下课啦，抽晚上时间整理的笔记，没有太完善，请大佬们查阅！！！！！！！！

字符串 str (string)

作用:

用来记录文字信息

表示方法:

在非注释中凡是用引号括起来的部分都是字符串

' 单引号

" 双引号

''' 三单引号

""" 三双引号

空字符串的字面值的表示方式

''

""

''''''

""""""

注:空字符串的布尔测试值bool(x)为 False

非空字符串的字面值

'hello'

"hello"

'''hello'''

"""hello"""

单引号和双引号的区别:

单引号内的双引号不算结束符

双引号内的单引号不算结束符

示例:

print("I'm a student")

print('I am "shuaigege"')

三引号字符串

以'''或"""开头,以'''或"""结尾的字符串

作用:

三引号字符串中的换行会自动转换为换行符'\n'

三引号内可以包含单引号和双引号

示例:

# 要表示如下的文字内容:

welcome to beijing.

I like python!

I am studing.

s = 'welcome to beijing.\nI like python!\nI am studing.'

s2 = '''welcome to beijing.

I like python!

I am studing.'''

隐式字符串字面值的拼接

"I'm student." 'I am "shuaigege"'

得到的字符串等同于

'''I'm student.I am " shuaigege "'''

用转义序列代表特殊字符

字符串字面值中用字符反斜杠 \ 后跟一些字符代表特殊的一个字符

转义字符表:

\' 代表一个单引号

\" 代表一个双引号

\n 代表一个换行符

\\ 代表一个反斜杠

\r 返回光标至行首

\t 水平制表符

\f 换页

\v 垂直制表符

\b 倒退(backspace)

\0 字符串,字符值为0

\0oo oo为两位八进制表示的字符

\xXX XX为两位十六进制表示的字符

\uXXXX Unicode 16的十六进制的字符

\UXXXXXXXX Unicode 32 的十六进制表示的字符

ASCII 编码:

# 查看 ASCII编码表

$ man ascii # 仅linux可用

常用的ASCII编码:

字符 十进制 十六进制

'0' 48 0x30

'A' 65 0x41

'a' 97 0x61

序列的概念

字符串就是序列

求序列的长度的函数len

len(x) 返回字符串中字符的个数

字符串练习:

'1234abcd' 有几个字符 8

'5\'4"' 有几个字符 4

'\"A\x34' 有几个字符 3

'\n\bc\t\r' 有几个字符 5

raw 字符串(原始字符串)

格式:

r"字符串内容"

r'字符串内容'

r"""字符串内容"""

r'''字符串内容'''

作用:

让转义字符 '\' 无效

示例:

a = "C:\newfile\test.py"

print(a)

b = r"C:\newfile\test.py"

print(b)

print(len(a)) # 16

print(len(b)) # 18

字符串的运算:

运算符:

+ += \* \*=

+ 加号运算符用于拼接字符串

+= 用于拼接运算后改变原变量的绑定关系

示例:

x = "abcd" + "efg"

print(x) # abcdefg

x += '123'

print(x) # abcdefg123

\* 运算符生成重复的字符串

x = "ABCD" \* 3 # x = 'ABCDABCDABCD'

y = 2 \* "123" # y = '123123'

注: 字符串只能和整数相乘

\*= 生成重复的字符串并改变原变量的绑定关系

x = '123'

x \*= 2 # x = '123123'

y = 3

y \*= "ABC" # y = 'ABCABCABC'

练习:

写一个程序,打印一个高度为4行的矩形方框

要求,输入一个整数,此整数代表矩形的宽度.

如:

输入矩形宽度: 10

打印如下:

##########

# #

# #

##########

如:

输入矩形宽度: 5

打印如下:

#####

# #

# #

#####

X=int(input(“请输入数字：”))

Print(“#”\*x)

Print(“#”,” ”\*x,”#”,sep=””)

Print(“#”,” ”\*(x-2),”#”,sep=””)

Print(“#”,” ”\*(x-2),”#”,sep=””)

Print(“#”,” ”\*x,”#”,sep=””)

字符串的比较运算

运算符:

< <= > >= == !=

示例:

'A' < 'B' # True

'AB' <= 'AC' # True

'ABC' > 'ACB' # False

'AD' > 'ABC' # True

'AB' < 'ABC' # True

'ABC' == 'abc' # False

'ABCD' != 'DCBA' # True

in / not in 运算符

作用:

in 用于序列,字典,集合中,用于判断某个值是否存在于容器中,如果存在返回True, 否则返回False

说明:

not in 返回结果与in 相反

格式:

对象 in 序列

示例:

x = 'welcome to tarena!'

'to' in x # True

'e t' in x # True

'hello' in x # False

'hello' not in x # True

写程序,输入一个长长的字符串,然后判断你的姓名是否存在于这个字符串中.

