

Python Cookbook 3rd Edition Documentation

Contents:

- [Copyright](#)
- [前言](#)
 - [项目主页](#)
 - [译者的话](#)
 - [作者的话](#)
 - [这本书适合谁](#)
 - [这本书不适合谁](#)
 - [在线示例代码](#)
 - [使用示例代码](#)
 - [联系我们](#)
 - [致谢](#)
- [第一章：数据结构和算法](#)
 - [1.1 解压序列赋值给多个变量](#)
 - [1.2 解压可迭代对象赋值给多个变量](#)
 - [1.3 保留最后 N 个元素](#)
 - [1.4 查找最大或最小的 N 个元素](#)
 - [1.5 实现一个优先级队列](#)
 - [1.6 字典中的键映射多个值](#)
 - [1.7 字典排序](#)
 - [1.8 字典的运算](#)
 - [1.9 查找两字典的相同点](#)
 - [1.10 删除序列相同元素并保持顺序](#)
 - [1.11 命名切片](#)
 - [1.12 序列中出现次数最多的元素](#)
 - [1.13 通过某个关键字排序一个字典列表](#)
 - [1.14 排序不支持原生比较的对象](#)
 - [1.15 通过某个字段将记录分组](#)
 - [1.16 过滤序列元素](#)
 - [1.17 从字典中提取子集](#)
 - [1.18 映射名称到序列元素](#)
 - [1.19 转换并同时计算数据](#)
 - [1.20 合并多个字典或映射](#)
- [第二章：字符串和文本](#)
 - [2.1 使用多个界定符分割字符串](#)
 - [2.2 字符串开头或结尾匹配](#)
 - [2.3 用Shell通配符匹配字符串](#)

- 2.4 字符串匹配和搜索
- 2.5 字符串搜索和替换
- 2.6 字符串忽略大小写的搜索替换
- 2.7 最短匹配模式
- 2.8 多行匹配模式
- 2.9 将Unicode文本标准化
- 2.10 在正则式中使用Unicode
- 2.11 删除字符串中不需要的字符
- 2.12 审查清理文本字符串
- 2.13 字符串对齐
- 2.14 合并拼接字符串
- 2.15 字符串中插入变量
- 2.16 以指定列宽格式化字符串
- 2.17 在字符串中处理html和xml
- 2.18 字符串令牌解析
- 2.19 实现一个简单的递归下降分析器
- 2.20 字节字符串上的字符串操作
- 第三章：数字日期和时间
 - 3.1 数字的四舍五入
 - 3.2 执行精确的浮点数运算
 - 3.3 数字的格式化输出
 - 3.4 二八十六进制整数
 - 3.5 字节到大整数的打包与解包
 - 3.6 复数的数学运算
 - 3.7 无穷大与NaN
 - 3.8 分数运算
 - 3.9 大型数组运算
 - 3.10 矩阵与线性代数运算
 - 3.11 随机选择
 - 3.12 基本的日期与时间转换
 - 3.13 计算最后一个周五的日期
 - 3.14 计算当前月份的日期范围
 - 3.15 字符串转换为日期
 - 3.16 结合时区的日期操作
- 第四章：迭代器与生成器
 - 4.1 手动遍历迭代器
 - 4.2 代理迭代
 - 4.3 使用生成器创建新的迭代模式
 - 4.4 实现迭代器协议
 - 4.5 反向迭代
 - 4.6 带有外部状态的生成器函数
 - 4.7 迭代器切片
 - 4.8 跳过可迭代对象的开始部分
 - 4.9 排列组合的迭代
 - 4.10 序列上索引值迭代

