【白话python连载（4.1）】python的数据类型--迈步万里长城（一）

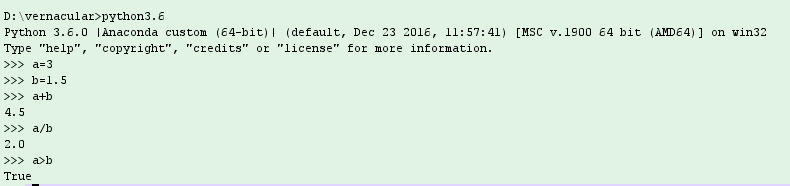
从之前的介绍我们知道，python作为计算机程序之一，可以用来处理数学计算、文本、图像、视频、网页内容等等。不同内容背后对应的数据类型是不同的，弄明白不同的数据类型及其特点，才能迈出最重要的一步，完成更好的程序组织。python的数据类型内容稍微有点多，所以分两小节进行介绍，本节主要介绍基本的数据类型和序列的数据类型。下面直接进入主题，来依次了解python中的不同数据类型。

* 基本的数据类型介绍

python中最基本的数据类型（Number）包括：整型、布尔型、浮点型、复数等。

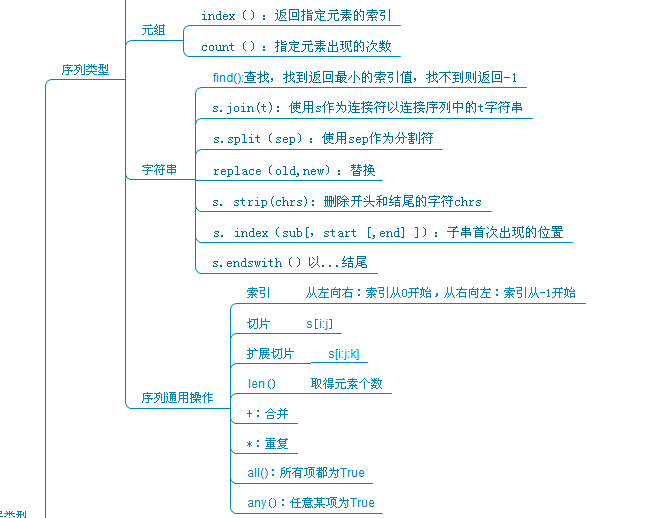
|  |  |  |
| --- | --- | --- |
| 基本的数据类型 | 特点 | 实例 |
| 整型（int） | python3中只有长整型一种（基本数值） | a=3，b=-2，c=0 |
| 布尔型（bool） | 一般用于判断中，可以返回表达式计算的正误、数据结构是否为空、字符是否存在等。 | None→False，a>c→True  0，{},[],(),以及None均为False，反之则为True。 |
| 浮点型（float） | 带小数点的数值 | e=32.3e+18,f=-3.0 |
| 复数(complex) | 负数中的实部和虚部都是浮点数 | 0.2 - 3j,或者complex(2,-3) |

python中使用变量不需要提前指明数据类型，在赋值时将自动确定变量的类型。使用range函数可以生成常用的整数序列。python中有一个特殊的类型，空值，被称作 Null 对象或者 NoneType，它只有一个值，那就是 None。它不支持任何运算也没有任何内建方法。None和任何其他的数据类型比较永远返回False。









* 组合的数据类型介绍

在python中有多重组合的数据类型，包括序列类型、映射类型和集合类型三大类。其中python包含6中内建的序列类型，即列表、元组、字符串、Unicode字符串、buffer对象和xrange对象。序列中的每个元素拥有一个序号即元素的位置——索引。

• 序列的数据类型介绍

最常用的序列类型是列表、元组和字符串。它们拥有通用的序列操作方法：索引、切片、扩展切片、长度、序列相加、乘法、判断、成员

列表→[list]

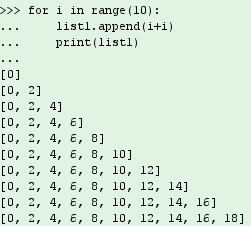
**创建列表**

list作为python的内置数据类型，通过方括号[]进行包裹，可以基于已有列表生成新的列表，主要借助列表定义、列表操作方法，或列表生成式创建新列表。除此也可以使用列表生成器，但是列表生成器并不是直接创建一个列表，而是通过迭代器不断的返回列表中的数据。

