python中str常用操作

**1. 字符串的操作**

字符串的连接操作

符号： +

格式：str1 + str2

例如：str1 = 'I Love'

str2 = 'You!'

print(str1 + str2)

>>> 'I Love You!'

返回值：str

字符串的复制操作

符号： \*

格式：str \* num

例如：str = 'abc'

num = 3

print(str1 \* num)

>>> 'abcabcabc'

返回值：str

字符串的索引操作

符号： []

格式：str[index]

例如：str = 'abcde'

print(str[2])

>>> 'c'

返回值：str

字符串的切片操作

符号： [::]

格式：str[index:index:step]

例如：str = 'abcdefg'

print(str[:])

>>> 'abcdefg'

#取字符串所有内容，开头的0可以忽略

print(str[:3])

>>> 'abc'

#取出字符串索引0-3的内容(顾头不顾尾，不包含索引3的对象)

print(str[:5:2])

>>> 'ace'

#取出字符串索引0-5的内容，每2个索引取一次

print(str[3:])

>>> 'defg'

#取出字符串索引3开始到结尾的内容

print(str[-2:])

>>> 'fg'

#取出字符串倒数第2开始到结尾的内容

返回值：str

**2. 内置方法**

连接

格式：x.\_\_add\_\_(y)等同于x+y

返回值：str

包含

格式：x.\_\_contains\_\_(y)

返回值：bool

相等

格式：x.\_\_eq\_\_(y)等同于x==y

返回值：bool

大于等于

格式：x.\_\_ge\_\_(y)等同于x>=y

返回值：bool

访问属性方法(文件操作时用)

格式：x.\_\_getattribute\_\_(self,name)

返回值：tuple

获取键值对(字典操作时用)

格式：x.\_\_getitem\_\_(key)

返回值：键的值

大于

格式：x.\_\_gt\_\_(y)等同于x>y

返回值：bool

哈希值

格式：x.\_\_hash\_\_()

返回值：哈希值，int类型

迭代器

格式：x.\_\_iter\_\_()

返回值：迭代器

小于等于

格式：x.\_\_le\_\_(y)等同于x<=y

返回值：bool

长度

格式：x.\_\_len\_\_()等同于len(x)

返回值：int类型

小于

格式：x.\_\_lt\_\_()

返回值：布尔类型

复制

格式：x.\_\_mul\_\_(y)等同于x\*y

返回值：str

注：：此处的y必须是int类型

不等于

格式：x.\_\_ne\_\_(y)等同于x!=y

返回值：bool

右->左 复制

格式：x.\_\_rmul\_\_(y)等同于y\*x

返回值：str

注：此处的y必须是int类型

**3. 常用方法**

首字母大写，后面的小写

格式：x.capitalize()

#开头第一个单词首字母大写，后面的所有字符串全部小写

例如：x = = 'i am A 好 boy'

print(x.capitalize())

>>> 'I am a 好 boy'

返回值：str

全部字符小写

格式：x.casefold()

#字符串中所有单词的所有字母全部小写

例如：x = = 'i am A 好 boy'

print(x.casefold())

>>> 'i am a 好 boy'

返回值：str

居中，两边默认以空格填充

格式：x.center()

#定义字符串的长度，不足长度时，两边以指定字符串进行填充

例如：x = 'abc'

print(x.center(20,'\*'))

>>> '\*\*\*\*\*\*\*\*abc\*\*\*\*\*\*\*\*\*'

返回值：str

计数(默认全文计数)

格式：x.count(str,index1,index2)

#指定开始和结束范围来统计某字符串的个数

例如：x = 'sffefwsf'

print(x.count('sf'),0,8)

>>> 2

返回值：int类型

编码

格式：x.encode()

#指定字符串的编码格式

例如:x.encode(encoding='utf-8')#转换为utf-8格式

返回值：bytes

以什么为结尾

格式：x.endswith(str,index1,index2)

#指定字符串的开始和结束范围，判断所选区域是否是以指定字符串结尾

例如：x = 'adfd'

print(x.endswith('fd',2,3))

>>> True

返回值：bool

把\t转换为空格

格式：x.expandtabs()

#默认开头到\t为8个字节，不足以空格填充

例如：x = 'i am\t A\t 好 boy'

print(x.expandtabs())

>>> 'i am A 好 boy'

返回值：str

查找

格式：x.find(str,index1,index2)

#指定开始和结束的范围，查找指定区域内是否由指定的字符串（只返回查找到第一个的索引值）

例如：x = 'asdfdsfsafsaf'

print(x.find('df',1,8))

>>> 2

返回值：int类型

注：如果find未查找到，将返回为-1

格式化

格式：x.format(\*args)

#字符串的格式化可以有参数，可以无参数，可以是索引值参数，也可以是关键字参数

1. 无参数

例如：s1 = 'I {} {} {}'

print(s1.format('love','you','!'))

