

马哥教育



主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

马哥教育

www.magedu.com





Python对象类型 及其运算

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

马哥教育

www.magedu.com



❖ Python程序中保存的所有数据都是围绕对象这个概念展开的

- ➡ 程序中存储的所有数据都是对象
- ➡ 每个对象都有一个身份、一个类型和一个值
 - 例如，`school="MaGe Linux"`会以“MaGe Linux”创建一个字符串对象，其身份是指向它在内存中所处位置的指针(其在内存中的地址)，而`school`就是引用这个具体位置的名称
- ➡ 对象的类型也称对象的类别，用于描述对象的内部表示及它支持的方法和操作
- ➡ 创建特定类型的对象时，有时也将该对象称为该类型的实例
- ➡ 实例被创建后，其身份和类型就不可改变
 - 如对象值是可修改的，则称为可变对象
 - 如果对象的值不可修改，则称为不可变对象
- ➡ 如果某个对象包含对其它对象的引用，则将其称为容器；
- ➡ 大多数对象都拥有大量特有的数据属性和方法
 - 属性：与对象相关的值
 - 方法：被调用时将在对象上执行某些操作的函数
 - 使用点(.)运算符可以访问属性和方法



对象的身份与类型

- ❖ Python内置函数`id()`可返回一个对象的身份，即该对象在内存中的位置
 - ➔ `is`运算符用于比较两个对象的身份
 - ➔ `type()`用于返回一个对象的类型
 - ➔ 对象类型本身也是一个对象，称为对象的类
 - 该对象的定义是唯一的，且对于某类型的所有实例都是相同的
 - 所有类型对象都有一个指定的名称，可用于执行类型检查，如`list`、`dict`

```
if a is b:  
    statements  
if a == b:  
    statements  
if type(a) is type(b):  
    statements
```



对象类型

- ❖ 数字
- ❖ 字符串
- ❖ 列表
- ❖ 字典
- ❖ 元组
- ❖ 文件
- ❖ 集合
- ❖ 其它类型
- ❖ 编程单元类型
- ❖ 与实现相关的类型

例子

3077, 3.14, 300000
'magedu.com', "spam"
['one', 'two', 'three']
{'course':'linux', 'tutor':'mage'}
(32, 'spam', 'eggs')
myFile=open('/tmp/tfile','r')
set('abc'), {'a', 'b', 'c'}
类型、None、布尔型
函数、模块、类
编译的代码堆栈跟踪



用于表示数据的内置类型

类型分类	类型名称	描 述
None	Type (None)	null对象None
数字	int	整数
	Long	任意精度的整数（仅在Python 2中使用）
	float	浮点数
	complex	复数
序列	bool	布尔值（True或False）
	str	字符串
	unicode	Unicode字符串（仅在Python 2中使用）
	list	列表
	tuple	元组
	xrange	xrange()函数创建的整数范围（在Python 3中称为range）
映射 集合	dict	字典
	set	可变集合
	frozenset	不可变集合



数字类型操作

- ❖ Python使用5种数字类型：布尔型、整型、长整型、浮点型和复数，所有数字类型均为不可变
- ❖ 数字操作

操 作	描 述	操 作	描 述
$x + y$	加法	$x ** y$	乘方 (x^y)
$x - y$	减法	$x \% y$	取模 ($x \bmod y$)
$x * y$	乘法	$-x$	一元减法
x / y	除法	$+x$	一元加法
$x // y$	截断除法		

- ❖ 比较运算

操 作	描 述	操 作	描 述
$x << y$	左移	$x y$	按位或
$x >> y$	右移	$x ^ y$	按位异或
$x \& y$	按位与	$\sim x$	按位求反



- ❖ 序列表示索引为非负整数的有序对象集合，包括字符串、列表和元组
 - ➡ 字符串是字符的
 - ➡ 列表和元组是任意Python对象的序列
- ❖ 字符和元组属于不可变序列，而列表则支持插入、删除和替换元素
- ❖ 所有序列都支持迭代

