

python

split()

字符串的分割。

```
#,n 分割为n个字符
#','以为分割
#[n]取第n个字符作为后缀

s='www.dod.com.cn'
print(s.split(',',[0][0]))
print(s.split(',',[1][1]))
print(s.split(',',[2][2]))
print(s.split(',',[4][3]))
print(s.split(',',-1)[0])

#输出
www.dod.com.cn
dod.com.cn
com.cn
cn
www
>>>

s='www.dod.com.cn'
#分割三次并将分割的字符串保存到三个文件内
s1,s2,s3= s.split(',')
print(s1)
print(s3)
print(s2)
#输出
www
com.cn
dod
>>>

子主题

s='www.dod.com.cn'
# 去掉换行符 '\n \t'
c = '''hello
world'''
print(c)
print(c.split('\n'))
print(c.split('\t'))
#输出
hello
world
['hello', ' world']
['hello\n world']
>>>
```

count()

COUNT 顾名思义，就是计数的意思，此函数用处：对表中记录数进行计数。

```
var = "中华人民共和国万岁，huangyang"
var =print(var.count("中华人民"))
#输出
1
>>>
```

replace()

replace() 方法用于在字符串中用一些字符替换另一些字符，或替换一个与正则表达式匹配的子串。
stringObject.replace(要替换的文本,替换成的文本)

```
var = "终古人民共和国，终古人民"
var =print(var.replace("终古","中国"))
#输出
中国人民共和国，中国人民
>>>
```

center()

描述：返回一个长度为width,两边用fillchar(单字符)填充的字符串，即字符串str居中，两边用fillchar填充。若字符串的长度大于width,则直接返回字符串str

```
• str = "i love python"
print(str.center(1,"*"))
print(str.center(1,"*"))
print(str.center(20,"8"))
输出
i love python
„i love python...
?????????????????????????????????????i love python????????????????????????????????????????
```

strip()

子主strip() 方法用于移除字符串头尾指定的字符（默认为空格或换行符）或字符序列。
注意：该方法只能删除开头或是结尾的字符，不能删除中间部分的字符。

```
str = "123abcrunoob321"
print (str.strip('12 ')) # 字符序列为 12

# result

3abcrunoob3
```

join()

join(): 连接字符串数组。将字符串、元组、列表中的元素以指定的字符(分隔符)连接生成一个新的字符串

```
a1 = ['hello','good','boy','xiaoming']
print(''.join(a1))
print(':'join(a1))
#输出
hello good boy xiaoming
hello:good:boy:xiaoming print(':'join(a1))

a2 = 'hello good boy xiaoming'
print(':'join(a2))
#输出
h:el:l:o: :g:o:o:d: :b:o:y: :x:i:a:o:m:i:n:g
```

format()

通过位置，通过关键字，通过映射，填充与对齐，精度保留，千位分隔符，十进制转换

```
#通过位置
print ('{0},{1}'.format('chuhao',20))
print ('{},{1}'.format('chuhao',20))
print ('{1},{0},{1}'.format('chuhao',20))
#通过关键字参数
print ('{name},{age}'.format(age=18,name='chuhao'))
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def __str__(self):
        return 'This guy is {self.name},is {self.age} old'.format(self=self)

print (str(Person('chuhao',18)))
#通过映射 list
a_list = ['chuhao',20,'china']
print ('my name is {0[0]},from {0[2]},age is {0[1]}'.format(a_list))
#my name is chuhao,from china,age is 20

#通过映射 dict
b_dict = {'name':'chuhao','age':20,'province':'shaxi'}
print ('my name is {name}, age is {age},from {province}'.format(**b_dict))
#my name is chuhao, age is 20,from shaxi

#填充与对齐
print ('{:>8}'.format('189'))
# 189
print ('{:5>8}'.format('189'))
#00000189
print ('{:a>8}'.format('189'))
#aaaaa189

#精度与类型f
#保留两位小数
print ('{:2f}'.format(321.33345))
#321.33

#用来做金额的千位分隔符
print ('{:,}'.format(1234567890))
#1,234,567,890

#其他类型 主要就是进制了，b、d、o、x分别是二进制、十进制、八进制、十六进制。

print ('{:b}'.format(18)) #二进制 10010
print ('{:d}'.format(18)) #十进制 18
print ('{:o}'.format(18)) #八进制 22
print ('{:x}'.format(18)) #十六进制12
```

len()

参数：str：要计算的字符串、列表、字典、元组等
返回值：字符串、列表、字典、元组等元素的长度

```
>>> s="hello huangyang"
>>> len(s)
15
```

str()

1.在字符串中使用整数时，需要显式地将这个整数用作字符串，让其与两侧的字符串数据类型保持一致。即调用 str() 函数
2. str函数将对象转化为适于人阅读的形式。将整形转化为字符串形式

```
age = 20
sss = age + "岁快乐"
print(sss)
```

也许你以为这就输出了：
20岁快乐
但是事与愿违

Traceback (most recent call last):
File "C:/Users/19847/Desktop/66666.py", line 2, in <module>
sss = age + "岁快乐"
TypeError: unsupported operand type(s) for +: 'int' and 'str'

上面这些就是运行结果书的时数字和字符串不能识别
那这种情况下我们就需要用str()这一函数了

```
`age = 20
sss = str(age) + "岁快乐"
print(sss)
```

运行结果就是
"20岁快乐"

```
>>> list=["abcd","123456"] >>> a=123
>>> str(list) >>> str(a)
"['abcd','123456']" '123'
```

hex()

hex() 函数用于将10进制整数转换成16进制，以字符串形式表示。

```
>>>hex(255)
'0xff'
>>> hex(-42)
'-0x2a'
```

oct()

oct() 用于将一个十进制的整数转换成八进制字符串

```
>>> oct(24)
'0o30'
```

ord()

ord()函数主要用来返回对应字符的ASCLL码

```
>>> ord("a")
97
```

chr()

chr()h函数从指定的ASCLL码返回字符

```
>>> chr(97)
'a'
```

lower()

ower() 用于把字符串中的大写字母转换成小写字母

```
>>> str="HGJB"
>>> str.lower()
'hgjb'
```

upper()

upper() 用于把字符串中的小写字母转换成大写字母

```
>>> str.upper()
'HUGF'
```