String là một chuỗi ký tự

PHP strlen () chức năng trả về độ dài của một chuỗi

PHP str\_word\_count () chức năng đếm số từ trong một chuỗi:

PHP strrev () chức năng đảo ngược một chuỗi:

**strpos($string, $needle)**: Tìm vị trí của chuỗi $needle trong chuỗi $string, kết quả trả về vị trí đầu tiên của $needle nếu tìm thấy và false nếu không tìm thấy.PHP str\_replace () chức năng thay thế một số nhân vật có một số nhân vật khác trong một chuỗi.

Substr: lấy chuỗi

substr(string,start,length)

strcmp:so sánh 2 chuỗi

strcmp(string1,string2)

Str\_split:chuyển đổi một chuỗi thành một mảng

$str = "Hello Friend";  
  
$arr1 = str\_split($str); or $arr2 = str\_split($str,3);

Strtolower():chuyển đổi một chuỗi sang chữ thường

Strtoupper():chuyển đổi một chuỗi sang chữ hoa

**trim($string, $character)**; Xóa ký tự $character nằm ở đầu và cuối chuỗi $str, nếu ta không nhập $character thì mặc định nó hiểu là xóa khoảng trắng.

**explode($delimiter, $string)**: chuyển một chuỗi string thành một mảng các phần tử với ký tự tách mảng là delimiter.

**strstr( $string, $needle)**: Tách một chuỗi bắt đầu từ $needle cho đến hết chuỗi.

PHP - Regular Expressions

biểu thức thông thường là không có gì hơn là một chuỗi hoặc mẫu của nhân vật chính. Họ cung cấp nền tảng cho các chức năng mô hình khớp.

Sử dụng biểu hiện thường xuyên, bạn có thể tìm kiếm một chuỗi cụ thể bên trong một chuỗi khác, bạn có thể thay thế một chuỗi bằng một chuỗi khác và bạn có thể tách một chuỗi thành nhiều khối.

PHP cung cấp các chức năng cụ thể để hai bộ chức năng biểu hiện thường xuyên, mỗi tương ứng với một loại nhất định của biểu thức chính quy. Bạn có thể sử dụng bất kỳ của họ dựa vào sự thoải mái của bạn.

* POSIX Regular Expressions
* PERL Phong cách Regular Expressions

Brackets

[0-9] matches any decimal digit from 0 through 9.  
[a-z] matches any character from lowercase a through lowercase z.  
[A-Z] matches any character from uppercase A through uppercase Z.  
[A-Za-z] matches any character from uppercase A through lowercase z.

Quantifiers

p + phù hợp với bất kỳ chuỗi có chứa ít nhất một p. •

p \* khớp với bất kỳ chuỗi chứa số không hoặc nhiều p.

p? phù hợp với bất kỳ chuỗi chứa số không hay một p.

p {2} phù hợp với bất kỳ chuỗi có chứa một chuỗi của hai p.

p {2,3} khớp với bất kỳ chuỗi có chứa một chuỗi của hai hoặc ba p.

p {2,} phù hợp với bất kỳ chuỗi có chứa một chuỗi của ít nhất hai p.

p $ phù hợp với bất kỳ chuỗi với p tại kết thúc của nó.

Tuy nhiên lá cờ khác có thể được chèn vào trước và trong vòng một chuỗi ký tự:

^ p phù hợp với bất kỳ chuỗi với p vào đầu nó. •

[^ a-zA-Z] khớp với bất kỳ chuỗi không chứa bất kỳ ký tự khác nhau, từ a đến z và A đến Z.

p.p phù hợp với bất kỳ chuỗi chứa p, tiếp theo bất kỳ ký tự, lần lượt theo sau bởi p khác.

Bạn cũng có thể kết hợp ký tự đặc biệt để tạo thành biểu thức phức tạp hơn. Hãy xem xét những điều sau đây ví dụ: • ^. {2} $ phù hợp với bất kỳ chuỗi nào chứa đúng hai nhân vật. • <b> (. \*) </ B> phù hợp với bất kỳ chuỗi đặt trong <b> và </ b>. • p (hp) \* khớp với bất kỳ chuỗi có chứa một p theo sau bởi không hay nhiều trường hợp của các chuỗi mã lực

Predefined Character Ranges (Character Classes)

[: Alpha:]: Chữ thường và ký tự chữ cái in hoa. Đây cũng có thể là

quy định như [A-Za-z].

[: Alnum:]: ký tự chữ cái viết thường và chữ hoa và chữ số.

Điều này cũng có thể được quy định như [A-Za-z0-9].

[: Cntrl:]: Điều khiển nhân vật như tab, chạy thoát, hoặc xóa.

[:digit:]: bằng số chữ số từ 0 đến 9. Điều này cũng có thể được quy định như [0-9].

[: Graph:]: nhân vật có thể in được tìm thấy trong phạm vi ASCII 33-126.

[: Lower:]: chữ thường ký tự chữ cái. Điều này cũng có thể được quy định như [a-z].

[: Punct:]: ký tự chấm câu, bao gồm ~ `! @ # $% ^ & \* () - \_ + = {} []:; '

<>,. ? và /.

[: Upper:]: ký tự chữ cái viết hoa. Điều này cũng có thể được quy định như [A-Z].

[: Space:]: ký tự khoảng trắng, bao gồm cả không gian, tab ngang, tab dọc,

dòng mới, thức ăn dạng, hoặc vận chuyển trở lại.

