

第三章-列表简介

- 列表是什么
 - 由一系列按照特定顺序排列的元素组成，用方括号表示，并用逗号进行分割。
 - 怎么访问？
 - such as: `bicycles[0]`
 - 注意：索引是从0开始而不是1.
- 修改、增加和删除元素
 - 修改
 - 直接重写: `bicycles[0]='ducati'`
 - 增加
 - 列表末尾
 - `bicycles.append('ducati')`
 - 列表中插入
 - `bicycles.insert(0, 'ducati')`
 - 0代表位置，意味着插入的元素已经成了第0位
 - 删除
 - `del` 语句
 - `del bicycles[0]`
 - 可以删除任何一个位置的元素
 - `pop ()`
 - 这个方法能让你删除列表末尾的元素，并能让你接着使用它
 - 使用场景：某个列表是没有做完的事，把它们一个个移入已经做完的事情的列表，那就可以用`pop ()` 创建一个中间值，不断转移
 - `popped_bicycle = bicycles.pop()`
 - 用`pop`删除任意位置元素
 - 只要在`pop (1)` 的括号中输入索引即可
 - 根据值删除元素
 - `remove('ducati')`
 - 注意里面是有单引号的
 - 如果用一个中间变量来指代'ducati'的话，就可以反复使用
 - 组织列表
 - 使用`sort ()` 永久性排序
 - 只能用`bicycles.sort()`这种方式来读取，不能用函数的方式（为什么呢？）

- 按字母排序，传递reverse=True的函数可以反向排序
- 使用sorted () 临时排序
 - 同上，但是可以用函数的方式读取
- 倒着打印列表
 - bicycles.reverse ()
- 确定列表长度
 - len ()
 - 注意，这是输出的是int，要想用和其他字符一起用print () 的话要str ()