* range（）函数可以生成一组有规律的数组，然后再用数据类型转换方法list（range（））将生成的数组转变为列表类型。

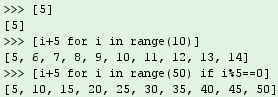


通过循环依次获取需要的数据，并且在已经申明好的列表中完成添加而形成新的列表。



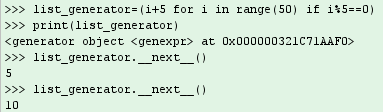
* 为了方便操作，可以使用列表生成式

基本格式是[ 元素表达式 依次循环原有列表 判断条件]，列表也可以直接指定值初始化生成，其中循环原有列表和判断条件可以按照实际需求进行指定。



* 为了减少列表所占用的内容，可以通过列表的生成器创建动态列表。并且调用生成器的next方法依次获取生成器中的每个值。

基本格式是（ 元素表达式 依次循环原有列表 判断条件），类似于列表生成式，但是要用圆括号包裹相应内容。

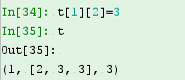


**列表的常用方法如下**：

|  |  |
| --- | --- |
| 方法与说明 | 实例 |
| .append(obj)  在列表的末尾进行元素追加 |  |
| .clear()  清空所有值 |  |
| .copy()  复制列表中的所有值 |  |
| .count(value)  统计元素的个数 |  |
| .extent(iterable)  在列表之后添加另一个列表的值 |  |
| .index(value，)  找出某个值第一个匹配的位置 |  |
| .insert(index,object)  在列表指定位置插入元素 |  |
| .pop(object)，  移除列表中指定位置的元素，  并且返回该元素的值 |  |
| .remove(value)  移除列表中该值的第一个匹配项 |  |
| .reverse()  反转列表中的所有元素 |  |
| .sort(key=None，reverse=False)  对原列表按照一定规则进行排序  （Unicode字符） |  |

元组→（tuple）

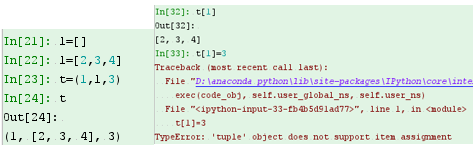
元组的特点与列表很像，是通过圆括号()包裹表示。**创建元组的**方式很简单，只需要在括号括号中添加元素并且依次用逗号隔开即可。而且为了避免歧义对于单元素的元组在创建时，也需要在元素后面添加逗号。



|  |  |
| --- | --- |
| 方法 | 说明 |
| .count(value) | 统计元组中某个元素的个数 |
| .index(value,) | 返回某个元素第一个匹配的位置 |

但是元组初始化之后对应元素索引不可改变。因此元组没有添加删除扩展等操作方法。只能进行元素计数和查找。

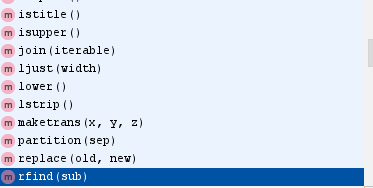
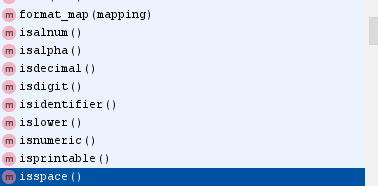
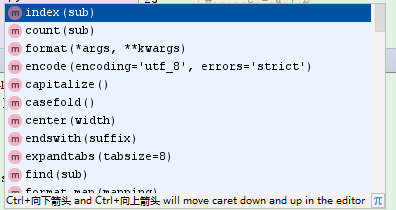
而且要注意的是如果元组索引下的元素类型是可变的，则元素内容是可变的，否则元素内容是不可变的。以嵌套列表的元组为例，初始化一个元组t后，其本身索引所对应的内容不可修改，但是索引中l列表中的内容可修改。最终元组的值发生改变，是因为列表的特点导致的。

