字符串序列操作：

例：程序string\_test/string\_test1.py

str1 = 'Spam'  
print(len(str1))  
print(str1[0])  
print(str1[-1]) # 最后一个  
print(str1[-2]) # 倒数第2个  
print()  
  
# 字符串切片  
print(str1[1:3]) # pa  
print(str1[1:]) # pam  
print(str1[:-1]) # Spa  
print(str1[:]) # Spam  
  
print(str1 + 'xyz') # Spamxyz  
print(str1 \* 2) # SpamSpam  
  
# 字符串为不可变对象  
# str1[0] = 'z' # Error

字符串方法：

程序string\_test/string\_test2.py

s = 'Spam'  
print('position ', s.find('pa'))  
  
# 字符串具有不可变性  
s1 = s.replace('pa', 'XYZ')  
print(s) # Spam  
print(s1) # SXYZm  
  
line = 'aaa,bbb,ccccc,dd'  
print(line.split(',')) # 返回列表  
  
# 格式化输出  
print('%s, eggs, and %s' % ('spam', 'SPAM!')) # spam, eggs, and SPAM!  
print('{0}, eggs, and {1}'.format('spam', 'SPAM!')) # spam, eggs, and SPAM!  
print()  
  
# 查找字符串的方法  
print(dir(s))

输出为：

position 1

Spam

SXYZm

['aaa', 'bbb', 'ccccc', 'dd']

spam, eggs, and SPAM!

spam, eggs, and SPAM!

字符串的切片扩展

例：string\_test/string\_test3.py

s = 'abcdefghijklmnop'  
  
# 从2取到10，步长为2  
# 取1,3,5,7,9  
print(s[1:10:2]) # bdfhj  
  
s1 = 'hello'  
# 负值步长，从尾到头  
print(s1[::-1]) # olleh  
  
s2 = 'abcdefg'  
# 取值范围为cdef，从右至左  
#步长为2  
print(s2[5:1:-2]) # fd

字符串函数：

例：string\_test/string\_test4.py

print('spam', repr('spam')) # spam 'spam'  
  
s = 'xxxxSPAMxxxxSPAMxxxx'  
where = s.find('SPAM')  
print(where) # 4  
s1 = s[:where] + 'EGGS' + s[(where+4):]  
print(s1) # xxxxEGGSxxxxSPAMxxxx  
print()  
s2 = s.replace('SPAM', 'EGGS')  
print(s2) # xxxxEGGSxxxxEGGSxxxx  
print()  
s3 = s.replace('SPAM', 'EGGS', 1)  
print(s3) # xxxxEGGSxxxxSPAMxxxx

字符串格式化：老版

例：string\_test/string\_test5.py

# Python中的格式化类似C语言中的格式化  
x = 1234  
print('integers:...%d...' % x) # integers:...1234...  
  
# -6d，从左对齐，小于6位，用空格补足低位  
print('integers:...%-6d...' % x) # integers:...1234 ...  
  
# 06d，小于6位，高位补0  
print('integers:...%06d...' % x) # integers:...001234...  
  
y = 1.23456789  
# -6.2f，小数点后保留2位有效数字，从左对齐，小于6位，用空格补足低位  
print('integers:...%-6.2f...' % y) # integers:...1.23 ...  
  
# 05.2f，小数点后保留2位，小于5位，高位补0  
print('integers:...%05.2f...' % y) # integers:...01.23...  
print()  
  
# 基于字典的格式化  
reply = """Greetings...  
Hello %(name)s!  
Your age squared is %(age)s"""  
values = {'name': 'Bob', 'age': 40}  
print(reply % values)

输出为：

Greetings...

Hello Bob!

Your age squared is 40

字符串格式化：

string\_test/string\_test6.py

import sys  
  
template = '{0}, {1} and {2}'  
print(template.format('spam', 'ham', 'eggs')) # spam, ham and eggs  
template = '{motto}, {pork} and {food}'  
print(template.format(pork='ham', motto='spam', food='eggs')) # spam, ham and eggs  
template = '{motto}, {0} and {food}'  
print(template.format('ham', motto='spam', food='eggs')) # spam, ham and eggs  
print()  
x = '{motto}, {0} and {food}'.format(42, motto=3.14, food=[1, 2])  
print(x) # 3.14, 42 and [1, 2]  
print(x.split(' and ')) # ['3.14, 42', '[1, 2]']  
print(x.replace('and', 'but under no')) # 3.14, 42 but under no [1, 2]  
  
# My laptop runs win32  
print('My {config[spam]} runs {sys.platform}'.format(sys=sys, config={'spam': 'laptop'}))  
print()  
# {0:10}, 0代表第一个参数，10字符宽  
# spam = 123.4567  
print('{0:10} = {1:10}'.format('spam', 123.4567))  
  
# {0:>10}:第一个参数，10字符宽，右对齐  
# {1:<10}：第二个参数，10字符宽，左对齐  
# spam = 123.4567  
print('{0:>10} = {1:<10}'.format('spam', 123.4567))  
  
# win32 = laptop  
print('{0.platform:>10} = {1[item]:<10}'.format(sys, {'item': 'laptop'}))  
print()  
  
# 3.141590e+00, 3.142e+00  
print('{0:e}, {1:.3e}'.format(3.14159, 3.14159))