Python

zz

October 16, 2020

Contents

1 Python基础

1.1 程序输入

input_var = input("Please input something:")

函数 input() 让程序暂停运行，等待用户输入一些文本。获取用户输入后，Python将其存储在一个变量中。函数 input() 接受一个参数：即要向用户显示的“提示”或者“说明”，让用户知道该怎么做。可以将参数先存储在变量中，然后将变量传递给 input()。

使用函数 input() 时，Python将用户输入解读为“字符串”。因此输入整数、浮点数等时需要强制格式转换。

1.2 条件语句

if语句

if boolean_exp :

exp;

if-elif-else结构依次检查每个测试条件，直到遇到通过了的条件，Python将执行紧跟在其后边的代码块，并跳过余下的代码块。else是一条包罗万象的语句，只要不满足任何 if 和 elif 的条件测试，其中的代码块就会被执行，这可能会引入无效甚至恶意的数据。如果知道最终要测试的条件，应考虑使用一个 elif 来代替 else 。这样你可以肯定仅当满足相应的条件时，你的代码才会被执行。

在 if 语句中将列表名用在条件表达式中时，Python将在列表至少包含一个元素时返回True，并在列表为空时返回False。

1.3 循环语句

for循环

for value in iterator:

use value do something

for循环可以遍历任何可迭代对象iterator中的所有元素。每次返回一个元素赋值给for关键字后的变量名value。

while循环

while boolean_exp:

do something

while循环当布尔表达式为真时一直运行。

在要求很多条件都满足时才继续运行的程序中，使用一个变量作为标志，用于判断整个程序是否处于活动状态是一个不错的做法。任何一个事件导致标志位False都将结束循环。

break

break语句跳出当前整个循环，不在继续执行余下的循环轮次。

continue

continue语句跳出当前循环轮次，回到循环的开头，判断循环条件后进行下一次循环。

for循环是一种遍历列表的有效方式，但是for循环中不应该修改列表，否则将导致Python难以跟踪其中的元素。要在遍历列表的同时对其进行修改，可以使用while循环。通过while循环同列表和字典结合使用，可以收集、存储并组织大量的输入。

1.4 Python注释

单行注释：‘#’
多行注释：三引号

1.5 Python标准数据类型

Python3 中有六个标准的数据类型：
- Number（数字）
- String（字符串）
- List（列表）
- Tuple（元组）
- Set（集合）
- Dictionary（字典）

Python3 的六个标准数据类型中：
不可变数据（3 个）：Number（数字）、String（字符串）、Tuple（元组）
可变数据（3 个）：List（列表）、Dictionary（字典）、Set（集合）

1.6 Python关键字

False：Bool值，逻辑“假”
True：Bool值，逻辑“真”
and：表示逻辑‘与’
or：表示逻辑“或”
not：表示逻辑‘非’
if：if条件语句用于选择分支，依据条件选择执行那个语句块
else：if语句的逻辑假分支
elif：和if配合使用的，if语句中的一个分支用elif表示
for：for循环
while：while循环
break：终止循环
continue：跳过continue后面循环块中的语句，继续进行下一轮循环
from：导入相应的模块，用import或者from...import...
import：导入模块
as： 导入模块时起别名，import...as...；与with连用，with open(“file”,’r’) as f
class：声明类关键字

def：定义类、函数等
in：查找列表中是否包含某个元素
is：is判断的是a对象是否就是b对象，是通过id来判断的；而==判断的是a对象的值是否和b对象的值相等，是通过value来判断的
print：输出
return：函数返回关键字
pass：pass的意思就是什么都不做。
with：上下文管理，在with关键字管理的语句块中，打开的文件在执行完with语句块后释放
None：None有自己的数据类型NoneType，并且这个类型永远只有它一个。
assert：表示断言，用于声明某个条件为真，如果该条件不是真的，则抛出异常：AssertionError
del：用于删除一个或者连续几个元素
try：和try一起使用，用来捕获异常
except：与try关键字一起使用，捕获异常
finally：看到finally语句，必然执行finally语句的代码块
global：定义全局变量
lambda：匿名函数，不用想给函数起什么名字。提升了代码的简洁程度
nonlocal：与global关键字的区别见下边。
raise：raise可以显示地引发异常。一旦执行raise语句，后面的代码就不执行了
yield：用起来和return很像，但它返回的是一个生成器

