

5-10 成员运算符

成员运算符 有两个: in not in

主要特点 判断一个元素是否在另外一组元素中 并且返回一个 bool 类型的值

in 元素是否在另一组元素中

a in [1,2,3,4,5] ==> Error 因为 a 没有定义

a=1 a in [1,2,3,4,5] ==> True a 等于 1, 1 是不是在列表[1,2,3,4,5]中 返回结果为真
b=6 b in [1,2,3,4,5] ==> False b 等于 6, 6 是不是在列表[1,2,3,4,5]中 返回结果为假
=====

not in 元素是否不在另一组元素中

a=1 a not in [1,2,3,4,5] ==> False a 等于 1, 1 是不是不在列表[1,2,3,4,5]中
1 在列表中 返回结果为假
b=6 b not in [1,2,3,4,5] ==> True b 等于 6, 6 是不是不在列表[1,2,3,4,5]中
6 不在列表中 返回结果为真
=====

以上举例了 成员运算符在 list 中的使用 下面介绍在 string tuple set 和 dict 中的使用

str b='h' b in 'hello' ==> True 字符 h 在 字符 hello 中 对不对

tuple b='h' b not in (1,2,3,4,5) ==> True 字符 h 在元组(1,2,3,4,5)中 对不对

set b='h' b not in {1,2,3,4,5} ==> True 字符 h 在集合{1,2,3,4,5}中 对不对

dict b='a' b in {'c':1} ==> False
b='1' b in {'c':1} ==> False
b='c' b in {'c':1} ==> True

字典 dict 的成员运算是针对 key:Value 的 Key 并不是 Value

5-11 身份运算符

(这里需要理解对象的定义)(现在只是一个简单的了解)

身份运算符: `is` `is not` 并且返回一个 `bool` 值 用于比较两个对象的存储单元(内存地址)

`is` 是判断 2 个表示符是不是引用子一个对象

`x is y` 类似于 `id(x) == id(y)`

如果引用的是同一对象 则返回 `True` 否则(两个表示符不是同一对象) 则返回 `False`

`x is not y` 类似于 `id(x) != id(y)`

如果引用的不是同一对象 则返回 `True` 否则(两个表示符引用的是同一对象) 则返回 `False`

```
>>> a=1          >>> a=1          >>> a='hello'    >>> c='hello'
>>> b=2          >>> b=1          >>> b='world'    >>> a is c
>>> a is b       >>> a is b       >>> a is b       True
False           True           False
```

```
>>> a=1
>>> b=1.0
>>> a==b
True          # 这是单纯比较 a 和 b 的值 就是 1==1 True
>>> a is b
False         # 这是比较 a 和 b 的内存地址 所以会返回 False
```

```
>>> id(a)
271112368
>>> id(b)
55816320
```

```
>>> a={1,2,3}
>>> b={2,1,3}
>>> a==b
True
>>> a is b
False
```

集合 `set` 是无序的 所以只比较值

```
a==b ==> True
id(a) != id(b)
a is b ==> False
```

```
>>> c=(1,2,3)
>>> d=(2,1,3)
>>> c==d
False
>>> c is d
False
```

`c` 与 `d` 是元组 `tuple` 是有序的 要按照顺序来 所以

```
c != d ==> False
id(c) != id(d) ==> False
```

5-12 如何判断变量的值, 身份 与 类型

值 Value 身份 id 类型 type

type 类型的判断

```
>>> a=1  
>>> type(a)==int  
True
```

如果这样做就太麻烦了

函数(Function) isinstance(变量名称要判断的对象,类型名称甚至可以为元组) 返回一个 bool 类型

```
isinstance(a,int) ==> True
```

如果为元组:

```
isinstance(a,(int,str,float))
```

a 是不是 int str float 中的一种? 同样也会返回 一个 bool 类型

```
>>> isinstance(a,(int,str,float))  
True
```

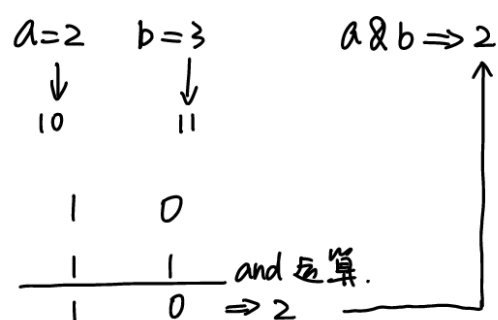
5-13 位运算符

&		^	~	<<	>>
按位与	按位或	按位异或	按位取反	左移动	右移动

所有的位运算符 都是把数字当做二进制数字进行运算
如果是其他进制的数 会先转成二进制再运算

按位与 (and) $a=2 \Rightarrow 10$
 $b=3 \Rightarrow 11$
 $a \& b \Rightarrow 2$

按位与 (and)



按位或 (or) $a=2 \Rightarrow 10$
 $b=3 \Rightarrow 11$
 $a | b \Rightarrow 3$

按位或 (or)

