

shUnit2

佐野 尚之



本ドキュメントのライセンスについて

この作品は、クリエイティブ・コモンズのAttribution 3.0 Unportedライセンスの下でライセンスされています。

この使用許諾条件を見るには、<http://creativecommons.org/licenses/by/3.0/>をチェックするか、クリエイティブ・コモンズに郵便にてお問い合わせください。

住所は：171 Second Street, Suite 300, San Francisco, California 94105, USA です。



原作者のクレジット（氏名、作品タイトルとURL）を表示することを守れば、改変はもちろん、営利目的での二次利用も許可される最も自由度の高いCCライセンス。

<http://creativecommons.jp/>



変更履歴

■第1版・・・2013/03/20

本ドキュメントは、オープンソースの「LibreOffice 4.0.1.2」を使用して作成。

目次

- ・ OS、開発環境などの対象バージョン 5
- ・ 各ソフトウェアのインストール先 および 作業フォルダについて 6
- ・ shUnit2について 7
- ・ 機能説明(一部) 8
- ・ インストール 10
- ・ プログラム作成&shUnit2実行 12
- ・ 参考情報 15



OS、開発環境、対象バージョン

■動作確認環境

Acer Aspire 1410

Windows 7 Home Premium(64bit版)

Intel Celeron processor SU2300(1.2GHz, 800MHz FSB)。8GBメモリに変更。HDD250GB。

■開発環境

Cygwin (**事前にインストールしておいてください**)

<参考情報>

Cygwinのインストール&設定方法

<http://musashi.sourceforge.jp/cygwin/cygwin.html>

Cygwin VIエディタのインストール

http://blogs.yahoo.co.jp/dadadadaddy_blog/3566369.html

各ソフトウェアのインストール先 および 作業フォルダについて

本ドキュメントの指定通りの場所ではなくても問題はありません。別のドライブやフォルダにインストールした場合は、ドライブ名やフォルダ名を読み替えてインストール後の設定を行ってください。

shUnit2について

- log4shの複数シェル対応を自動テストすることを目的として開発された。
- JUnitと同様の方法で動作するように設計されている。
- シェルスクリプト用の xUnit ユニット（単体）テストフレームワーク。
- テスト失敗した時に赤にならない。改造すれば赤くなる。以下のサイト参照。

シェルスクリプトでユニットテストを書いてみよう

<http://a4p.me/archives/1157>

機能説明(一部) (1/2)

■ Setup/Teardown

- oneTimeSetUp . . . 任意のテストが実行される前に一度だけ呼び出されます。
- oneTimeTearDown . . . 全てのテストが実行された後に一度だけ呼び出されます。
- setUp . . . 各テストが実行される前に呼び出されます。
- tearDown . . . 各テストが完了した後に呼び出されます。

■ Failures(実際の値比較はしないみたいなので、注意してください！)

- fail . . . ただちにテストを失敗させます。
- failNotEquals . . . ただちにテストを失敗させ、引数1と引数2の値が等しくないことをレポートします。
- failSame . . . ただちにテストを失敗させ、引数1と引数2の値が等しいことをレポートします。
- failNotSame . . . ただちにテストを失敗させ、引数1と引数2の値が等しくないことをレポートします。

failNotEqualsと同じ？

機能説明(一部) (2/2)

■ Asserts

- assertEquals
- assertEquals
- assertEquals . . . この関数は機能的に assertEquals と等しい関数のようです。
- assertEquals . . . この関数は機能的に assertEquals と等しい関数のようです。
- assertEquals
- assertEquals
- assertEquals
- assertEquals

インストール (1/2)

- Cygwinを起動
- 「`wget -P ~/download http://shunit2.googlecode.com/files/shunit2-2.1.6.tgz`」 を実行

```
sano@sano-PC ~
$ wget -P ~/download http://shunit2.googlecode.com/files/shunit2-2.1.6.tgz
--2013-03-20 15:24:31-- http://shunit2.googlecode.com/files/shunit2-2.1.6.tgz
shunit2.googlecode.com (shunit2.googlecode.com) を DNS に問いあわせています... 173
.194.72.82, 2404:6800:4008:c01::52
shunit2.googlecode.com (shunit2.googlecode.com) |173.194.72.82|:80 に接続していま
す... 接続しました,
HTTP による接続要求を送信しました、応答を待っています... 200 OK
長さ: 61558 (60K) [application/x-gzip]
`/home/sano/download/shunit2-2.1.6.tgz' に保存中

100%[=====>] 61,558      263K/s 時間 0.2s

2013-03-20 15:24:31 (263 KB/s) - `/home/sano/download/shunit2-2.1.6.tgz' に保存
完了 [61558/61558]
```