1.7 Python运算符

2 Number

3 String

3.1 Python字符串

字符串是 Python 中最常用的数据类型。我们可以使用引号(’或”)来创建字符串。 Python 不支持单字符类型，单字符在 Python 中也是作为一个字符串使用。Python 访问子字符串时，可以使用方括号来截取字符串中的字符或子字符串。

Python使用加号（+）来拼接字符串。

函数 str() 可以将非字符串的值转换为字符串。

3.2 Python转义字符

转义字符	描述
\(在行尾时)	续行符
\\	反斜杠符号
\′	单引号
\″	双引号
\b	退格(Backspace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页

3.3 Python字符串运算符

下表实例变量 a 值为字符串 “Hello”，b 变量值为 “Python”

操作符	描述	实例
+	字符串连接	>>>a + b 'HelloPython'
*	重复输出字符串	>>>a * 2 'HelloHello'
[ ]	通过索引获取字符串中字符	>>>a[1] 'e'
[:]	双引号	>>>a[1:4] 'ell'
in	如果字符串中包含给定的字符返回True	>>>'H' in a True
not in	如果字符串中不包含给定的字符返回True	>>>'H' not in a False
r/R	所有的字符串都是直接按照字面的意思来使用	>>>print r'\n' \n
%	格式字符串	

3.4 Python三引号

Python 中三引号可以将复杂的字符串进行赋值。

Python 三引号允许一个字符串跨多行，字符串中可以包含换行符、制表符以及其他特殊字符。

三引号的语法是一对连续的单引号或者双引号（通常都是成对的用）。

三引号让程序员从引号和特殊字符串的泥潭里面解脱出来，自始至终保持一小块字符串的格式是所谓的WYSIWYG（所见即所得）格式的。一个典型的用例是，当你需要一块HTML或者SQL时，这时当用三引号标记，使用传统的转义字符体系将十分费神。

3.5 字符串内建函数

1. str.capitalize()

功能:

返回一个当前整个字符串的第一个字母大写的版本，原字符串不变。
参数:
无

2. str.casefold()
功能:
返回一个当前整个字符串的全小写的版本，原字符串不变。与str.lower()的区别为后者只能用于ASCII码字符串，前者可以用于多国语言。
参数:
无

3. str.center(width[, fillchar])
功能:
返回一个指定的宽度 width 使当前字符串居中的新字符串，fillchar 为填充的字符，默认为空格。如果 width 小于字符串宽度直接返回字符串，不会截断。fillchar 只能是单个字符，默认为空格。
参数:
width — 新字符串的宽度
fillchar — 填充字符

4. str.count(sub[, start[, end]])
功能:
返回子字符串在字符串中出现的次数。方法用于统计字符串里某个字符出现的次数。可选参数为在字符串搜索的开始与结束位置。
参数:
sub — 搜索的子字符串
start — 字符串开始搜索的位置。默认为第一个字符,索引值为0。
end — 字符串结束搜索的位置。默认为字符串的最后一个位置。

5. str.encode(encoding='utf-8', errors='strict')
功能:
返回编码后的字符串。该方法以 encoding 指定的编码格式编码字符串。errors 参数可以指定不同的错误处理方案。
参数:
encoding — 要使用的编码，默认为"UTF-8"。
errors — 设置不同错误的处理方案,默认为 'strict',意为编码错误引起一个UnicodeError。 其他可能的值有 'ignore', 'replace', 'xmlcharrefreplace', 'backslashreplace' 以及通过 codecs.register_error() 注册的任何值。

6. str.endswith(suffix[, start[, end]])
功能:
返回一个bool值。方法用于判断字符串是否以指定后缀结尾，如果以指定后缀结尾返回True，否则返回 False。可选参数 "start" 与 "end" 为检索字符串的开始与结束位置。
参数:
suffix — 被匹配的字符串。该参数可以是一个字符串或者是几个字符串组成的元组。
start — 字符串中的开始位置。
end — 字符中结束位置。

7. str.expandtabs(tabsize=8)
功能:
返回字符串中的 tab 符号('\t')转为空格后生成的新字符串。方法把字符串中的 tab 符号('\t')转为空格，tab 符号('\t')默认的空 格数是 8。添加的空格数为一个制表符大小减去已有的字符个数。
参数:
tabsize – 指定转换字符串中的 tab 符号('\t')转为空格的字符数。

