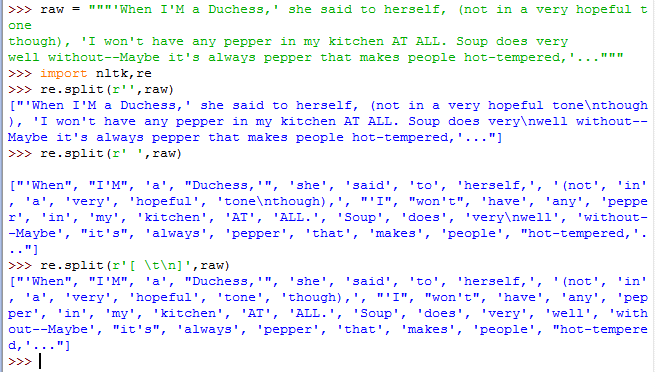
分词是将字符串切割成可识别的构成语言数据的语言单元。

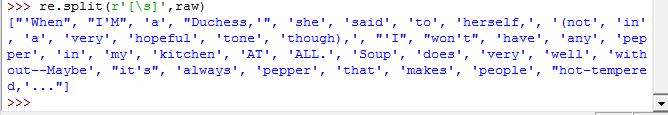
文本分词的简单方法

可以使用split()在空格符出分割原始文本。

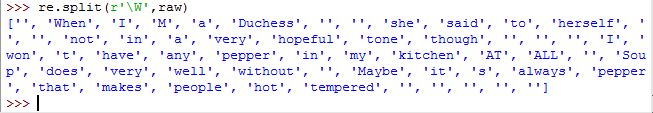


光用空格来分词是不够的，就会出现上面tone\nthough这样的错误，所以制表符，换行符都必须考虑。

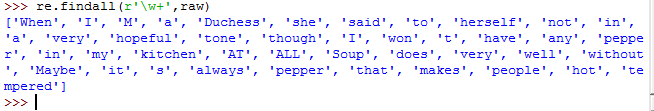
另外，re库中\s表示所有空白字符，可以起到同样效果，\S是\s的补集



\w相当于[a-zA-Z0-9\_] \W为其补集



用findall可以去除空格



**Table 3.4**:

Regular Expression Symbols

|  |  |
| --- | --- |
| Symbol | Function |
| \b | Word boundary (zero width) |
| \d | Any decimal digit (equivalent to [0-9]) |
| \D | Any non-digit character (equivalent to [^0-9]) |
| \s | Any whitespace character (equivalent to [ \t\n\r\f\v]) |
| \S | Any non-whitespace character (equivalent to [^ \t\n\r\f\v]) |
| \w | Any alphanumeric character (equivalent to [a-zA-Z0-9\_]) |
| \W | Any non-alphanumeric character (equivalent to [^a-zA-Z0-9\_]) |
| \t | The tab character |
| \n | The newline character |

nltk的正则表达式分词器

函数nltk.regexp\_tekenize()与re.findall()类似，然而nltk.regexp\_tekenize()效率更高，且避免了括号特殊处理的需要。为了增强可读性，将正则表达式分几行写，（？x）告诉Python去掉嵌入的空白字符和注释。

**>>> text = 'That U.S.A. poster-print costs $12.40...'  
>>> pattern = r'''(?x) # set flag to allow verbose regexps  
... ([A-Z]\.)+ # abbreviations, e.g. U.S.A.  
... | \w+(-\w+)\* # words with optional internal hyphens  
... | \$?\d+(\.\d+)?%? # currency and percentages, e.g. $12.40, 82%  
... | \.\.\. # ellipsis  
... | [][.,;"'?():-\_`] # these are separate tokens; includes ], [  
... '''  
>>> nltk.regexp\_tokenize(text, pattern)**

**['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']**