Python勉強会@HACHINONE

第7章

例外とアサーション

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。

例外 exception

Python勉強会@HACHINOHE

- 「例外」とは「何かが基準に適合しないこと」
  - 配列の要素数の範囲外、型が合わない、参照した名前が存在しないなどなど
- 例外が起こると、プログラムが停止する
  - バグが顕在的になる
- 自分で例外を発生させることもできる: raise
- 例外が起きても、プログラムを停止させずに、処理する(handle)ことができる: try〜except

try～except

Python勉強会@HACHINOHE

- 例外を処理する: ゼロ除算の例

```
try:
    successFailureRatio = numSuccess / float(numFailures)
    print ‘成功/失敗の割合は', successFailureRatio
except ZeroDivisionError:
    print '失敗していないので、成功/失敗の割合が計算できません'
print '今ここ'
```

例外の発生する可能性があるブロック

例外の処理をするブロック

- exceptの書き方
  - 一つの例外 `except 例外:`
  - 複数の例外 `except (例外1, 例外2, ...):`
  - すべての例外 `except:`

指練習7.1

Python勉強会@HACHINOHE

- 文字列中の数字の合計

```
# -*- coding: utf-8 -*-
def sumDigits(s):
    """sを文字列とする。
       sの中の数字の合計を返す。
       例えば、sが'a2b3c'ならば5を返す"""
    sum = 0
    for c in s:
        try:
            sum += int(c)
        except ValueError: # int(c)が失敗したとき
            continue # pass # 何もしない
    return sum
```

```
'' => 0
'1' => 1
'12' => 3
'a' => 0
'ab' => 0
'a1b2' => 3
```

多相的なpolymorphic 関数

Python勉強会@HACHINOHE

- 異なる型に対応

```
# -*- coding: utf-8 -*-
def readInt():
    while True:
        val = raw_input('整数を入力してください: ')
        try:
            val = int(val)
            return val
        except ValueError:
            print str(val) + 'は整数ではありません'
```

```
# -*- coding: utf-8 -*-
def readVal(valType, requestMsg, errorMsg):
    while True:
        val = raw_input(requestMsg + ' ')
        try:
            val = valType(val)
            return val
        except ValueError:
            print str(val) + errorMsg
```

指練習7.2

Python勉強会@HACHINOHE

- 偶数を検索

```
# -*- coding: utf-8 -*-
def findAnEven(l):
    """lをint型の要素を持つリストとする。
       lの中で最初に現れた偶数を返す。
       lが偶数を含まなければValueErrorを引き起こす。"""
    for i in l:
        if i % 2 == 0:
            return i
    raise ValueError('findAnEvenは偶数を含まないリストを引数に呼び出されました。')
```

```
[] => ValueError
[0] => 0
[1] => ValueError
[2] => 2
[1, 2] => 2
[1, 2, 3] => 2
[1, 2, 3, 4] => 2
[4, 3, 2, 1] => 4
```

try-exceptによるフロー制御

Python勉強会@HACHINOHE

- リストの比を求める

```
# -*- coding: utf-8 -*-
def getRatios(vect1, vect2):
    """vect1とvect2を同じ長さのリストとする。
    vet1[i]/vect2[i]を意味する値からなるリストを返す。"""
    ratios = []
    for index in range(len(vect1)):
        try:
            ratios.append(vect1[index] / float(vect2[index]))
        except ZeroDivisionError:
            ratios.append(float('nan')) # nan = Not a Number
        except:
            raise ValueError('getRatiosは不適切な引数で呼び出されました')
    return ratios
```

```
try:
    print getRatios([1.0, 2.0, 7.0, 6.0], [1.0, 2.0, 0.0, 3.0])
    print getRatios([], [])
    print getRatios([1.0, 2.0], [3.0])
except ValueError, msg:
    print msg
```

```
[1.0, 1.0, nan, 2.0]
[]
getRatiosは不適切な引数で呼び出されました
```

try-exceptによらないフロー制御

Python勉強会@HACHINOHE

- 例外(不適切な値)の処理で行数が増える

```
# -*- coding: utf-8 -*-
def getRatios(vect1, vect2):
    """vect1とvect2を同じ長さのリストとする。
    vet1[i]/vect2[i]を意味する値からなるリストを返す。"""
    ratios = []
    if len(vect1) != len(vect2): # リストの要素数が異なる
        raise ValueError('getRatiosは不適切な引数で呼び出されました')
    for index in range(len(vect1)):
        vect1Elem = vect1[index]
        vect2Elem = vect2[index]
        if (type(vect1Elem) not in (int, float)) \
           or (type(vect2Elem) not in (int, float)): # 要素は整数か小数
            raise ValueError('getRatiosは不適切な引数で呼び出されました')
        if vect2Elem == 0.0: # ゼロ除算
            ratios.append(float('NaN'))
        else:
            ratios.append(vect1Elem / vect2Elem)
    return ratios
```

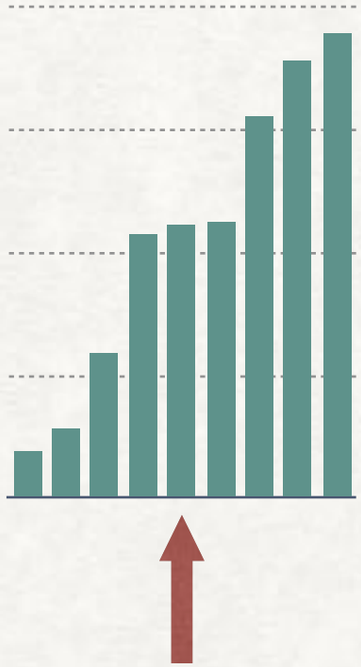
try-exceptによるフロー制御

Python勉強会@HACHINOHE

- 成績を読み込み、処理(中央値の表示)する

```
# -*- coding: utf-8 -*-
def getGrades(fname):
    try:
        gradesFile = open(fname, 'r') # 読み込むファイルを開く
    except IOError: # ファイルが開けないとか
        raise ValueError('getGradesは' + fname + 'を開けません')
    grades = []
    for line in gradesFile:
        try:
            grades.append(float(line))
        except:
            raise ValueError('行を小数に変換できません')
    return grades

try:
    grades = getGrades('quiz1grades.txt')
    grades.sort()
    median = grades[len(grades) // 2] # 2で割って小数点以下を切り捨て
    print '成績の中央値は' + str(median)
except ValueError, errorMsg:
    print 'うわーい！', errorMsg
```



中央値

9個値があるなら

前から5番目(添字4)

assert 表明

Python勉強会@HACHINOHE

- 仕様を表明する
  - 表明対象は論理式

    ```
    assert 論理式
    ```

    - TrueにならないとAssertionError表明違反が発生
  - 表明違反時のメッセージを指定

    ```
    assert 論理式，表明違反時のメッセージ
    ```
- プログラムの中にassert文を入れることで不適な値をチェックできる
  - 適切な場合、何も表示しないので、print文デバッグよりも便利な場合もある

まとめ

Python勉強会@HACHINOHE

- バグやエラーを検知、対処する
- 例外
  - 何かが基準に合わない場合に発生したり、自分で発生させる
    - ValueError、ZeroDivisionError、IOError、...
  - try〜except〜で例外を処理する
- アサーション / 表明
  - プログラムの中で値が不適かどうかを診断させる