Test Driving Pytest

```
1  git clone https://github.com/2achary/guest_list.git
```

Project
Guest List
https://github.com/2achary/guest_list.git

```
2  git clone https://github.com/2achary/guest_list.git
```

Pytest Installation

```
pip install pytest
```

```
git clone https://github.com/2achary/guest_list.git
```

Project structure

```
guest_list
├── guest_list.py
└── tests
    └── test_guest_list.py
```

```
4  git clone https://github.com/2achary/guest_list.git
```

Test Discovery

Pytest will check inside a python module if the name starts with **`test_*`** or ends with **`*_test`**

Inside a test module, pytest considers these test cases:

- Functions and methods starting with **`test_*`**
- Classes starting with **`Test*`**


```
git clone https://github.com/2achary/guest_list.git
```

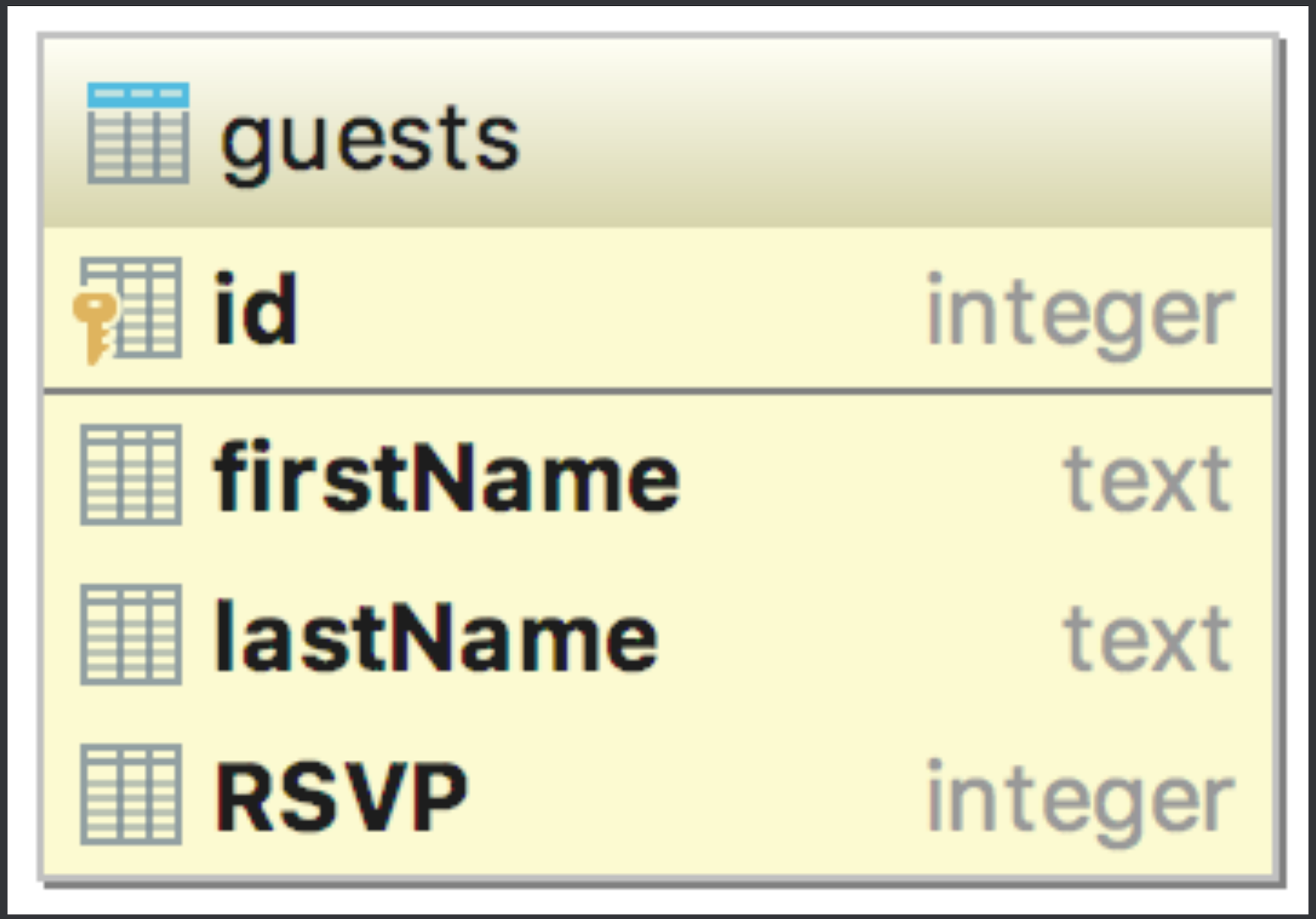
Functions

guest_list.py

- insert_guest
- delete_guest
- get_guest
- get_guest_list
- rsvp


```
git clone https://github.com/2achary/guest_list.git
```

Guests table




```
7  git clone https://github.com/2achary/guest_list.git
```

guest_list.py

```
import sqlite3


conn = sqlite3.connect('guest_list.db')
cursor = conn.cursor()
```

```
def insert_guest(first_name, last_name):
    query = """INSERT INTO guests VALUES (null, ?, ?, null)"""

    with conn:
        result = cursor.execute(query, (first_name, last_name,))
        return result.lastrowid
```



```
git clone https://github.com/2achary/guest_list.git
```

For a first test:

- Call the **`insert_guest`** function with a first and last name
- Check that guest list has one guest in it
- Check that the one guest is the one we inserted

```
9  git clone https://github.com/2achary/guest_list.git
```

tests/test_guest_list.py

```
def test_insert_guest_saves_guest():

    # Call the function being tested

    # Get the guest list straight from the
    # db and make sure there's only 1 result

    # Assert the database row is what we inserted
    pass
```

 git clone https://github.com/2achary/guest_list.git

```python
from guest_list import insert_guest

def test_insert_guest_saves_guest():
    # Call the function being tested
    inserted_id = insert_guest('Rick', 'Grimes')

    # Get the guest list straight from the
    # db and make sure there's only 1 item in it

    # Assert the database row is what we inserted
    pass
```

11 `git clone https://github.com/2achary/guest_list.git`

