python学习手册 第四版;

Mark Lutz编著;

---------------------------------------------Part 1, 使用入门-----------------------------------------------------

Chapter 1, 问答环节;

人们为何使用Python;

软件质量; 开发效率; (2021-5-7)

Python是脚本语言吗?

好吧, python的缺点是什么呢?

如今谁在使用python;

使用python可以做些什么;

系统编程; 用户图形接口; internet脚本; 组件集成; 数据库编程; 快速原型; 游戏、图像、人工智能、XML、机器人等; (2021-5-7)

python如何获得支持;

Python有哪些技术上的优点;

面向对象; 免费; 可移植; 功能强大; 可混合; 简单易用; 简单易学; (2021-5-7)

python和其他语言比较起来怎么样;

Chapter 2, python如何运行程序;

python解释器简介;

程序执行;

程序员视角; python的视角; (2021-5-7)

执行模块的变体;

Chapter 3, 如何运行程序;

交互提示模式下编写代码;

系统命令行和文件;

第一段脚本;

使用命令行运行文件;

使用命令行和文件;

UNIX可执行脚本;

点击文件图标;

在windows中点击图标; input的技巧; 图标点击的其他限制; (2021-5-8)

模块导入和重载;

模块的显要特性: 属性; import和reload的使用注意事项; (2021-5-8)

使用excel运行模块文件;

IDLE用户界面;

IDLE基础; 使用IDLE; 高级IDLE工具; (2021-5-8)

其他的IDE;

其他启动选项;

嵌入式调用; 冻结二进制的可执行性; 文本编辑器启动的选择; 其他的启动选择; 未来的可能; (2021-5-8)

我应该选用哪种;

---------------------------------------------Part 2, 类型和运算-----------------------------------------------------

Chapter 4, 介绍python对象类型;

为什么使用内置类型;

python的核心数据类型;

列表[]; 字典{}; 元组(); (2021-5-8)

数字;

字符串;

序列的操作; 不可变性; 类型特定的方法; 寻求帮助; 编写字符串的其他方法; 模式匹配; (2021-5-8)

列表;

序列操作; 边界检查; 嵌套(二维数组); 列表解析; (2021-5-8)

字典;

映射操作(键值对); 重访嵌套; 键的排序for循环; 迭代和优化; 不存在的键, if测试; (2021-5-8)

元组;

为什么要用元组; (2021-5-8)

文件;

其他文件类工具; (2021-5-8)

其他核心类型;

如何破坏代码的灵活性; 用户定义的类; 剩余的内容; (2021-5-8)

Chapter 5, 数字;

python的数字类型;

数字常量; 内置数学工具和扩展; python表达式操作符(混合操作所遵循的操作符优先级, 括号分组的子表达式, 混合类型自动升级, 预习运算符重载); (2021-5-10)

在实际应用中的数字;

变量和基本的表达式; 数字显示的格式; 比较, 一般的和连续的; 除法, 传统除法, floor除法和真除法; 整数精度; 复数; 十六进制, 八进制和二进制记数; 位操作; 其他的内置数学工具; (2021-5-10)

其他数字类型;

小数数字; 分数类型; 集合; 布尔型; (2021-5-10)

数字扩展;

Chapter 6, 动态类型简介;

缺少类型声明语句的情况;

变量, 对象和引用(a=3, 变量a是对象3的一个引用); 类型属于对象, 而不是变量; 对象的垃圾收集; (2021-5-12)

共享引用;

共享引用和在原处修改; 共享引用和相等; (2021-5-12)

动态类型随处可见;

Chapter 7, 字符串;

字符串常量;

单双引号字符串是一样的; 用转义序列代表特殊字节; raw字符串抑制转义; 三重引号编写多行字符串块; (2021-5-12)

实际应用中的字符串;

基本操作; 索引和分片; 字符串转换工具; 修改字符串; (2021-5-12)

字符串方法;

修改字符串; 文本解析; 其他常见字符串方法; (2021-5-12)

字符串格式化表达式;

更高级的字符串格式化表达式; 基于字典的字符串格式化; (2021-5-12)

