三元判断关联

例： temp = x if x > y else y #如果x>y,temp=x,否则temp=y

for循环

for 目标 in 表达式:

循环体

member = [1,2,3,4,5]

member.append() #向列表末端添加元素，但一次只能添加一个元素

member.extend() #扩展列表，向列表里添加一个列表，即向列表添加多个元素,extend([元素,元素,元素])

member.insert() #向列表插入一个元素，insert(插入位置，元素名)

member.remove() #移除列表里某个元素，remove(元素名) 即知道元素名的移除

del member[] #删除列表里某个元素，member[元素序号]，即知道元素序号的移除

member.pop() #从列表中取出最后一个元素，并返回元素值，若 pop(元素序号)，则取出指定元素

member[:] #分（切）片，从列表中拷贝出某个片段序列，member[1:3]=[2,3]

member.count() #统计某个元素在列表中的个数

member.index() #找出某个元素在整个列表中第一次出现的位置，index(元素名，范围起始编号，范围结束编号)

member.reverse()#使列表中的元素逆序排序，member.reverse() = [5,4,3,2,1]

member.sort() #使列表中的元素按升序排序

memeber.sort(reverse=True) #使列表中的元素按降序排序，默认reverse=Flase

PS：分片，即member[:]是拷贝出一个新的列表，对原列表操作后不影响拷贝出的新列表。

而member1[]=member[]仅仅是添加一个新的列表名指向member[]，对原列表所进行的操作会影响member1[]。

tuple #元组函数，元组是不可以修改的列表，只能用切片的方式来更新和删除

sorted(numbers) #升序排序

list(reversed(numbers)) #降序排序

list(enumerate(numbers))#为每个元素的索引序号与元素组成元组并构成序列[(0,)(1,)(2,)(3,)(4,)]

list(zip(a,b)) #把a，b两个列表里的元素一一对应组成元组并构成序列 [(,)(,)(,)(,)]

pow(x,y) #返回x的y次幂

bin(x) #十进制转换为二进制

global x #在局部变量中x修改后的值赋值到全局变量中

|  |  |
| --- | --- |
| **符号** | **说明** |
| %c | 格式化字符及其 ASCII 码 # '%c' % 97 输出 a ASCII码数值转换字符 |
| %s | 格式化字符串 |
| %d | 格式化整数 |
| %o | 格式化无符号八进制数 |
| %x | 格式化无符号十六进制数 |
| %X | 格式化无符号十六进制数（大写） |
| %f | 格式化浮点数字，可指定小数点后的精度 |
| %e | 用科学计数法格式化浮点数 |
| %E | 作用同 %e，用科学计数法格式化浮点数 |
| %g | 根据值的大小决定使用 %f 或 %e |
| %G | 作用同 %g，根据值的大小决定使用 %f 或者 %E |

|  |  |
| --- | --- |
| **符号** | **说明** |
| m.n | m 是显示的最小总宽度，n 是小数点后的位数 |
| - | 用于左对齐 |
| + | 在正数前面显示加号（+） |
| # | 在八进制数前面显示 '0o'，在十六进制数前面显示 '0x' 或 '0X' |
| 0 | 显示的数字前面填充 '0' 取代空格 |

