Регулярные выражения

Общая задача механизма регулярных выражений - находить или не находить совпадения строки или ее части с шаблоном. В шаблон входят как обычные буквы, так и специальные символы. Каждый символ в отдельности называется литералом, каждый специальный символ называется метасимволом. В класс литералов входят все символы латиницы a, b, c, d и так далее до z, кириллицы а, б, … я, а так же цифры 0, 1, 2 и так до 9.

- *Обычные символы называются "литералами".*
- *Специальные символы называются "метасимволами".*
- *Литералы означают сами себя.*
- *Метасимволы предназначены для описания диапазона литералов, каких-то применимых к литералам условий, свойств литералов, их количества.*
- *Все литералы можно классифицировать, собрав их вместе по какому-то признаку.*

Функции обработки регулярных выражений:

- preg_replace - заменить
- preg_match - найти соответствие
- preg_split - разбить

Функция preg_match() реализует обращение к механизму обработки регулярных выражений, поиск совпадения в строке и возвращает совпадения в массив:

```
preg_match("шаблон_поиска", "строка_в_которой_проводится_поиск",
 массив_с_результами_поиска);
```

Чтобы указать поиск букв можно перечислить их все [abc....xyz], либо написать интервал [a-z], с русскими буквами [а-я], а так же с цифрами [0-9]. Но есть еще заглавные [A-Z], т.е. чтобы получить символьный класс со всеми буквами латинского алфавита надо поставить в шаблоне [a-zA-Z].

Квантификаторы

Один символьный класс может совпасть только с одним символом!

Условие: Найти в строках участки, которые состоят из четырех любых букв латинского алфавита, после которых следуют пять любых цифр.

```
abcd12345efg
fghi56789qwe
```

можно написать: [a-z][a-z][a-z][a-z][0-9][0-9][0-9][0-9][0-9].

Квантификатор - число, которое выражает количество символов в условии поиска. Упрощаем условие поиска при помощи квантификаторов: [a-z]{4}[0-9]{5}.

Каждый символьный класс описывает только один символ, количество схожих символов идущих подряд описывается квантификаторами.

[a-z]{1,3} означает, что подряд может идти от одного до трех букв латинского алфавита.
[a-z]{2,} означает, что может идти минимум 2 буквы латинского алфавита подряд.

[a-z]* означает, что подряд может идти сколь угодно букв латинского алфавита, в том числе ни одной, идентично [a-z]{0,} .

[a-z]+ означает, что обязательно подряд должна идти минимум одна буква латинского алфавита, но максимальное количество не указано, идентично [a-z]{1,}.

[a-z]? означает, что количество латинских букв не должно превышать 1, буква также может вообще отсутствовать, идентично [a-z]{0,1}.

Символьные классы

В символьный класс может входить любой литерал, а так же интервалы литералов. Для описания интервалов литералов используется символ '-', который ставится между первым символом интервала и последним.

Например:

[1-5] - числа в диапазоне от 1 до 5.

[a-f] - буквы латинского алфавита от a до f.

[a-fq-x] - буквы латинского алфавита от a до f и от q до x.

Описать условие, что в определенном месте строки могут стоять символы: либо a, либо g, либо 7, либо 4: символьный класс: [ag47].

В символьном классе можно перечислять допустимые в условии поиска литералы. Перечисление литералов можно совмещать с указанием интервалов:

[14a-kz] - это означает, что символ в строке может совпадать с 1, 4, буквами латинского алфавита с a по k, а так же с буквой z. Естественно литералами могут быть не только буквы и цифры, а так же знаки препинания, математические знаки, например ',' (запятая), '!' (восклицательный знак), '+' (плюс), '-' (минус).

[-,a-z] - означает, что в символьный класс входят минус, запятая, а так же буквы латинского алфавита от a до z.

Символьный класс, в который входят все символы, кроме заданных? Например, все кроме a, b, c? Для этого существует специальный спецсимвол отрицания: ^ (крышка).

[^abc] - все символы, кроме букв латинского алфавита a, b, c.

При поиске в реальной программе надо поставить в начале условия поиска символ '^' (начало строки), а в конце условия поиска '$' (конец строки). Поэтому пример с пятью буквами и четырьмя цифрами (как и все остальные после них вышеприведенные) надо переписать: ^[a-z]{5}[0-9]{4}$.

