

数据类型概述

1

数据类型概述

—— 数字

基本数字类型

- int : 有符号整数
- bool : 布尔值
 - True : 1
 - False : 0
- float : 浮点数
- complex : 复数

数字表示方式

- python默认以十进制数显示
- 数字以0o或0O开头表示为8进制数
- 数字以0x或0X开头表示16进制数
- 数字以0b或0B开头表示2进制数

2

数据类型概述

—— 序列对象

定义字符串

- python中字符串被定义为引号之间的字符集合
- python支持使用成对的单引号或双引号
- 无论单引号，还是双引号，表示的意义相同
- python还支持三引号（三个连续的单引号或者双引号），可以用来包含特殊字符
- python不区分字符和字符串

字符串切片

- 使用索引运算符[]和切片运算符[:]可得到子字符串
- 第一个字符的索引是0，最后一个字符的索引是-1
- 子字符串包含切片中的起始下标，但不包含结束下标

```
>>> py_str = 'python'
>>> py_str[0]
'p'
>>> py_str[-2]
'o'
>>> py_str[2:4]
'th'
>>> py_str[2:]
'thon'
>>> py_str[:4]
'Pyth'
```

字符串连接操作

- 使用+号可以将多个字符串拼接在一起
- 使用*号可以将一个字符串重复多次

```
>>> py_str = 'python'
>>> is_cool = 'is Cool'
>>> print py_str + ' ' + is_cool
python is Cool
>>> py_str * 2
'pythonpython'
```


定义列表

- 可以将列表当成普通的“数组”，它能保存任意数量任意类型的python对象
- 像字符串一样，列表也支持下标和切片操作
- 列表中的项目可以改变

```
>>> alist = [1, "tom", 2, "alice"]  
>>> alist[1] = 'bob'  
>>> alist[2:]
```

列表操作

- 使用in或not in判断成员关系
- 使用append方法向列表中追加元素

```
>>> alist = [1, "tom", 2, "alice"]
>>> 'tom' in alist
True
>>> 'alice' not in alist
False
>>> alist.append(3)
>>> alist[5] = 'bob'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list assignment index out of range
```

元组的定义及操作

- 可以认为元组是“静态”的列表
- 元组一旦定义，不能改变

```
>>> atuple = (1, "tom", 2, "alice")
>>> 'tom' in atuple
True
>>> atuple[0] = 3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

3

数据类型概述

—— 字典

字典的定义及操作

- 字典是由键-值(key-value)对构成的映射数据类型
- 通过键取值，不支持下标操作

```
>>> user_dict = {'name':'bob', 'age':23}
>>> use_dict['gender'] = 'male'
>>> 'bob' in user_dict
False
>>> 'name' in user_dict
True
>>> user_dict[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 0
```

数据类型比较

- 按存储模型分类
 - 标量类型：数值、字符串
 - 容器类型：列表、元组、字典
- 按更新模型分类：
 - 可变类型：列表、字典
 - 不可变类型：数字、字符串、元组
- 按访问模型分类
 - 直接访问：数字
 - 顺序访问：字符串、列表、元组
 - 映射访问：字典



更多精彩...



<http://bj.linux.tedu.cn/>
企业QQ：86198501