

Python内置数据类型总结（字符串）

字符串是Python中最常用的数据类型之一，使用单引号、双引号、三引号来括起来表示，三引号表示多行字符串，平常使用单引号或双引号就行。如有单、双引号嵌套时，可使用反斜杠【\】进行转义，或者使用不是嵌套中的引号，例如 `var = "This is 'dog'!"`。

字符串是不可变的序列数据类型，不能修改字符串本身，和数字类型一样！Python完全支持Unicode编码，所有的字符串都是Unicode字符串。虽然字符串本身不可变，但是可以像列表一样进行切片和截取子串操作，但不会引起字符串本身的变化。

1.字符串运算：

下表中a为"Xiao",b为"Shouhua"

操作	描述	例子	结果
+	字符串拼接（速度慢，少用）	<code>a + b</code>	"XiaoShouhua"
*	重复字符串，相当于乘法	<code>a * 2</code>	"XiaoXiao"
[]	通过索引获取字符串中的字符	<code>a[2]</code>	"a"
[:]	截取、切片	<code>a[2:]</code>	"ao"
in	成员运算符，如果字符串中包括给定的字符串，返回True	<code>"X" in a</code>	True
not in	同上，不包含返回True	<code>"X" not in a</code>	False
r/R	原始字符串，不会转义特殊字符	<code>print(r"\n")</code>	\n

2.转义字符串：

转义字符串都有特殊含义，见下表：

转义字符	描述
(在行尾时)	续行符
\	反斜杠
\'	单引号
\"	双引号
\a	响铃
\b	退格
\e	转义

转义字符	描述
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\o33	颜色控制

3、字符串内置方法

下表从常用的字符串内置方法开始梳理，常用的都要记住如何使用，不常用的要有印象，以*的个数表示常用的程度。

方法	描述	使用频率
<code>string.split(sep, maxsplit=-1)</code>	字符串分割，返回对象为分割后的子串列表。默认使用空格分割，可指定分隔符，分隔符不包含在分割后的子串中；默认贪婪分割，可指定分割次数。	*
<code>string.splitlines([keepends])</code>	按照('\r','r\n','\n')分割字符串，返回一个包含各行作为元素的列表，如果参数keepends为False，不包含换行符，否则保留换行符	***
<code>string.strip([chars])</code>	去除两边空格: 默认返回去除字符串两边空格的字符串，其中有 <code>string.lstrip()</code> 和 <code>string.rstrip()</code> 是分别去除左、右各边的空格。可指定去除的字符	*
<code>string.find(sub[, start[, end]])</code>	返回查找子串出现的第一个位置索引，可以指定查找范围，没有找到子串返回-1。其中 <code>string.rfind()</code> 是从右边开始查找	*
<code>string.index(sub[, start[, end]])</code>	返回查找子串出现的一个位置索引，可指定查找范围，没有找到会异常ValueError，其中 <code>string.rindex()</code> 是从右边开始检索	
<code>string.count(sub[, start[, end]])</code>	返回查找子串在字符串中出现的次数，可指定查找范围，没有找到，返回0	
<code>string.lower()</code>	返回一个全部为小写的字符串	
<code>string.upper()</code>	返回一个全部为大写的字符串	
<code>string.startswith(prefix[, start[, end]])</code>	返回在给定范围中是否以指定字符串开头，是返回True，否则返回False	

