**unicodedata模块**

**str = unicodedata.category(char)**

返回char在Unicode中的类型，下面列出一部分

一共分为七大类：Letter, Mark, Number, Punctuation, Symbol, Seperator, Other，每个大类下面还有进一步的二级分类

[Cc] Other, Control  
[Cf] Other, Format  
[Cn] Other, Not Assigned (no characters in the file have this property)  
[Co] Other, Private Use  
[Cs] Other, Surrogate  
[LC] Letter, Cased  
[Ll] Letter, Lowercase  
[Lm] Letter, Modifier  
[Lo] Letter, Other  
[Lt] Letter, Titlecase  
[Lu] Letter, Uppercase  
[Mc] Mark, Spacing Combining  
[Me] Mark, Enclosing  
[Mn] Mark, Nonspacingz 组合字符  
[Nd] Number, Decimal Digit  
[Nl] Number, Letter  
[No] Number, Other  
[Pc] Punctuation, Connector  
[Pd] Punctuation, Dash  
[Pe] Punctuation, Close  
[Pf] Punctuation, Final quote (may behave like Ps or Pe depending on usage)  
[Pi] Punctuation, Initial quote (may behave like Ps or Pe depending on usage)  
[Po] Punctuation, Other  
[Ps] Punctuation, Open  
[Sc] Symbol, Currency  
[Sk] Symbol, Modifier  
[Sm] Symbol, Math  
[So] Symbol, Other  
[Zl] Separator, Line  
[Zp] Separator, Paragraph  
[Zs] Separator, Space

**str = unicodedata.normalize(form,unistr)**

把Unicode编码的字符串标准化，确保字符串中所有字符在底层有相同的表示

form表示规范化形式，取值可能为'NFD'，'NFKD'，'NFC'，'NFKC'

NFD和NFKD将可能的组合字符进行拆解

NFC和NFKC将可为组成的字符进行组合

该函数有3个作用(已知)

·将组合字符拆解出来(form='NFD')

·将ç变为c

·去除字符串首尾的某些陌生的unicode字符

s = 'réparai.'

s[1] >>> 'é'

str = ''.join(c for c in unicodedata.normalize('NFD', s))

str >>> 'réparai.'

str[2] >>>'́'

unicodedata.category(str[2]) >>> 'Mn'

str1 = ''.join(c for c in unicodedata.normalize('NFD', s) if unicodedata.category(c)!='Mn')

str1 >>>'reparai.'