马哥教育

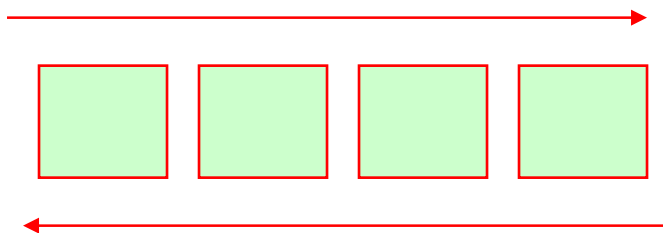
www.magedu.com



适用于所有序列的操作和方法

- ❖ `s[i]`: 索引运算符
- ❖ `s[i:j]`为切片运算符, `s[i:j:stride]`为扩展切片运算符
- ❖ `min(s)`和`max(s)`只适用于能够对元素排序的序列
- ❖ `sum(s)`只适用于数字序列

项 目	描 述
<code>s[i]</code>	返回一个序列的元素 <i>i</i>
<code>s[i:j]</code>	返回一个切片
<code>s[i:j:stride]</code>	返回一个扩展切片
<code>len(s)</code>	<i>s</i> 中的元素数
<code>min(s)</code>	<i>s</i> 中的最小值
<code>max(s)</code>	<i>s</i> 中的最大值
<code>sum(s [, initial])</code>	<i>s</i> 中各项的和
<code>all(s)</code>	检查 <i>s</i> 中的所有项是否为True
<code>any(s)</code>	检查 <i>s</i> 中的任意项是否为True



马哥教育

www.magedu.com



适用于可变序列的操作

项 目	描 述
<code>s[i] = v</code>	项目赋值
<code>s[i:j] = t</code>	切片赋值
<code>s[i:j:stride] = t</code>	扩展切片赋值
<code>del s[i]</code>	项目删除
<code>del s[i:j]</code>	切片删除
<code>del s[i:j:stride]</code>	扩展切片删除

马哥教育

www.magedu.com



可用于列表的方法

❖ **list(s)**可将任意可迭代类型转换为列表，而如果**s**已经是一个列表，则该函数构造的新列表是**s**的一个浅复制

方 法	描 述
<code>list(s)</code>	将 s 转换为一个列表
<code>s.append(x)</code>	将一个新元素 x 追加到 s 末尾
<code>s.extend(t)</code>	将一个新列表 t 追加到 s 末尾
<code>s.count(x)</code>	计算 s 中 x 的出现次数
<code>s.index(x, [, start [, stop]])</code>	当 s[i] == x 时返回最小的 i ，可选参数 stop 用于指定搜索的起始和结束索引
<code>s.insert(i, x)</code>	在索引 i 处插入 x
<code>s.pop([i])</code>	返回元素 i 并从列表中移除它。如果省略 i ，则返回列表中最后一个元素
<code>s.remove(x)</code>	搜索 x 并从 s 中移除它
<code>s.reverse()</code>	颠倒 s 中的所有元素的顺序
<code>s.sort([key [, reverse]])</code>	对 s 中的所有元素进行排序。 key 是一个键函数。 reverse 是一个标志，表明以倒序对列表进行排序。 key 和 reverse 应该始终以关键字参数的形式指定



适用于字符串的操作

❖ Python 2提供两种字符串对象类型

➡ 字节字符串：字节(8bit数据)序列

➡ Unicode字符串：Unicode字符(16bit数据)序列

➡ Python可以使用32 bit整数保存Unicode字符，但此为可选特性

方 法	描 述
<code>s.capitalize()</code>	首字符变大写
<code>s.index(sub [, start [,end]])</code>	找到指定子字符串 <code>sub</code> 首次出现的位置，否则报错
<code>s.join(t)</code>	使用 <code>s</code> 作为分隔符连接序列 <code>t</code> 中的字符串
<code>s.lower()</code>	转换为小写形式
<code>s.replace(old, new [,maxreplace])</code>	替换一个子字符串
<code>s.split([sep [,maxsplit]])</code>	使用 <code>sep</code> 作为分隔符对一个字符串进行划分。 <code>maxsplit</code> 是划分的最大次数
<code>s.strip([chrs])</code>	删掉 <code>chrs</code> 开头和结尾的空白或字符
<code>s.upper()</code>	将一个字符串转换为大写形式

www.magedu.com



序列操作总结

❖ 序列类型支持的操作符

操 作	描 述	操 作	描 述
$s + r$	连接	<code>for x in s:</code>	迭代
$s * n, n * s$	制作 s 的 n 个副本, n 为整数	<code>all(s)</code>	如果 s 中的所有项都为True, 则返回True
$v1, v2, \dots, vn = s$	变量解包 (unpack)	<code>any(s)</code>	如果 s 中的任意项为True, 则返回True
$s[i]$	索引	<code>len(s)</code>	长度
$s[i:j]$	切片	<code>min(s)</code>	s 中的最小项
$s[i:j:stride]$	扩展切片	<code>max(s)</code>	s 中的最大项
$x \text{ in } s, x \text{ not in } s$	从属关系	<code>sum(s [, initial])</code>	具有可选初始值的项的和