8. str.find(sub[, start[, end]])
功能:
返回包含的子字符串开始的索引值，如果不包含子字符串返回-1。
参数:
sub – 指定检索的字符串
start – 开始的索引，默认为0。
end – 结束的索引，默认为字符串的长度。

9. str.format()
功能:
字符串的格式化。详情见字符串格式化。
参数:
无。

10. str.format_map(mapping)
功能:
字符串的格式化。详情见字符串格式化。
参数:
mapping – 格式化字符串需要的一个字典对象。

11. str.index(sub, start=None, end=None)
功能:
如果包含子字符串,则返回开始的索引值，否则抛出异常ValueError。(和find()类似，但是会抛出异常)。
参数:
sub – 指定检索的字符串。
start – 开始的索引，默认为0。
end – 结束的索引，默认为字符串的长度。

12. str.isalnum()
功能:
如果 string 至少有一个字符并且所有字符都是[字母或数字]则返回 True,否则返回 False。
参数:
无。

13. str.isalpha()
功能:
如果字符串至少有一个字符并且所有字符都是字母则返回 True，否则返回 False。
参数:
无。

14. str.isdecimal()
功能:
检查字符串是否只包含十进制字符。这种方法只存在于unicode对象。注意:定义一个十进制字符串，只需要在字符串前添加 'u' 前缀即可。
参数:
无。

15. str.isdigit()
功能:
如果字符串只包含数字则返回 True 否则返回 False。
参数:
无。

16. str.isidentifier()
功能:
如果字符串是有效的 Python 标识符返回 True，否则返回 False。可用来判断变量名是否合法。
参数:
无。

17. str.islower()
功能:
如果字符串中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False。
参数:
无。

18. str.isnumeric()
功能:
如果字符串中只包含数字字符，则返回 True，否则返回 False。这种方法是只针对unicode对象。
参数:
无。

19. str.isprintable()
功能:
如果字符串中为空或者字符串中所有的字符都是可打印字符，则返回 True，否则返回 False。
参数:
无。

20. str.isspace()
功能:
如果字符串中只包含空格，则返回 True，否则返回 False。
参数:
无。

21. str.istitle()
功能:
如果字符串中所有的单词拼写首字母是否为大写，且其他字母为小写则返回 True，否则返回 False。
参数:
无。

22. str.isupper()
功能:
如果字符串中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是大写，则返回 True，否则返回 False。
参数：无。

23. str.join(iterable)
功能:
用于将序列中的元素以指定的字符(str)连接生成一个新的字符串。
参数:
iterable — 要连接的元素序列。

24. str.ljust(width, fillchar=None)
功能:
返回一个原字符串左对齐,并使用空格填充至指定长度的新字符串。如果指定的长度小于原字符串的长度则返回原字符串。
参数:
width — 指定字符串长度。
fillchar – 填充字符，默认为空格。

25. str.lower()
功能:
返回将字符串中所有大写字符转换为小写后生成的字符串拷贝。
参数:
无。

26. str.lstrip(chars=None)
功能:
返回截掉字符串左边的空格或指定字符后生成的新字符串。
参数:
chars –指定截取的字符,默认为空格。

27. str.maketrans(*args, **kwargs)
功能:
功能：maketrans() 方法用于给 translate() 方法创建字符映射转换表。可以只接受一个参数，此时这个参数是个字典类型（暂不研究）。对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串，表示转换的目标。两个字符串的长度必须相同，为一一对应的关系。在Python3中可以有第三个参数，表示要删除的字符，也是字符串。一般 maketrans() 方法需要配合translate() 方法一起使用。
参数:
intab – 需要转换的字符组成的字符串。
outtab – 转换的目标字符组成的字符串。
delchars – 可选参数，表示要删除的字符组成的字符串。
示例:

```
intab = "aeiou"
outtab = "12345"
deltab = "thw"
trantab1 = str.maketrans(intab,outtab)
trantab2 = str.maketrans(intab,outtab,deltab)
test = "this is string example....wow!!!"
print(test.translate(trantab1))
```

print(test.translate(trantab2))
输出：
th3s 3s str3ng 2x1mpl2....w4w!!!
3s 3s sr3ng 2x1mpl2....4!!!

