

# ※ 标准数据类型-Set

## ※ 标准数据类型-Set

开篇

### 6. 集合 Set

6.1 集合的操作 (集合是 无序 的)

6.2 字典本身是序列, 可以进行索引与切片

6.3 Dict 字典内置函数

## 开篇

什么是代码

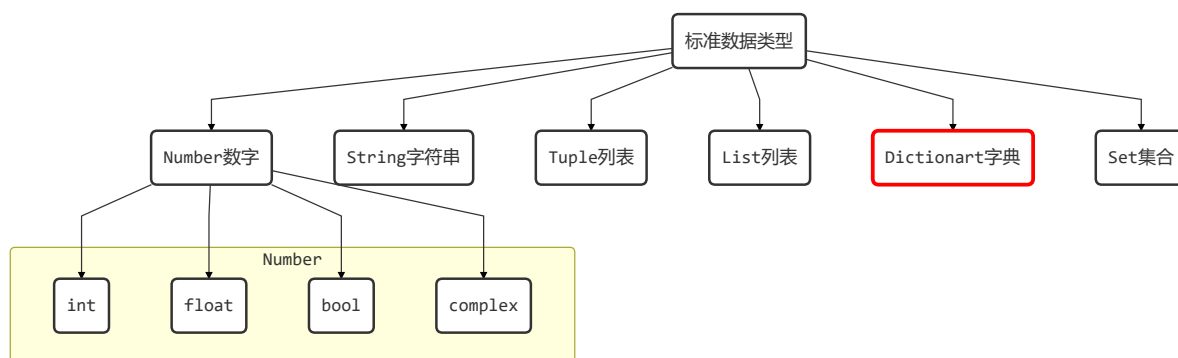
现实世界事物在计算机世界中的映射

什么是写代码

用计算机语言来描述现实中的事物

画画\摄影本身就是通过图形轮廓,色彩光影对现实世界中事物进行描述,

所以同样道理,如果我们想要用计算语言对现实进行描述时,这些元素就是计算语言的标准数据类型.



- **不可变数据 (3 个)** : Number (数字) 、String (字符串) 、Tuple (列表) ;
- **可变数据 (3 个)** : List (列表) 、Dictionary (字典) 、Set (集合) 。

## 6. 集合 Set

```

1 set() #创建一个空集合
2
3 {1,'2',False,(1,3)} #花括号之内元素之间以逗号隔开
4
5 set1={1,'2',False,(1,3)} # 创建一个set集合赋值给变量set1
6
7 type(set1) #检查set1的类型

```

## 6.1 集合的操作 (集合是无序的)

- `set1[0]` 集合(set)不支持索引成员值
- `set1[2:4]` 集合(set)不支持切片操作
- `set1[0]=3` 集合(set)不支持修改成员值

操作符	描述	实例	结果
in	成员运算符 - 如果对象中包含给定的字符返回 True	1 in {1,20}	True
not in	成员运算符 - 如果对象中不包含给定的字符返回 True	1 not in {1,20}	False

## 6.2 字典本身是序列，可以进行索引与切片

```

1 set1.update('b') #集合(set)允许增加成员
2 print(set1)
3
4 set1.remove('b') #集合(set)允许删除成员
5 print(set1)
6
7 for x in set1:print(x,end=',') #可以通过迭代方法遍历集合(set)成员
8
9 set1={1,'2',False,(1,3)}
10 set2={1,'2',False,(1,3),3}
11 print(set2-set1) #元组的求差操作,必须是大的集合减小集合
12
13 print(set2|set1) #集合set1与set2的所有元素
14
15 print(set2 & set1) #集合set1与set2都包含的元素
16
17 print(set2 ^ set1) #集合set1与set2不同时包含的元素
18
19 1 in set1 #检查成员操作
20
21 1 not in set1 #检查成员操作
22
23 del(set1) #通过del()方法删除集合

```

## 6.3 Dict 字典内置函数

```
1  help(set) #通过help查看set内置函数
2
3  set1=set(('old',))
4  set2=set1.copy() #将集合set1复制到新的集合set2中
5  print(id(set1),id(set2))
6
7  set1=set(('old',))
8  set1.add('new') # 增加一个元素到集合set中
9  print(set1)
10
11 set1={'old', 'new'}
12 set2={'s', 'old'}
13 set.difference(set1,set2) # 输出set1包含,但set2不包含的成员值
14
15 set1={'old', 'new'}
16 set2={'s', 'old'}
17 set.difference_update(set1,set2) # set1包含,但set2不包含的成员值,并更新set1为该值
18 print(set1)
19
20 set1={'old', 'new', 's'}
21 set1.discard('s') # 传入指定的值,如果集合中包含该值则删除该成员,如果没有则不抛异常
22 print(set1)
23
24 set1={'old', 'new'}
25 set2={'s', 'old'}
26 set3=set.intersection(set1,set2) #返回set1与set2相同值的新集合
27 print(set3)
28
29 set1={'old', 'new'}
30 set2={'s', 'old'}
31 set.intersection_update(set1,set2) #得到set1与set2的相同值并更新set1
32 print(set1)
33
34 set1={'old', 'new'}
35 set2={'s', 'old'}
36 set.isdisjoint(set1,set2) #判断set1与set2是否无交集
37
38 set1={'old', 'new', 'ds'}
39 set2={'old', 'new'}
40 set.issubset(set1,set2) #判断set1是否从属于set2
41
42 set1={'old', 'new', 'ds'}
43 set2={'old', 'new'}
44 set.issuperset(set1,set2) #判断set1是否包含set2
45
46 set1={'old', 'new', 'ds'}
47 set1.pop() #随机删除一个成员
48 print(set1)
49
50 set1={'old', 'new', 's'}
51 set1.remove('s') # 传入指定的值,如果集合中包含该值则删除该成员,如果没有则抛出异常
52 print(set1)
53
```

```
54 set1={'old', 'new'}
55 set2={'s', 'old'}
56 set3=set.symmetric_difference(set1,set2) #返回去掉set1与set2相同值的新集合set
57 print(set3)
58
59 set1={'old', 'new'}
60 set2={'s', 'old'}
61 set.symmetric_difference_update(set1,set2) #得到去掉set1与set2相同值的集合并更新set1
62 print(set1)
63
64 set1={'old', 'new'}
65 set2={'s', 'old'}
66 set3=set.union(set1,set2) #将set1与set2合并
67 print(set3)
68
69 set1={'old', 'new'}
70 set2={'s', 'old'}
71 set.update(set1,set2) #将set1与set2合并,并更新set1的值
72 print(set1)
```