|  |  |  |
| --- | --- | --- |
| **符号** | **说明** | |
| \' | 单引号 | |
| \" | 双引号 | |
| \a | 发出系统响铃声 | |
| \b | 退格符 | |
| \n | 换行符 | |
| \t | 横向制表符（TAB） | |
| \v | 纵向制表符 | |
| \r | 回车符 | |
| \f | 换页符 | |
| \o | 八进制数代表的字符 | |
| \x | 十六进制数代表的字符 | |
| \0 | 表示一个空字符 | |
| \\ | 反斜杠 | |
| **函数** | | **说明** |
| capitalize() | | 把字符串的第一个字符改为大写 |
| casefold() | | 把整个字符串的所有字符改为小写 |
| center(width) | | 将字符串居中，并使用空格填充至长度 width 的新字符串 |
| count(sub[, start[, end]]) | | 返回 sub 在字符串里边出现的次数，start 和 end 参数表示范围，可选。 |
| encode(encoding='utf-8', errors='strict') | | 以 encoding 指定的编码格式对字符串进行编码。 |
| endswith(sub[, start[, end]]) | | 检查字符串是否以 sub 子字符串结束，如果是返回 True，否则返回 False。start 和 end 参数表示范围，可选。 |
| expandtabs([tabsize=8]) | | 把字符串中的 tab 符号（\t）转换为空格，如不指定参数，默认的空格数是 tabsize=8。 |
| find(sub[, start[, end]]) | | 检测 sub 是否包含在字符串中，如果有则返回索引值，否则返回 -1，start 和 end 参数表示范围，可选。 |
| index(sub[, start[, end]]) | | 跟 find 方法一样，不过如果 sub 不在 string 中会产生一个异常。 |
| isalnum() | | 如果字符串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。 |
| isalpha() | | 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。 |
| isdecimal() | | 如果字符串只包含十进制数字则返回 True，否则返回 False。 |
| isdigit() | | 如果字符串只包含数字则返回 True，否则返回 False。 |
| islower() | | 如果字符串中至少包含一个区分大小写的字符，并且这些字符都是小写，则返回 True，否则返回 False。 |
| isnumeric() | | 如果字符串中只包含数字字符，则返回 True，否则返回 False。 |
| isspace() | | 如果字符串中只包含空格，则返回 True，否则返回 False。 |
| istitle() | | 如果字符串是标题化（所有的单词都是以大写开始，其余字母均小写），则返回 True，否则返回 False。 |
| isupper() | | 如果字符串中至少包含一个区分大小写的字符，并且这些字符都是大写，则返回 True，否则返回 False。 |
| join(sub) | | 以字符串作为分隔符，插入到 sub 中所有的字符之间。 |
| ljust(width) | | 返回一个左对齐的字符串，并使用空格填充至长度为 width 的新字符串。 |
| lower() | | 转换字符串中所有大写字符为小写。 |
| lstrip() | | 去掉字符串左边的所有空格 |
| partition(sub) | | 找到子字符串 sub，把字符串分成一个 3 元组 (pre\_sub, sub, fol\_sub)，如果字符串中不包含 sub 则返回 ('原字符串', '', '') |
| replace(old, new[, count]) | | 把字符串中的 old 子字符串替换成 new 子字符串，如果 count 指定，则替换不超过 count 次。 |
| rfind(sub[, start[, end]]) | | 类似于 find() 方法，不过是从右边开始查找。 |
| rindex(sub[, start[, end]]) | | 类似于 index() 方法，不过是从右边开始。 |
| rjust(width) | | 返回一个右对齐的字符串，并使用空格填充至长度为 width 的新字符串。 |
| rpartition(sub) | | 类似于 partition() 方法，不过是从右边开始查找。 |
| rstrip() | | 删除字符串末尾的空格。 |
| split(sep=None, maxsplit=-1) | | 不带参数默认是以空格为分隔符切片字符串，如果 maxsplit 参数有设置，则仅分隔 maxsplit 个子字符串，返回切片后的子字符串拼接的列表。 |
| splitlines(([keepends])) | | 在输出结果里是否去掉换行符，默认为 False，不包含换行符；如果为 True，则保留换行符。。 |
| startswith(prefix[, start[, end]]) | | 检查字符串是否以 prefix 开头，是则返回 True，否则返回 False。start 和 end 参数可以指定范围检查，可选。 |
| strip([chars]) | | 删除字符串前边和后边所有的空格，chars 参数可以定制删除的字符，可选。 |
| swapcase() | | 翻转字符串中的大小写。 |
| title() | | 返回标题化（所有的单词都是以大写开始，其余字母均小写）的字符串。 |
| translate(table) | | 根据 table 的规则（可以由 str.maketrans('a', 'b') 定制）转换字符串中的字符。 |
| upper() | | 转换字符串中的所有小写字符为大写。 |
| zfill(width) | | 返回长度为 width 的字符串，原字符串右对齐，前边用 0 填充。 |