字符串格式化调用方法;

基础知识; 添加键、属性和偏移量; 添加具体格式化; 与%格式化表达式比较; 为什么用新的格式化方法; (2021-5-12)

通常意义下的类型分类;

同样分类的类型共享其操作集合; 可变类型能够在原处修改; (2021-5-12)

Chapter 8, 列表与字典;

列表;

实际应用中的列表;

基本列表操作; 列表迭代和解析; 索引、分片和矩阵; 原处修改列表(索引与分片的赋值, 列表方法调用, 其他常见列表操作); (2021-5-12)

字典;

通过键而不是偏移量来读取; (2021-5-13)

实际应用中的字典;

字典的基本操作; 原处修改字典; 其他字典方法; 语言表; 字典用法注意事项; 创建字典的其他方法; (2021-5-13)

Python 3.0中的字典变化;

Chapter 9, 元组、文件及其他;

元组;

实际应用中的元组; 为什么有了列表还要元组; (2021-5-13)

文件;

打开文件; 使用文件; 实际应用中的文件; 其他文件工具; (2021-5-13)

重访类型分类;

对象灵活性;

引用VS拷贝;

比较、相等性和真值;

Python 3.0的字典比较; python中真和假的含义; python的类型层次; type对象; (2021-5-13)

python中的其他类型;

内置类型陷阱; 赋值生成引用, 而不是拷贝; 重复能够增加层次深度; 留意循环数据结构; 不可变类型不可以在原处改变; (2021-5-14)

--------------------------------------------- Part 3, 语句和语法 -----------------------------------------------------

Chapter 10, python语句简介;

重访python程序结构;

python的语句; 两个if的故事; python增加了什么; python删除了什么; 为什么使用缩进语法; 几个特殊实例; (2021-5-14)

简短实例: 交互循环;

一个简单的交互式循环; 对用户输入数据做数学运算; 用测试输入数据来处理错误; 用try语句处理错误; 嵌套代码三层; (2021-5-14)

Chapter 11, 赋值、表达式和打印;

赋值语句;

赋值语句的形式; 序列赋值; python 3.0中的扩展序列解包; 多目标赋值语句; 增强赋值语句; 变量命名规则; (2021-5-14)

表达式语句;

表达式语句和在原处的修改; (2021-5-15)

打印操作;

Python 3.0的print函数; python 2.6的print语句; 打印流重定向; 版本独立的打印; (2021-5-15)

Chapter 12, if测试和语法规则;

if语句;

通用格式; 基本例子; 多路分支; (2021-5-15)

python语法规则;

代码块分隔符; 语句的分隔符; 一些特殊情况; (2021-5-15)

真值测试;

If/else三元表达式;

Chapter 13, while和for循环;

while循环;

一般格式; 例子; break, continue, pass和循环else; 一般循环格式; pass(以后会填上); continue(应该少用); break; 循环else; (2021-5-15)

for循环;

一般格式; 例子; (2021-5-15)(-------------)

编写循环的技巧;

循环计数器, while和range; 非完备遍历, range和分片; 修改列表, range; 并行便利, zip和map; 产生偏移和元素, enumerate(返回(index, value)元组); (2021-5-17)

Chapter 14, 迭代器和解析, 第一部分;

迭代器: 初探;

文件迭代器; 手动迭代, iter和next; 其他内置类型迭代器; (2021-5-17)

列表解析: 初探;

列表解析基础知识; 在文件上使用列表解析; 扩展的列表解析语法; (2021-5-17)

其他迭代环境;

Python 3.0中的新的可迭代对象;

range迭代器; map, zip和filter迭代器; 多个迭代器VS单个迭代器; 字典视图迭代器; (2021-5-17)

其他迭代器主题;

Chapter 15, 文档;

Python文档资源;

#注释; dir函数; 文档字符串: \_\_doc\_\_; PyDoc, help函数; PyDoc, HTML报表; 标准手册集; 网络资源; 已出版书籍; (2021-5-17)

常见编写代码的陷阱;

-------------------------------------------------- Part 4, 函数 ----------------------------------------------------------

