模块与包

模块

在 Python 中,一个.py文件就称之为一个模块(Module)。

- 大大提高了代码的可维护性;
- 编写代码不必从零开始。当一个模块编写完毕,就可以被其他地方引用;

模块

```
import sys
def test():
    args = sys.argv
    if len(args) == 1:
        print 'Hello world'
    elif len(args) == 2:
        print 'Hello %s!' %args[1]
    else:
        print 'Too many arguments!'
if __name__ == '__main__':
    test()
```

调用模块时用import xxx

包

如果不同的人编写的模块名相同怎么办?为了避免模块名冲突,Python 又引入了按目录来组织模块的方法,称为包(Package)

包

创建包的步骤:

创建一目录为包名;

在该文件夹下创建__init__.py文件存放包的信息，该文件可以为空;

根据需要存放脚本文件，已编译的扩展及子包;

可以用import,import as,from import等语句导入模块和包;

作用域

在一个模块中,我们可能会定义很多函数和变量,但有的函数和变量我们希望给别人使用,有的函数和变量我们希望仅仅在模块内部使用。在 Python 中,是通过 _ 前缀来实现。比如__author__ , __name__ 就是特殊变量的，__func__,__fun为私有函数，不能直接引用。

安装第三方模块

- 需要联网；
- 通过 setuptools 这个工具完成;
- pip install 包名 或 pycharm中安装;

模块搜索路径

- 当我们试图加载一个模块时,Python 会在指定的路径下搜索对应的.py 文件;

- 默认情况下,Python 解释器会搜索当前目录、所有已安装的内置模块和第三方模块,搜索路径存放在 sys 模块的 path 变量中;

over !