The `pytest` Package

> Prerequisite: Automated Testing

Reference:

- https://github.com/pytest-dev/pytest/
- https://docs.pytest.org/en/latest/
- https://docs.pytest.org/en/latest/getting-started.html#our-first-test-run
- https://docs.pytest.org/en/latest/goodpractices.html
- https://docs.pytest.org/en/latest/usage.html#dropping-to-pdb-python-debugger-on-failures
- http://python-guide-pt-br.readthedocs.io/en/latest/writing/tests/#py-test
- https://docs.pytest.org/en/latest/capture.html

Installation

If you are using Pip to manage software packages (recommended), install Pytest, as necessary:

```
pip install pytest
```

Otherwise, if you are using Pipenv, you will want to first navigate inside your repository's root directory before installing Pytest:

```
cd path/to/my-repo/
pipenv install pytest --dev # NOTE: the --dev flag denotes this package will
be used in development only
```

Usage

The Pytest package is generally used as a command-line utility for running pre-defined files of "test" code. Follow the "Testing 1,2,3" Exercise to get acclimated with Pytest.

Example invocations:

```
pytest

pytest --disable-pytest-warnings # disables warnings

pytest  -s #> see print statements
```

```
pytest test/parser_test.py -k 'test_my_thing' # test a certain file /
function
```

Expecting Errors

The Pytest package can be imported to facilitate assertions that errors will be raised:

```
import pytest

def test_divide_by_zero():
    with pytest.raises(Exception) as e_info:
        result = 2 / 0 # we expect this code will raise the error (division
by zero)
```

Fixtures

The Pytest package can be imported to facilitate the construction of test fixtures (for example, to be placed in the "conftest.py" file):

```
import pytest

# prevents unnecessary or duplicative language model loading
# fixture only loaded when a specific test needs it
# module-level fixture only invoked once for all tests
@pytest.fixture(scope="module")
def nlp():
    import spacy
    print("LOADING THE LANGUAGE MODEL...")
    return spacy.load("en_core_web_md")

def test_my_thing(nlp):
    doc = nlp(reconstructed_text)
    # etc
```