## 正则笔记

1. 正则表达式简介

正则表达式是用于描述字符排列和匹配模式的一种语法规则。它主要用于字符串的分割、匹配、查找及替换操作。到目前为止，我们前面所用过的精确（文本） 匹配也是一种正则表达式。

在PHP中，正则表达式一般是由正规字符和一些特殊字符（类似于通配符）联合构成的一个文件模式的程序性描述。

PHP中，正则表达式有三个作用：

— 匹配：也常常用于从字符串中析取信息。

— 用新文本代替匹配文本。

— 将一个字符串拆分为一组更小的信息块。

一个正则表达式中至少包含一个原子。

1. PCRE提供的正则函数库要比POSIX提供的正则函数库功能和语法强大，并且PCRE正则函数库的语法支持更多特性。
2. Perl语言兼容的正则表达式处理函数

preg\_match() 进行正则表达式匹配

preg\_match\_all() 进行全局正则表达式匹配

preg\_replace() 执行正则表达式的搜索和替换

preg\_split() 用正则表达式分割字符串

preg\_grep() 返回与模式匹配的数组单元

Preg\_replace\_callback() 用回调函数执行正则表达式的搜索和替换

1. 定界符

在程序语言中，使用与Perl兼容的正则表达式，通常都需要将模式表达式放入定界符之间，如 “/”。

作为定界符常使用反斜线 “/”，如 “/apple/”。用户只要把需要匹配的模式内容放入定界符之间即可。作为定界的字符也不仅仅局限于 “/”。除了字母、数字和斜线“\”以外的任何字符都可以作为定界符，像 ‘#’、’|’、’!’等都可以的

1. 原子是正则表达式的最基本的组成单元，而且在每个模式中最少要少包含一个原子。原子是由所有那些未显示指定为原字符的打印和非打印字符组成，具体分为5类：
2. 普通字符作为原子：如a-z A-Z 0-9
3. 一些特殊字符和转义后元字符作为原子

如: \” \’ \\* \+ \. \?等

1. 一些非打印字符作为原子：如： \f \n \r \t \v \cx
2. 使用通用字符类型作为原子： 如: \d \D \w \W \s \S
3. 自定义原子表([]) 作为原子：如：’/[aj]sp/’ ‘/[^aj]sp/’
4. 正则表达式中常用的非打印字符

\cx ：匹配由x致命的控制字符。如\cM匹配一个Control-M或回车符。x的值必须为A-Z或a-z之一。

\f ：匹配一个换页符。等价于\x0c或者\cl

\n ：匹配一个换行符。等价于 \x0a 或 \cJ

\r ：匹配一个回车符。等价于\x0d或 \cM

\t ：匹配一个制表符。等价于\x09或\cI

\v ：匹配一个垂直制表符。等价于\x0b或\cK

\d ：匹配任意一个十进制数字，等价于[0-9]

\D ：匹配任意一个除十进制以外的字符，等价于[^0-9]

\s ：匹配任意一个空白符，等价于[\f\n\r\t\v]

\S ：匹配出空白符以外任何字符，等价于[^\f\n\r\t\v]

\w ：匹配任意一个数字、字母、下划线，等价于 [0-9a-zA-Z\_]

\W ：匹配一个除数字、字母或下划线以外的任意一个字符，等价于[^0-9a-zA-Z\_]

1. 元字符

\* ：匹配 0 次、1 次或多次其前面的原子

+ ：匹配 1 次、或多次其前的原子

？：匹配0次或1次其前面的原子

. ：匹配除了换行符外的任意一个字符

| ： 匹配两个或多个分支选择

{n} : 表示其前面的原子恰好出现n次

{n,} : 表示其前面的原子出现次数不小于n次

{n,m} ：表示其前面的原子最少出现n次，最多出现m次

^或\A ：匹配输入字符串的开始位置（或在多行模式下行的开头，即紧随一个换行符之后）

$或\Z ：匹配输入字符串的结束位置（或在多行模式下行的结尾，紧随一个换行符之前）

\b ：匹配单词的边界

\B ：匹配除单词边界以外的部分

[] ：匹配方括号中指定的任意一个原子

[^] ：匹配除方括号中的原子以外的任意一个字符

() ：匹配其整体为一个原子，即模式单元。可以理解为有多个单个原子组成的大原子

字符串边界限制 ：在某些情况下 ，需要对匹配方位进行限定。已获得更准确的匹配结果。 ^ $ 表示指定字符串的开始和结束

模式单元的作用：

1. 将括号中的内容作为一个整体来看待

例如：([\/\-\.])\d{2} 圆括号中的字符为一个整体

1. 可以进行分组，便于后向引用

当把正则表达式中的一部分放到圆括号中时，会在缓存中生成一个缓冲区（1-99），每个区块中存放的内容是对应编号的圆括号中匹配的内容，如果想使用对应括号中匹配的内容，需要使用 [\\n(n代表对应的区块编号)，如果想使用第一个括号中匹配的内容，则使用](\\\\n(n代表对应的区块编号)，如果想使用第一个括号中匹配的内容，则使用) [\\1，第二个括号中的内容](\\\\1，第二个括号中的内容) [\\n。。。以此类推](\\\\n。。。以此类推)(这就是后向引用)

模式匹配的优先级：

1. \ 转义字符
2. () (?:) (?=) [] 模式单元和原子表
3. \* + ? {n} {n,} {n,m} 重复匹配
4. ^ $ \b \B \A \Z 边界限制
5. | 模式选择
6. 模式修正符

i ：在和模式进行匹配时不区分大小写

m ：将字符串视为多行。默认的正则开始 “^” 和结束 “$” 将目标字符串作为单一的 “行” 字符。加上m后，那么开始和结束将会指字符串的每一行