String底层用unicode编码字符串，双字节

**一、创建字符串**

"string" 字符串字面量

'string' 字符串字面量

**二、字符串遍历**

[index] 返回对应索引字符

string.count(str <, begin, length>) 返回指定子串出现的次数

string.find(str <, begin, length>) 返回指定子串对应索引，未找到时-1

string.index(str <, begin, length>) 同find，未找到时抛异常

string.rfind(str <, begin, length>) 同find，区别从右向左

string.rindex(str <, begin, length>) 同index，区别从右向左

max(str) 返回最大的字母

min(str) 返回最小的字母

**三、字符串判断**

"a" in "abc" 判断是否包含

"a" not in "bcd" 判断不包含

string.endswith(obj <, begin, length>) 判断以指定子串为结尾

string.isalnum() 判断只包含字母和数字

string.isalpha() 判断只包含字母

string.isdecimal() 判断只包含十进制数字

string.isdigit() 判断只包含数字

string.islower() 判断字母都为小写

string.isnumeric() 判断只包含数字

string.istitle() 判断所有单词首字母大写

string.isupper() 判断字母都为大写

string.startwith(obj <, begin, length>) 判断以指定子串为开头

**四、字符串操作**

**1内容操作**

+ 字符串连接

\* 重复输出字符串

[index:length] 返回截取字符串

string.join(str\_arr) 用string拼接字符串数组

string.replace(origin, target <, limit>) 替换字符串，不超过limit次数

string.split(str <, limit>) 以指定字符分割字符串，不超过limit次数

string.splitlines(keepend) 以换行分割字符串，keepend是否保留换行符

**2格式操作**

大小写

string.capitalize() 将首字母大写

string.lower() 将字母转化小写

string.swapcase() 翻转字母大小写

string.title() 将单词首字母大写

string.upper() 将字母转化为大写

补全字符

string.center(int) 原字符串按指定长度居中

string.expandtabs(tabsize=8) 转化tab符为空格，默认8个空格

string.ljust(width) 以空格补全尾部字符串，width为总长度

string.rjust(width) 以空格补全头部字符串，width为总长度

string.zfill(width) 以0补全头部字符串，width为总长度

"string: %s number: %d" % ("abc", 123) # => "string: abc; number: 123"

注意：格式化符号参见C语言数据类型

"{} {}".format("abc", "123") # => "abc 123"

"{1} {0}".format("abc", "123") # => "123 abc"

string.lstrip() 去掉头部空字符

string.rstrip() 去掉尾部空字符

**五、字符编码**

ord(char) 返回字符对应unicode索引

chr(int) 返回unicode索引对应的字符

string.encode(encoding) 编码字符串为字节码

string.decode(encoding) 解码字节码为字符串