

day18: 运算符重载

笔记本: Python基础

创建时间: 2018/7/1 14:47

更新时间: 2018/7/2 19:39

作者: liuchang_0412@163.com

1.id() 函数:

id(obj) 返回对象的标识(identity)

2.运算符重载

定义:

用自定义的规则实现实例之间的运算符操作或函数操作

作用:

让实例像数学表达式一样进行运算操作

让实例像内建对象一样, 进行内建函数操作

让程序简洁易读

3.对象转字符串函数重载:

①repr(obj)

repr()函数的重载方法:

__repr__(self):

...

②str(obj)

str()函数的重载方法:

__str__(self):

...

注: 如果对象没有__str__方法, 则用repr(obj)函数的结果代替

示例: 见mynumber.py

#练习: 自定义一个类Student, 有学生姓名, 年龄, 成绩:

Class Student:

....

s1=Student("张三", 21, 98)

L = []

L.append(s1)

s1 = Student("李四", 20, 88)

L.append(s1)

for x in L:

print(x)

4.算术运算符的重载:

①重载方法:

+ __add__

- __sub__

* __mul__

```
/      __truediv__
//     __floordiv__
%      __mod__
**     __pow__
```

②二元运算符重载的格式:

```
def __xxx__(self, other):
```

...

注: 二元运算符的重载方法的参数列表中只能有两个参数

③重载说明:

运算符重载方法的参数已经有固定的含义, 不可改变原有意义

除__call__方法外, 其他重载方法的参数个数不可改变

#练习: 自定义两个列表类, 实现两个自定义列表相加。

```
class MyList:
```

....

```
L1=MyList([1,2,3])
```

```
L2=MyList([4,5,6])
```

```
L3=L1+L2
```

```
print(L3)  #MyList([1,2,3,4,5,6])
```

```
L3=L2+L1
```

```
print(L3)  #MyList([4,5,6,1,2,3])
```

5.方向算术运算符重载:

```
__radd__(self, lhs)  #加法 lhs + self.data (lhs --> left hand side)
```

```
__rsub__(self, lhs)  #减法 lhs - self.data
```

```
__rmul__(self, lhs)  #乘法 lhs * self.data
```

```
__rtruediv__(self, lhs)  #除法 lhs / self.data
```

```
__rfloordiv__(self, lhs)  #地板除法 lhs // self.data
```

```
__rmod__(self, lhs)  #求余 lhs % self.data
```

```
__rpow__(self, lhs)  #求幂 lhs ** self.data
```

6.复合赋值运算符重载:

```
__iadd__(self, rhs)  #求和 self.data + rhs
```

```
__isub__(self, rhs)  #求差 self.data - rhs
```

```
__imul__(self, rhs)  #求积 self.data * rhs
```

```
__itruediv__(self, rhs)  #除法 self.data / rhs
```

```
__ifloordiv__(self, rhs)  #地板除 self.data // rhs
```

```
__imod__(self, rhs)  #求余 self.data % rhs
```

```
__ipow__(self, rhs)  #求幂 self.data ** rhs
```

7.一元运算符重载:

```
__neg__  #负号 -
```

`__pos__` #正号 +

重载方法:

```
def __xxx__(self):
```

...

示例:

见mylist2.py

8.比较运算符的重载:

`__lt__` #小于 <

`__le__` #小于等于 <=

`__gt__` #大于 >

`__ge__` #大于等于 >=

`__eq__` #等于 ==

`__ne__` #不等于 !=

比较运算符通常返回True或者False

9.位运算符重载

`__invert__` #取反 ~

`__and__` #位与 &

`__or__` #位或 |

`__xor__` #位异或 ^

`__lshift__` #左移 <<

`__rshift__` #右移 >>

10.内建函数重载:

`__abs__` abs(obj) 函数调用

`__len__` len(obj)

`__reversed__` reversed(obj)

`__round__` round(obj)

...

#练习:

1.自定义一个类Mylist, 实现 + , += , * , *=操作。实现能够用len(x)返回自定义列表的元素个数

2.实现有序集合OrderSet类, 能实现两个集合的交集(&), 并集(|), 补集(-), 对称补集(^), ==, != 等操作。功能与集合相同。(要求: 内部用列表来存储)

例如:

```
s1 = OrderSet([1,2,3,4])
```

```
s2 = OrderSet([3,4,5])
```

```
print(s1 & s2) #OrderSet({3,4})
```

```
print(s1 | s2) #OrderSet({1,2,3,4,5})
```

```
print(s1 ^ s2) #OrderSet({1,2,5})
```

```
if OrderSet({1,2,3}) != OrderSet({1,2,3,4}):  
    print("不相等")
```