python的基本数据类型

2020年11月27日

15:21



空对象：None

布尔值：True、False

**数字类型--int,float,complex**

1.数字类型（复数除外）支持的运算



例子：

x = 12.2

y = 9.8

print(x+y,x-y,x\*y,x/y,x//y,x%y)

print(abs(x)) #x的绝对值

print(int(x))

print(float(x))

c = complex(10,12) #复数 (10+12j)

print(c)

print(c.conjugate())#复数c的共轭 (10-12j)

print(divmod(x,y)) #(x//y,x%y)

print(pow(x,y)) #x的y次幂

print(x\*\*y)#x的y次幂

2.



例子：

import math

a = 12.256

b = 12.625

print(math.trunc(a)) #12

print(math.trunc(b)) #12

print(round(a,1)) #12.3 四舍五入

print(round(b)) #13

print(math.floor(a)) # 12 小于等于a的最大整数

print(math.ceil(a))  #13大于等于a的最大整数

**序列类型--list,tuple,string**

1.通用序列操作



例子：

string = 'hello'

list1 = [1,2,3,'hello']

tuple1 = (1,2,3,'world')

print('h' in string)

print('h' not in string)

print(string + string)

print(list1 + list1)

print(tuple1 + tuple1)

print(string \* 6)

print(list1 \* 6)

print(tuple1 \* 6)

print(string[1])

print(string[1:3])

print(string[:6:2])

print(len(string))

print(min(string))

print(max(string))

print(string.index('l'))

print(string.count('l'))

2.可变序列--list



例子：

s = ['h','l','o','1','2','3','4','5','6']

s[0]='e'

print(s) #['e', 'l', 'o', '1', '2', '3', '4', '5', '6']

s[1:3] = ['a','b','c']

print(s) #['e', 'a', 'b', 'c', '1', '2', '3', '4', '5', '6']

del s[6:]

print(s) #['e', 'a', 'b', 'c', '1', '2']

s[3:5:1] = ['0','0']

print(s) #['e', 'a', 'b', '0', '0', '2']

del s[3:5:1]

print(s) #['e', 'a', 'b', '2']

s.append('u')

print(s) #['e', 'a', 'b', '2', 'u']

t = s.copy()

print(t) #['e', 'a', 'b', '2', 'u']

s.clear()

print(s) #[]

t.extend('s')

print(t) #['e', 'a', 'b', '2', 'u', 's']

t\*=2

print(t) #['e', 'a', 'b', '2', 'u', 's', 'e', 'a', 'b', '2', 'u', 's']

t.insert(0,'w')

print(t) #['w', 'e', 'a', 'b', '2', 'u', 's', 'e', 'a', 'b', '2', 'u', 's']

print(t.pop(6)) #s

print(t)  #['w', 'e', 'a', 'b', '2', 'u', 'e', 'a', 'b', '2', 'u', 's']

t.remove('e')

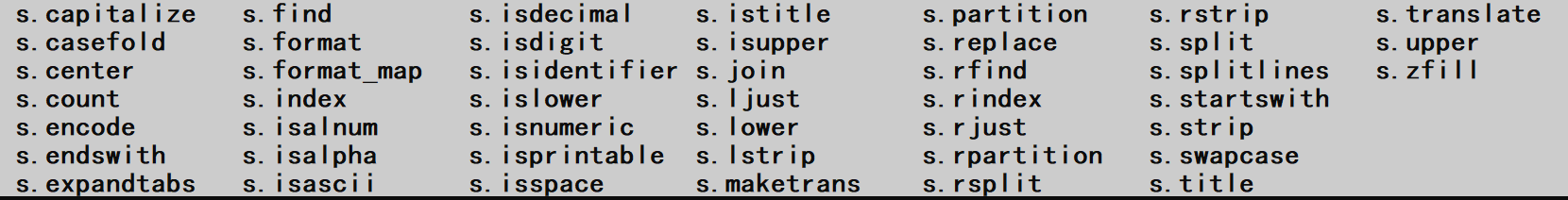
print(t)  #['w', 'a', 'b', '2', 'u', 'e', 'a', 'b', '2', 'u', 's']

t.reverse()

print(t) #['s', 'u', '2', 'b', 'a', 'e', 'u', '2', 'b', 'a', 'w']

3.不可变序列--tuple

4.文本序列类型--str



例子：

str = 'hello World!'

