6 字符串

一、字符串

字符串就是由若干个不同的unicode字符组成的不可变序列

1 .字符串创建

```
#单引号字符串
str1 = '天要下雨，娘要嫁人，由他去吧'
str2 = str()  #空串
str3 = str([10,20,30])

#双引号字符串
str2 = "天要下雨，娘要嫁人，由他去吧"

#三引号字符串 多行字符串
str3 = '''
好雨知时节
当春乃发生
'''
str4 = """
随风潜入夜
润物细无声
"""
```


2.获取字符

因为是不可变序列，所以不能修改单个字符；**凡是修改字符串的操作都产生新串**

```
str1 = '天要下雨，娘要嫁人，由他去吧'
print(str1[0],str1[-1]) #天 吧
```

3.字符串转义

- 常见转义字符

转移字符	说明	转移字符	说明
`\'`	单引号	`\"`	双引号
\n	换行	\r	回车
\t	tab	\\	\

- 原生字符

  如果在字符串中不把\当做转移字符，可以使用

```
str1 = r'c:\wh1803\course\1'
print(str1)  #c:\wh1803\course\1
```

4.序列通用操作

```
#1 字符串拼接
#对于字符串字面值，只要相邻两个字符串中间没有其他字符就会自动拼接为一个字符串
str1 = '中美达成共识'
        '不打贸易战'
print(str1) #中美达成共识不打贸易战

#其他情况使用+运算符拼接字符串
str1 = '你好'
str2 = ' 树先生'
print(str1 + '世界')
print(str1 + str2)

#2.字符串重复
str1 = '汪' * 3
print(str1) #汪汪汪

#3.成员操作
str1 = '天要下雨，娘要嫁人，由他去吧'
if '天' in str1:
    print('是成员')
else:
    print('不是成员')

#4.字符串截取（切片）
```

```
str1 = '123456'
print(str1[0:2]) #'12'
print(str1[1:]) #'23456'
print(str1[::2]) #'135'
print(str1[:]) #'123456'
print(str1[::-1]) #'654321'

#5.字符串长度
print(len(str1))
```

4 字符串常用函数

4.1 字符串查找和替换

str1 = 'a fox jumped over the fence'

方法名	说明	示例
str.count(sub,start=0,end=len(string))	查找子串sub出现的次数；start从指定下标开始查，end结束下标	str1.count('f')
str.**find**(str, beg=0, end=len(string))	从左向右检测字符串中是否包含子字符串 str,如果包含返回下标，否则返回-1。beg和end是可选参数，指定查找范围	str1.find('fox')
str.index(sub[, start[, end]])	作用类似find，但子串sub不存在会报错ValueError	str1.index('fox')
str.**replace**(old, new[, count])	返回一个新字符串，原串中的old被替换为new，可选参数count指定替换次数。	str1.replace('a','many')

4.2 字符串分隔和组合

方法名	说明	示例
str.**split**([sep[, num=count(sep)]])	将字符串拆分为以sep为分隔符的列表，如果指定num,则最多拆分num次	str1.split(' ')
str.partition(seq)	将字符串拆分为一个有三个元素的元组（seq前的字符串，seq，seq后的字符串）。	
str.**splitlines**([keepends])	拆分一个包含多行的字符串，以每行为一个元素返回一个列表。keepends是一个True字符或非零整数，表示保留行尾标志（即换行符）	
str.**join**(seq)	以指定字符串str作为分隔符，将seq对象中所有的元素(字符串表示)合并为一个新的字符串;seq可以是字符串、列表等	

4.3 字符串判断

方法	说明	示例
str.**isalpha**()	判断字符串是否由字母构成并且只包含字母，是返回True，否返回False	str1.isalpha()
str.isalnum()	检测字符串是否由字母和数字组成，如果都是数字或字母返回True，否则返回False	
str.**isdigit**()	检测字符串是否由数字构成，可检测byte类型	
str.**isdecimal**()	检测字符串是否由数字构成	
str.**isnumeric**()	检测字符串是否由数字构成, 可以检测汉字数字：十	
str.isspace()	检测字符串是否只有空格或tab构成	
str.**islower**()	检测字符串中的字母字符是否全部由小写字母组成	
str.**isupper**()	检测字符串中的字母字符是否全部由大写写字母组成	
str.startswith(suffix[, start[, end]])	用于判断字符串是否以指定子字符串开头，如果是则返回True，否则返回False。	
str.endswith(suffix[, start[, end]])	用于判断字符串是否以指定子字符串 结尾，如果是则返回True，否则返回False。	

4. 4字符串转换

方法	说明	示例
str.**lower**()	字符串转小写	
str.**upper**()	字符串转大写	
str.swapcase()	把字符串中的大小写字母互换，大写转换成小写，小写转换成大写。不去管非字母类字符。	
str.capitalize()	将字符串的第一个字符转换为大写， 其余转换为小写	
str.title()	字符串中每个单词的首字母大写，其余小写。	
str.lstrip([chars])	去除字符串左边指定的字符，默认是去除空格	
str.rstrip([chars])	去除字符串左边指定的字符，默认是去除空格	
str.**strip**([chars])	去除字符串两边边指定的字符，默认是去除空格	

4.5 其它方法

```
#1. 将其他类型转换为字符串
print(str(90)) #'90'
print(str([20,30])) #'[20,30]'

#2 ord(x) 返回一个字符所对应的码值
print(ord('a')) #97
print(ord('中')) #20013

#3 chr(x) 输入一个unicode码，返回一个对应的字符。
print(chr(20013))  #中
```