如果存在,则打印: '您的名字出现了'

否则提示: '没出现你的名字'

X=str(input(“请输入你的字符串:”))

A=str(input(“请输入你的名字：”))

If a in x

Print(“你的名字出现了”)

Else：

Print(”你的名字没出现”)

索引 index

python字符串是不可以改变的序列,

所有的序列都可以通过索引来获取其中的数据元素

语法:

字符串[整数表达式]

说明:

1. 序列的正向索引是从0开始的,第二个索引为1,最后一个索引为len(s)-1

2. 序列的反向索引是从-1开始的,-1代表最后一个,-2代表倒数第二个,以此类推,第一个是-len(s)

示例:

s= 'ABCDE'

s[0] # 'A'

s[-1] # 'E' 反向索引

s[-3] # 'C'

s[-5] # 'A'

s[-8] # IndexError 越界错误

切片 slice

从字符串序列中取出相应的元素重新组成一个字符串序列

语法:

s[(开始索引b):(结束索引e)(:(步长s))]

注: 小括号() 括起的部分代表可省略

说明:

1. 开始索引b是切片切下的位置,0代表第一个元素,1代表第二个元素 -1代表最后一个元素...

2. 结束索引是切片的终止索引(但不包含终止点)

3. 步长是切片每次获取完当前元素后移动的方向和偏移量

3.1 没有步长,相应于步长为1(默认为1)

3.2 当步长为正整数时,取正向切片,开始索引默认为0,结束索引为最后一个元素的下一个位置

3.3 当步长为负数时,取反向切片.

反向切片时,默认的起始位置为最后一个元素.终止位置是第一个元素的前一个位置

示例:

s = 'ABCDE'

a = s[1:4] # a---> 'BCD'

a = s[1:] # a---> 'BCDE'

a = s[0:2] # a---> 'AB'

a = s[:2] # a---> 'AB'

a = s[:] # a---> 'ABCDE'

a = s[1:1] # '' 空

a = s[4:1] # '' 空

a = s[1:4:2] # 'BD'

a = s[::2] # 'ACE'

a = s[4:0:-1] # a---> 'EDCB'

a = s[::-1] # 'EDCBA'

a = s[0::-1] # 'ECA'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

练习:

1. 写一个程序,输入一个字符串,把字符串的第一个字符和最后一个字符去掉后,打印出处理后的字符串

2. 输入任意一个字符串,判断这个字符串是否是回文

回文示例:

上海自来水来自海上

ABCCBA

12321

回文是指中心对称的文字

# x=str(input("请输入："))

# print(x[1:len(x)-1])

# a=str(input("请输入："))

# if a[:]==a[::-1]:

# print("shihuiwen")

# else:

# print("bushi ")

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python 3中能用于字符串的函数

len(x) 返回字符串的长度

max(x) 返回字符串串中编码值最大的字符

min(x) 返回最小值的字符

示例:

s = "Hello"

len(s) # 5

max(s) # 'o'

min(s) # 'H'

字符串编码转换函数:

ord(c) 返回一个字符c的Unicode编码值

chr(i) 返回i这个值所对应的字符

示例:

print(ord('A')) # 65

print(ord('中')) # 20013

print(chr(65)) # A

print(chr(2345)) # ???

整数转字符串的函数

bin(i) 将整数转换为二进制字符串

oct(i) 将整数转换为八进制字符串

hex(i) 将整数转换为十六进制字符串

示例:

bin(9) # '0b1001'

oct(10000) # '0o23420'

hex(10000) # '0x2710'

字符串的构造(创建)函数

str(obj='') 将对象转换为字符串

帮助函数 help(xxx)

>>> help(函数名)

>>> help(类型名)

python3 中常用的字符串方法(method)

方法的调用语法:

对象.方法名(方法传参)

示例:

'abc'.isalpha() #True判断'abc'是否是字母

123.isalpha() # 报错

详见:

help(str)

练习:

输入一个字符串:

1. 判断您输入的字符串有几个空格

2. 将源字符串左右空白字符去掉,打印出有效的字符的个数

3. 判断您输入的是否是数字

3.1 如果是数字,判断您输入的数字是否大于100

# x=str(input("请输入："))

# print(x[1:len(x)-1])

# a=str(input("请输入："))

# if a[:]==a[::-1]:

# print("shihuiwen")

# else:

# print("bushi ")

练习:

1. 用字符串 \* 运算符打印三角形

要求输入一个整数,此整数代表最长的一行星离左侧的字节数

如:

请输入离左侧的距离: 3

打印如下:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

# a=int(input("请输入数字："))

# print(a\*" "," \*",sep='')

# print(a\*" "," \*\*\*",sep='')

# print(a\*" "," \*\*\*\*\*",sep='')

# print(a\*" ","\*\*\*\*\*\*\*",sep='')

2. 输入一个字符串,把输入的字符串中的空格全部去掉.打印出处理后的字符串的长度和字符串内容

# # a=str(input("请输入："))

# # b=a.replace(" ",'')

# # print(b,len(b))

3. 输入三行文字,让这三行文字在一个方框内居中显示

如输入(不要输入中文)

hello!

I'm studing python!

I like python

打印如下:

+-------------------------------+

| hello! |

| I'm studing python! |

| I like python |

+--------------------------------+

# a=str(input("请输入："))

# b=str(input("请输入："))

# c=str(input("请输入："))

# max=len(a)

# if len(b) >max:

# max =len(b)

# if len(c)>max:

# max =len(c)

# v=a.center(max+5)

# j=b.center(max+5)

# p=c.center(max+5)

# print("+",(max+5)\*'-',"+")

# print("|",v,"|")

# print("|",j,"|")

# print("|",p,"|")

# print("+",(max+5)\*'-',"+")