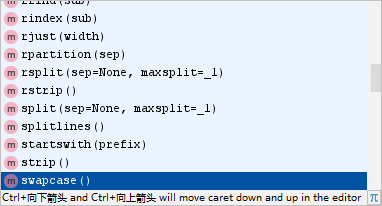
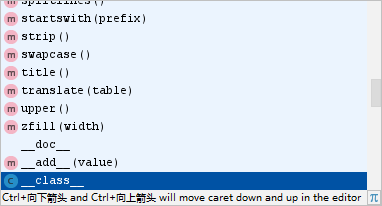


因为元组的不可更改的特性，可以很好地保证代码数据的安全性，但是也会因此降低代码的灵活性。因此在实际编程操作过程中，应当按照任务要求选择列表或者元组类型。

字符串

字符串是python中最常用的序列数据类型。直接通过单引号或双引号包裹就可以生成字符串。字符串有40多种内建函数，这里介绍最常用的几种，剩下的大家可自行测试。



|  |  |
| --- | --- |
| .count(str, beg, end)  返回 str 在 string 里面出现的次数，如果 beg 或者 end 指定，则返回指定范围内 str 出现的次数。**（直接指定索引号即可）** |  |
| .endswith(suffix, beg, end)  检查字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False. |  |
| .encode(encoding=’utf-8’,errors=’strict’)  以 encoding 指定的编码格式编码字符串，如果出错默认报一个ValueError 的异常，除非 errors 指定的是'ignore'或者'replace' | 在连载2中关于编码的问题有详细介绍  python默认Unicode编码运行，但是保存方式需指定为需要的编码方式（默认utf-8为python的存储方式） |
| .find(str, beg, end)  检测 str 是否包含在字符串中，如果指定范围 beg 和 end ，则检查是否包含在指定范围内，如果包含返回开始的索引值，否则返回-1 |  |
| .index(str, beg, end)  跟find()方法一样，只不过如果str不在字符串中会报一个异常. |  |
| .islower()  如果字符串中包含至少一个区分大小写的字符，并且所有这些字符都是小写，则返回 True，否则返回 False |  |
| .join(iterable：str)  以指定字符串作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串 | 字符串本身是序列数据类型，可迭代 |
| .replace(old, new [, max]) （从头开始匹配）  将字符串中的 str1 替换成 str2,如果 max 指定，则替换不超过 max 次。 |  |
| .split(str=None,maxsplit=\_1)  默认以空格为分隔符截取字符串，如果第二个参数有指定值，则仅截取指定个子字符串 |  |
| . strip()【lstrip(),rstrip()】  删除字符串字符串开头和末尾的空格 |  |

了解python中字符串的常规方法和编码问题后，可以在使用过程中，更灵活的处理文本内容，进行字符串格式化规范化的存储。

• 序列数据类型的通用操作

接下来我们来看这三种常用的序列数据类型的通用操作。



|  |  |
| --- | --- |
| 通用操作（初始化） |  |
| 索引：从左向右：索引从0开始，  从右向左：索引从-1开始  超过索引值后返回结果会报错  通过索引获取字符串中字符 |  |
| 切片：s[i:j]  截取部分内容，j最大默认字符串长度 |  |
| 扩展切片：s[i:j:k]  指定截取内容的方式，每隔几个截取 |  |
| len():取得元素的个数 |  |
| +：合并数据（只用于字符串之间的操作） |  |
| \*：重复合并（只用于字符串之间的操作） |  |
| all()：所有项都为False，则返回Ture  any()：任何一项为False，则返回Ture |  |
| in&not in：成员判断 |  |

• 总结

本次主要介绍了：

python中基本的四种数据类型：整型、布尔型、浮点型和复数（重点要注意布尔型为false的一些值：None，0，[],()）

python中的序列数据类型：列表，元组和字符串。（重点要了解每种数据类型的特点和一些常规方法，而且在使用字符串时了解编码的相关问题(可去第二节查看）

最后重点介绍了序列数据类型的通用方法，方便进行数据的查询、判断和处理等。