#1 str.capitalize()返回原字符串的副本，其首个字母大写，其余为小写

print(str.capitalize()) #Hello world.!

#2 str.casefold() 返回原字符串消除大小写的副本。消除大小写的字符串可用于忽略大小写的匹配

print(str.casefold()) #hello world.!

#3 str.center(width[,fillchar])返回长度为width的字符串，原字符串在其正中。使用指定的fillchar填充两边的空位

#（默认使用ASCII空格符）。如果width小于等于len(s)则返回原字符串的副本。

print(str.center(16,'\*')) #\*\*hello World!\*\*

print(str.center(10)) #hello World!

#4 str.count(sub[,start[,end]])返回子字符串sub在[start,end]范围内非重叠出现的次数。可选参数start与end会被解读为切片表示法

str2 = 'pananananam'

print(str2.count('an')) #4

print(str2.count('an',2,8)) #2

#5 str.encode(encoding='utf-8',errors='strict')返回原字符串编码为字节串对象的版本。

# 默认编码为 'utf-8'。 可以给出 errors 来设置不同的错误处理方案。 errors 的默认值为

# 'strict'，表示编码错误会引发 UnicodeError。 其他可用的值为 'ignore', 'replace',

# 'xmlcharrefreplace', 'backslashreplace' 以及任何其他通过 codecs.register\_error() 注册的值

#6 str.endswith(suffix[, start[, end]])如果字符串以指定的 suffix 结束返回 True，否则返回 False。

# suffix 也可以为由多个供查找的后缀构成的元组。 如果有可选项 start，将从所指定位置开始检查。

#  如果有可选项 end，将在所指定位置停止比较。

print(str2.endswith('a',2,4)) #True

#7 str.expandtabs(tabsize=8)返回字符串的副本，其中所有的制表符会由一个或多个空格替换，具体取决于当前列位置和给定的制表符宽度。

str3 = '01\t012\t0123\t01234'

print(str3.expandtabs())

#8 str.find(sub[,start[,end]]) 返回子字符串sub在s[start:end]切片内被找到的最小索引

print(str2.find('na'))

#9 str.format(\*args,\*kwargs)执行字符串格式化操作

print('The sum of 1 + 2 is {0}'.format(1+2)) #The sum of 1 + 2 is 3

#10 str.index(sub[,start[,end]])  类似于 find()，但在找不到子类时会引发 ValueError。

print(str2.index('na'))

#11 isalpha() 如果字符串中的所有字符都是字母，并且至少有一个字符，返回 True ，否则返回 False

print(str2.isalpha())

#12 str.isalnum() 如果字符串中的所有字符都是字母或数字且至少有一个字符，则返回 True ， 否则返回 False 。

print(str2.isalnum())

#13 str.isascii() 如果字符串为空或字符串中的所有字符都是 ASCII ，返回 True ，否则返回 False 。ASCII 字符的码点范围是 U+0000-U+007F 。

print(str2.isascii())

#14 str.lower() 返回原字符串的副本，其所有区分大小写的字符 4 均转换为小写。

print(str2.lower())

#15 str.upper() 返回原字符串的副本，其中所有区分大小写的字符 4 均转换为大写。

print(str2.upper())

#16 str.split(sep=None, maxsplit=-1) 返回一个由字符串内单词组成的列表，使用 sep 作为分隔字符串。

# 如果给出了 maxsplit，则最多进行 maxsplit 次拆分（因此，列表最多会有 maxsplit+1 个元素）。

print(str.split())