>>> 'I love you !'

2. 索引参数

例如：s1 = 'I {0} {1} {0} {1}'

print(s1.format('love','you'))

>>> 'I love you love you'

注：使用索引参数时，只能按顺序，从索引0开始

3. 关键字参数

例如：s1 = 'I {m} {n}'

print(s1.format(m = 'love',n = 'you'))

>>> 'I love you'

4. 格式限定符

填充常和对齐一起使用：

^ < > 分别是居中，左对齐和右对齐，后面带宽度

: 后面带填充的字符，只能是一个字符，默认是空格

例如：

无(位置)参数：s1 = 'I love you {}'

print(s1.format(', very much !'))

>>> 'I love you , very much !'

默认居中方法：s1 = 'I love you {:^18}'

print(s1.format(', very much !'))

>>> 'I love you , very much ! '

指定字符居中:s1 = 'I love you {:\*^18}'

print(s1.format(', very much !'))

>>> 'I love you \*\*, very much !\*\*\*'

指定字符左对齐:s1 = 'I love you {:\*<18}'

print(s1.format(', very much !'))

>>> 'I love you , very much !\*\*\*\*\*'

指定字符右对齐:s1 = 'I love you {:\*>18}'

print(s1.format(', very much !'))

>>> 'I love you \*\*\*\*\*, very much !'

5. 精度与类型f

例如：s1 = '圆周率大概是{}'

print(s1.format(3.1415926))

>>> '圆周率大概是3.1415926'

s1 = '圆周率大概是{:.2f}'

print(s1.format(3.1415926))

>>> '圆周率大概是3.14'

s1 = '圆周率大概是{:.2f}'

print(s1.format(3.1415926))

>>> '圆周率大概是3.1416'

注：精度一般和浮点一起使用，取值时使用四舍五入法

6. 进制操作

主要的进制为b、d、o、x，分别是二、十、八、十六进制

例如：

十进制：s1 = 'The pen values {} yuan!'

s1 = 'The pen values {:d} yuan!'

print(s1.format(17))

>>> 'The pen values 17 yuan!'

二进制：s1 = 'The pen values {:b} yuan!'

print(s1.format(17))

>>> 'The pen values 10001 yuan!'

八进制：s1 = 'The pen values {:o} yuan!'

print(s1.format(17))

>>> 'The pen values 21 yuan!'

十六进：s1 = 'The pen values {:x} yuan!'

print(s1.format(17))

>>> 'The pen values 11 yuan!'

7. 金融字符，千分位 即,

例如：s1 = 'The phone is {}$ !'

print(s1.format(10000000))

>>> 'The phone is 10000000$ !'

s1 = 'The phone is {:,}$ !'

print(s1.format(10000000))

>>> 'The phone is 10,000,000$ !'

返回值：str

查找

格式：x.index(str,index1,index2)

#用法和find一样

返回值：int

注：：当index未查找到，程序将报错

非符号字符串

格式：x.isalnum()

#字符串中可以有大小写，可以有数字，但不可以有符号

返回值：bool

纯字母的字符串

格式：x.isalpha()

#字符串中可以有大小写，但不可以有数字和符号

返回值：bool

纯数字的字符串

格式：x.isdecimal()

#字符串中只可以数字

返回值：bool

纯数字的字符串

格式：x.isdigit()

#字符串中只可以数字

返回值：bool

开头字母的字符串

格式：x.isidentifier()

#字符串以字母开头，可以是大小写，后面可以有数字，但数字不能开头

返回值：bool

全部小写

格式：x.islower()

#字符串全部是小写，也可以有数字，数字可以开头，但不能全数字

格式：x.isnumeric()

#字符串中可以有数字，也可以有中文大小写数字

返回值：bool

可打印

格式：x.isprintable()

#打印为空，则为假

返回值：bool

空格

格式：x.isspace()

#字符串中只能是空格

返回值：bool

标题

格式：x.istitle()

#字符串中每个单词首字母大写

返回值：bool

全部大写

格式：x.isupper()

#字符串中可以有数字，数字可以开头，但必须有大写字母

返回值：bool

拼接

格式：x.join(str)

#把字符串以指定字符串进行相连

例如：x = 'abcde'

print('\_'.join(x))

>>> 'a\_b\_c\_d\_e'

返回值：str

左对齐

格式：x.ljust(str)