❖ 列表类型支持的操作

操 作	描 述	操 作	描 述
$s[i] = x$	索引赋值	<code>del s[i]</code>	删除一个元素
$s[i:j] = r$	切片赋值	<code>del s[i:j]</code>	删除一个切片
$s[i:j:stride] = r$	扩展切片赋值	<code>del s[i:j:stride]</code>	删除一个扩展切片

Python

映射类型

- ❖ 映射类型表示一个任意对象的集合，而且可以通过另一个几乎是任意键值的集合进行索引
- ❖ 与序列不同，映射是无序的，它通过键进行索引
 - ➡ 任何不可变对象都可用作字典的键，如字符串、数字、元组等
 - ➡ 包含可变对象的列表、字典和元组不能用作键
- ❖ `m[k]`运算符用于选择映射对象中的一项，其中“k”是一个键值
 - ➡ 引用不存在的键会引发`KeyError`异常

马哥教育

www.magedu.com



字典的方法和操作

项 目	描 述
<code>len(m)</code>	返回 <code>m</code> 中的项目数
<code>m[k]</code>	返回 <code>m</code> 中键 <code>k</code> 的项
<code>m[k]=x</code>	将 <code>m[k]</code> 的值设为 <code>x</code>
<code>del m[k]</code>	从 <code>m</code> 中删除 <code>m[k]</code>
<code>k in m</code>	如果 <code>k</code> 是 <code>m</code> 中的键，则返回True
<code>m.clear()</code>	删除 <code>m</code> 中的所有项目
<code>m.copy()</code>	返回 <code>m</code> 的一个副本
<code>m.fromkeys(s [,value])</code>	创建一个新字典并将序列 <code>s</code> 中的所有元素作为新字典的键，而且这些键的值均为 <code>value</code>
<code>m.get(k [,v])</code>	返回 <code>m[k]</code> ，如果找不到 <code>m[k]</code> ，则返回 <code>v</code>
<code>m.has_key(k)</code>	如果 <code>m</code> 中存在键 <code>k</code> ，则返回True，否则返回False (Python 3已废弃，使用 <code>in</code> 运算符。只能用于Python 2)
<code>m.items()</code>	返回由 (key, value) 对组成的一个序列
<code>m.keys()</code>	返回键值组成的一个序列
<code>m.pop(k [,default])</code>	如果找到 <code>m[k]</code> ，则返回 <code>m[k]</code> 并从 <code>m</code> 中删除它；否则，如果提供了 <code>default</code> 的值，则返回这个值，如果没有提供，则引发 <code>KeyError</code> 异常
<code>m.popitem()</code>	从 <code>m</code> 中删除一个随机的 (key, value) 对，并把它返回为一个元组
<code>m.setdefault(k [, v])</code>	如果找到 <code>m[k]</code> ，则返回 <code>m[k]</code> ；否则返回 <code>v</code> ，并将 <code>m[k]</code> 的值设为 <code>v</code>
<code>m.update(b)</code>	将 <code>b</code> 中的所有对象添加到 <code>m</code> 中
<code>m.values()</code>	返回 <code>m</code> 中所有值的一个序列

- ❖ 集合是唯一一项的无序集
- ❖ 与序列不同，集合不提供索引或切片操作
- ❖ 与字典不同，集合不存在相关的键值
- ❖ 放入集合中的项目必须是不可变的
- ❖ 集合有两种类型：
 - ➔ 可变集合：**set**
 - ➔ 不可变集合：**frozenset**

