

python in action

Lesson 7 - 模块, 包和命令行输入

凤凰山 github.com/gxtrobot/pyinaction

2019-11

Lesson 7 - 模块, 包和命令行输入

课程内容

今天的课程将介绍 Python 的模块与包的使用, 以及程序的输入输出多种方式, 包括从命令行接收用户输入.

课程目标

- 了解并使用模块
- 了解包的使用
- 了解命令行输入
- 运用所学知识写个小程序

```
hello.py

import sys
name = 'world'

print('module name', __name__)
if __name__ == '__main__':
    if len(sys.argv) > 1:
        name = sys.argv[1]
    print('hello', name)
```

模块的定义

- 一个 python 文件 (.py) 就是一个模块
- 一个模块可以直接被执行 `python hello.py`
- 特殊变量 `__name__` 定义了模块的名字
- 当模块直接被 python 当脚本程序执行是, `__name__` 为 `__main__`
- 模块也可以被其他模块导入, 被导入时 `__name__` 为模块名称, 上例为 `hello`
- 定义只在模块当做脚本运行时的代码, 放入所谓的 `main` 代码块中

模块的介绍

模块的作用域

```
n = 100

def double_it():
    print('double it: ', n)
    # n = 2 * n

def double_it_global():
    global n
    print('double it global: ', n)
    n = 2 * n

double_it()
print('n:', n)
double_it_global()
print('n:', n)
```

模块的作用域

- 定义在模块中最外层（不在任何函数，或类之内）的为全局变量
- 定义在其他地方为非全局变量（局部变量）
- 全局变量可以被所以函数访问
- 但函数修改全局变量需要使用 `global` 关键字，否则会报错

包的定义

- 包是组织模块的方式
- 一个目录里有多个模块，可以组成一个包
- 目录里必须定义 `__init__.py` 文件，可以为空
- 目录可以嵌套，所以包也可以嵌套

模块，包的导入

- 按照包，模块，函数的层级导入

模块的多种导入方式

- `import xxx(模块名)`
- `from xxx(包名) import xxx(模块名)`
- 可以用 `as` 重命名
 - `import xxx(模块名) as yyy(别名)`
 - `from xxx(包名) import xxx(模块名) as yyy(别名)`

函数或其他对象的多种导入方式

- `from xxx(模块名) import xxx(函数名)`
- `from xxx(包名).xxx(模块名) import xxx(函数名)`
- 可以用 `as` 重命名
 - `from xxx(模块名) import xxx(函数名) as yyy(别名)`
 - `from xxx(包名).xxx(模块名) import xxx(函数名) as yyy(别名)`

模块，包的一些特点

- 模块在导入时会执行所有最外层的代码
- 包在导入时会执行 `__init__.py` 里的代码（只执行一次）
- 模块重复导入无效，只有第一次有效（代码更新后已导入模块不会更新）
- 模块，包的命名都为小写，满足 `python` 命名规范

- 当前程序执行目录（运行 `python xxx.py` 命令的地方）
- `sys.path` 里的所有目录

实验下各种模块，函数的导入方法假设定义如下包和模块

- 包: testpkg
 - 模块: `__init__.py`
 - 模块: `moduleA`
 - 函数: `testA()`
 - 模块: `moduleB`
 - 函数: `testB()`
 - 模块: `moduleC`
 - 函数: `testC()`

```
from testpkg import moduleA
moduleA.testA()
from testpkg.moduleA import testA
testA()
from testpkg import moduleB
from testpkg import moduleC as moudC
moudC.testC()
```

从命令行读取输入

用 input 函数函数获取用户输入

```
a = input('plese enter a number: ')\nprint(a)
```

- input 函数可以接收一个提示字符串参数，用于提示用户输入信息
- 用户按回车后函数返回
- 返回值为用户输入的字符串

改写下 `calc.py`, 使用 `input` 接收参数

- 创建 testpkg 包，并测试各种模块，包的导入
- 编写一个模块，里面有 report 函数，并使用 input 函数获取一个数字列表（如 10 100 90），并使用 report 打印列表（提示：可以使用无限循环）

