# [python3中str（字符串）的常用方法示例](http://blog.csdn.net/zhongbeida_xue/article/details/51211506)

字符串是 Python 中最常用的数据类型。我们可以使用引号('或")来创建字符串。

创建字符串很简单，只要为变量分配一个值即可

**1.capitalize()** 将字符串的第一个字符改为大写

s='i love you'  
print(s.capitalize())

**2.casefold()** 将字符串所有字符改为小写，比lower()支持范围更广，支持外语，我也不懂，我只认识中文

S='I LOVE YOU'  
print(S.casefold())

**3.center(width)** 将字符串居中，并用空格将字符串填充至width长度，空格均匀分布在两侧，当width<len(s)时没有效果

S='I LOVE YOU'  
print(S.center(20))  
print(len(S.center(20)))

**4.count(sub[,start[,end]])** 返回sub在字符串里出现的次数，start，end为可选参数，决定范围

S='I LOVE YOU'  
print(S.count('O'))

**5.encode(encoding='utf-8',errors='strict')** 以encoding指定的编码格式对字符串进行编码

S='I LOVE YOU'  
print(S.encode(encoding='utf-8',errors='strict'))

**6.endswith(sub[,start[,end]])** 检查字符串是否以sub结尾，是返回True,否返回False，start，end为可选参数，决定范围

S='I LOVE YOU'  
print(S.endswith('U'))

**7.expandtabs([tabsize=8])** 把字符串的tab字符（\t）转化为空格，如不指定tabsize,默认为8个空格

s='\t i love you'  
print(s.expandtabs())  
print(len(s.expandtabs()))

**8.find(sub[,start[,end]])** 检测sub是否在字符串中，如果在则返回index，否则返回-1，start，end为可选参数，决定范围。这里返回的是sub的index，同时start,end都是包含的。

s='i love you'  
print(s.find('y'))

**9.index(sub[,start[,end]])**类似find()，不同在于如果sub不在字符串中，返回的不是-1而是异常

s='i love you'  
print(s.index('y'))

10.**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False

s='i love you'  
print(s.isalnum())

11.**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False

s='i love you'  
print(s.isalpha())

12.**isdigit()** 如果字符串只包含数字则返回True,否则返回False

s='i love you'  
print(s.isdigit())

13.**isdecimal()** 如果字符串只包含十进制数字则返回True，否则返回False

s='i love you'  
print(s.isdecimal())

14.**islower()** 如果字符中至少包含一个能区分大小写的字符，并且这些字符都是小写则返回True,否则返回Flase

s='i love you'  
print(s.islower())

15.**isupper()**如果字符中至少包含一个能区分大小写的字符，并且这些字符都是大写则返回True,否则返回Flase

s='i Love you'  
print(s.isupper())

16.**isnumeric()** 如果字符串只包含数字字符，则返回True,否则返回False

初一看感觉和isdigit()是一样的，但是：汉字的数字，同样的还有罗马数字等是返回true的

s='三'  
print(s.isidentifier())

17.**isidentifier()**判断字符串是否包含该语言的保留字

s='i love you'  
print(s.isidentifier())

18.**isprintable()** 判断字符串中所有的字符串都是可以通过repr表示成字符串，或者字符串是空的，都返回True,否则返回False

s='\n i love you'  
print(s.isprintable()) #false

print('ces'.islower())

print(chr(1000000).isprintable()) #false

这里使用一个超出字符编码范围的数字去转化成字符，测试其是否可以打印，显然，答案是不行。

19.**isspace()**判断字符串，至少有一个字符的字符串中所有字符是否都是空格，不是则返回False

s='i love you'  
print(s.isspace())

print(' '.isspace())

20.**istitle()**如果字符串中所有的单词拼写首字母是否为大写，且其他字母为小写则返回 True，否则返回 False.

s='Ia'  
print(s.istitle())

 21.**isupper()** 判断字符串是否全部是大写

S='I LOVE YOU'  
print(S.isupper())

 22.**join()** 返回一个用指定字符串分隔的字，或者是将指定字符加入到另一个字符中。

s='i love you'  
print(','.join(s))

 23.**lower()**返回的是指定字符串的拷贝，并转化成小写

S='I LOVE YOU'  
print(S.lower())

 24.**ljust()** 可以指定宽度，以及填充字符串，返回的是按宽度，填充字符串格式化后的左对齐的字符串。

print('a'.ljust(10, 'A'))

