

《用 Python 玩转数据》之序列运算、操作、函数/方法简介

1.序列

(1) 序列的标准类型运算

<、>、<=、>=、==、!=	值比较
is、is not	对象身份比较
and、or、not	逻辑运算

(2) 通用序列类型操作

seq[start: end]	切片操作
*	重复组合序列数据
+	连接 2 个序列
in、not in	判断元素是否存在序列中

(3) 序列常用函数

函数	描 述
list(iter)	将可迭代对象 iter 转换成列表
tuple(iter)	将可迭代对象 iter 转换成元组
str(obj)	将对象 obj 转换成字符串表示
len(sequence)	返回 sequence 的长度，为整型类型
sorted(iter, key, reverse)	返回可迭代对象 iter 排序后的列表,key 用来指定排序的规则，reverse 用来指定顺序还是逆序排列
reversed(sequence)	返回序列 sequence 逆序排列后的迭代器
sum(iter, start)	将 iter 中的数值和 start 参数的值相加，返回 float 类型数值
max(iter)	返回可迭代对象 iter 中的最大值
min(iter)	返回可迭代对象 iter 中的最小值
enumerate(iter[, start])	返回一个 enumerate 对象，可生成一个迭代器，该迭代器的元素是由参数 iter 元素的索引和值组成的元组
zip(iter1 [, iter2 [...]])	返回一个 zip 对象，可生成一个迭代器，该迭代器的第 n 个元素是每个可迭代对象的第 n 个元素组成的元组

2.字符串

字符串常用方法

方法	描述
<code>s.capitalize()</code>	返回字符串 <code>s</code> 首字母大写其余小写的形式
<code>s.lower()</code>	返回字符串 <code>s</code> 的小写形式
<code>s.upper()</code>	返回字符串 <code>s</code> 的大写形式
<code>s.title()</code>	返回字符串 <code>s</code> 的标题形式即单词首字母大写形式
<code>s.format(*args, **kwargs)</code>	格式化字符串操作
<code>s.count(sub[, start[, end]])</code>	返回指定字符在[指定位置的]字符串 <code>s</code> 中出现的次数
<code>s.find(sub[, start[, end]])</code>	返回指定字符在[指定位置的]字符串 <code>s</code> 中出现的索引号, 找不到则返回-1
<code>s.index(sub[, start[, end]])</code>	与 <code>find()</code> 类似, 不同的是如果找不到会引发 <code>ValueError</code> 异常
<code>s.replace(old, new[, count])</code>	把字符串 <code>s</code> 中的 <code>old</code> (旧字符串) 替换成 <code>new</code> (新字符串)。如果指定第三个参数 <code>count</code> , 则仅仅替换前 <code>count</code> 次出现的子串
<code>s.lstrip([chars])</code>	移除字符串 <code>s</code> 左边的指定字符(默认为空格), 返回移除字符串 <code>s</code> 左边指定字符后生成的新字符串
<code>s.rstrip([chars])</code>	移除字符串 <code>s</code> 末尾的指定字符(默认为空格), 返回移除字符串 <code>s</code> 末尾指定字符后生成的新字符串
<code>s.strip([chars])</code>	移除字符串 <code>s</code> 头尾指定的字符(默认为空格), 返回移除字符串 <code>s</code> 头尾指定字符后生成的新字符串
<code>s.join(iterable)</code>	用指定的字符串 <code>s</code> 连接元素为字符串的可迭代对象
<code>s.split(sep=None, maxsplit=-1)</code>	以指定的字符作为分隔符(默认为空格)分割字符串 <code>s</code> , <code>maxsplit</code> 指分割次数(默认为不限制次数)
<code>s.endswith(suffix[, start[, end]])</code>	判断字符串 <code>s</code> [的指定位置]是否以后缀 <code>suffix</code> 结尾
<code>s.startswith(prefix[, start[, end]])</code>	判断字符串 <code>s</code> [的指定位置]是否以前缀 <code>prefix</code> 开头

3.列表

列表常用方法

方法	描述
<code>l.append(obj)</code>	在列表 <code>l</code> 末尾添加新的对象
<code>l.count(obj)</code>	统计某个元素在列表 <code>l</code> 中出现的次数
<code>l.extend(seq)</code>	在列表 <code>l</code> 末尾一次性追加另一个序列 <code>seq</code> 中的多个值(用新列表扩展原来的列表)
<code>l.index(obj)</code>	从列表 <code>l</code> 中找出某个值第一个匹配项的索引位置, 索引从 0 开始
<code>l.insert(index, obj)</code>	将对象 <code>obj</code> 插入列表 <code>l</code> 中索引为 <code>index</code> 的元素前
<code>l.pop(index)</code>	移除列表 <code>l</code> 中索引为 <code>index</code> 的一个元素(默

	认为最后一个元素), 并且返回该元素的值
<code>l.remove(obj)</code>	移除列表 <code>l</code> 中某个值的第一个匹配项
<code>l.reverse()</code>	将列表 <code>l</code> 中的元素反转
<code>l.sort(key=None, reverse=False)</code>	对原列表 <code>l</code> 进行排序, 可通过参数 <code>key</code> 指定排序依据, 通过参数 <code>reverse</code> 指定顺序 (默认方式) 或逆序排列

4.元组

元组常用函数

函数	描述
<code>len(t)</code>	计算元组 <code>t</code> 的元素个数
<code>max(t)</code>	返回元组 <code>t</code> 中元素的最大值
<code>min(t)</code>	返回元组 <code>t</code> 中元素的最小值
<code>tuple(seq)</code>	将序列 <code>seq</code> 转换为元组