# 组合数据类型

1. **填空题**
2. 使用内置的 list 函数可创建一个列表。
3. Python中列表的元素可通过 索引 或 切片 两种方式访问。
4. 使用内置的 tuple 函数可创建一个元组。
5. 字典元素由 键 和 值 组成。
6. 通过Python的内置方法 keys() 可以查看字典键的集合。
7. 调用items()方法可以查看字典中的所有 值的集合 。
8. **判断题**
9. 列表只能存储同一类型的数据。（ ×）
10. 元组支持增加、删除和修改元素的操作。（× ）
11. 列表的索引从1开始。（× ）
12. 字典中的键唯一。（√ ）
13. 集合中的元素无序。（√ ）
14. 字典中的元素可通过索引方式访问。（× ）
15. **选择题**
16. 下列方法中，可以对列表元素排序的是（ A）。
17. sort()
18. reverse()
19. max()
20. list()
21. 阅读下面的程序：

li\_one = [2, 1, 5, 6]

print(sorted(li\_one[:2]))

运行程序，输出结果是（A ）。

1. [1 ,2]
2. [2 ,1]
3. [1 ,2 ,5 ,6]
4. [6 ,5 ,2 ,1]
5. 下列方法中，默认删除列表最后一个元素的是（ C）。
6. del
7. remove()
8. pop()
9. extend()
10. 阅读下面程序：

lan\_info = {'01': 'Python', '02': 'Java', '03': 'PHP'}

lan\_info.update({'03': 'C++'})

print(lan\_info)

运行程序，输出结果是（B ）。

1. {'01': 'Python', '02': 'Java', '03': 'PHP'}
2. {'01': 'Python', '02': 'Java', '03': 'C++'}
3. {'03': 'C++','01': 'Python', '02': 'Java'}
4. {'01': 'Python', '02': 'Java'}
5. 阅读下面程序：

set\_01 = {'a', 'c', 'b', 'a'}

set\_01.add('d')

print(len(set\_01))

运行程序，输出结果是（ C）。

1. 5
2. 3
3. 4
4. 2
5. **简答题**
6. 列举Python中常用的组合数据类型，简单说明它们的异同。

Python中有包括以下几种常用的组合数据类型：列表（Lists）、元组（Tuples）、集合（Sets）、和字典（Dictionaries）。

异同点：

列表和元组都是有序的序列，但列表是可变的，而元组是不可变的。

集合是无序的，其中不允许重复元素，而列表和元组可以包含重复元素。

字典是键值对的集合，用于建立键和值之间的映射关系。

列表和元组通常用于存储有序的数据，而集合用于处理独特的元素，字典用于存储键-值对的数据。

不同类型的组合数据类型具有不同的性能和适用场景，根据需求选择合适的数据类型非常重要。

1. 简单介绍删除字典元素的几种方式。

使用del语句

使用pop()方法：可以使用del语句来删除字典中的指定元素，通过指定要删除的键来实现。

使用pop()方法：

字典对象提供了pop(key, default)方法，它可以删除指定键的元素并返回其值。如果键不存在，可以通过default参数指定一个默认值，而不会引发KeyError异常。

使用popitem()方法：

popitem()方法用于删除字典中的最后一个元素，并以元组的形式返回被删除的键-值对。这个方法通常用于随机删除元素，因为字典中的元素是无序的。