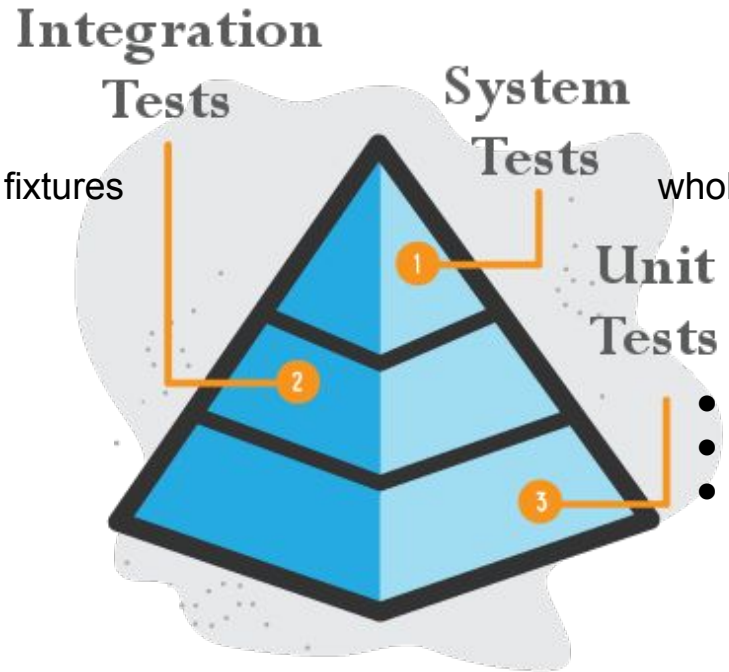
pytest

Что умеет?



shared fixtures

whole-app fixtures

- monkeypatching
- mocking
- fixtures

- automatic discovery
- plugins
- documentation

Преимущества

pytest	unittest
cross-python	tied to installed python
pythonic style	junit style
test resource injection	xUnit setup only

```
$ pip install pytest

$ py.test --version
This is pytest version 2.9.2
```

- Появился как часть **py library**
- После переноса в отдельный проект название сохранили для обратной совместимости

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_sample.py
def func(x):
  return x + 1

def test_answer():
  assert func(3) == 5

Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_sample.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 1 items

test_sample.py F

================================== FAILURES ===================================
_____________________________________ test_answer _____________________________________

    def test_answer():
>     assert func(3) == 5
E     assert 4 == 5
E      +  where 4 = func(3)

test_sample.py:5: AssertionError
========================== 1 failed in 0.05 seconds ===========================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_smart.py
def test_list_compare():
  l = list(range(10))
  l2 = list(range(5)) + list(range(7, 10))
  assert l == l2
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_smart.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 1 items

test_smart.py F

================================== FAILURES ===================================
_____________________________ test_list_compare _____________________________

    def test_list_compare():
      l = list(range(10))
      l2 = list(range(5)) + list(range(7, 10))
>     assert l == l2
E     assert [0, 1, 2, 3, 4, 5, ...] == [0, 1, 2, 3, 4, 7, ...]
E       At index 5 diff: 5 != 7
E       Left contains more items, first extra item: 8
E       Use -v to get the full diff

test_smart.py:4: AssertionError
=========================== 1 failed in 0.05 seconds ==========================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_strings.py
def test_strings():
  x = 'qwertyuiop'
  y = 'qwertAuiop'
  assert x == y
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_strings.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 1 items

test_strings.py F

================================== FAILURES ===================================
_____________________________________ test_strings _____________________________________

    def test_strings():
      x = 'qwertyuiop'
      y = 'qwertAuiop'
>     assert x == y
E     assert 'qwertyuiop' == 'qwertAuiop'
E       - qwertyuiop
E       ?       ^
E       + qwertAuiop
E       ?       ^

test_strings.py:4: AssertionError
========================== 1 failed in 0.11 seconds ===========================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_foo.py
class Foo:
  def __init__(self, val):
    self.val = val

def test_compare():
  f1 = Foo(1)
  f2 = Foo(2)
  assert f1 == f2

Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat conftest.py
from test_foo import Foo

def pytest_assertrepr_compare(op, left, right):
  if isinstance(left, Foo) and isinstance(right, Foo) and op == '==':
    return ['Comparing Foo val attributes:',
            '    vals: %s != %s' % (left.val, right.val)]
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_foo.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 1 items

test_foo.py F

================================== FAILURES ===================================
_____________________________________ test_compare _____________________________________

    def test_compare():
      f1 = Foo(1)
      f2 = Foo(2)
>     assert f1 == f2
E     assert Comparing Foo val attributes:
E           vals: 1 != 2

test_foo.py:8: AssertionError
========================== 1 failed in 0.03 seconds ===========================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_simplefactory.py
def test_function(myfuncarg):
  assert myfuncarg == 17

def pytest_funcarg__myfuncarg(request):
  return 42

Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_simplefactory.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 1 items

test_simplefactory.py F

================================== FAILURES ===================================
_____________________________ test_function ______________________________

myfuncarg = 42

    def test_function(myfuncarg):
>     assert myfuncarg == 17
E     assert 42 == 17

test_simplefactory.py:2: AssertionError
========================== 1 failed in 0.03 seconds ===========================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_parametrize.py
import pytest

@pytest.mark.parametrize('numiter', range(10))
def test_func(numiter):
  assert numiter < 9

Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test test_parametrize.py
============================= test session starts =============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 10 items

test_parametrize.py .........F

================================== FAILURES ===================================
_____________________________________ test_func[9] _____________________________________

numiter = 9

    @pytest.mark.parametrize('numiter', range(10))
    def test_func(numiter):
>     assert numiter < 9
E     assert 9 < 9

test_parametrize.py:5: AssertionError
==================== 1 failed, 9 passed in 0.07 seconds =====================
```

```
Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ cat test_xdist.py
import time
import pytest

@pytest.mark.parametrize('x', range(5))
def test_xdist(x):
  time.sleep(x)

Anastasia@Asus_X200 /cygdrive/d/Projects/pytest-review
$ py.test --durations=4 test_xdist.py
============================= test session starts ============================
platform win32 -- Python 3.5.3, pytest-2.8.5, py-1.4.31, pluggy-0.3.1
rootdir: D:\Projects\pytest-review, inifile:
collected 5 items

test_xdist.py .....

========================== slowest 4 test durations ==========================
4.00s call     test_xdist.py::test_xdist[4]
3.00s call     test_xdist.py::test_xdist[3]
2.00s call     test_xdist.py::test_xdist[2]
1.00s call     test_xdist.py::test_xdist[1]
========================== 5 passed in 10.06 seconds ==========================
```