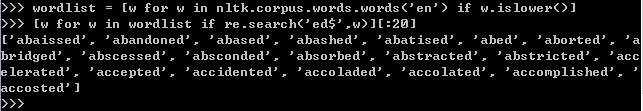
在Python中使用正则表达式，需要使用import re导入re函数库，还需要用于搜索的词汇链表。

$用来表示以XX结尾。

import re,nltk

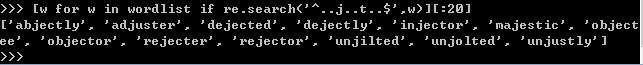
wordlist = [w for w in nltk.corpus.words.words('en') if w.islower()]

[w for w in wordlist if re.search('ed$',w)][:20]

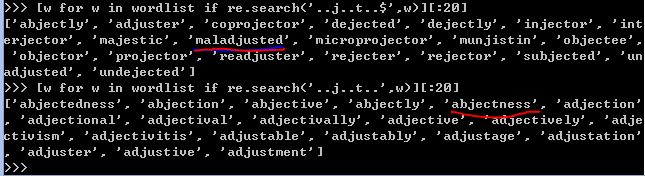


通配符 . 可以用来匹配任何单个字符。

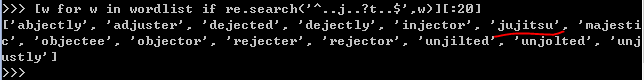
[w for w in wordlist if re.search('^..j..t..$',w)][:20]