25.**partition():**在指定字符串中查找sep,如果找到了返回该字符前面的部分，sep,及后面的部分，

如果没找到则返回sep及两个空字符中，类似于split,但又有不同

print('ssaafdaf'.partition('f'))

 26.**replace()** ,用指定字符串替换指定字符串，如果不指定替换次数，仅替换第一个。

print('ssfaafda'.replace('f','m'))

27.**rfind()**: 返回指定子串的最高索引，如果没找到则返回-1，可以指定要开始替换的起始，结束位置。

print('ssfaafda'.rfind('f'))

 28.**rindex(),**与上面的rfind一样，只是如果没找到不是返回-1，而是触发错误

print('ssfaafda'.rindex('f'))

 29.**rjust()**;与ljust()相对应

print('ssfaafda'.rjust(15))

 30.**rpartition()**与partition一样，但是是从右边开始

print('ssfaafda'.rpartition('f'))

31.**rsplit()**，与split作用相同，但是从右侧开始

print('ssfaafda'.rsplit('f'))

 但是讲真，如果不仔细考虑你是不会发现它与split有什么不同的，只有当你指定了最大切割次数时才会有效果。

print('ssfaafda'.rsplit('f',1))

32.**rstrip(),** 从右侧移除指定字符

print(' ssfaafda '.rstrip())

 33.**split()**， 按指定字符串对目标字符串进行切割，可以指定切割次数

print('this is a test'.split('i', 1))

 感觉它与partition的不同在于它返回的结果中移除了指定的字符串

34.**splitlines(),**返回字符串的行，按换行符切割，如果没指定keepends=True，则会将其从结果中移除

print('this is a string\n this is a test'.splitlines())

print('this is a string\n this is a test'.splitlines(keepends=True))

35.**startswith(str)**,判断字符串是否以某个字符开头

print('this is a string\n this is a test'.startswith('T'))

36.**strip()** 移除字符串两侧的指定字符串，默认移除空格，需要注意的是可以指定多个字符

print('this is a test'.strip('t'))

37.**swapcase()** 转换大小写

print('this is A test'.swapcase())

38.**title()**， 标题格式，就是首字母大写，其它字符小写

print('this is a test'.title())

39.**upper()**,将字符全部转成大写

print('this is a test'.upper())

40.**zfill()**,这里的z指zero，用0将字符填充到指定长度

print('aa'.zfill(10))

41. **maketrans(intab, outtab) ，**方法用于创建字符映射的转换表，对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串表示转换的目标。

两个字符串的长度必须相同，为一一对应的关系。

**translate(table，delete)，** 方法根据参数table给出的表(包含 256 个字符)转换字符串的字符,要过滤掉的字符放到 **deletechars** 参数中。

intab = "aeiou"  
outtab = "12345"  
trantab = str.maketrans(intab, outtab)  
str = "this is string example....wow!!!"  
print (str.translate(trantab))

# 制作翻译表  
bytes\_tabtrans = bytes.maketrans(b'abcdefghijklmnopqrstuvwxyz', b'ABCDEFGHIJKLMNOPQRSTUVWXYZ')  
# 转换为大写，并删除字母o  
print(b'runoob'.translate(bytes\_tabtrans, b'o'))

# python3中dict（字典）的常用方法示例

**1.clear()：**清空字典内容

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.clear())

2. **copy()：拷贝字典**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
stu2 = stu.copy()  
print(stu2)

3. **fromkeys(seq: Sequence[\_T], value: \_S)：指定一个列表，把列表中的值作为字典的key,生成一个字典**

print(dict.fromkeys(name))  
print(dict.fromkeys(name,25)) #指定默认值

4. **get(key)：指定key，获取对应的值**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.get('num2'))

5. **items()：返回由“键值对组成元素“的列表**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(list(stu.items()))

6. **keys()：获取字典所有的key**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(list(stu.keys()))

7.**values()：获取字典所有的value**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.values())

8. **pop(key)：获取指定key的value，并在字典中删除**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.pop('num1'))

9. **popitem()：随机获取某个键值对，并在字典中删除**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.popitem())

10. **setdefault(获取指定key的value，如果key不存在，则创建)**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.setdefault('num4'))  
print(stu)

11．**update({key:value})：添加键 - 值对到字典**

stu = {  
 'num1':'Tom',  
 'num2':'Lucy',  
 'num3':'Sam',  
}  
print(stu.update({'num4':'Ben'}))  
print(stu.update({'num2':'Ben'}))