```
import sqlite3
...
def test_insert_guest_saves_guest():
    # Set up the connection and table
    conn = sqlite3.connect('guest_list.db')
    cursor = conn.cursor()

    with conn:
        cursor.execute("DROP TABLE IF EXISTS guests")
        cursor.execute('''CREATE TABLE if not exists guests
                       (id INTEGER PRIMARY KEY firstName TEXT,
                       lastName TEXT, RSVP integer)''')

    # Call the function being tested
    inserted_id = insert_guest('Rick', 'Grimes')

    # Get the guest list
    with conn:
        guest_list = list(cursor.execute("SELECT * FROM guests"))

        # verify there's only 1 item in the guest list
        assert len(guest_list) == 1

        # Assert the database row is what we inserted
        assert guest_list[0] == (1, 'Rick', 'Grimes', None)
```

```
12 git clone https://github.com/2achary/guest_list.git
```

Project structure

```
$ tree -L 2 guest_list
guest_list <--- WORKING DIRECTORY
├── guest_list.py
└── tests <--- DIRECTORY TO POINT PYTEST TO
    └── test_guest_list.py <--- starts with test_*
```

```
13 git clone https://github.com/2achary/guest_list.git
```

$ pytest tests

```
14 git clone https://github.com/2achary/guest_list.git
```

Import error

```
(guest_list) zswift@zachs-MacBook-Pro:guest_list$ pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 0 items / 1 errors

==================================== ERRORS ====================================
_________________ ERROR collecting tests/test_guest_list.py _________________
ImportError while importing test module '/Users/zswift/projects/personalProjec
ts/guest_list/tests/test_guest_list.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
tests/test_guest_list.py:2: in <module>
    from guest_list import insert_guest
E   ImportError: No module named 'guest_list'
!!!!!!!!!!!!!!!!!!!! Interrupted: 1 errors during collection !!!!!!!!!!!!!!!!!!!!
=========================== 1 error in 0.13 seconds ============================
```

 git clone https://github.com/2achary/guest_list.git

Project structure

```
$ tree -L 2 guest_list
guest_list
├── guest_list.py
└── tests
    └── test_guest_list.py <--- we're trying to import
                               a guest_list.py funciton
                               from here which won't work
```



```
git clone https://github.com/2achary/guest_list.git
```

