

### 例 3.1 字符串运算符示例

```
1 a, b="study", "python"
2 print(a+b) # 输出: studypython
3 print(a*3) # 输出: studystudystudy
4 print("on" in b) # 输出: True
5 print(r"进入目录: c:\\python\\study") # 输出: 进入目录: c:\\python\\study
6 print(f"进入目录: c:\\python\\study") # 输出: 进入目录: c:\python\study
```

### #step 为正数的例子，正步长则往右方向找字符

#start 必须在 end 下标的左边，因为它是从左往右开始截取的。否则结果为空字符串

#如果 start 缺省，start 下标为首字符位置;如果 stop 缺省，默认为尾字符位置

```
s='0123456789'
```

```
print(s[1:8:2],s[1:-2:2],s[-9:8:2],s[-9:-2:2]) # 结果为: 1357 1357 1357 1357
```

```
print(s[:8:2], s[:-2:2]) # 结果为: 0246 0246 。start=1, 默认
```

```
print(s[2::2],s[-8::2]) # 结果为: 2468 2468
```

```
print(s[::2]) # 结果为: 02468
```

### #step 为负数的例子，负步长则往左方向找字符

#start 必须在 end 下标的右边，因为它是从右往左开始截取的。否则结果为空字符串

#如果 start 缺省，start 下标为尾字符位置;如果 stop 缺省，默认为首字符位置

```
s='0123456789'
```

```
print(s[8:2:-2],s[8:-8:-2],s[-2:2:-2],s[-2:-8:-2]) # 结果为: 864 864 864 864
```

```
print(s[:4:-2], s[:-6:-2]) # 结果为: 975 975
```

```
print(s[8::-2],s[-2::-2]) # 结果为: 86420 86420
```

```
print(s[::-1]) # 结果为: 9876543210
```

### #没有步长参数 step 的示例

```
s='python! '
```

```
print(s[0:6],s[0:-1],s[-7:6],s[-7:-1]) # 结果为: python python python python
```

```
print(s[2:],s[2:],s[:]) # 结果为: py thon! Python
```

### # 字符串函数处理

```
str='python Book'
```

```
# 全部大写: str.upper()
```

```
# 全部小写: str.lower()
```

```
# 大小写互换: str.swapcase()
```

```
# 首字母大写，其余小写: str.capitalize()
```

```
# 首字母大写: str.title()
```

```
print("{0}.lower( )='{1}'".format('str',str.lower()))
```

```
print("{0}.upper( )='{1}'".format('str',str.upper()))
```

```
print("{0}.swapcase( )='{1}'".format('str',str.swapcase()))
print("{0}.capitalize( )='{1}'".format('str',str.capitalize()))
print("{0}.title( )='{1}'".format('str',str.title()))
```

# 字符串搜索相关

```
# 搜索指定字符串，没有返回-1: str.find('t')
# 指定起始位置搜索: str.find('t',start)
# 指定起始及结束位置搜索: str.find('t',start,end)
# 从右边开始查找: str.rfind('t')
# 搜索到多少个指定字符串: str.count('t')
# 上面所有方法都可用 index 代替，不同的是使用 index 查找不到会抛异常，而 find 返回-1
print("str.find('nono')={}".format(str.find('nono')))
print("str.find('th')={}".format(str.find('th')))
print("str.find('th',1)={}".format(str.find('th',1)))
print("str.find('th',1,5)={}".format(str.find('th',1,5)))
print("str.rfind('th')={}".format(str.rfind('th')))
print("str.count('th')={}".format(str.count('th')))
```

# 字符串替换相关

```
# 替换 old 为 new: str.replace('old','new')
# 替换指定次数的 old 为 new: str.replace('old','new',maxReplaceTimes)
print("str.replace('o', '*')={}".format(str.replace('o', '*')))
print("str.replace('o', '*',1)={}".format(str.replace('o', '*',1)))
```

# 字符串去空格及去指定字符

```
# 去两边空格: str.strip()
# 去左空格: str.lstrip()
# 去右空格: str.rstrip()
# 去两边字符串: str.strip('d')，相应的也有 lstrip，rstrip
s=' python String function '
print("s.strip( )='{1}'".format(s.strip()))
str='python String function'
print("s.strip('p')='{1}'".format(s.strip('p')))
```

# 按指定字符分割字符串为数组: str.split(' ')

# 默认按空格分隔

```
print("str.split( )='{1}'".format(str.split()))
str1='a-b-c-de'
print("'a-b-c-de'.split('-')='{1}'".format(str1.split('-')))
```

# 字符串判断相关

```
# 是否以 start 开头: str.startswith('start')
# 是否以 end 结尾: str.endswith('end')
# 是否全为字母或数字: str.isalnum()
```

```
# 是否全字母: str.isalpha()
# 是否全数字: str.isdigit()
# 是否全小写: str.islower()
# 是否全大写: str.isupper()

print("str.startswith('python')={}".format(str.startswith('python')))
print("str.endswith('python')={}".format(str.endswith('python')))
print("str.isalnum( )={}".format(str.isalnum()))
print("str.isalpha( )={}".format(str.isalpha()))
print("str.isupper( )={}".format(str.isupper()))
print("str.islower( )={}".format(str.islower()))
print("'python123'.isdigit( )={}".format('python123'.isdigit()))
print("'3423'.isdigit( )={}".format('3423'.isdigit()))

# 格式化相关,对齐, 指定宽度
# 右对齐, 左边不够用空格补齐: str.ljust(width)
# 左对齐, 右边不够用空格补齐: str.rjust(width)
# 中间对齐, 两边不够用空格补齐: str.center(width)
# 右对齐, 左边不足用 0 补齐:str.zfill(width)
s='abcd'
print("'abcd'.ljust(10)='{}'.format(s.ljust(10)))
print("'abcd'.rjust(10)='{}'.format(s.rjust(10)))
print("'abcd'.center(10)='{}'.format(s.center(10)))
print("'abcd'.zfill(10)='{}'.format(s.zfill(10)))
```