

Python勉強会@HACHINONE

第6章

テストとデバッグ

お知らせ

Python勉強会@HACHINOHEでは、ジョン・V・グッターグ『Python言語によるプログラミングイントロダクション』近代科学社、2014年をみんなで勉強しています。

この本は自分で読んで考えて調べると力が付くように書かれています。

自分で読んで考えて調べる前に、このスライドを見るのは、いわば**ネタバレ**を聞かされるようなものでもったいないです。

是非、本を読んでからご覧ください。

テスト

Python勉強会@HACHINOHE

- 適切なテスト・スイート(組)を考えよう
 - あらゆる入力をテストすることはできないから
 - 手順
 - あり得る入力の集合
 - 同じ種類の結果を返すと想像されるものに分割
 - 分割した集合の中で、それぞれ代表的な値を選ぶ
- 2種類のテスト
 - グラスボックス(ガラス箱)・テスト: プログラムを見て作る
 - ブラックボックス・テスト: 仕様を見て作る

テスト・スイートの例

Python勉強会@HACHINOHE

- 大小判定の関数

```
def isBigger(x, y):  
    """xとyはint  
    xがyより小さければTrue  
    そうでなければFalseを返す"""
```

- テスト・スイート

- $x > 0, y > 0$
- $x < 0, y < 0$
- $x > 0, y < 0$
- $x < 0, y > 0$
- $x = 0, y = 0$
- $x = 0, y \neq 0$
- $x \neq 0, y = 0$

ブラックボックス・テストの例

Python勉強会@HACHINOHE

- 平方根を求める関数

```
def sqrt(x, epsilon):  
    """x>=0, epsilon>0のfloat  
    x - epsilon <= result * result <= x + epsilon  
    を満たすresultを返す"""
```

- テスト・スイート

x	epsilon
0.0	0.0001
25.0	0.0001
0.5	0.0001
2.0	0.0001
2.0	1.0/2.0**64
1.0/2.0**64	1.0/2.0**64
2.0**64.0	1.0/2.0**64
1.0/2.0**64	2.0**64.0
2.0**64.0	2.0**64.0

典型例

極端な例

※epsilonが1.0/2.0**64は
本当に必要か？

エイリアシングの罠

Python勉強会@HACHINOHE

- エイリアシング(同じものを参照)に留意
- リストのコピー
- L1とL2が同じリストを参照しているとうまく動かない

```
def copy(L1, L2):  
    """L1とL2はリスト  
    L2にL1をコピーする"""  
    while len(L2) > 0: # L2からすべての要素について  
        L2.pop() # L2の最後の要素を削除  
    for e in L1: # L1のすべての要素について  
        L2.append(e) # L2に追加
```

コードを見ないと気づかないかもしれないバグ

Python勉強会@HACHINOHE

- 素数判定

```
def isPrime(x):  
    """xは負でないint  
    xが素数ならTrue、そうでなければFalseを返す"""  
    if x <= 2:  
        return False  
    for i in range(2, x):  
        if x % i == 0:  
            return False  
    return True
```

- 2は素数だが、Falseが返る

経路完全

Python勉強会@HACHINOHE

- 経路完全とは、プログラムのすべての経路をテストすること
 - 通常は不可能
- 経路完全でも、発見できないバグもある

```
def abs(x):  
    """xはint  
    x>=ならx、そうでなければ-xを返す"""  
    if x < -1:  
        return -x  
    else:  
        return x
```

- -2と2をテスト・スイートにすると、経路完全になる
 - しかし、-1のとき-1が返るバグを発見できない

ガラスボックス・テストに関する経験則

Python勉強会@HACHINOHE

- すべてのif文の両方の分岐を試す
- except節を試す
- for文は、ループに入らない、1回実行、2回以上実行を試す
- while文は、forの場合に加え、ループを抜けるすべての場合を試す
- 再帰もfor文と同様

テストの管理

Python勉強会@HACHINOHE

- テストの段階
 - 単体テスト→統合テスト
- テスト・ドライバ
 - テスト対象を呼び出すプログラム
- テスト・スタブ
 - テスト対象が呼び出す下位プログラムの代用プログラム
- 回帰テスト
 - コードを修正したために、他の部分にエラーがでないかをテストする

デバッグ

Python勉強会@HACHINOHE

- バグの種類
 - 顕在的バグ、潜在的バグ
 - 継続的バグ、断続的バグ
- 守備的プログラミング
 - バグが発生するなら、顕在的で継続的になるようにプログラムを書くこと
- デバッグの方法
 - テスト結果とコードを検討する
- バグの探索
 - 二分探索的にコードを調べるなど

二分探索デバッグ例:1

Python勉強会@HACHINOHE

- 回文判定

この辺をチェック
print resultで['b']

```
# -*- coding: utf-8 -*-
def isPal(x):
    """xはリスト
       リストが回文ならTrue、そうでなければFalseを返す"""
    temp = x
    temp.reverse
    if temp == x:
        return True
    else:
        return False

def silly(n):
    """nは正のint
       n個の入力を受け取り、
       入力文字列が回文なら'Yes'、そうでなければ'No'を出力"""
    for i in range(n):
        result = []
        elem = raw_input('要素を入力: ')
        result.append(elem)
    if isPal(result):
        print 'Yes'
    else:
        print 'No'

silly(2) # プロンプトにはaとbを入力
```


二分探索デバッグ例:2

Python勉強会@HACHINOHE

- 回文判定

この辺をチェック

print temp, xで

['a', 'b'] ['a', 'b']

```
# -*- coding: utf-8 -*-
def isPal(x):
    """xはリスト
       リストが回文ならTrue、そうでなければFalseを返す"""
    temp = x
    temp.reverse
    if temp == x:
        return True
    else:
        return False

def silly(n):
    """nは正のint
       n個の入力を受け取り、
       入力文字列が回文なら'Yes'、そうでなければ'No'を出力"""
    result = []
    for i in range(n):
        elem = raw_input('要素を入力: ')
        result.append(elem)
    print result
    if isPal(result):
        print 'Yes'
    else:
        print 'No'
```

silly(2) # プロンプトにはaとbを入力

二分探索デバッグ例:3

Python勉強会@HACHINOHE

- 回文判定

```
# -*- coding: utf-8 -*-
def isPal(x):
    """xはリスト
    リストが回文ならTrue、そうでなければFalseを返す"""
    temp = x[:]
    temp.reverse()
    print temp, x
    if temp == x:
        return True
    else:
        return False

def silly(n):
    """nは正のint
    n個の入力を受け取り、
    入力文字列が回文なら'Yes'、そうでなければ'No'を出力"""
    result = []
    for i in range(n):
        elem = raw_input('要素を入力: ')
        result.append(elem)
    print result
    if isPal(result):
        print 'Yes'
    else:
        print 'No'
```

silly(2) # プロンプトにはaとbを入力

まとめ

Python勉強会@HACHINOHE

- テスト
 - 完璧なテストを作ることは難しい
 - テストには、ブラックボックス・テストとガラスボックス・テストがある
 - テストのためのうまい入力のスィート(組)を見つけよう
- デバッグ
 - テストと結果を見てよく考える
 - 二分探索的にバグを見つける
 - 修正することでプログラムを壊さないように