
    return 'hello'


class TestSimple(unittest.TestCase):
    def test_hello_function(self):
        self.assertEqual('hello', hello())


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

Test again: OK

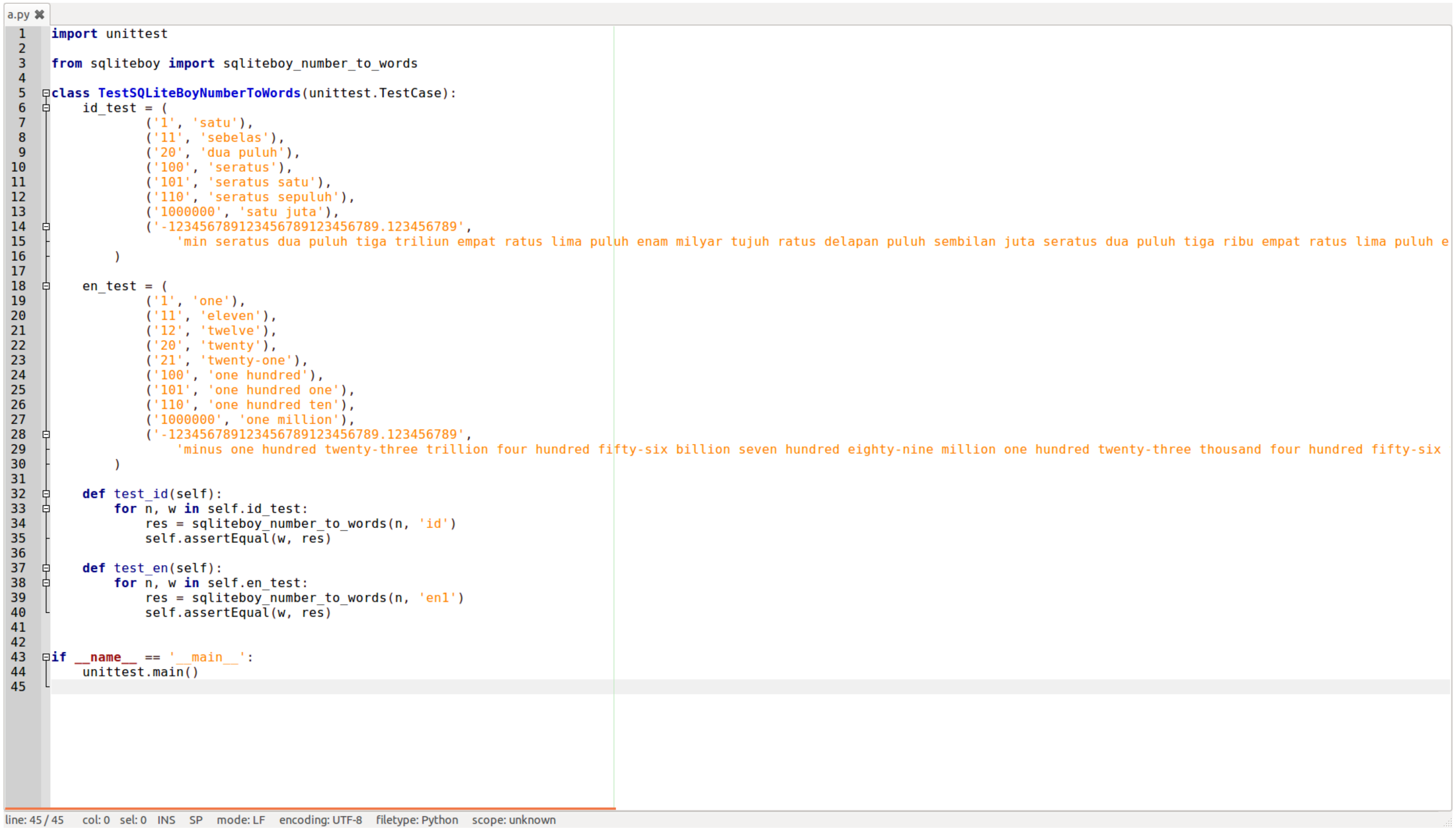


```
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```

SQLiteBoy

- https://github.com/nopri/sqliteboy
- Simple Web SQLite Manager/Form/Report Application
- Single Python file (> 11,000 lines of code), without any modern test case
- Old codes, non-descriptive variable/function names
- Lets test number to words functionality

Test case: better late than never

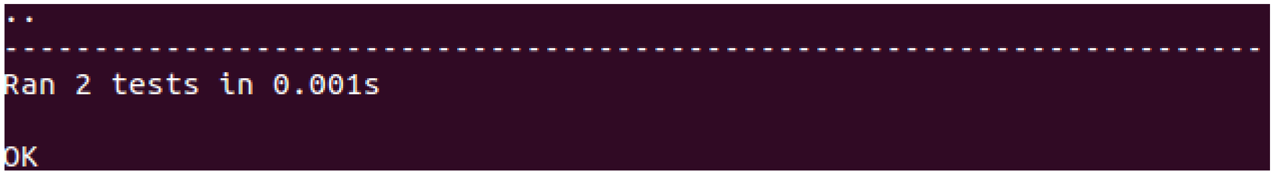


a.py

```python
import unittest

from sqliteboy import sqliteboy_number_to_words

class TestSQLiteBoyNumberToWords(unittest.TestCase):
    id_test = (
            ('1', 'satu'),
            ('11', 'sebelas'),
            ('20', 'dua puluh'),
            ('100', 'seratus'),
            ('101', 'seratus satu'),
            ('110', 'seratus sepuluh'),
            ('1000000', 'satu juta'),
            ('-123456789123456789123456789.123456789',
                'min seratus dua puluh tiga triliun empat ratus lima puluh enam milyar tujuh ratus delapan puluh sembilan juta seratus dua puluh tiga ribu empat ratus lima puluh e
        )

    en_test = (
            ('1', 'one'),
            ('11', 'eleven'),
            ('12', 'twelve'),
            ('20', 'twenty'),
            ('21', 'twenty-one'),
            ('100', 'one hundred'),
            ('101', 'one hundred one'),
            ('110', 'one hundred ten'),
            ('1000000', 'one million'),
            ('-123456789123456789123456789.123456789',
                'minus one hundred twenty-three trillion four hundred fifty-six billion seven hundred eighty-nine million one hundred twenty-three thousand four hundred fifty-six
        )

    def test_id(self):
        for n, w in self.id_test:
            res = sqliteboy_number_to_words(n, 'id')
            self.assertEqual(w, res)

    def test_en(self):
        for n, w in self.en_test:
            res = sqliteboy_number_to_words(n, 'en1')
            self.assertEqual(w, res)


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

line: 45 / 45   col: 0   sel: 0   INS   SP   mode: LF   encoding: UTF-8   filetype: Python   scope: unknown

Test case: OK



```
..
----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK
```