



内置数据结构2

北京理工大学计算机学院 高玉金

2019年3月



集合类型结构：集合set

- 集合类型与数学中集合的概念一致，即包含0个或多个数据项的**无序**组合
- 集合中元素**不可重复**，元素类型只能是固定数据类型，例如：整数、浮点数、字符串、元组等
- 列表、字典和集合类型本身都是**可变数**据类型，不能作为集合的元素出现
- 由于集合是无序组合，它没有索引和位置的概念，不能分片，集合中元素可以动态增加或删除



集合的生成

- 集合用大括号 ({}) 表示
- 可以用赋值语句生成一个集合，如 `s = {3, "4", 5.0}`
- 可以用 `set(x)` 函数生成集合，`x` 为序列类型（字符串、list 或元组）
- 集合的打印效果与定义顺序可以不一致

```
In [131]: st2=set("Hello")
```

```
In [132]: st2
```

```
Out[132]: {'H', 'e', 'l', 'o'}
```

```
In [133]: st2
```

```
Out[133]: {'H', 'e', 'l', 'o'}
```

```
In [134]: print(st2)
```

```
{'o', 'H', 'l', 'e'}
```



集合的使用

- 成员关系测试 (in 和 not in)
- 元素去重
- 删除数据项

```
In [143]: s2 = set(s)-{"H"}
```

```
In [144]: s2
```

```
Out[144]: {'d', 'e', 'l', 'o', 'r', 'w'}
```

```
In [139]: s=list("Helloworld")
```

```
In [140]: len(set(s))
```

```
Out[140]: 7
```

```
In [141]: set(s)
```

```
Out[141]: {'H', 'd', 'e', 'l', 'o', 'r', 'w'}
```



映射类型结构：字典

- 通过任意键信息查找一组数据中值信息的过程叫映射，Python语言中通过字典实现映射
- Python语言中的字典可以通过大括号({})建立：
 {<键1>:<值1>, <键2>:<值2>, ... , <键n>:<值n>}
- 键和值通过冒号连接，不同键值对通过逗号隔开
- 字典打印出来的顺序与创建之初的顺序不同
- 字典是集合类型的延续，各个元素并没有顺序之分



字典中键的要求

- 每个键只能对应一个值，不允许一键对应多个值
- 当字典键重复赋值时，取最后的赋值
- 键必须是可哈希的，**可变类型对象如**列表和字典不可哈希，所以不能作为字典的键

```
In [80]: hash("Hello")
Out[80]: 8387971725452392808
```

```
In [81]: s=list("Hello")
```

```
In [82]: s
Out[82]: ['H', 'e', 'l', 'l', 'o']
```

```
In [83]: hash(s)
Traceback (most recent call last):
```

TypeError: unhashable type: 'list'



字典的生成

- 赋值生成，如 `d={ “name” :” Diego” ,” age” :12}`
- 特殊字符串转字典 `dt = eval(“ { “a” :5} ”)`
- 列表式生成

示例：有20个学生，成绩在60-100之间。请筛选出成绩在90分以上的学生

```
stuInfo = { 'student' + str(i):random.randint(60,100) for i in range(20) }  
  
{name:score for name,score in stuInfo.items() if score > 90}
```



字典的生成

```
{ 'student3': 99,  
  'student4': 100,  
  'student5': 92,  
  'student8': 96,  
  'student9': 95,  
  'student13': 98,  
  'student18': 91,  
  'student19': 96 }
```




字典的主要方法

函数和方法	描述
<code><d>.keys()</code>	返回所有的键信息
<code><d>.values()</code>	返回所有的值信息
<code><d>.items()</code>	返回所有的键值对（注意返回值类型）
<code><d>.get(<key>,<default>)</code>	键存在则返回相应值，否则返回默认值
<code><d>.pop(<key>,<default>)</code>	键存在则返回相应值，同时删除键值对，否则返回默认值
<code><d>.popitem()</code>	随机从字典中取出一个键值对，以元组(key, value)形式返回
<code><d>.clear()</code>	删除所有的键值对
<code>del <d>[<key>]</code>	删除字典中某一个键值对
<code><key> in <d></code>	如果键在字典中返回True，否则返回False



词典元素的遍历

与其他组合类型一样，字典可以通过for...in语句对其元素进行遍历，基本语法结构：

for <变量名> in <字典名>/keys()/values()/items():

语句块

```
>>> D={"中国":"北京", "美国":"华盛顿", "法国":"巴黎"}
>>> for k,v in D.items():
    print(k+":"+v)
```

```
中国:北京
美国:华盛顿
法国:巴黎
```

```
for d in D:
    print(d+":"+D[d])
```