马哥教育

www.magedu.com



集合类型的方法和操作

项 目

描 述

<code>len(s)</code>	返回s中项目数
<code>s.copy()</code>	制作s的一份副本
<code>s.difference(t)</code>	求差集。返回所有在s中，但不在t中的项目
<code>s.intersection(t)</code>	求交集。返回所有同时在s和t中的项目
<code>s.isdisjoint(t)</code>	如果s和t没有相同项，则返回True
<code>s.issubset(t)</code>	如果s是t的一个子集，则返回True
<code>s.issuperset(t)</code>	如果s是t的一个超级，则返回True
<code>s.symmetric_difference(t)</code>	求对称差集。返回所有在s或t中，但又不同时在这两个集合中的项
<code>s.union(t)</code>	求并集。返回所有在s或t中的项

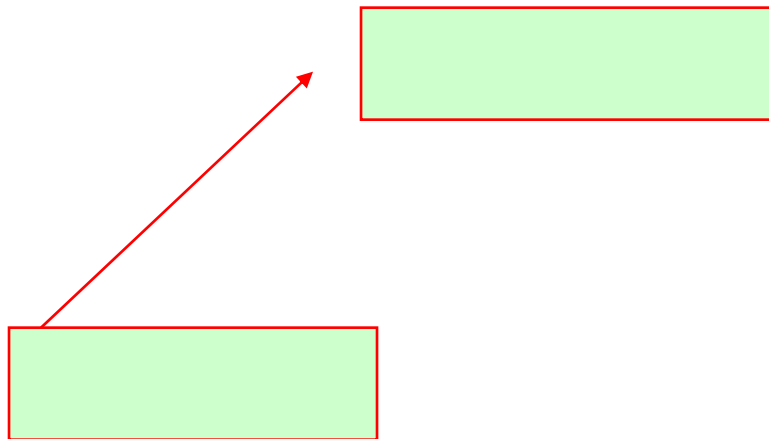
马哥教育

❖ 集合操作

操 作	描 述	操 作	描 述
<code>s t</code>	s和t的并集	<code>len(s)</code>	集合中项数
<code>s & t</code>	s和t的交集	<code>max(s)</code>	最大值
<code>s - t</code>	求差集	<code>min(s)</code>	最小值
<code>s ^ t</code>	求对称差集		



❖ l2 = l1



马哥教育
www.magedu.com



可变集合类型set的方法

项 目	描 述
<code>s.add(item)</code>	将 <code>item</code> 添加到 <code>s</code> 中。如果 <code>item</code> 已经在 <code>s</code> 中，则无任何效果
<code>s.clear()</code>	删除 <code>s</code> 中的所有项
<code>s.difference_update(t)</code>	从 <code>s</code> 中删除同时也在 <code>t</code> 中的所有项
<code>s.discard(item)</code>	从 <code>s</code> 中删除 <code>item</code> 。如果 <code>item</code> 不是 <code>s</code> 的成员，则无任何效果
<code>s.intersection_update(t)</code>	计算 <code>s</code> 与 <code>t</code> 的交集，并将结果放入 <code>s</code>
<code>s.pop()</code>	返回一个任意的集合元素，并将其从 <code>s</code> 中删除
<code>s.remove(item)</code>	从 <code>s</code> 中删除 <code>item</code> 。如果 <code>item</code> 不是 <code>s</code> 的成员，则引发 <code>KeyError</code> 异常
<code>s.symmetric_difference_update(t)</code>	计算 <code>s</code> 与 <code>t</code> 的对称差集，并将结果放入 <code>s</code>
<code>s.update(t)</code>	将 <code>t</code> 中的所有项添加到 <code>s</code> 中。 <code>t</code> 可以是另一个集合、一个序列或者支持迭代的任意对象

马哥教育

www.magedu.com



马哥教育
www.magedu.com



马哥教育

www.magedu.com



引用计数与垃圾收集

❖ 所有对象都有引用计数

- ➡ 给对象分配一个新名称或将其放入一个容器中，其引用计数都会增加
- ➡ 使用`del`语句或为变量名重新赋值时，对象的引用计数会减少
- ➡ `sys.getrefcount()`可以获得对象的当前引用计数

❖ 一个对象的引用计数器归零时，它将被垃圾收集机制回收

马哥教育

www.magedu.com



- ❖ 博客: <http://magedu.blog.51cto.com>
- ❖ 主页: <http://www.magedu.com>
- ❖ QQ: 2813150558, 1661815153, 113228115
- ❖ QQ群: 203585050, 279599283



马哥教育

Thank You!