$ python -m pytest tests

The only difference, is this way adds the CWD to python's search path

```
17 git clone https://github.com/2achary/guest_list.git
```

Standard passing test output

```
(guest_list) zswift@zachs-MacBook-Pro:guest_list$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 1 items

tests/test_guest_list.py .

=========================== 1 passed in 0.01 seconds ===========================
```


git clone https://github.com/2achary/guest_list.git

Setting up the db connection

Creating the table

```
import sqlite3
from guest_list import insert_guest

def test_insert_guest_saves_guest():
    conn = sqlite3.connect('guest_list.db')
    cursor = conn.cursor()

    TABLEDEF = '''CREATE TABLE if not exists guests
                  (firstName text, lastName text, RSVP integer)'''
    with conn:
        cursor.execute(TABLEDEF)

    # Call the function being tested
    result = insert_guest('Rick', 'Grimes')

    # Get the guest list straight from the db and make sure there's only 1
    # item in it
    with conn:
        guest_list = list(cursor.execute("SELECT * FROM guests"))

    assert len(guest_list) == 1

    # Assert the database row is what we inserted
    pass
```

Name strings

Getting all the guests in a list

Fixtures

- Fixtures are functions with a **`@pytest.fixture`** decorator
- Test function can use fixtures by including the fixture name in their argument list.
- Inside the test case, the name of the fixture evaluates to what the fixture returns

 `git clone https://github.com/2achary/guest_list.git`

A Fixture for the sqlite connection

```
21 git clone https://github.com/2achary/guest_list.git
```

```
...

import pytest

@pytest.fixture
def conn():
    return sqlite3.connect('guest_list.db')

def test_insert_guest_saves_guest(conn):
    cursor = conn.cursor()

    ....
```



```
git clone https://github.com/2achary/guest_list.git
```

A Fixture for the cursor

```
23 git clone https://github.com/2achary/guest_list.git
```

```
...
@pytest.fixture
def conn():
    return sqlite3.connect('guest_list.db')

@pytest.fixture
def cursor(conn):
    return conn.cursor()

def test_insert_guest_saves_guest(conn, cursor):
    TABLEDEF = '''
        CREATE TABLE if not exists guests
        (firstName text, lastName text, RSVP integer)'''
    with conn:
        cursor.execute(TABLEDEF)
...
```

 git clone https://github.com/2achary/guest_list.git

conftest.py

Magical file where all test modules know to look for fixtures

- If a fixture is defined in this file, test modules don't even need to import it, they can simply ask for it in their function arguments

```
25 git clone https://github.com/2achary/guest_list.git
```

Adding a conftest.py file to the tests directory

```
$ tree -L 2 guest_list
guest_list
├── guest_list.py
└── tests
    ├── conftest.py
    └── test_guest_list.py
```

 `git clone https://github.com/2achary/guest_list.git`

Move the fixtures to conftest.py

`tests/conftest.py`

```
import pytest
import sqlite3


@pytest.fixture()
def conn():
    return sqlite3.connect('guest_list.db')


@pytest.fixture
def cursor(conn):
    return conn.cursor()
```



```
git clone https://github.com/2achary/guest_list.git
```

Make sure the test still passes

```
(guest_list) zswift@zachs-MBP:guest_list$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/guest_list, inifile:
collected 1 items

tests/test_slides.py .

=========================== 1 passed in 0.02 seconds ===========================
```

