

Регулярные выражения

Регулярные выражения

```
<?php
```

```
// наша строка для испытаний
```

```
$string = 'abcdefghijklmnopqrstuvwxyz0123456789';
```

```
// вывод
```

```
echo $string;
```

```
?>
```

Регулярные выражения

- Если нам нужно просто узнать есть ли шаблон 'abc' в строке \$string мы можем набросать такой код:

```
<?php
$string = 'abcdefghijklmnopqrstuvwxyz0123456789';

echo preg_match("/abc/", $string);
?>
```

Начало строки

```
<?php
$string = 'abcdefghijklmnopqrstuvwxy0123456789';
```

```
// тест на начало строки
if(preg_match("/^abc/", $string))
{
    // окей, строка начинается с абс
    echo 'The string begins with abc';
}
else
{
    echo 'это фэйл';
}
?>
```

Теперь мы желаем узнать, начинается ли строка с 'abc'. Символ начала строки в регулярках - '^' (caret - знак вставки).

Пример выведет:
The string begins with abc

Оборачивающие слэши - разделители, содержат регулярное выражение. Это могут быть любые парные символы, например @regex@, #regex#, /regex/ и .т.п.

Символ ^ сразу после первого разделителя указывает что выражение начинается сначала строки и НИКАК иначе.

Что делать с регистром символов (строчные-прописные)

```
<?php
$string = 'abcdefghijklmnopqrstuvwxyz0123456789';
echo preg_match("/^ABC/", $string) ? 'Совпадение' : 'Не думаю';
?>
```

Скрипт вернет:
Не думаю

Все потому что поиск регистро-зависимый. Шаблон 'abc' не тоже самое что 'ABC'.
Чтобы найти оба варианта, нужно использовать модификатор. В регулярных выражениях для этого применяется модификатор 'i', который нужно указать за закрывающим разделителем регулярного выражения.

Регулярные выражения

```
<?php
$string = 'abcdefghijklmnopqrstuvwxyz0123456789';

if (preg_match("/^ABC/i", $string)) {
    echo 'Совпадение, строка начинается с abc';
} else {
    echo 'Не думаю';
}
?>
```

Теперь скрипт найдет паттерн 'abc'. Также теперь будут попадать под шаблон строки вида abc, ABC, Abc, aBc, и т.п.

Позже будет рассказано подробнее о модификаторах.

Как указать в паттерне конец строки

Делается это также как и в случае с поиском начала строки.

Распространенная ошибка, допускаемая многими прогерами - использование символа \$ для указания конца строки в шаблоне.

Это неверно, правильное решение - использовать утверждение \z. Посмотрите на этот код

```
<?php
$string = 'abcdefghijklmnopqrstuvwxyz0123456789';

// паттерн в конце строки?
if (preg_match("/89\z/i", $string)) {
    echo 'Совпадение, строка заканчивается на 89';
} else {
    echo 'Не думаю';
}
?>
```

preg_match_all();

```
<?php
$string = 'abcefg hijklm nopqrst uvwxyz0123456789';

// ищем BCE совпадения
preg_match_all("/^[b]/", $string, $matches);

// выведем все совпадения в цикле foreach
foreach ($matches[0] as $value) {
    echo $value;
}
?>
```