- 4.11 同时迭代多个序列
- 4.12 不同集合上元素的迭代
- 4.13 创建数据处理管道
- 4.14 展开嵌套的序列
- 4.15 顺序迭代合并后的排序迭代对象
- 4.16 迭代器代替while无限循环
- 第五章：文件与IO
 - 5.1 读写文本数据
 - 5.2 打印输出至文件中
 - 5.3 使用其他分隔符或行终止符打印
 - 5.4 读写字节数据
 - 5.5 文件不存在才能写入
 - 5.6 字符串的I/O操作
 - 5.7 读写压缩文件
 - 5.8 固定大小记录的文件迭代
 - 5.9 读取二进制数据到可变缓冲区中
 - 5.10 内存映射的二进制文件
 - 5.11 文件路径名的操作
 - 5.12 测试文件是否存在
 - 5.13 获取文件夹中的文件列表
 - 5.14 忽略文件名编码
 - 5.15 打印不合法的文件名
 - 5.16 增加或改变已打开文件的编码
 - 5.17 将字节写入文本文件
 - 5.18 将文件描述符包装成文件对象
 - 5.19 创建临时文件和文件夹
 - 5.20 与串行端口的数据通信
 - 5.21 序列化Python对象
- 第六章：数据编码和处理
 - 6.1 读写CSV数据
 - 6.2 读写JSON数据
 - 6.3 解析简单的XML数据
 - 6.4 增量式解析大型XML文件
 - 6.5 将字典转换为XML
 - 6.6 解析和修改XML
 - 6.7 利用命名空间解析XML文档
 - 6.8 与关系型数据库的交互
 - 6.9 编码和解码十六进制数
 - 6.10 编码解码Base64数据
 - 6.11 读写二进制数组数据
 - 6.12 读取嵌套和可变长二进制数据
 - 6.13 数据的累加与统计操作
- 第七章：函数
 - 7.1 可接受任意数量参数的函数
 - 7.2 只接受关键字参数的函数

- 7.3 给函数参数增加元信息
- 7.4 返回多个值的函数
- 7.5 定义有默认参数的函数
- 7.6 定义匿名或内联函数
- 7.7 匿名函数捕获变量值
- 7.8 减少可调用对象的参数个数
- 7.9 将单方法的类转换为函数
- 7.10 带额外状态信息的回调函数
- 7.11 内联回调函数
- 7.12 访问闭包中定义的变量
- 第八章：类与对象
 - 8.1 改变对象的字符串显示
 - 8.2 自定义字符串的格式化
 - 8.3 让对象支持上下文管理协议
 - 8.4 创建大量对象时节省内存方法
 - 8.5 在类中封装属性名
 - 8.6 创建可管理的属性
 - 8.7 调用父类方法
 - 8.8 子类中扩展property
 - 8.9 创建新的类或实例属性
 - 8.10 使用延迟计算属性
 - 8.11 简化数据结构的初始化
 - 8.12 定义接口或者抽象基类
 - 8.13 实现数据模型的类型约束
 - 8.14 实现自定义容器
 - 8.15 属性的代理访问
 - 8.16 在类中定义多个构造器
 - 8.17 创建不调用init方法的实例
 - 8.18 利用Mixins扩展类功能
 - 8.19 实现状态对象或者状态机
 - 8.20 通过字符串调用对象方法
 - 8.21 实现访问者模式
 - 8.22 不用递归实现访问者模式
 - 8.23 循环引用数据结构的内存管理
 - 8.24 让类支持比较操作
 - 8.25 创建缓存实例
- 第九章：元编程
 - 9.1 在函数上添加包装器
 - 9.2 创建装饰器时保留函数元信息
 - 9.3 解除一个装饰器
 - 9.4 定义一个带参数的装饰器
 - 9.5 可自定义属性的装饰器
 - 9.6 带可选参数的装饰器
 - 9.7 利用装饰器强制函数上的类型检查
 - 9.8 将装饰器定义为类的一部分

- 9.9 将装饰器定义为类
- 9.10 为类和静态方法提供装饰器
- 9.11 装饰器为被包装函数增加参数
- 9.12 使用装饰器扩充类的功能
- 9.13 使用元类控制实例的创建
- 9.14 捕获类的属性定义顺序
- 9.15 定义有可选参数的元类
- 9.16 *args和**kwargs的强制参数签名
- 9.17 在类上强制使用编程规约
- 9.18 以编程方式定义类
- 9.19 在定义的时候初始化类的成员
- 9.20 利用函数注解实现方法重载
- 9.21 避免重复的属性方法
- 9.22 定义上下文管理器的简单方法
- 9.23 在局部变量域中执行代码
- 9.24 解析与分析Python源码
- 9.25 拆解Python字节码
- 第十章：模块与包
 - 10.1 构建一个模块的层级包
 - 10.2 控制模块被全部导入的内容
 - 10.3 使用相对路径名导入包中子模块
 - 10.4 将模块分割成多个文件
 - 10.5 利用命名空间导入目录分散的代码
 - 10.6 重新加载模块
 - 10.7 运行目录或压缩文件
 - 10.8 读取位于包中的数据文件
 - 10.9 将文件夹加入到sys.path
 - 10.10 通过字符串名导入模块
 - 10.11 通过钩子远程加载模块
 - 10.12 导入模块的同时修改模块
 - 10.13 安装私有的包
 - 10.14 创建新的Python环境
 - 10.15 分发包
- 第十一章：网络与Web编程
 - 11.1 作为客户端与HTTP服务交互
 - 11.2 创建TCP服务器
 - 11.3 创建UDP服务器
 - 11.4 通过CIDR地址生成对应的IP地址集
 - 11.5 创建一个简单的REST接口
 - 11.6 通过XML-RPC实现简单的远程调用
 - 11.7 在不同的Python解释器之间交互
 - 11.8 实现远程方法调用
 - 11.9 简单的客户端认证
 - 11.10 在网络服务中加入SSL
 - 11.11 进程间传递Socket文件描述符

