模块基础

1

模块基础

—— 定义模块

模块基本概念

- 模块是从逻辑上组织python代码的形式
- 当代码量变得相当大的时候，最好把代码分成一些有组织的代码段，前提是保证它们的彼此交互
- 这些代码片段相互间有一定的联系，可能是一个包含数据成员和方法的类，也可能是一组相关但彼此独立的操作函数

创建模块

- 模块物理层面上组织模块的方法是文件，每一个以.py作为结尾的python文件都是一个模块
- 模块名称切记不要与系统中已存在的模块重名
- 模块文件名字去掉后面的扩展名（.py）即为模块名

导入模块（import）

- 使用import导入模块
- 模块被导入后，程序会自动生成pyc的字节码文件以提升性能
- 模块属性通过“模块名.属性”的方法调用
- 如果仅需要模块中的某些属性，也可以单独导入

```
>>> import sys
>>> import os, string
>>> string.digits
'0123456789'
>>> from random import randint
>>> randint(1, 10)
3
```

模块加载（load）

- 一个模块只被加载一次，无论它被导入多少次
- 只加载一次可以阻止多重导入时代码被多次执行
- 如果两个文件相互导入，防止了无限的相互加载
- 模块加载时，顶层代码会自动执行，所以只将函数放入模块的顶层是良好的编程习惯

模块导入的特性

- 模块具有一个__name__特殊属性
- 当模块文件直接执行时，__name__的值为'__main__'
- 当模块被另一个文件导入时，__name__的值就是该模块的名字

```
[root@zzghost1 day02]# vim foo.py
#!/usr/bin/env python3
print(__name__)
[root@py01 bin]# ./foo.py
__main__
[root@zzghost1 day02]# python
>>> import foo
foo
```

应用案例

- 生成8位随机密码
  1. 编写一个能生成8位随机密码的程序
  2. 使用random的choice函数随机取出字符
  3. 改进程序，用户可以自己决定生成多少位的密码

更多精彩...

http://bj.linux.tedu.cn/
企业QQ： 86198501