Software Testing in Python

Dr. Kristian Rother & Dr. Magdalena Rother

„The program works perfectly.”

Count the words in that sentence.

„That #§&%$* program still doesn't work!\nI already de-bugged it 3 times, and still numpy.array keeps raising AttributeErrors. What should I do?”

Count the words in that sentence.

Objectives

1) Automatic Tests

Implement tests with nose

2) Test-Driven-Development

Apply TDD:write a test first.

3) Tests in the development process

Calculate test coverage.

Materials: www.academis.eu/testing

Part I: Automated Tests with nose

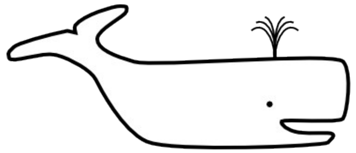
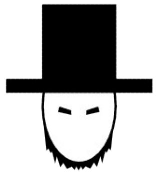
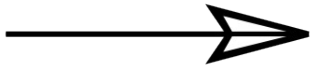
1 **Unit Tests**
test one piece of code in isolation

1+1 **Integration Tests**
test whether two units work together

☑ **Acceptance Tests**
test a user-level feature

Materials: www.academis.eu/testing

Word Frequencies

word	***frequency***
 *whale*	𝍸
 *captain*	𝍷𝍷𝍷𝍷
 *harpoon*	𝍷

Materials: www.academis.eu/testing

Task 1.1: Unit Tests

Download and unpack the example code sniplets.

Run the first example:

```
nosetests test_example.py
```

Task 1.2: TestCase subclasses

Fill in the blanks in the file `test_unit.py`

```
nosetests test_unit.py
```

Task 1.3: Integration Tests

Fix the code so that the tests pass

```
nosetests test_integration.py
```

Task 1.4: Acceptance Tests

Run the acceptance test example

```
nosetests test_acceptance.py
```

Task 1.5: Write your own tests

Test the module `word_report.py`

Write an Integration Test and an Acceptance Test.

Bonus question:
How would you write a Unit Test for `get_top_words()` that does not use `TextBody`?

Part II: Test-Driven Development

Border cases
write quality tests

Fixtures
prepare and clean up

Test generators
cover many examples with few lines

TDD
Write a test first, then the code

Materials: www.academis.eu/testing

Task 2.1: Border Cases

Fill in the assert functions

```
nosetests test_border_cases.py
```

Use all functions that are imported.

Task 2.2: Fixtures

Run the code example

```
nosetests test_fixtures.py
```

Task 2.3: Test Generators

Run the code example

```
nosetests test_generator.py
```

ACADEMIS

Task 2.4: New Feature

Make TextBody remove special characters automatically.

1. Write a test
2. Run and make sure it fails
3. Write the code
4. Run the test and make sure it passes
5. Repeat until time runs out

Part III: Tests in the Development Process

Test Coverage
How much of your code is tested?

Test Suites
running a lot of tests

Packages
testing in a setup script

Materials: www.academis.eu/testing

Task 3.1: Test Coverage

Calculate test coverage

```
nosetests test_generator.py
```

(see handout)

Task 3.2: Test suites

Run all tests

```
nosetests -v
```

Task 3.3: Packaging

Write a script that runs all tests when you type:

```
python setup.py test
```

(see handout)

ACADEMIS

Task 3.4: User Acceptance

Use the program

`word_counter.py`

to find out whether Melville used 'whale' or 'captain' more frequently in his book.

Materials: www.academis.eu/testing

Other Testing Topics

- Performance tests
- Concurrency tests (gevent)
- Web testing (splinter, selenium)
- Doctests
- Other frameworks: py.test, unittest
- Continuous Integration (CI)

Objectives

1) Automatic Tests

Implement tests with nose

2) Test-Driven-Development

Apply TDD:write a test first.

3) Tests in the development process

Calculate test coverage.

Materials: www.academis.eu/testing

Testing Strategies

Types of Tests

- Unit Tests
- Integration Tests
- Acceptance Tests
- Regression tests

Quality Tests

- Write tests first
- 1 assert per test
- Border cases
- Test data
- Mock objects

Development

- Test coverage
- Test suite
- Setup script
- User Acceptance

Features of nose

- Test functions
- assert_xxx functions from nose.tools
- TestCase subclasses
- Fixtures
- Test generators
- Test detection
- Test selection

Tests help you when

- Writing code
- Debugging
- Refactoring
- Maintaining software
- Teamwork

ACADEMIS **_Dr. Kristian Rother_**

Teaching & Training

Software Testing

Leadership

www.academis.eu **_@k_rother_**