

1.定界符 //表示正则的开始和结束

2.原子，正则中的最小匹配单位

2.1 可见原子-键盘输出后可以看得见的字符，如符号，英文，数字等

2.2 不可见原子-键盘输出后看不见的字符，如换行(\n)，回车(\r),制表符(\t) 垂直制表符(\v) 换页符(\f)

3.原子筛选

3.1 | 竖线两边任意匹配

3.2 [] 括号中任意一个原子，可以写区间,[a-z0-9]

3.3 [^] 括号中任意一个原子之外的原子

4.原子集合

4.1 . 任意字符

4.2 \d 任意数字 [0-9]

4.3 \D 任意非数字 [^0-9]

4.4 \w 任意数字，字母，下划线[0-9a-zA-Z_]

4.5 \W 任意非数字，字母，下划线[^0-9a-zA-Z_]

4.6 \s 空白符

4.7 \S 非空白符

5.量词-限定前面原子出现的次数

5.1 {n} 前面原子恰好出现n次

5.2 {n,} 前面原子最少出现n次

5.3 {n,m} 前面原子最少出现n次,最多出现m次

5.4 + 前面原子最少出现1次 {1,}

5.5 ? 前面原子最少出现0次,最多出现一次 {0,1}

5.6 * 前面原子最少出现0次 {0,}

6.贪婪模式、非贪婪模式

贪婪模式 尽可能多的匹配(默认)

```
'12345'.replace(/\d{2,4}/,'X');  
"X5"
```

非贪婪模式 尽可能少的匹配，在量词后添加?

```
'12345'.replace(/\d{2,4}?/, 'X');  
"X345"
```

7.边界控制

^匹配字符串开始的位置,限制必须以其后面的原子开头，前面不能有任何东西

\$匹配字符串结尾的位置，限制必须以其前面的原子结束，后面不能有任何东西

()把括号里面的正常表达匹配的结果当作一个原子来看

8.修正模式

i 忽略大小写

g 匹配所有的