6.4 字典类型定义：

* 字典类型定义
* 字典处理函数及方法
* 字典类型应用场景

# 字典类型定义

理解“映射”

* 映射是一种键(索引)和值(数据)的对应

序列类型是由0…N整数作为数据的默认类型

而映射类型则是由用户为数据定义索引

字典类型是“映射“的体现

* 键值对：键是数据索引的扩展
* 字典是键值对的集合，键值对之间无序
* 采用大括号{}和dict()创建，键值对用冒号:表示

# 字典类型的用法

在字典变量中，通过键获得值

D = {“中国”:”北京,”美国”:”华盛顿”,”法国”:”巴黎”}

集合，序列和字典的比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | 集合 | 序列 | 字典 |
| 创建 | 建立集合类型使用{}或set()  注：  建立空集合类型使用,必须使用set() |  | 大括号{}和dict()创建 |
|  | 无序，不存在相同元素，且不可更改 |  |  |
| 操作函数和方法 |  |  |  |
|  |  | 5个函数和方法：  len(s)  min(s) 序列中的元素可以比较，否则报错  max(s)  s.index(x) 或 s.index(x,i,j) 返回序列中s从i开始到j结束第一次出现x的位置  s.count(x) 返回s中出现x的总次数 |  |
| 操作实例 |  |  | 1、定义空字典： d = {}  2、向d新增2个键值对元素： d = {‘1’:”北京”,’2’:”日本”} 或 d[“a”] = 1;d [“b”] = 2  3、修改第2个元素: d[“2”] = “美国”  4、判断字符”c”是否是d的键: “c” in d.keys() 或者“c” in d  5、计算d的长度 : len(d)  6、 清空d: d.clear() |
| 应用场景 | - 包含关系的比较  "p" in {"p","y","123"} - 数据去重  ls = ["p","p","y","y",123] # 有重复元素的列表  s = set(ls)  print(s)  lt = list(s)# 无重复元素的列表 |  | 映射的表达：   * 映射无处不在，键值对无处不在 * 例如：统计数据出现的次数，数据是键，次数是值 * 最主要作用：表达键值对数据，进而操作它们   元素遍历：  For k in d:  <语句块> |
|  |  |  |  |