Символ **\s** соответствует литералу 'пробел', либо литералу 'перевод строки', либо литералу 'табулятор'.

`preg_match("/^[a-z0-9]/", $string,$mathces);` - сначала идет слеш, потом символ начала строки, потом идет символьный класс, потом идет снова слеш. Вот именно первый и последний слеш символизируют, что внутри них заключено регулярное выражение.

Пример1. Проверка имени пользователя (только буквы и цифры).

```
<?php
$user = $_POST['username'];
if(!preg_match("/^[a-zA-Z0-9]+$/", $user)) {
    echo "Имя пользователя задано в неправильном формате";
} else {
    echo "Имя пользователя задано в правильном формате";
}
?>
```

Так как регистрационное имя пользователя должно состоять из латинских букв, а также цифр, то надо написать символьный класс, который будет удовлетворять этому условию: `[a-zA-Z0-9]` в этот символьный класс входит три интервала, первый интервал a-z (все символы от маленькой буквы а до маленькой буквы z), второй интервал A-Z (аналогично, но с большими буквами), третий интервал 0-9 (цифры от 0 до 9). Мы описали только одну букву, из которой может состоять регистрационное имя, но таких букв может быть... А теперь как раз надо ответить на вопрос, сколько таких букв может быть? Да сколько угодно, скажете вы, а я скажу, что вы неправы.

Регистрационное имя должно состоять минимум из одной буквы! и я считаю, чти это обязательное условие при прохождении регистрации, поэтому надо данный факт описать. Вспоминаем про квантификаторы: `[a-zA-Z0-9]+`

Плюс '+', как раз тот квантификатор, который говорит, что в строковой переменной $user должен быть минимум один символ, который соответствует условию.

А теперь надо сказать регулярному выражению, что условию должна соответствовать вся строка, от начала до конца, поэтому добавляем в регулярное выражение символ начала строки '^' вначале регулярного выражения и символ конца строки '$' в конец: `^[a-zA-Z0-9]+$`

Теперь надо объяснить функции preg_match, что строка `^[a-zA-Z0-9]+$` является регулярным выражением, надо поставить ограничители, я ставлю слеш '/':

`^[a-zA-Z][a-zA-Z0-9]*$` - условие, определяющее, что первый символ - буква.

Спецсимволы

\s	описывает либо пробел, либо символ табуляции, либо символ новой строки [a-z\s] – буквы латиницы с пробелом
\S	в своем большинстве это видимые символы, т.е. все, что не совпадает с \s
\w	все символы, которые могут входить в слово, обычно это [a-zA-Z_]
\W	все что не входит в определение \w
\d	все цифры, т.е. уже известный вам символьный класс [0-9]
\D	все, что не является цифрой

Для поля ввода адреса e-mail добавим в список разрешенных символов знаки "@" и "." и "-", иначе пользователь не сможет корректно ввести адрес. Зато уберем русские буквы и пробел:

```
if (preg_match("/[^(\w)|(\@)|(\.)|(\-)]/",$usermail)) {
        echo "invalid mail";
        exit;
}
```

Проверка на число

```
if(preg_match('/^\d+$/', $var)) echo $var;
```

Состоит ли строка только из букв, цифр и "_", длиной от 8 до 20 символов:

```
if (preg_match("/^[a-za-я0-9_]{8,20}$/",$string)) echo "yes"; else echo "no";
```

Есть ли в строке цифра и заглавная буква?

```
if (!preg_match("/(?=.*[0-9])("/(?=.*[A-Z])/", $pass)) echo "НЕТ цифр";
```

Историческая справка

Алгоритм поиска с использованием регулярных выражений был впервые разработан одним из создателей UNIX Кеном Томпсоном. Интересно, что изначально регулярные выражения появились не в теории вычислительных систем, а в нейрофизиологии. Основу теории регулярных выражений заложили нейрофизиологи У. Мак-Каллох и У. Питтс, работавшие над способами математического описания нервных процессов. Позднее математик С. Клини, основываясь на этих исследованиях, опубликовал работу "Представление событий в нейронных сетях", в которой и было введено понятие регулярных выражений. Кен Томпсон, основываясь на этих работах, адаптировал теорию регулярных выражений для алгоритмов поиска информации. Именно начиная с его работ, регулярные выражения стали использоваться в текстовых редакторах и вошли во многие языки программирования.