5. 字符串格式化

- 用%格式化

```
%[flags][width][.precision]typecode
flags:对其方式，-左对齐 +右对齐(默认),0表示用0填充(只针对数值型)，默认是用空格填充
width：所占宽度，单位是列
.precision：精度，如果带小数点，可以指定带几位小数，指定后会四舍五入
typecode：d 将数值转换为整型显示；f 将数值转换为浮点数显示 s将数值转换为字符串显示
#大家好，我叫    王尼玛，我今年35岁了,我有5000000.69
print("大家好，我叫%+6s，我今年%d岁了,我有%10.2f" % ('王尼玛',35,5000000.687))
```

- 用format格式化

:	<填充>	<对齐>	<宽度>	,	<.精度>	<类别>
	用于填充的单个字符	< 左对齐 > 右对齐 ^ 居中对齐	槽的设定输出宽度	数字的千位分隔符 适用于整数和浮点数	浮点数小数部分的精度或字符串的最大输出长度	整数类型 B,c,d,o,x,X 浮点数类型 e,E,f,%

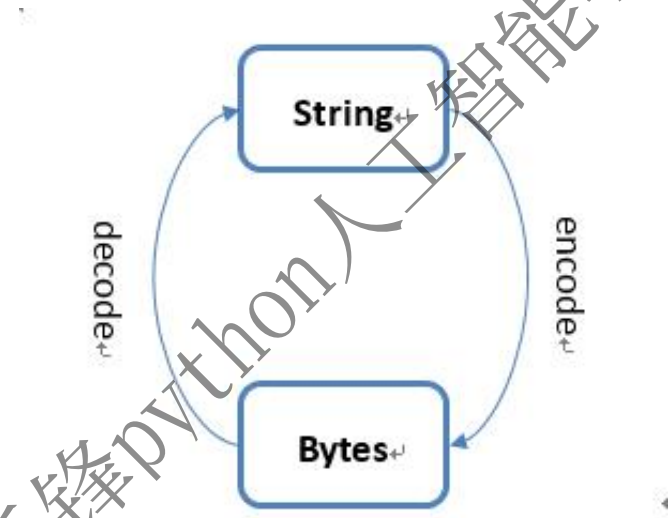
```
[[fill]align][sign][#][width][.precision][type]
fill：填充字符，可选
align：对齐方式  <左对齐  >右对齐  ^居中对齐
sign：显示符号，+正数显示正号，负数显示符号；-正数不显示符号，负数显示符号
#：  对于2、8、16进制会显示0b 0o 0x
width：宽度
, 千分位分隔符
.precision：精度
type：  s字符串  d整型  f浮点数

tp1 = "I am {}, age {}, {}".format("seven", 18, 'alex')
tp2 = "I am {name}, age {age}, really {name}".format(name="seven", age=18)
tp3 = "I am {:s}, age {:d}, money {:.0f}".format("seven", 18, 88888.1)
print(tp1) #I am seven, age 18, alex
print(tp2) #I am seven, age 18, really seven
print(tp3) #I am seven, age 18, money 88888
```


二、字节

在python3中最重要的特性是对文本和二进制数据做了更加清晰的区分，文本总是Unicode，由字符类型表示，而二进制数据则由byte类型表示，python3不会以任意隐式方式混用字节型和字符型，也因此在python3中不能拼接字符串和字节包（python2中可以，会自动进行转换），也不能在字节包中搜索字符串，也不能将字符串传入参数为字节包的函数。

Bytes 对象是由单个字节作为基本元素（8位，取值范围 0-255）组成的序列，为不可变对象。 bytes对象只负责以二进制字节序列的形式记录所需记录的对象，至于该对象到底表示什么（比如到底是什么字符）则由相应的编码格式解码所决定。Python3中，bytes通常用于网络数据传输、二进制图片和文件的保存等等。可以通过调用bytes()生成bytes实例，其值形式为 b'xxxxx'，其中 'xxxxx' 为一至多个转义的十六进制字符串（单个 x 的形式为：\x12，其中\x为小写的十六进制转义字符，12为二位十六进制数）组成的序列，每个十六进制数代表一个字节（八位二进制数，取值范围0-255），对于同一个字符串如果采用不同的编码方式生成bytes对象，就会形成不同的值.



1 创建字节

```
#创建字节
b1 = b'hello'
b2 = b"ello"
b3 = b'''hello'''
b4 = bytes('中文','utf-8')
```

2 字符串和字节的转换

```
#字符串转字节
s1 = "中文"
s2 = s1.encode('utf-8')  #str.encode()
print(type(s2))  #<class 'bytes'>
print(s2)  #b'\xe4\xb8\xad\xe6\x96\x87'

#字节转字符串
s3 = s2.decode('utf-8')  #bytes.decode()
print(type(s3))  #<class 'str'>
print(s3)  #中文
```

三、列表推导式

1.随机数

import random

函数名	函数说明
choice(seq)	返回一个序列（列表、元组，字符串）中返回一个随机元素
randrange(start,end,step)	start 指定范围的起始值 包含本身 end 指定范围的结束值 不包含本身 step 步长
randint()	返回一个随机整数
shuffle(seq)	将序列元素随机排列（打乱顺序）

2.列表推导式

运用列表推导式，可以快速生成list，可以通过一个list推导出另一个list，而代码却十分简洁。

```
#列表推导式语法：
[exp for iter_var in iterable]
执行for-in循环时，通过iter_var遍历iterable每一项，exp表达式中可以直接使用iter_var，每遍历一项，产生一个新的列表元素。
#生成[0,1,4,9,16,25]
```

```
[x*x for x in range(6)]

#生成[0,4,16,36,64]
l2 = [x*x for x in range(9) if x % 2 ==0]
print(l2)

#可以使用双重循环
suit = ['♥','♦','♣','♠']
face = ['A','2','3','4','5','6','7','8','9','10','J','Q','K']
poke = [[x,y] for x in suit for y in face]
```