字符串（str）

```
字符：一个字符 a  b
字符串：
    一个字符串对象
Python：'a',"a","""a"""
Java：
    字符：'a'
    字符串："abc" 'abc'--错误的
1. 天生具有跨平台特性---操作系统（os）
2. 不可变类型
3. 字符串是可迭代对象
        s='abcdef'
        for  i in s:
            print(i)
 4. 字符串支持索引和分片操作
 5. 支持的操作符
        拼接：+
        重复：*
        比较（布尔）：>  < ==  <=  >= !=
        逻辑：not and or
        成员关系：  in  not in
```

字符串的创建

```
1. s='abc'
    s="abc"   s="""abc""" s='''a'''
        三引号：可以换行 ，表示注释
2. str()创建

二进制：bytes
    python2
        str---Unicode---bytes
    python3
        str--str--bytes
```

- 字符串的方法

```
1. str(object='')
        创建一个字符串对象
2. capitalize()
        把字符串的第一个字符改为大写，后面的小写
3. casefold（）
        把整个字符串小写
4. center（width,[字符]）
        居中
        width:宽度
```

[字符]：默认是空格

5. encode()
   编码 str--bytes
6. decode()
   解码 bytes--str
7. endswith (suffix,[start],[end])
   判断字符串是否是一suffix
   [end]：结束下标---取不到

   >>> s4.endswith('b',0,1)   #下标
   False
   >>> s4.endswith('b',0,2)
   True
8. count(sub,[start],[end])
   返回sub字符出现的次数
9. find(sub,[start],[end])
   返回sub这个字符第一次出现的下标,如果查不到返回-1
10. index(sub,[start],[end])
    返回sub这个字符第一次出现的下标,如果查不到--报错
11. isalnum()
    判断字符串是否都是数字或字母
12. isalpha()
    判断是否都是字母
13. isdecimal()
    判断是否都是数字
14. isdigit()
    判断是否都是数字
15. isnumeric()
    判断是否都是数字
16. islower()
    判断是否都是小写
17. isupper ( )
    判断是否都是大写
18. isspace()
    判断是否都是空格
19. istitle()
    判断开头是否是大写
20. lstrip()
    去掉左侧的空格
21. rstrip()
    去掉右侧的空格
22. split(str)
    用str进行分割字符串，返回一个列表，str不存在
    >>> s='1012301230123'
    >>> s.split('0')
    ['1', '123', '123', '123']
    >>> s='1012301230123'
23. join(str)
    用str每个字符分割字符串
24. upper()
    将字符串转成大写
25. lower ( )
    将字符串转成小写

- 字符串的格式化

按照某个格式进行输出
format(*args,**kwargs)
    *args:可变长参数---元组
    **kwargs:可变长参数---字典
1. ***字符串中用{数字}表示占位***
        和参数的位置有关（字符串向参数要值---下标）
        s='{0}love{1}'
        print(s.format('刘能','赵四'))

         s='{2}love{1}{1}'
        print(s.format('刘能','赵四','广坤'))
2. {}直接将format里面的参数按照顺序进行替换
        s='{}love{}{}{}'
        print(s.format('刘能','赵四','广坤','刘大脑袋','长贵'))
        索要的参数不够则报错
3. 字符串中使用{字母}表示占位
        和关键字有关
        s='{a}love{b}{c}{d}'
        print(s.format(a='刘能',b='赵四',c='广坤',d='刘大脑袋',e='长贵'))
4. 位置参数和关键字参数混用时
        位置参数在前，关键字参数在后
        s='{d}love{b}{c}{0}'
        print(s.format('刘能',b='赵四',c='广坤',d='刘大脑袋',))
5. format进阶
        s='{0:.2f}'
        .1:保留小树几位 #四舍五入小数点后一位
        f:定点数
            将小树的位置固定在数据的最高位，或者固定最低位
            最高位：   .123--定点小树
            最低位：   123.--定点整数

- 字符串的格式化操作

定义：字符串通过 *'%*' 的形式表示占位，通过 %参数 的形式进行传参数*
        将参数格式化成字符串
        和位置有关，多个参数用逗号隔开并且用括号包一下
        参数一一对应，不能多也不能少
1. *%c：格式化编码*
        '%c'%97
        '%c%c'%(97,98)
2. *%s:格式化字符串*
        最常用（参数可以是任何值）
        #九九乘法表
        for i in range(1,10):
         for j in range(1,i+1):
             print('%s * %s = %s'%(i,j,i*j),end='\t')
         print()
3. *%d：格式化整数*
        >>> '%d'%56

```
        '56'
        >>> '%d'%100.923  ---向下取整（舍去小数部分）
        '100'
4. %o：格式化无符号八进制
        将参数转化成八进制
            7二进制---0111--0代表正数1111   1--代表负数  -7 ----1111---15
5. %x：格式化无符号十六进制
        >>> '%x'%10
        'a'
6. %X：同上
        ‘A’
7. %f：格式化定点数
        默认取小数点后六位（四舍五入）
8. %e：用科学计数法格式化定点数
        默认取小数点后六位（四舍五入）
        >>> '%e'%123.1324657
        '1.231325e+02'
        >>> '%e'%0.000001231324657
        '1.231325e-06'
9. %E：同上
10. %g： 自动选择使用%f或者%e
11. %G：同上
```

- 格化式的辅助命令

```
1. m.n
        m:显示的是最小宽度
        n：小数点后的位数
        >>> '%5.1f'%1.12345678
        '  1.1' #位数不够空格来凑
        >>> '%5.3f'%1.12345678
        '1.123'
        >>> '%5.4f'%1.12345678
        '1.1235'
2. -
        用于左对齐
        >>> '%-5.1f'%1.12345678
        '1.1  '

        >>> '%-5d'%1.12345678
        '1    '
3. # 显示八进制或者十六进制
        在八进制前显示0o--一种符号
        在十六进制前显示0x
        >>> '%o'%10
        '12'
        >>> '%#o'%10
        '0o12'
```

转移字符

\n:换行符
\t:横向制表符
\':单引号
\":双引号
\a:发出声音
\b:退格符
\v:纵向制表符
\f:换页符
\0:空串
\\: \

原始字符串：r 防止发生转意：r'C:\new\now'

```
s=5+3+2\  #字符串也可以
   +10\
   +12

print(s)
```