28 **git clone https://github.com/2achary/guest_list.git**

```python
@pytest.fixture
def clean_guest_list(conn, cursor):
    # define nested function to drop table
    def drop_table():
        with conn:
            cursor.execute("DROP TABLE IF EXISTS guests")

    # drop the table before creating a new one
    drop_table()

    # create a new table
    TABLEDEF = '''CREATE TABLE IF NOT EXISTS guests(
                  id INTEGER PRIMARY KEY, firstName TEXT,
                  lastName TEXT, RSVP INTEGER)'''
    with conn:
        cursor.execute(TABLEDEF)

    yield # <-- The fixture pauses here lets the test run

    # drop the table after the test is finished
    drop_table()
```



```
git clone https://github.com/2achary/guest_list.git
```

```python
from guest_list import insert_guest

def test_insert_guest_saves_guest(clean_guest_list, conn, cursor):
    # Call the function being tested
    result = insert_guest('Rick', 'Grimes')

    # Get the guest list straight from the db and make sure
    # there's only 1 item in it
    with conn:
        guest_list = list(cursor.execute("SELECT * FROM guests"))

    assert len(guest_list) == 1

    # Assert the database row is what we inserted
    assert guest_list[0] == (1, 'Rick', 'Grimes', None)
```


git clone https://github.com/2achary/guest_list.git

```python
@pytest.fixture
def first_name():
    return 'Rick'

@pytest.fixture
def last_name():
    return 'Grimes'

@pytest.fixture
def database_id():
    return 1
...
```

```python
@pytest.fixture
def expected_db_row(database_id, first_name, last_name):
    return database_id, first_name, last_name, None

@pytest.fixture
def guest_list_from_db(conn, cursor):
    def inner():
        with conn:
            return list(cursor.execute("SELECT * FROM guests"))
    return inner # So that it can be called multiple times
```



```
git clone https://github.com/2achary/guest_list.git
```

Using all the fixtures

```
from guest_list import insert_guest

def test_insert_guest_saves_guest(clean_guest_list, first_name,
                                  last_name, guest_list_from_db,
                                  expected_db_row, database_id):
    # Call the function being tested
    inserted_id = insert_guest(first_name, last_name)
    assert inserted_id == database_id

    # Get the guest list straight from the db and make sure there's only 1
    # item in it
    guest_list = guest_list_from_db()
    assert len(guest_list) == 1

    # Assert the database row is what we inserted
    assert guest_list[0] == expected_db_row
```



```
git clone https://github.com/2achary/guest_list.git
```

delete_guest

guest_list.py

```
def delete_guest(first_name, last_name):
    query = "DELETE FROM guests WHERE firstName=? and lastName=?"
    with conn:
        result = cursor.execute(query, (first_name, last_name,))
        if result.rowcount == 0:
            raise ValueError("Guest not found")
```



```
git clone https://github.com/2achary/guest_list.git
```

Asserting an Exception is raised

tests/test_guest_list.py

```
from guest_list import insert_guest, delete_guest
...
def test_delete_guest_raises(clean_guest_list, first_name, last_name):
    # There hasn't been data pre-populated, so this should throw and error
    with pytest.raises(ValueError) as e:
        delete_guest(first_name, last_name)
    assert 'Guest not found' in str(e)
```


git clone https://github.com/2achary/guest_list.git

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
Kitematic (Beta)
========================= test session starts ==========================

platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 2 items

tests/test_guest_list.py ..

======================== 2 passed in 0.03 seconds ======================
```

35 **git clone https://github.com/2achary/guest_list.git**

tests/test_guest_list.py

```
def test_delete_guest_deletes(clean_guest_list, insert_one_guest,
                              guest_list_from_db, first_name, last_name,
                              expected_db_row):
    # Insert a guest
    insert_one_guest()

    # Assert there is a guest <--sanity check that the fixture worked
    guest_list = guest_list_from_db()
    assert len(guest_list) == 1
    assert guest_list[0] == expected_db_row

    # call delete_guest
    delete_guest(first_name, last_name)

    # assert there are no guests in the db
    guest_list = guest_list_from_db()
    assert len(guest_list) == 0
```


git clone https://github.com/2achary/guest_list.git

test/conftest.py

```
@pytest.fixture
def insert_one_guest(first_name, last_name, conn, cursor):
    def inner():
        with conn:
            cursor.execute('''INSERT INTO guests VALUES
                              (null, ?, ?, null)''', (first_name, last_name,))
    return inner
```

```
37 git clone https://github.com/2achary/guest_list.git
```

```
========================== test session starts ==========================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 3 items

tests/test_guest_list.py ...