方法	描述	使用频率
string.endswith(prefix[, start[, end]])	返回在给定范围中是否以指定字符串结尾，是返回True，否则返回False	
string.replace(old, new[, count])	返回使用字符串new替换字符串old count次的新的字符串	
string.encode(encoding='utf-8', errors='strict')	返回以encoding指定的编码格式编码的bytes对象，如果出错会报一个UnicodeEncodeError异常，除非指定errors是'ignore'或者'replace'。	
bytes.decode(encoding='utf-8', errors='strict')	和string.encode是逆向过程，将bytes以指定编码格式解码为string，如果出错会报一个UnicodeEncodeError异常，除非指定errors是'ignore'或者'replace'。	
string.format()	格式化字符串,常用方式为位置参数和关键字参数	*
string.join(seq)	以指定的string为分隔符，将序列seq中的元素（其中的元素必须是以字符串类型的形式才可以）合并为一个新的字符串	***
string.center(width)	返回一个原字符串居中，并默认使用空格填充至长度width的新字符串，可指定填充字符串	
string.ljust(width)	返回一个原字符串左对齐，并默认使用空格在右侧填充至长度width的新字符串，可指定填充字符串	***
string.rjust(width)	返回一个原字符串右对齐，并默认使用空格在左侧填充至长度width的新字符串，可指定填充字符串	**
string.zfill(width)	返回长度为width的字符串，原字符串右对齐，前面填充0	**
string.expandtabs(tabsize=8)	把字符串string中的tab符号转化为空格，tab默认空格数是8	**
string.capitalize()	返回首字母大写的字符串	*
string.isalnum()	如果 string 至少有一个字符并且所有字符都是字母或数字则返回 True,否则返回 False	***
string.isalpha()	如果string中至少出现一个字符且所有字符都是字母的则返回True，否则返回 False	***
string.isdigit()	如果 string 只包含数字则返回 True 否则返回 False.	***

方法	描述	使用频率
string.isdecimal()	如果 string 只包含十进制数字则返回 True 否则返回 False.	***
string.isnumeric()	如果 string 中只包含数字字符, 则返回 True, 否则返回 False	**
string.islower()	只要string中不包含大写字符并且包含至少一个小写字符就会返回True, 否则返回False	***
string.isupper()	只要string中不包含小写字符并且包含至少一个大写字符就会返回True, 否则返回False	***
string.istitle()	如果string是标题式的字符串, 返回 True, 否则返回False, 标题式是: 字符串中的单词首字母大写, 例如"Title 10Dd"	***
string.isspace()	如果string中只包含空格类型, 就返回 True, 否则返回False. 空格类型: '\n\t\v\r\f'	**
string.title(width)	返回'标题化'的string, 即单词的首字母全部大写, 其余字母小写。	**
string.swapcase()	翻转大小写/交换大小写	*
str.maketrans(intab, outtab)	接受两个长度相同的字符串, 第一个字符串是需要转换的字符串, 第二个字符串是转化的目标字符串, 用于创建字符串映射的转化表	**
string.translatte	使用str.maketrans()方法转化的转化表进行字符串的转化	**
string.partition(str)	按照指定字符分割字符串/从左边找str出现的第一个位置起, 把字符串string分成一个3元素的元祖(string_pre_str, str, string_post_str), 如果string中不包含str, 则string_pre_str == string	
string.rpartition(str)	从右边找str出现的第一个位置起, 把字符串string分成一个3元素的元祖(string_pre_str, str, string_post_str), 如果string中不包含str, 则string_pre_str == string	

4.编码问题

计算机只能处理二进制数01, 所以产生了字符编码, 用来将我们人类可读的数据类型转化为二进制。常用编码包括ASCII码、Unicode编码、utf-8编码、GB2312、gbk等。

但在python3中运行时全部使用Unicode编码不会产生任何编码问题，对于编码，要记住三条规则：

操作系统运行时，在内存中统一使用的Unicode编码，当需要将数据保存到硬盘或者网络传输时，就转化为utf-8编码，进行保存和传输；

用文本编辑器的时候，从文件系统或者说硬盘上读取的utf-8编码字符被转换为Unicode编码字符到内存中，程序或操作系统才能使用。编辑完成后，保存时再转换为uft-8进行保存；

浏览网页时，服务器会把Unicode内容转化为utf-8进行传输给客户端浏览器。

```
class Person(object):

    def init(self, name, gender, **kw):

        self.name = name

        self.gender = gender

    for k,v in kw.items():

        setattr(self, k, v) __
```