

Click Here



Python 列表 描述 sort() 函数用于对原列表进行排序，如果指定参数，则使用比较函数指定的比较函数。语法 sort(方法语法：list.sort(cmp=None, key=None, reverse=False) 参数 cmp -- 可选参数，如果指定了该参数会使用该参数的方法进行排序。key -- 主要是用来进行比较的元素，只有一个参数，具体的函数的参数就是取自于可迭代对象中，指定可迭代对象中的一个元素来进行排序。reverse -- 排序规则，reverse = True 降序，reverse = False 升序（默认）。返回值 该方法没有返回值，但是会对列表的对象进行排序。实例 以下实例展示了 sort() 函数的使用方法：aList = ['123', 'Google', 'Runoob', 'Taobao', 'Facebook']; aList.sort(); print("List : " + print(aList)) 以上实例输出结果如下：List : ['123', 'Facebook', 'Google', 'Runoob', 'Taobao'] 以下实例降序输出列表：vowels = ['e', 'a', 'u', 'o', 'i'] vowels.sort(reverse=True) print(vowels) 以上实例输出结果如下：降序输出：['u', 'o', 'i', 'e', 'a'] 以下实例演示了通过指定列表中的元素排序来输出列表：def takeSecond(elem): return elem[1] random = [(2, 2), (3, 4), (4, 1), (1, 3)] random.sort(key=takeSecond) print('排序列表 : ') print(random) 以上实例输出结果如下：排序列表：[(4, 1), (2, 2), (1, 3), (3, 4)] Python 列表 parallelStream, removeIf, stream, toArray int size() 返回此列表中的元素数。如果此列表包含超过Integer.MAX_VALUE 元素，则返回Integer.MAX_VALUE。Specified by: size in interface Collection 结果 此列表中的元素数量 boolean isEmpty() Specified by: isEmpty在界面 Collection 结果 true如果此列表不包含任何元素 Object[] toArray() 以适当的顺序（从第一个元素到最后一个元素）返回包含此列表中所有元素的数组。返回的数组将是“安全的”，因为此列表不会保留对它的引用。（换句话说，即使此列表由数组支持，此方法也必须分配新数组）。因此调用者可以自由修改返回的数组。此方法充当基于阵列和基于集合的API之间的桥梁。Specified by: toArray在界面 Collection 结果 一个数组，以适当的顺序包含此列表中的所有元素。另请参见：Arrays.asList(Object[]) T[] toArray(T[] a) 以适当的顺序返回包含此列表中所有元素的数组（从第一个元素到最后一个元素）；返回数组的运行时类型是指定数组的运行时类型。如果列表适合指定的数组，则返回其中。否则，将为新数组分配指定数组的运行时类型和此列表的大小。如果列表适合指定的数组，并且有空余空间（即，数组的元素多于列表），则紧跟在列表末尾之后的数组中的元素将设置为null。（仅当调用者知道列表不包含任何null元素时，这在确定列表长度时很有用。）与toArray()方法一样，此方法充当基于阵列和基于集合的API之间的桥梁。此外，该方法允许精确控制输出阵列的运行时类型，并且在某些情况下可以用于节省分配成本。假设x是已知仅包含字符串的列表。以下代码可用于将列表转换到新分配的String数组中：String[] y = x.toArray(new String[0]); 请注意，toArray(new Object[0])功能与toArray()相同。Specified by: toArray在界面 Collection 参数类型 T - 要包含集合的数组的组件类型 参数 a - 存储此列表元素的数组（如果足够大）；否则，为此目的分配相同运行时类型的数组。结果 包含此列表元素的数组 异常 ArrayStoreException - 如果指定数组的运行时类型不是此列表中每个元素的运行时类型的超类型 NullPointerException - 如果指定的数组为null default void replaceAll(UnaryOperator operator) 将该列表的每个元素替换为将运算符应用于该元素的结果。操作员抛出的错误或运行时异常将转发给调用者。实现要求：对于此list，默认实现等效于：final ListIterator li = list.listIterator(); while (li.hasNext()) { li.set(operator.apply(li.next())); } 如果列表的listIterator不支持set操作，则在替换第一个元素时将抛出UnsupportedOperationException。参数 operator - 要应用于每个元素的运算符 异常 UnsupportedOperationException - 如果此列表不可修改。如果无法替换元素或者通常不支持修改，则实现可能会抛出此异常 NullPointerException - 如果指定的运算符为null或运算符结果为空值且此列表不允许null元素（optional） 从以下版本开始：1.8 default void sort(Comparator