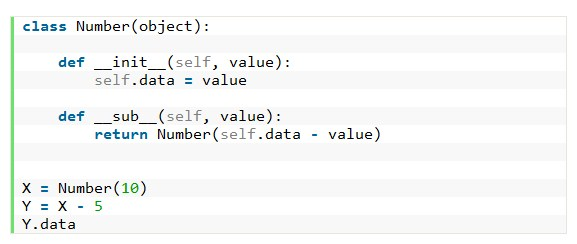
**构造函数与表达式： \_\_init\_\_, \_\_sub\_\_**



**常见运算符重载方法**

|  |  |  |
| --- | --- | --- |
| method | overload | call |
| \_\_init\_\_ | 构造函数 | 对象创建: X = Class(args) |
| \_\_del\_\_ | 析构函数 | X对象收回 |
| \_\_add\_\_ | 云算法+ | 如果没有\_iadd\_， X+Y， X+=Y |
| \_\_or\_\_ | 运算符| | 如果没有\_ior\_，X|Y, X|=Y |
| \_repr\_\_, \_\_str\_\_ | 打印，转换 | print(X)，repr(X)，str(X) |
| \_\_call\_\_ | 函数调用 | X(\*args, \*\*kwargs) |
| \_\_getattr\_\_ | 点号运算 | X.undefined |
| \_\_setattr\_\_ | 属性赋值语句 | X.any=value |
| \_\_delattr\_\_ | 属性删除 | del X.any |
| \_\_getattribute\_\_ | 属性获取 | X.any |
| \_\_getitem\_\_ | 索引运算 | X[key]，X[i:j] |
| \_\_setitem\_\_ | 索引赋值语句 | X[key]，X[i:j]=sequence |
| \_\_delitem\_\_ | 索引和分片删除 | del X[key]，del X[i:j] |
| \_\_len\_\_ | 长度 | len(X)，如果没有\_\_bool\_\_，真值测试 |
| \_\_bool\_\_ | 布尔测试 | bool(X) |
| \_\_lt\_\_, \_\_gt\_\_,  \_\_le\_\_, \_\_ge\_\_,  \_\_eq\_\_, \_\_ne\_\_ | 特定的比较 | X<Y，X>Y，X<=Y，X>=Y，  X==Y，X!=Y  注释：（lt: less than, gt: greater than,    le: less equal, ge: greater equal,    eq: equal, ne: not equal  ） |
| \_\_radd\_\_ | 右侧加法 | other+X |
| \_\_iadd\_\_ | 实地（增强的）加法 | X+=Y(or else \_\_add\_\_) |
| \_\_iter\_\_, \_\_next\_\_ | 迭代环境 | I=iter(X), next() |
| \_\_contains\_\_ | 成员关系测试 | item in X(任何可迭代) |
| \_\_index\_\_ | 整数值 | hex(X), bin(X),  oct(X) |
| \_\_enter\_\_, \_\_exit\_\_ | 环境管理器 | with obj as var: |
| \_\_get\_\_, \_\_set\_\_,  \_\_delete\_\_ | 描述符属性 | X.attr, X.attr=value, del X.attr |
| \_\_new\_\_ | 创建 | 在\_\_init\_\_之前创建对象 |

**索引和分片： \_\_getitem\_\_, \_\_setitem\_\_**

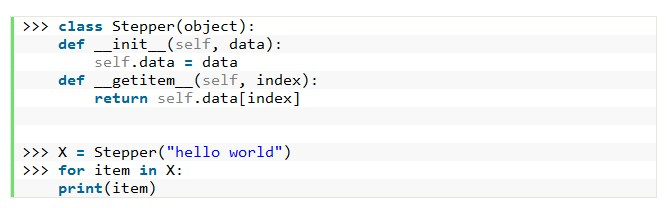
如果在类中定义的话，则对于实例的索引运算，会自动调用\_\_getitem\_\_。当实例X出现X[i]这样的索引运算时，Python会自动调用\_\_getitem\_\_方法



