

https://woaielf.github.io/
—— made by ZY

字符串

操作符

标准类型操作符 e.g. 比较操作 按照 ASCII 值来比较大小

序列操作符 + 可用连接多个字符串
... 或 join() 性能更佳

更推荐 .format() 方法

格式化作操作符 (%)

原始字符串操作符 (r/R) r'' '' 内部字符 默认「不转义」

Unicode 字符串操作符 (u/U) u''

字典参数 print "I'm %(key)s" % (key: value)

flags <sp> 一个空格 在正数的左侧填充一个空格 从而 与负数对齐

辅助指令

width 显示宽度

precision 小数点后精度

typecode %d 整数 %2d 占两个字符 %02d 占两个字符并用 0 补全 %f 浮点数 %2f 两位小数 %s 字符串 %e 科学计数法

1 特殊字符串 转义字符 \ \n 换行 \t 制表符 \\ 反斜杠 \r 回车

2 ... 多行内容

3 不可变性 指向永远不变 改变一个字符串元素 需要新建「字符串」

注意 单引号和双引号用法相同 均表示字符串 字符串 可看作 特殊的「元组」

独特特性

编码问题

概念

1 bit (位) = 0 / 1 = 2 种可能性

1 byte (字节) = 8 bit = 2^8 = 256 种可能性

ASCII

1 个字节 8 位

英文字符 + 常见符号

Unicode

2 个字节 16 位

字符集 世界文字字符

变长编码方式

UTF-8 英文 1 个字节 中文 3 个字节 生僻字符 4-6 字节

UTF-16 统一 2 个字节

汉字

GB2312

DBCS

GBK

GB18030

127 号

之前 "半角" 字符

之后 "全角" 字符

需要 文本 -> 数字

Unicode (字符集) -> UTF-8 (可变长编码方式)

Python

类型

UTF-8 3.X 版本「默认编码」

ASCII 2.X 版本「默认编码」

指定方式 e.g. #-*- coding: #-

转换 -> UTF-8 编码 u'ABC'.encode('utf-8') 'ABC'

e.g. 中国 = 'zhongguo'

print(中国)

BIF

按照 ASCII 值来比较大小 cmp()

len() / enumerate() / zip()

input()

工厂函数 str() & unicode()

dir(str) 查看所有 BIF

更改显示方式

首字母大写, 其余小写 capitalize()

每个单词首字母大写 title()

全大 (小) 写 lower() / upper()

反转大小写 swapcase()

居中 center(width)

左 (右) 对齐 ljust(width) / rjust()

前面用 0 填充至 width 长度 zfill(width)

decode() / encode()

检查

find(str) / rfind()

index(str) / rindex()

endswith(obj) / startswith()

isalnum() / isalpha() / isdigit() / islower() / isspace() / istitle() / isupper()

tab 符 -> 空格 expandtabs()

创建 (str 中的 a -> A) 的新对象后返回 replace('a', 'A')

strip() / lstrip() / rstrip()

split() / rsplit() / splitlines()

根据 str 分割字符串为「三元组」 partition(str) / rpartition()

join(iter)

字符串 BIF

格式化作字符串

str.format()

参数

位置参数 'b, a, b' '(1, 0), (1).format('a', 'b')

关键字参数 '(name, age).format(age=18, name='kzc')

下标 p=['kzc', 18] '0[0], 0[1]'.format(p)

格式限定符

<填充> <对齐> <宽度> , <精度> <类型>

< 左 > 右 ^ 居中

千位分隔符

整数 b, c, d, o, x, X

浮点数 e, E, f, %

e.g.

'189' '(>8).format('189')

'00000189' '{:0>8}.format('189')

'aaaaa189' '{a>8}.format('189')

'321.33' '{:.2f}.format(321.33345)

精度 & 类型

二进制、十进制、八进制、十六进制 b, d, o, x

数字类型

'10001' '{b}'.format(17)

'1,234,567,890' '{:,}'.format(1234567890)

金额的千位分隔符

转换字符串的字符

通过「maketrans 方法」转换而来

要过滤掉的字符 del 参数

trantab = maketrans(intab, outtab)

from string import maketrans

intab = "aeiou"

outtab = "12345"

str = "this is string example...wow!!!"

th3s 3s str3ng 2x1mpl2....w4w!!! print(str.translate(trantab))

th3s 3s str3ng 21pl2....w4w!!! print(str.translate(trantab, 'xm'))