

实现跨模块全局变量

日前在做基于 Django 的网站，在不同模块中需要频繁定义同一个变量：判断系统环境，针对不同的环境做不同的处理。因此想 get 一个跨模块的全局变量

定义全局变量管理模块

借鉴global关键字的思路，既然在一个文件里面可以生效的话，那么我们就专门为全局变量定义一个“全局变量管理模块”就好了

- 全局变量模块 global_demo.py

```
def _init(): # 初始化
    global _global_dict
    _global_dict = {}

def set_value(key,value):
    """ 定义一个全局变量 """
    _global_dict[key] = value

def get_value(key,defValue=None):
    """ 获得一个全局变量,不存在则返回默认值 """
    try:
        return _global_dict[key]
    except KeyError:
        return defValue
```

- 设置全局变量 write.py

```
import global_demo as gl
gl._init() # 初始化（只需在定义变量时初始化一次即可）

gl.set_value('CODE','UTF-8')
gl.set_value('PORT',80)
gl.set_value('HOST','127.0.0.1')
```

- 由此，即可在任意模块引用定义过的变量

```
import global_demo as gl

#不需要再初始化了
PORT = gl.get_value('PORT') # ROOT = 80
CODE = gl.get_value('CODE') # CODE = UTF-8
```

但是在实际使用的时候感觉并没有简化多少 XD
每次需要 引用模块、取值、再判断
之前需要 导入 os 模块、取系统类型、再判断
方法有待精进

Reference

[python 3 实现定义跨模块的全局变量和使用](#)
[Python 3 实现定义跨模块的全局变量和使用](#)

2019.03.04