^表示匹配开始，$表示匹配结束，删去后，开始结尾是任意的

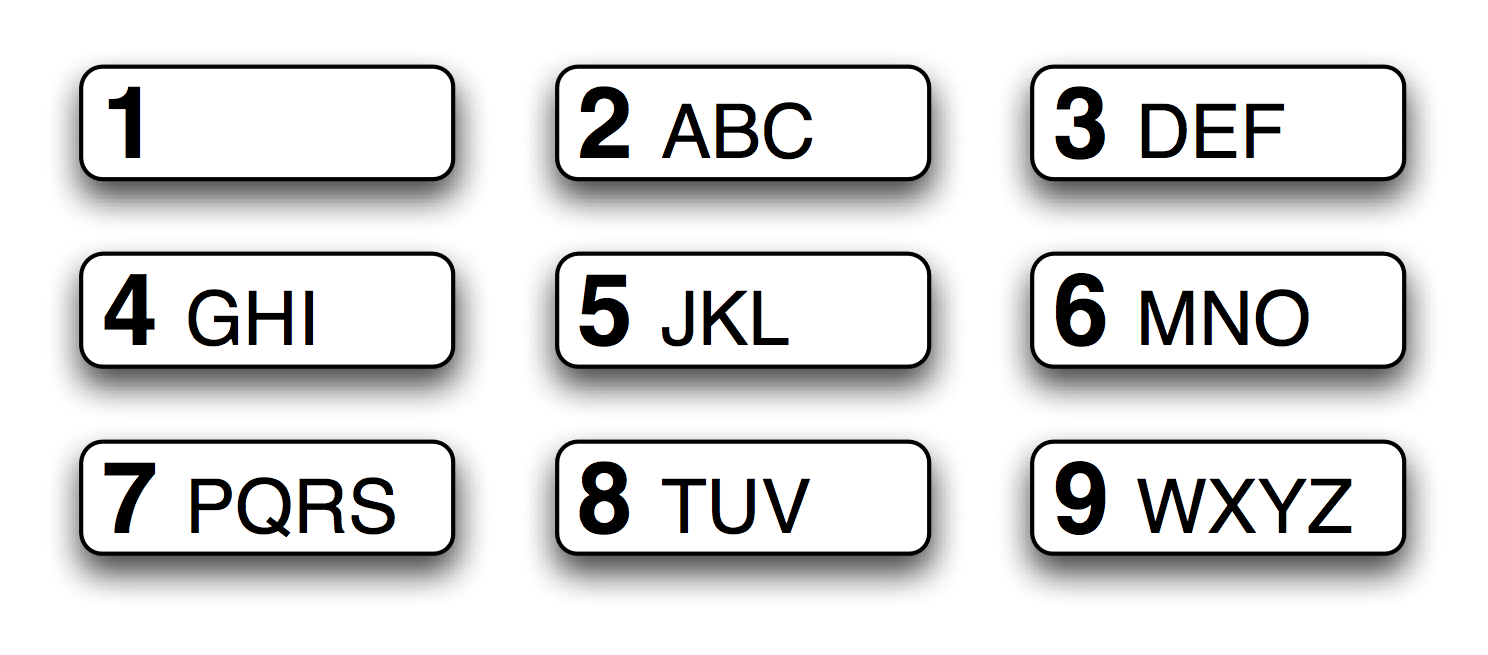


？表示前面的字符是可选的



范围与闭包

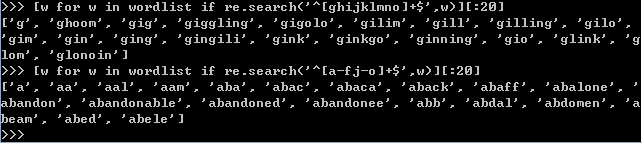
hole和golf在九宫格上的输入序列都是相同的



我们可以通过

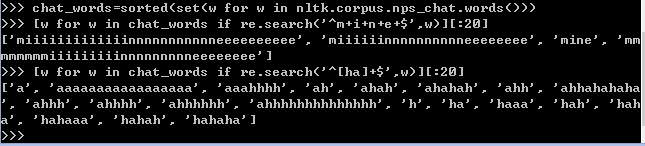


查看有哪些类似的词

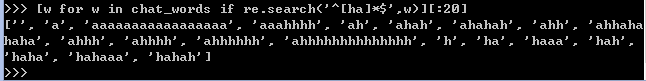


可以用-表示省略，后面加+表示闭包。

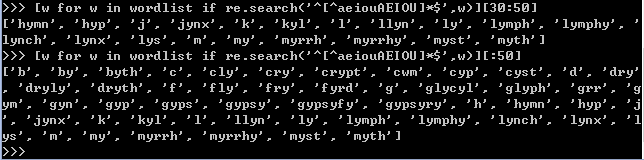
[]表示集合



使用\*闭包会有不同的结果



^也可以表示除去



+\表示一元闭包，{n}表示n元闭包，{n,m}表示n到m元闭包，|表示或者（）表示操作符范围

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **>>> wsj = sorted(set(nltk.corpus.treebank.words())) >>> [w for w in wsj if re.search('^[0-9]+\.[0-9]+$', w)] ['0.0085', '0.05', '0.1', '0.16', '0.2', '0.25', '0.28', '0.3', '0.4', '0.5', '0.50', '0.54', '0.56', '0.60', '0.7', '0.82', '0.84', '0.9', '0.95', '0.99', '1.01', '1.1', '1.125', '1.14', '1.1650', '1.17', '1.18', '1.19', '1.2', ...] >>> [w for w in wsj if re.search('^[A-Z]+\$$', w)] ['C$', 'US$'] >>> [w for w in wsj if re.search('^[0-9]{4}$', w)] ['1614', '1637', '1787', '1901', '1903', '1917', '1925', '1929', '1933', ...] >>> [w for w in wsj if re.search('^[0-9]+-[a-z]{3,5}$', w)] ['10-day', '10-lap', '10-year', '100-share', '12-point', '12-year', ...] >>> [w for w in wsj if re.search('^[a-z]{5,}-[a-z]{2,3}-[a-z]{,6}$', w)] ['black-and-white', 'bread-and-butter', 'father-in-law', 'machine-gun-toting', 'savings-and-loan'] >>> [w for w in wsj if re.search('(ed|ing)$', w)] ['62%-owned', 'Absorbed', 'According', 'Adopting', 'Advanced', 'Advancing', ...]**  **Table 3.3:**  **Basic Regular Expression Meta-Characters, Including Wildcards, Ranges and Closures**   |  |  | | --- | --- | | **Operator** | **Behavior** | | **.** | **Wildcard, matches any character** | | **^abc** | **Matches some pattern *abc* at the start of a string** | | **abc$** | **Matches some pattern *abc* at the end of a string** | | **[abc]** | **Matches one of a set of characters** | | **[A-Z0-9]** | **Matches one of a range of characters** | | **ed|ing|s** | **Matches one of the specified strings (disjunction)** | | **\*** | **Zero or more of previous item, e.g. a\*, [a-z]\* (also known as *Kleene Closure*)** | | **+** | **One or more of previous item, e.g. a+, [a-z]+** | | **?** | **Zero or one of the previous item (i.e. optional), e.g. a?, [a-z]?** | | **{n}** | **Exactly *n* repeats where n is a non-negative integer** | | **{n,}** | **At least *n* repeats** | | **{,n}** | **No more than *n* repeats** | | **{m,n}** | **At least *m* and no more than *n* repeats** | | **a(b|c)+** | **Parentheses that indicate the scope of the operators** | | |

用r’....’表示正则表达式以免\n,\b这些被解释为转行和退格