**拦截分片**



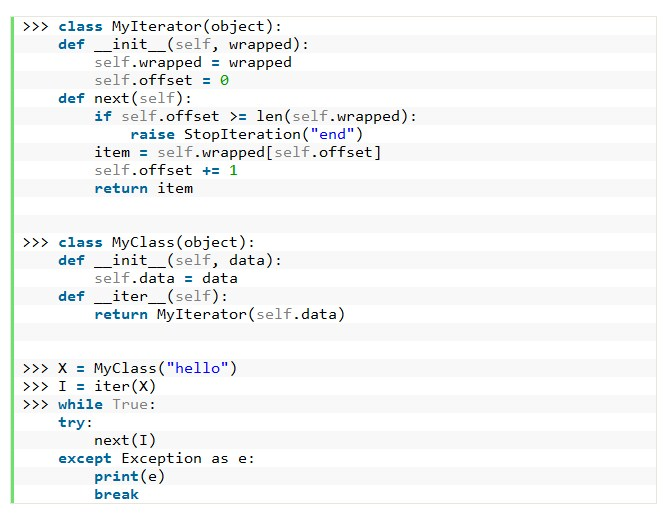
**索引迭代： \_\_getitem\_\_**



**迭代器对象: \_\_iter\_\_,  \_\_next\_\_**

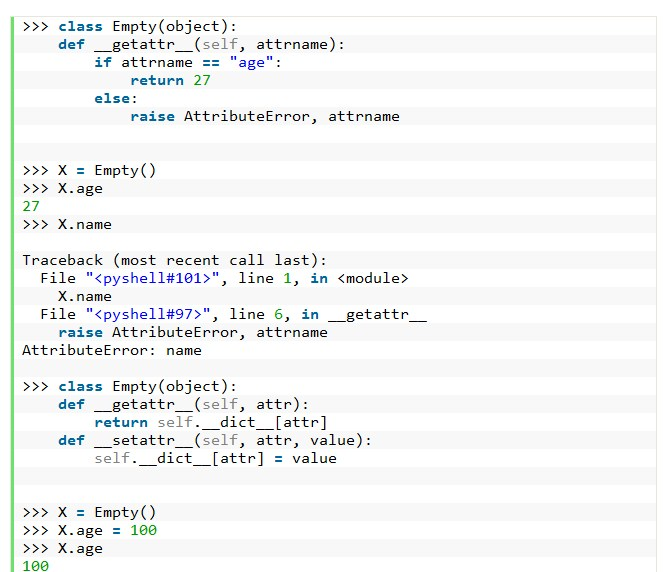
尽管上一节\_\_getitem\_\_是有效的，但它是迭代退而求其次的方法。Python所有的迭代环境会有优先尝试\_\_iter\_\_的方法，再尝试\_\_getitem\_\_。

从技术角度上讲，迭代环境是通过iter去尝试寻找\_\_iter\_\_方法来实现，而这种方法返回一个迭代器对象。如果已经提供了，python会重复调用迭代器对象的next()方法，直到发生StopIteration异常。如果没有找到\_\_iter\_\_，python会使用\_\_getitem\_\_机制。

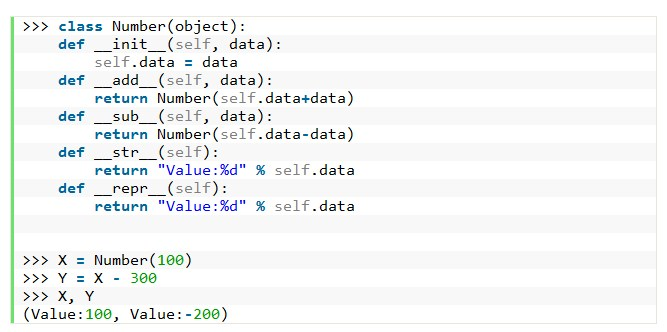


**\_\_getattr\_\_和\_\_setattr\_\_捕捉属性的的引用**

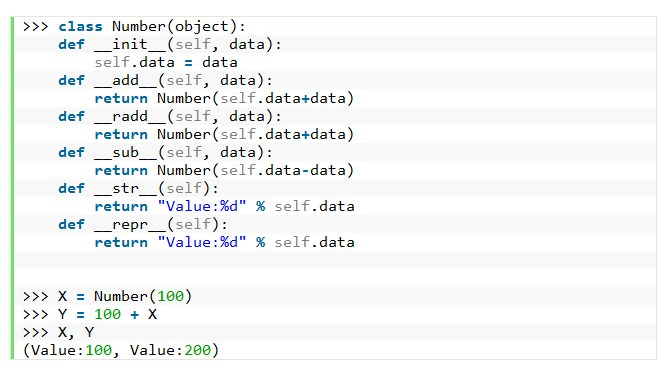
\_\_getattr\_\_拦截属性.运算符



**\_\_repr\_\_和\_\_str\_\_会返回字符串表达形式**

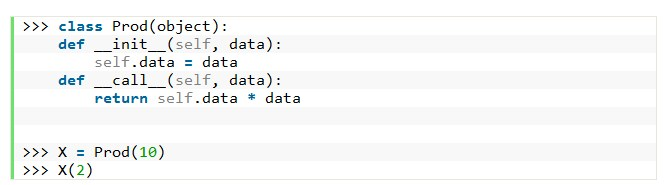


**\_\_radd\_\_处理右侧加法**



**\_\_call\_\_拦截调用**

当实例调用时，使用\_\_call\_\_方法



**\_\_del\_\_是析构器**

当实例创建时，就会调用\_\_init\_\_构造方法。当实例空间被收回时，析构函数\_\_del\_\_就会自动执行。