28. str.partition(sep)
功能：
方法用来根据指定的分隔符将字符串进行分割。返回一个3元的元组，第一个为分隔符左边的子串，第二个为分隔符本身，第三个为分隔符右边的子串。
参数：
sep — 指定的分割符。

29. str.replace(old, new, count=None)
功能：
返回字符串中的 old（旧字符串）替换成 new(新字符串)后生成的新字符串，如果指定第三个参数count，则替换不超过count次。
参数：
old — 将被替换的子字符串。
new — 新字符串，用于替换old子字符串。
count — 可选字符串, 替换不超过 count 次。

30. str.rfind(sub, start=None, end=None)
功能：
返回字符串最后一次出现的位置(从右向左查询)如果没有匹配项则返回 -1。
参数：
str — 查找的字符串。
start — 开始查找位置，默认为首字符，下标为0。
end — 结束查找位置，默认为字符串的长度。

31. str.rindex(sub, start=None, end=None)
功能：
返回字符串最后一次出现的位置(从右向左查询)，如果没有匹配项则抛出异常ValueError。
参数：
str — 查找的字符串。
start — 开始查找位置，默认为首字符，下标为0。
end — 结束查找位置，默认为字符串的长度。

32、str.rjust(width, fillchar=None)
功能：
返回一个原字符串右对齐,并使用空格填充至长度 width 的新字符串。如果指定的长度小于字符串的长度则返回原字符串。
参数：
width — 指定填充指定字符后中字符串的总长度.
fillchar — 填充的字符，默认为空格。

33、str.rpartition(sep)
功能:
方法从右边开始查找，根据指定的分隔符将字符串进行分割。返回一个3元的元组，第一个为分隔符左边的子串，第二个为分隔符本身，第三个为分隔符右边的子串。
参数:
sep — 指定的分割符。

34、str.rsplit(sep=None, maxsplit=-1)
功能:
通过指定分隔符对字符串从右边开始进行切片，如果参数 num 有指定值，则分隔成为 num+1 个子字符串。返回分割后的字符串列表。
参数:
sep — 指定的分割符。
maxsplit — 分割次数。

35. str.rstrip(chars=None)
功能:
返回截掉字符串右边的空格或指定字符后生成的新字符串。
参数:
chars — 指定截取的字符,默认为空格。

36. str.split()
功能:
通过指定分隔符对字符串从左边开始进行切片，如果参数 num 有指定值，则分隔成为 num+1 个子字符串。返回分割后的字符串列表。
参数:
sep — 指定的分割符。
maxsplit — 分割次数。

37. str.splitlines(keepends=None)
功能:
按照行('\r', '\r\n', '\n')分隔，返回一个包含各行作为元素的列表，如果参数 keepends 为 False，不包含换行符，如果为 True，则保留换行符。
参数:
keepends – 在输出结果里是否保留换行符，默认为 False。

38. str.startswith(prefix, start=None, end=None)
功能:
方法用于检查字符串是否是以指定子字符串开头，如果是则返回 True，否则返回 False。如果参数 start 和 end 指定值，则在指定范围内检查。
参数:
prefix – 检测的字符串。
start – 可选参数用于设置字符串检测的起始位置。
end – 可选参数用于设置字符串检测的结束位置。

39. str.strip(chars=None)
功能:
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。
参数:
chars – 移除字符串头尾指定的字符序列。

40. str.swapcase()
功能:
方法用于对字符串的大小写字母进行转换。返回大小写字母转换后生成的新字符串。
参数:
无。

41. str.title()
功能:
方法返回"标题化"的字符串,就是说所有单词都是以大写开始，其余字母均为小写。
参数:
无。

42. str.translate(table)
功能:
方法根据参数table给出的表(包含 256 个字符)转换字符串的字符。
参数:
table – 翻译表，翻译表是通过maketrans方法转换而来。

43. str.upper()
功能:
方法将字符串中的小写字母转为大写字母。返回小写字母转为大写字母的字符串。
参数:
无。

45. str.zfill(width)
功能:
方法返回指定长度的字符串，原字符串右对齐，前面填充0。
参数:
width — 指定字符串的长度。

4 String