======================== 3 passed in 0.04 seconds ========================
```


git clone https://github.com/2achary/guest_list.git

guest_list.py

```
def get_guest(first_name, last_name):
    query = "SELECT * FROM guests WHERE firstName=? AND lastName=?"
    with conn:
        result = cursor.execute(query, (first_name, last_name,))
        try:
            return next(result)
        except StopIteration:
            raise ValueError
```



```
git clone https://github.com/2achary/guest_list.git
```

tests/test_guest_list.py

```
def test_get_guest_raises_if_not_found(clean_guest_list, first_name,
                                       last_name):
    with pytest.raises(ValueError):
        get_guest(first_name, last_name)

def test_get_guest_finds_existing_guest(clean_guest_list, first_name,
                                        last_name, insert_one_guest,
                                        guest_list_from_db, expected_db_row):
    # Insert a guest so there is one to find
    insert_one_guest()
    assert len(guest_list_from_db()) == 1

    # Call the function and assert the return value is the guest inserted
    result = get_guest(first_name, last_name)
    assert result == expected_db_row
```

 git clone https://github.com/2achary/guest_list.git

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 5 items

tests/test_guest_list.py .....

=========================== 5 passed in 0.05 seconds ===========================
```



```
git clone https://github.com/2achary/guest_list.git
```

```
def get_guest_list():
    query = 'SELECT * FROM guests'
    with conn:
        return list(cursor.execute(query))
```

```
42 git clone https://github.com/2achary/guest_list.git
```

```python
def test_get_guest_list_returns_empty_list_if_no_guests(clean_guest_list):
    result = get_guest_list()
    assert result == []

def test_get_guest_list_returns_guest_list(clean_guest_list, insert_one_guest,
                                           expected_db_row):
    # Insert a couple guests (the same guest twice..)
    insert_one_guest()
    insert_one_guest()

    # Assert the guest list has two guests and they're both the same
    guest_list = get_guest_list()
    assert len(guest_list) == 2
    assert all(guest[1:] == expected_db_row[1:] for guest in guest_list)
```



```
git clone https://github.com/2achary/guest_list.git
```

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 7 items

tests/test_guest_list.py .......

=========================== 7 passed in 0.06 seconds ===========================
```



```
git clone https://github.com/2achary/guest_list.git
```

RSVP

For the last one, let's write the test first

The rsvp() function will need to take a first name, last name and rsvp boolean and update the matching guest's rsvp field in the database

- True means the guest is coming
- False means the guest is not coming

 `git clone https://github.com/2achary/guest_list.git`

tests/test_guest_list.py

```
from guest_list import (insert_guest,
                        delete_guest,
                        get_guest,
                        get_guest_list,
                        rsvp)

...

def test_rsvp_raises_if_guest_not_found(clean_guest_list,
                                        first_name,
                                        last_name):
    # Call rsvp without inserting any guests
    with pytest.raises(ValueError):
        rsvp(first_name, last_name, True)
```

 git clone https://github.com/2achary/guest_list.git

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 0 items / 1 errors

==================================== ERRORS ====================================
_____________________ ERROR collecting tests/test_guest_list.py _____________________
ImportError while importing test module '/Users/zswift/projects/personalProj
ects/guest_list/tests/test_guest_list.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
tests/test_guest_list.py:1: in <module>
    from guest_list import (insert_guest,
E   ImportError: cannot import name 'rsvp'
!!!!!!!!!!!!!!!!!!!! Interrupted: 1 errors during collection !!!!!!!!!!!!!!!!!!!!
=========================== 1 error in 0.14 seconds ============================
```


git clone https://github.com/2achary/guest_list.git

guest_list.py

```
def rsvp():
    pass
```

48 **git clone https://github.com/2achary/guest_list.git**

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 8 items

tests/test_guest_list.py .......F
```

SourceTree

```
=================================== FAILURES ===================================
_____________________ test_rsvp_raises_if_guest_not_found ______________________
clean_guest_list = None, first_name = 'Rick', last_name = 'Grimes'

    def test_rsvp_raises_if_guest_not_found(clean_guest_list, first_name,
                                            last_name):
        # Call rsvp without inserting any guests
        with pytest.raises(ValueError):
>           rsvp(first_name, last_name, True)
E           TypeError: rsvp() takes 0 positional arguments but 3 were given

tests/test_guest_list.py:100: TypeError
==================== 1 failed, 7 passed in 0.08 seconds ====================
```