- 11.12 理解事件驱动的IO
- 11.13 发送与接收大型数组
- 第十二章：并发编程
 - 12.1 启动与停止线程
 - 12.2 判断线程是否已经启动
 - 12.3 线程间通信
 - 12.4 给关键部分加锁
 - 12.5 防止死锁的加锁机制
 - 12.6 保存线程的状态信息
 - 12.7 创建一个线程池
 - 12.8 简单的并行编程
 - 12.9 Python的全局锁问题
 - 12.10 定义一个Actor任务
 - 12.11 实现消息发布/订阅模型
 - 12.12 使用生成器代替线程
 - 12.13 多个线程队列轮询
 - 12.14 在Unix系统上面启动守护进程
- 第十三章：脚本编程与系统管理
 - 13.1 通过重定向/管道/文件接受输入
 - 13.2 终止程序并给出错误信息
 - 13.3 解析命令行选项
 - 13.4 运行时弹出密码输入提示
 - 13.5 获取终端的大小
 - 13.6 执行外部命令并获取它的输出
 - 13.7 复制或者移动文件和目录
 - 13.8 创建和解压归档文件
 - 13.9 通过文件名查找文件
 - 13.10 读取配置文件
 - 13.11 给简单脚本增加日志功能
 - 13.12 给函数库增加日志功能
 - 13.13 实现一个计时器
 - 13.14 限制内存和CPU的使用量
 - 13.15 启动一个WEB浏览器
- 第十四章：测试、调试和异常
 - 14.1 测试stdout输出
 - 14.2 在单元测试中给对象打补丁
 - 14.3 在单元测试中测试异常情况
 - 14.4 将测试输出用日志记录到文件中
 - 14.5 忽略或期望测试失败
 - 14.6 处理多个异常
 - 14.7 捕获所有异常
 - 14.8 创建自定义异常
 - 14.9 捕获异常后抛出另外的异常
 - 14.10 重新抛出被捕获的异常
 - 14.11 输出警告信息

- 14.12 调试基本的程序崩溃错误
- 14.13 给你的程序做性能测试
- 14.14 加速程序运行
- 第十五章：C语言扩展
 - 15.1 使用ctypes访问C代码
 - 15.2 简单的C扩展模块
 - 15.3 编写扩展函数操作数组
 - 15.4 在C扩展模块中操作隐形指针
 - 15.5 从扩展模块中定义和导出C的API
 - 15.6 从C语言中调用Python代码
 - 15.7 从C扩展中释放全局锁
 - 15.8 C和Python中的线程混用
 - 15.9 用SWIG包装C代码
 - 15.10 用Cython包装C代码
 - 15.11 用Cython写高性能的数组操作
 - 15.12 将函数指针转换为可调用对象
 - 15.13 传递NULL结尾的字符串给C函数库
 - 15.14 传递Unicode字符串给C函数库
 - 15.15 C字符串转换为Python字符串
 - 15.16 不确定编码格式的C字符串
 - 15.17 传递文件名给C扩展
 - 15.18 传递已打开的文件给C扩展
 - 15.19 从C语言中读取类文件对象
 - 15.20 处理C语言中的可迭代对象
 - 15.21 诊断分段错误
- 附录A
 - 在线资源
 - Python学习书籍
 - 高级书籍
- 关于译者
- Roadmap