Chapter 16, 函数基础;

为何使用函数;

最大化的代码重用和最小化代码冗余; 流程的分解; (2021-5-18)

编写函数;

def语句; def语句是实时执行的; (2021-5-18)

第一个例子: 定义和调用;

定义; 调用; python中的多态; (2021-5-18)

第二个例子: 寻找序列的交集;

定义; 调用; 重访多态; 本地变量; (2021-5-18)

Chapter 17, 作用域;

python作用域基础;

作用域法则; 变量名解析, LEGB原则; 作用域实例; 内置作用域; (2021-5-18)

global语句;

最小化全局变量(尽可能避免使用全局变量); 最小化文件间的修改; 其他访问全局变量的方法; (2021-5-18)

作用域和嵌套函数;

嵌套作用域的细节; 嵌套作用域举例; (2021-5-18)

nonlocal语句;

nonlocal基础; nonlocal应用; 为什么使用nonlocal; (2021-5-18)

Chapter 18, 参数;

传递参数;

参数和共享引用; 避免可变参数的修改; 对参数输出进行模拟; (2021-5-18)

特定的参数匹配模型;

基础知识; 匹配语法; 细节; 关键字参数和默认参数的实例; 任意参数的实例; Python 3.0 keyword-only参数; min调用; 满分; 加分点; 结论; (2021-6-2)

一个更有用的例子: 通用set函数;

模拟python 3.0 print函数;

使用keyword-only参数; (2021-6-2)

Chapter 19, 函数的高级话题;

函数设计概念;

递归函数;

用递归求和; 编码替代方案; 循环语句VS递归(尽量用循环而不是递归); 处理任意结构; (2021-6-2)

函数对象: 属性和注解; (-------------)

间接函数调用; 函数内省; 函数属性; python 3.0中的函数注解; (2021-6-10)

匿名函数: lambda;

lambda表达式; 为什么使用lambda(更加简洁的代码结构); 如何(不要)让python代码变得晦涩难懂; 嵌套lambda和作用域(最好避免使用嵌套的lambda); (2021-6-10)

在序列中映射函数: map;

函数式编程工具: filter和reduce;

----------------------------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------------------------

Chapter 25, OOP: 宏伟蓝图;

为何实用类;

继承; 组合; (2021-9-22)

概览OOP;

属性继承搜索; (2021-9-22)

类和实例;

类方法调用;

编写类树;

OOP是为了代码重用;

多态是指运算的意义取决于运算对象; (2021-9-23)

Chapter 26, 类代码编写基础;

类产生多个实例对象;

类对象提供默认行为;

实例对象是具体的元素;

第一个例子;

位于类中的函数通常被称为方法; 类和实例是类树中通过继承搜索的相连的命名空间; (2021-11-25)

类通过继承进行定制;

类树中重新定义的、取代属性的动作称为重载; (2021-11-25)

类是模块内的属性;

类可以截获python运算符;

运算符重载; (2021-11-25)

第三个例子;

为什么要使用运算符重载;

世界上最简单的python类;

类与字典的关系;

Chapter 27, 更多实例;

步骤1: 创建实例;

编写构造函数;

实例对象属性; 构造函数是创建实例时第一次运行的代码; 赋值给self保持持久; (2021-11-26)

在进行中测试;

编写一些代码, 测试; 编写更多的代码, 再次测试; (2021-11-26)

以两种方式使用代码;

If \_name\_ == ‘\_main\_’ , 只有文件为了测试而运行; (2021-11-26)

步骤2: 添加行为方法;

编写方法;

封装就是把操作逻辑包装到界面之后; (2021-11-26)

步骤3: 运算符重载;

提供打印显示;

步骤4: 通过子类定制行为;

编写子类;

扩展方法: 好的方式;

扩展最初的giveRaise, 而不是完全替换它; (2021-11-26)

多态的作用;

继承、定制和扩展;

OPP: 大思路;

步骤5: 定制构造函数;

OPP比我们认为的要简单;

组合类的其他方式;

步骤6: 使用内省工具;

特殊类属性;