```
49 git clone https://github.com/2achary/guest_list.git
```

```
def rsvp(first_name, last_name, answer):
    pass
```

50 **git clone https://github.com/2achary/guest_list.git**

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 8 items

tests/test_guest_list.py .......F

=================================== FAILURES ===================================
_____________________ test_rsvp_raises_if_guest_not_found ______________________

clean_guest_list = None, first_name = 'Rick', last_name = 'Grimes'

    def test_rsvp_raises_if_guest_not_found(clean_guest_list, first_name,
                                            last_name):
        # Call rsvp without inserting any guests
        with pytest.raises(ValueError):
>           rsvp(first_name, last_name, True)
E           Failed: DID NOT RAISE <class 'ValueError'>

tests/test_guest_list.py:100: Failed
====================== 1 failed, 7 passed in 0.08 seconds ======================
```

```
51 git clone https://github.com/2achary/guest_list.git
```

```
def rsvp(first_name, last_name, answer):
    raise(ValueError)
```

```
52 git clone https://github.com/2achary/guest_list.git
```

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 8 items

tests/test_guest_list.py ........

========================== 8 passed in 0.06 seconds ===========================
```

```
53  git clone https://github.com/2achary/guest_list.git
```

```
def test_rsvp_changes_answer_field(clean_guest_list,
                                   first_name, last_name,
                                   insert_one_guest,
                                   guest_list_from_db):
    # Insert a guest so we have someone to submit an RSVP for
    insert_one_guest()

    # Submit an RSVP and assert the status is True
    rsvp(first_name, last_name, True)
    guest_list = guest_list_from_db()
    assert guest_list[0] == (1, first_name, last_name, True)

    # Change the RSVP status and assert it is updated accordingly
    rsvp(first_name, last_name, False)
    guest_list = guest_list_from_db()
    assert guest_list[0] == (1, first_name, last_name, False)
```


git clone https://github.com/2achary/guest_list.git

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
first_name = 'Rick', last_name = 'Grimes', answer = True

    def rsvp(first_name, last_name, answer):
>       raise(ValueError)
E       ValueError

guest_list.py:52: ValueError
====================== 1 failed, 8 passed in 0.09 seconds =
```

55 `git clone https://github.com/2achary/guest_list.git`

```
def rsvp(first_name, last_name, answer):
    answer = 1 if answer else 0
    query = "UPDATE guests SET rsvp=? WHERE firstName=? AND lastName=?"
    with conn:
        result = cursor.execute(query, (answer, first_name, last_name))
        if result.rowcount == 0:
            raise ValueError("Guest not found")
```

```
56 git clone https://github.com/2achary/guest_list.git
```

Right now we have 9 passing tests

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 9 items

tests/test_guest_list.py .........

=========================== 9 passed in 0.06 seconds ===========================
```

 git clone https://github.com/2achary/guest_list.git

```
def test_rsvp_changes_answer_field(clean_guest_list,
                                   first_name, last_name,
                                   insert_one_guest,
                                   guest_list_from_db):
    # Insert a guest so we have someone to submit an RSVP for
    insert_one_guest()

    # Submit an RSVP and assert the status is True
    rsvp(first_name, last_name, True)
    guest_list = guest_list_from_db()
    assert guest_list[0] == (1, first_name, last_name, True)

    # Change the RSVP status and assert it is updated accordingly
    rsvp(first_name, last_name, False)
    guest_list = guest_list_from_db()
    assert guest_list[0] == (1, first_name, last_name, False)
```


git clone https://github.com/2achary/guest_list.git

Parameterized fixtures

```python
@pytest.fixture(params=[True, False])
def rsvp_status(request):
    return request.param
```



