

创建列表

- C#中的列表是一种数组，可以存储任意数量的对象。列表是动态的，这意味着它们可以随着需要而增长或缩小。列表是使用List类实现的，其中T是列表中可以存储的对象的类型。

```
List<数据类型> name = new List<数据类型>();
```

添加元素到列表

- 使用Add()方法将元素添加到列表末尾

```
List<string> list1 = new List<string>();
list1.Add("王者荣耀");
list1.Add("王国之心");
```

- 使用Insert()方法将元素添加到列表指定索引处

```
List<int> list1 = new List<int>{1,2,3,4,5,6,7}; //可以列表创建时直接将元素添加到列表中
list1.Insert(2,9); //将元素添加到下标为2的地方，原来的元素往后移

结果：
12934567
```

- 使用AddRange()方法将数组中的元素添加到列表末尾

```
List<int> list1 = new List<int>{101,102,103};
int[] array = new int[]{104,105,106};

list1.AddRange(array);

结果：
101 102 103 104 105 106
```

列表中常用的方法

方法	描述
Add(T item)	将一个元素添加到列表的末尾
AddRange(IEnumerable collection)	将一个集合中的元素添加到列表的末尾
Clear()	从列表中删除所有元素

方法	描述
Contains(T item)	如果列表包含指定的元素，则返回 true
CopyTo(T[] array, int arrayIndex)	将列表中的元素复制到指定的数组中
ElementAt(int index)	获取列表中指定索引处的元素
ElementAtOrDefault(int index)	获取列表中指定索引处的元素，如果索引超出范围则返回默认值
Find(Func<T, bool> predicate)	从列表中查找满足指定条件的元素，并返回第一个找到的元素
FindAll(Func<T, bool> predicate)	从列表中查找满足指定条件的元素，并返回一个包含所有找到元素的列表
ForEach(Action action)	将指定的操作应用于列表中的每个元素
IndexOf(T item)	返回列表中指定元素的索引，如果不存在则返回 -1
Insert(int index, T item)	将一个元素插入到列表的指定索引处
Remove(T item)	从列表中删除指定的元素
RemoveAt(int index)	移除列表指定索引处的元素
RemoveAll(Func<T, bool> predicate)	从列表中删除满足指定条件的所有元素
Reverse()	将列表中的元素倒转
Sort()	将列表中的元素排序
ToArray()	将列表中的元素复制到一个数组中
TrimExcess()	从列表中删除所有空白元素

列表中常用的属性

属性	描述
IsReadOnly	确定列表是否只读
IsFixedSize	确定列表是否固定大小
IsSynchronized	确定列表是否同步
Capacity	列表的容量
Count	列表中元素的数量

属性	描述
SyncRoot	列表的同步根
Item	指定索引处的元素
this[int index]	指定索引处的元素