

# Pythonコード添削道場

柴田 淳 担当分

2006.6.30 Python Workshop the Edge 2007

# 添削の指針

- Pythonicであるかどうか
- 好き度(パーセント表示)
  - 空気が読める : 好き
  - 空気が読めない : 嫌い

# お題1:単語数カウント

英文のテキストファイルを読み込んで単語の出現数を数えるプログラムを作れ。

余力があれば出現頻度の多い順に出力するプログラムも書け。

# ポイント

- 単語カウントに辞書を使う
  - ハッシュを使って出現数をカウント
- 初出キーの扱い

# 添削対象その1 好き度:0%

投稿者 **小田切篤**

```
# -*- encoding: utf-8 -*-
```

```
import re
```

```
def theme2_1(target):  
    """ 単語数え上げ  
    """
```

```
    p = re.compile(r"^[A-Za-z]|\s")
```

```
    d = dict()
```

```
    for s in p.split(target):
```

```
        if len(s) != 0:
```

```
            s = s.lower()
```

```
            :
```

スタッフが投稿しては  
いけません！！

# 添削対象その2 好き度:50%

名前がナイス!

投稿者:磯野フネ

```
a=open('text.txt').read()
for x in',.!?():':a=a.replace(x,'')
a=a.split()
t={}
for x in a:
    if x not in t:t[x]=1
    else:t[x]+=1
a=t.items()
a.sort(lambda x,y:y[1]-x[1])
for x,y in a:print y,x
```

t[x]=t.get(x, 0)+1

# 添削対象その3 好き度:90%

投稿者:にしお

ちなみにテストデータとして英語の長い文章がほしければ

<http://www.gutenberg.org/etext/108>

こんなのはどうでしょう。

若いのに気が利く！！

# お題2:シングルトン

Pythonでシングルトンを作れ。

シングルトン:

一つしか存在できないインスタ  
ンス

# ポイント

- Pythonのクラス生成の仕組みを知る
  - `__new__()`を使うとスマートに実装可能
- Pythonのアトリビュートについて知る
  - 性善説的運用

# 添削対象その5 好き度:90%

投稿者:odz

```
class Singleton(object):  
    _instances = dict()  
  
    def __new__(clazz, *args, **kwargs):  
        if clazz not in clazz._instances:  
            clazz._instances[clazz] =  
object.__new__(clazz, *args, **kwargs)  
            return clazz._instances[clazz]  
  
class DerivedSingleton(Singleton):  
    pass
```

インスタンス生成済みのク  
ラスを保存する辞書

# 添削対象その5

# 続き

```
>>> s1=Singleton()  
>>> s2=Singleton()  
>>> s1==s2  
True  
>>> d1=DerivedSingleton()  
>>> d2=DerivedSingleton()  
>>> d1==d2  
True  
>>> Singleton._instances={}  
>>> s3=Singleton()  
>>> s1==s3  
False
```

クラスアトリビュートを外部から書き換える

# 添削対象その6 好き度:95%

投稿者:海坊主+odz

```
class Singleton(object):  
    __instances = dict()
```

```
    def __new__(clazz, *args, **kwargs):  
        if clazz not in clazz.__instances:  
            clazz.__instances[clazz] =  
object.__new__(clazz, *args, **kwargs)  
            return clazz.__instances[clazz]
```

```
class DerivedSingleton(Singleton):  
    pass
```

アンダーバー2つでアトリ  
ビュートを隠蔽

# 添削対象その6

# 続き

```
>>> s1=Singleton()  
>>> s2=Singleton()  
>>> s1==s2  
True  
>>> d1=DerivedSingleton()  
>>> d2=DerivedSingleton()  
>>> d1==d2  
True  
>>> Singleton._Singleton__instances={}  
>>> s3=Singleton()  
>>> s1==s3  
False
```

アンダーバー+クラス名が付け加わった名前に変わるだけ

# 添削対象その7 好き度:5%

引数にすることで書き換え不能になる

投稿者:morchin

```
# -*- coding: utf-8 -*-
```

```
def singleton(obj, inst={}):  
    """同一クラス名のオブジェクトが既に登録されていれば例外を発生させる"""  
    assert obj.__class__ not in inst, '%s is already instantiated.' %  
obj.__class__  
    inst[obj.__class__] = True
```

```
class Foo:  
    """任意のクラス"""  
    def __init__(self):  
        singleton(self) # ここでクラス名を登録
```

カンニングはダメ！

```
if __name__ == '__main__':  
    # 参考URL: http://newworld.ddo.jp/doc/python-iaq-ja/index\_html  
    Foo()  
    Foo() # 例外が発生する
```

開発者として

添削対象その8 好き度:95%

投稿者:(名無し)

mysingleton

import mySingleton

モジュールオブジェクトはひとつしか存在しないので、シングルトンとして扱える

人間としての好き度:10%

空気を読もう！

おしまい