```
git clone https://github.com/2achary/guest_list.git
```

```
def test_rsvp_changes_answer_field(clean_guest_list, first_name, last_name,
                                   insert_one_guest, guest_list_from_db,
                                   rsvp_status):
    # Insert a guest so we have someone to submit an RSVP for
    insert_one_guest()

    # Submit an RSVP and assert the status set correctly
    rsvp(first_name, last_name, rsvp_status)
    guest_list = guest_list_from_db()
    assert guest_list[0] == (1, first_name, last_name, rsvp_status)
```

```
60 git clone https://github.com/2achary/guest_list.git
```

Now we have 10 even though we actually have 9 test functions

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests
============================= test session starts ==============================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 10 items

tests/test_guest_list.py ..........

=========================== 10 passed in 0.08 seconds ==========================
```

 git clone https://github.com/zachary/guest_list.git

Test running options

```
62 git clone https://github.com/2achary/guest_list.git
```

-v Verbose. Displays status of each function

```
========================= 10 passed in 0.08 seconds ==========================
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests -v
Microsoft Outlook
========================= test session starts ==========================
platform darwin -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0 -- /U
sers/zswift/virtualenvs/guest_list/bin/python
cachedir: .cache
rootdir: /Users/zswift/projects/personalProjects/guest_list, inifile:
collected 10 items

tests/test_guest_list.py::test_insert_guest_saves_guest PASSED
tests/test_guest_list.py::test_delete_guest_raises PASSED
tests/test_guest_list.py::test_delete_guest_deletes PASSED
tests/test_guest_list.py::test_get_guest_raises_if_not_found PASSED
tests/test_guest_list.py::test_get_guest_finds_existing_guest PASSED
tests/test_guest_list.py::test_get_guest_list_returns_empty_list_if_no_guest
s PASSED
tests/test_guest_list.py::test_get_guest_list_returns_guest_list PASSED
tests/test_guest_list.py::test_rsvp_raises_if_guest_not_found PASSED
tests/test_guest_list.py::test_rsvp_changes_answer_field[True] PASSED
tests/test_guest_list.py::test_rsvp_changes_answer_field[False] PASSED

========================= 10 passed in 0.06 seconds ==========================
```

python -m pytest test/<path-to-function>

63 **git clone https://github.com/2achary/guest_list.git**

-vv Really verbose

Won't truncate Failure diffs

```
64 git clone https://github.com/2achary/guest_list.git
```

-q

Quiet Mode. Just gives you dots for passing tests, **F**, for failures and **E** for errors. Also displays the time it took to run the tests

```
(guest_list) zachs-MBP:guest_list zswift$ python -m pytest tests -q
..........
10 passed in 0.06 seconds
```

I'm addicted to dots. If you don't know what that means, you need to write more unit tests.

— Chad Whitacre, Pycon 2007

66 `git clone https://github.com/2achary/guest_list.git`

-s

Lets print statements through.

```
67 git clone https://github.com/2achary/guest_list.git
```

monkeypatch

Built-in fixture

```
from unittest.mock import MagicMock, ANY

def test_insert_guest_with_mock(monkeypatch):
    cursor_mock = Mock()
    monkeypatch.setattr('guest_list.sqlite3.connect', MagicMock())
    monkeypatch.setattr('guest_list.cursor', cursor_mock)

    insert_guest('test', 'test')
    cursor_mock.execute.assert_called_with(ANY, ('test', 'test',))
```

I usually go with patch from **unittest.mock.patch**

 git clone https://github.com/2achary/guest_list.git

Things I would have loved to go into

- Skipping integration tests
  - Making custom decorators to conditionally skip tests based on command line options
- Fixture scope options
  - **function**: Run once per test
  - **class**: Run once per class of tests
  - **module**: Run once per module
  - **session**: Run once per session

 `git clone https://github.com/2achary/guest_list.git`

- Things to watch out for when using unittest.mock with pytest
  - some mock methods have built-in assertions that you can't use an assert statement with
  - The **`@mock.patch`** decorator messes up the arg list by inserting the mock object at the beginning. I'd recommend using the context manager instead.

 `git clone https://github.com/2achary/guest_list.git`

```
71 git clone https://github.com/2achary/guest_list.git
```