#字符串左对齐，需要指定长度，不足长度时可以用指定字符串进行填充（默认以空格填充），当指定长度小于字符串长度，将会左对齐

例如：x = 'abcde'

print(x.ljust(12,'\*'))

>>> 'abcde\*\*\*\*\*\*\*'

返回值：str

全部小写

格式：x.lower()

#字符串中所有单词，所有字母全部小写

#与casefold()功能一样

返回值：str

去除左边空格

格式：x.lstrip()

#去除对字符串左边的指定字符串，字符串中间和结尾的指定字符串不做处理，默认去除的是空格

例如：x = 'aaafdfdfaaadfdsaaa'

print(x.lstrip('a'))

>>> 'fdfdfaaadfdsaaa'

返回值：str

字符串进行分段

格式：x.partition(str)

#把字符串以从左到右第一个指定字符串为元素进行分段，以元组形式展现

例如：x = 'acbadfsadfsdfsd'

print(x.partition('sa'))

>>> ('acbadf', 'sa', 'dfsdfsd')

返回值：tuple

字符串替换

格式：x.replace(old,new,count)

#把字符串中指定的字符串替换为新字符串，默认全部替换，也可以指定替换次数，如果次数超过存在的个数，将全部替换

例如：x = 'acbadfsadfsdfsd'

print(x.replace('df','A',7))

>>> 'acbaAsaAsAsd'

返回值：str

右→左 查找

格式：x.rfind(str,index1,index2)

#功能同find()一样

返回值：int

右→左 查找

格式：x.rindex(str,index1,index2)

#功能同index()一样

返回值：int

右对齐

格式：x.rjust(str)

#字符串右对齐，需要指定长度，不足长度时可以用指定字符串进行填充（默认以空格填充），当指定长度小于字符串长度，将会右对齐

例如：x = 'adc'

print(x.rjust(6,'%'))

>>> '%%%adc'

返回值：str

右→左 字符串分段

格式：x.rpartition(str)

#把字符串以从右到左第一个指定字符串进行分段，以元组形式展示

例如：x = 'abccbacbd'

print(x.rpartition('cb'))

>>> ('abccba', 'cb', 'd')

返回值：tuple

字符串切片

格式：x.rsplit(str)

#把字符串以指定字符串进行切片，并以列表的形式展现

例如：x = 'abccbacbd'

print(x.rsplit('cb'))

>>> ['abc', 'a', 'd']

返回值：list

去除右边空格

格式：x.rstrip(str)

#去除字符串最右边的指定字符串，左边和中间的指定字符串不做处理，默认去除的是空格

例如：x = 'aaafdfdfaaadfdsaaa'

print(x.rstrip('a'))

>>> 'aaafdfdfaaadfds'

返回值：str

切片

格式：x.split()

#把字符串以指定字符串进行切片，并以列表的形式展现

#功能同rsplit()一样

返回值：list

换行符分段

格式：x.splitlines()

#把字符串以换行符进行切片，并以列表的形式展现

返回值：list

以什么为开头

格式：x.startswith(str,index1,index2)

#指定字符串的开始和结束范围，判断指定区域是否是以指定字符串开头

例如：x = 'adafsdaf'

print(x.startswith('da',1,8))

>>> True

返回值：bool

去除两边空格

格式：x.strip(str)

#去除字符串两边的指定字符串，中间的指定字符串不做处理，默认去除的是空格

例如：x = 'aaadfsaaafdsfaaa'

print(x.strip())

>>> 'dfsaaafdsf'

返回值：str

大小写互转

格式：x.swapcase()

#字符串中所有的字母进行大小写相互转换

例如：x = 'aBcdE'

print(x.swapcase())

>>> 'AbCDe'

返回值：str

标题化

格式：x.title()

#对字符串中所有单词首字母大写，单词中间的大写全部转换为小写

例如：x = 'asd faDSdsf sdf'

print(x.title())

>>> 'Asd Fadsdsf Sdf'

返回值：str

全部变大写

格式：x.upper()

#字符串中所有字母全部转换为大写

例如：x = 'DFdgDdfdg'

print(x.upper())

>>> 'DFDGDDFDG'

返回值：str

左对齐，零填充

格式：x.zfill()

#字符串左对齐，指定字符串长度，不足部分以0填充

例如：x = 'afd'

print(x.zfill(5))

>>> '00afd'

返回值：str

映射函数

x.maketrans(\*args)

#把两个字符串进行一一对应，两个字符串长度必须一致

格式：str.maketrans(str1,str2)

bytearray.maketrans(str1,str2)

bytes.maketrans(str1,str2)

返回值：dict

映射函数

x.translate(\*args)

#把maketrans()得到的映射表应用出来

格式：x.translate(str.maketrans(str1,str2))

返回值：str