インストール (2/2)

- 「tar -xzvf ~/download/shunit2-2.1.6.tgz」を実行して解凍する。

```
$ tar -xzvf ~/download/shunit2-2.1.6.tgz
shunit2-2.1.6/
shunit2-2.1.6/doc/
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.2.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.5.txt
shunit2-2.1.6/doc/shunit2.txt
shunit2-2.1.6/doc/TODO.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.4.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.3.txt
shunit2-2.1.6/doc/CHANGES-2.1.txt
shunit2-2.1.6/doc/contributors.txt
shunit2-2.1.6/doc/README.html
shunit2-2.1.6/doc/README.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.0.txt
shunit2-2.1.6/doc/rst2html.css
shunit2-2.1.6/doc/shunit2.html
shunit2-2.1.6/doc/LGPL-2.1
shunit2-2.1.6/doc/design_doc.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.6.txt
shunit2-2.1.6/doc/RELEASE_NOTES-2.1.1.txt
shunit2-2.1.6/doc/coding_standards.txt
shunit2-2.1.6/examples/
shunit2-2.1.6/examples/math.inc
shunit2-2.1.6/examples/equality_test.sh
shunit2-2.1.6/examples/mkdir_test.sh
shunit2-2.1.6/examples/party_test.sh
shunit2-2.1.6/examples/lineno_test.sh
shunit2-2.1.6/examples/math_test.sh
shunit2-2.1.6/lib/
shunit2-2.1.6/lib/shflags
shunit2-2.1.6/lib/shlib
shunit2-2.1.6/lib/versions
shunit2-2.1.6/bin/
shunit2-2.1.6/bin/which
shunit2-2.1.6/bin/gen_test_results.sh
shunit2-2.1.6/src/
shunit2-2.1.6/src/shunit2_test_macros.sh
shunit2-2.1.6/src/shunit2_test_helpers
shunit2-2.1.6/src/shunit2_test_standalone.sh
shunit2-2.1.6/src/shunit2_test_asserts.sh
shunit2-2.1.6/src/shunit2
shunit2-2.1.6/src/shunit2_test.sh
shunit2-2.1.6/src/shunit2_test_failures.sh
shunit2-2.1.6/src/shunit2_test_misc.sh
```

プログラム作成&shUnit2実行 (1/3)

- viを起動

```
sano@sano-PC ~  
$ vi sample.sh
```

- 以下の内容を記述して、「ESC」キー→「wq」を入力

```
#!/bin/sh  
  
rm -f ./sample/test.txt  
if [ -f ./sample/test.txt ]; then  
    echo "ファイルが削除されていません。"  
    exit 1  
fi  
  
exit 0
```

プログラム作成&shUnit2実行 (2/3)

- ・テストシェル作成

```
oneTimeSetUp() {
  mkdir ./sample
}

oneTimeTearDown() {
  rm -rf ./sample
}

setUp() {
  echo -n > ./sample/test.txt
}

testNormalEnd()
{
  ./sample.sh
  rc=$?
  assertEquals 'return Code' ${rc} 0
}

. ./shunit2-2.1.6/src/shunit2
```

プログラム作成&shUnit2実行 (3/3)

- テストシェル実行

```
sano@sano-PC ~  
$ ./sample_test.sh  
testNormalEnd  
  
Ran 1 test.  
  
OK
```

参考情報 (1/2)

shunit2

<http://code.google.com/p/shunit2/>

shUnit2 2.1.x ドキュメント

<https://sites.google.com/site/paclearner/shunit2-documentation>

シェルスクリプトでユニットテストを書いてみよう

<http://a4p.me/archives/1157>

シェルスクリプトの自動テストライブラリ shUnit2 を調べてみた

<http://d.hatena.ne.jp/oknknkc/20110823/1314113501>

シェルスクリプトのユニットテストフレームワーク "shUnit2" を使ってみた

<http://d.hatena.ne.jp/yohei-a/20100202/1265058239>

shUnit2 を使ってシェルスクリプトのユニットテストをやってみた

<http://d.hatena.ne.jp/yohei-a/20100202/1265094899>

shunit2でシェルもテストファースト

<http://d.hatena.ne.jp/tbpg/20120225/1330177900>

参考情報 (2/2)

bashとBourneShellの日付計算とかshUnit2のメモ

<http://tenten0213.hatenablog.com/entry/2012/09/06/003319>

shunit2でテスト

<http://nomorework.fool.jp/wordpress/2012/12/07/shunit2でテスト/>

シェルスクリプトのユニットテスト

<http://d.hatena.ne.jp/wyukawa/20110221/1298296121>

シェルスクリプトのテスト

<http://d.hatena.ne.jp/wyukawa/20110712/1310467588>