

python 总结

列表 pop append del remove insert

f 字符串

元组

字典 get()

and not

if else elif 语句

** 优先级

sort sorted reverse reversed reverse = True

Lstrip strip.rstrip

for in / not in

range.

items. keys. values

切片索引法 [:] 复制子集 / 引用传参

* 创建空元组, 可作为函数参数

* 创建空字典, 可作为函数参数

py 函数

§ 将函数存储在模块中

模块 func.c

import .py 文件 导入整个模块; 调用函数要
导入特定函数

from module-name import function-name, ...
直接调用 func(), 即可

from module-name import *

可导入模块中的所有函数

使用 as 指定别名

from module-name import func-name as ...

~~from~~ import module-name as ...

类

面向对象编程

根据类来创建对象:实例化

在 Python 中,有字母大写的名称指的是类
类中的函数称为方法

方法 `__init__()`

“创建新实例时自动运行 (构造函数)”

`__init__()` 包含两个形参, `self` 和 ...

`self` 必须放在最前面,并指向

`self` 为指向实例本身的引用,让实例得以访问类中的属性和方法

创建实例时 `self` 会自动传递,不需要后面的形参提供值
属性

eg: `self.age = age` (`this → age = age`)

其他方法,若执行时不需要额外的信息
则只有一个形参 `self`

实例: ... = `Class()`

与 C++ 不同,不能 `Class --()`

修改属性的值

1. 直接通过实例修改

2. 通过方法进行设置

3. 通过方法进行递增 (增加特定的值)

属性可在类的定义中初始化 `__init__()`,也可在实例化后动态添加