

字符串

字符串的驻留机制

```
a='python'
b="python"
c='''python'''
print(a,id(a))
print(b,id(b))
print(c,id(c))
# id相同，驻留机制：仅保存一份相同且不可变字符串，在创建相同字符串时，不开辟新空间，而是把字符串的地址赋给新创建的变量。
```

驻留机制的几种情况

1. 字符串长度为0或者1
2. 符合标识符的字符串（字母数字下划线）
3. 字符串只在编译时驻留
4. -5~256之间的数字

sys中利用intern方法可以强制使得两个字符串指向同一个对象 pyCharm对字符串进行了优化处理，内容相同的就进行了驻留。

作用：在需要字符串拼接时，用join方法，

字符串的常用操作

字符串的查询操作

查询方法：

1. index 查找子串substr第一次出现的**位置**，查找不存在抛出valueerror
2. rindex 查找子串substr 最后一次出现的**位置**
3. find 查找子串第一次出现的**位置**，找不到返回-1
4. rfind ()： 查找子串最后一次出现的**位置** 查找不存在，返回-1

大小写转换

大小写转换

1. upper 所有字符都转换成大写
2. lower 所有字符都转换成小写
3. swapcase： 大写变小写，小写变大写
4. capitalize 第一个**字符**转换成大写，其余**字符**转换成小写
5. title **单词**的第一个字符转换成大写，每个**单词**的剩余字符转换成小写

转换之后，虽然有时候转换前后内容完全相同，产生一个新的字符串对象，id就不一样了

字符串对齐操作

1. center() 居中对齐，第一个参数指定宽度，第二个参数指定填充字符，第二个参数是可选的，默认为空格，设置宽度小于实际宽度则还原字符串
2. ljust() 左对齐 第一个参数指定宽度，第二个参数指定填充字符，第二个参数是可选的，默认为空格，设置宽度小于实际宽度则还原字符串
3. rjust 第一个参数指定宽度，第二个参数指定填充字符，第二个参数是可选的，默认为空格，设置宽度小于实际宽度则还原字符串
4. zfill 右对齐，左边用0填充，只接受一个参数，用来指定字符串宽度，指定宽度小于字符串，返回字符串本身。

字符串的劈分操作

1. split
 1. 从左边开始劈分，默认劈分字符是空格字符串，返回的值是一个列表
 2. 参数sep指定劈分字符串
 3. 参数 maxsplit指定劈分字符串的最大劈分次数，
2. rsplit

判断字符串的方法

1. 判断是否为合法的字符串（字母数字&下划线）：isidentifier
2. 是否全部由空白字符组成（回车换行水平制表）：isspace
3. 是否全部由字母组成 isalpha
4. 是否全由十进制的数字组成 isdecimal
5. 判断是否全由数字组成 isnumeric
- 6.

print('123四'.isnumeric()) #结果为true

7. 判断是否全部由数字和字母组成 isalnum

字符串的其他方法

替换操作：replace () 三个参数：第一个参数指定谁被换，第二个参数指定用谁换，第三个参数指定替换次数。 返回新的字符串，原字符串不变

合并操作：join 将列表或元组中的字符串合并成一个字符串

字符串的比较操作

运算符： > >=, < ,<=,==,!=

两字符比较的是其原始值

```python

print(ord('a'),ord('b')) #ord 函数可以得到指定字符的ordinal value。 与之对应的是内置函数 chr(),指定ordinary value可以获得对应的字符。

输出：

```
97 98
print(chr(97),chr(98))
输出:
a b

== 和is的区别 is是内存地址的比较
```

## 字符串的切片操作

不可变，不具备增删改操作 切片产生新的对象

```
a='pythonpyThon'
b='pythonniubi'

s=a[:2]
s2=a[3:]
s3='!'
news=s+s3+s2
#完整切片写法
a[1:2:3] #从1开始，到2结束，不到2，步长为3
a[::2] #默认从0开始 最后结束 步长为2
a[::-1] #逆序切片
```

## 格式化字符串

%s 字符串 %d 整数 %()

```
#法1 %占位符
print('我叫%s,我今年%d' % (name,age))
#法2 #{}
print('我叫{0},今年{1}岁'.format(name,age))
#法3 f-string
print(f'我叫{name}, 今年{age}岁了')

#10表示宽度
print('%10d' %99)
.3表示的是小数点后三位
print('%0.3f' %3.13342232)
#两种方式结合起来
print('%10.3f' %3.322222)

#使用{}也可以
print('{:10.3}'.format(3.32323232)) # :10.3表示宽度是10 一共是三个数
```

## 字符串的编码转换

为什么：str在计算机内以unicode表示，要编码之后以字节传输到其他主机 编码：字符串转换成二进制数据 解码：bytes数据转换为字符串类型

```
s='天涯共此时'
print(s.encode(encoding='GBK')) # GBK中一个中文占两个字节
print(s.encode(encoding='UTF-8')) #UTF-8一个中午占3个字节

#byte代表就是编码后的数据
byte=s.encode(encoding='GBK')
print(byte.decode(encoding='GBK'))
```