[: Xdigit:]: ký tự thập lục phân. Điều này cũng có thể được quy định như [a-fA-F0-9]

## Chức năng biểu thức chính quy POSIX PHP

PHP hiện cung cấp bảy chức năng để tìm kiếm chuỗi sử dụng POSIX kiểu biểu thức thông thường -

int ereg(string *pattern*, string *string* [, array *regs*])

example:

<?php  
$username = "jasoN";  
if (ereg("([^a-z])",$username))  
echo "Username must be all lowercase!";  
else  
echo "Username is all lowercase!";  
?>

<?php  
$url = "http://www.apress.com";  
// Break $url down into three distinct pieces:  
// "http://www", "apress", and "com"  
$parts = ereg("^(http://www)\.([[:alnum:]]+)\.([[:alnum:]]+)", $url, $regs);  
echo $regs[0]; // outputs the entire string "http://www.apress.com"  
echo "<br />";  
echo $regs[1]; // outputs "http://www"  
echo "<br />";  
echo $regs[2]; // outputs "apress"  
echo "<br />";  
echo $regs[3]; // outputs "com"  
?>

int eregi(string *pattern*, string *string*, [array *regs*])

<?php  
$pswd = "jasonasdf";  
if (!eregi("^[a-zA-Z0-9]{8,10}$", $pswd))  
echo "Invalid password!";  
else  
echo "Valid password!";  
?>

string ereg\_replace(string *pattern*, string *replacement*, string *string*)

<?php  
$text = "This is a link to http://www.wjgilmore.com/.";  
echo ereg\_replace("http://([a-zA-Z0-9./-]+)$",  
"<a href=\"\\0\">\\0</a>",  
$text);  
?>

This returns the following:  
This is a link to  
<a href="http://www.wjgilmore.com/">[http://www.wjgilmore.com/.</a](http://www.wjgilmore.com/.%3c/a)>.

$url = "Apress (http://www.apress.com)";  
$url = ereg\_replace("http://([a-zA-Z0-9./-]+)([a-zA-Z/]+)",  
"<a href=\"\\0\">\\0</a>", $url);  
echo $url;  
// Displays Apress (<a href="http://www.apress.com">http://www.apress.com</a>)

array split(string *pattern*, string *string* [, int *limit*])

<?php  
$text = "this is\tsome text that\nwe might like to parse.";  
print\_r(split("[\n\t]",$text));  
?>  
This returns the following:  
Array ( [0] => this is [1] => some text that [2] => we might like to parse. )

## PERL Phong cách Regular Expressions

Perl kiểu biểu thức thông thường tương tự như trong POSIX của họ. Cú pháp POSIX có thể được sử dụng gần như thay thế cho nhau với các chức năng biểu hiện thường xuyên Perl-phong cách. Trong thực tế, bạn có thể sử dụng bất kỳ định lượng được giới thiệu trong phần POSIX trước.

Cho phép đưa ra giải thích cho vài khái niệm được sử dụng trong PERL biểu thức thông thường. Sau đó chúng tôi sẽ giới thiệu với bạn wih chức năng biểu hiện liên quan thường xuyên.

### nhân vật meta

Một nhân vật siêu đơn giản chỉ là một chữ cái trước bởi một dấu gạch chéo mà hành động để cung cấp cho sự kết hợp giữa một ý nghĩa đặc biệt.

Ví dụ, bạn có thể tìm kiếm khoản tiền lớn bằng cách sử dụng '\ d' nhân vật meta: **/ ([\ d] +) 000 /** , đây **\ d** sẽ tìm kiếm cho bất kỳ chuỗi ký tự số.

Sau đây là danh sách các ký tự meta có thể được sử dụng trong PERL Phong cách biểu hiện thường xuyên.

**Character Description**

. a single character

\s a whitespace character (space, tab, newline)

\S non-whitespace character

\d a digit (0-9)

\D a non-digit

\w a word character (a-z, A-Z, 0-9, \_)

\W a non-word character

[aeiou] matches a single character in the given set

[^aeiou] matches a single character outside the given set

(foo|bar|baz) matches any of the alternatives specified

### modifiers

Một số bổ có sẵn mà có thể làm cho công việc của bạn với regexps dễ dàng hơn nhiều, như trường hợp nhạy cảm, tìm kiếm trong nhiều dòng, vv

**Modifier Description**

i Makes the match case insensitive

m Specifies that if the string has newline or carriage

return characters, the ^ and $ operators will now

match against a newline boundary, instead of a

string boundary

o Evaluates the expression only once

s Allows use of . to match a newline character

x Allows you to use white space in the expression for clarity

g Globally finds all matches

cg Allows a search to continue even after a global match fails

## Biểu thức chính quy PERL Chức năng tương thích của PHP

PHP cung cấp các chức năng sau đây để tìm kiếm chuỗi sử dụng biểu thức thông thường Perl-tương thích -

|  |  |
| --- | --- |
| **Sr.No** | **Mô tả chức năng** |
| 1 | [**preg\_match ()**](https://www.tutorialspoint.com/php/php_preg_match.htm)  Các preg\_match () chức năng tìm kiếm chuỗi cho mô hình, trở về đúng nếu mô hình tồn tại, và false. |
| 2 | [**preg\_match\_all ()**](https://www.tutorialspoint.com/php/php_preg_match_all.htm)  Các preg\_match\_all () chức năng phù hợp với tất cả các lần xuất hiện của mô hình trong chuỗi. |
| 3 | [**preg\_replace ()**](https://www.tutorialspoint.com/php/php_preg_replace.htm)  Các preg\_replace () chức năng hoạt động giống như ereg\_replace (), ngoại trừ biểu thức thông thường có thể được sử dụng trong các tham số đầu vào mô hình và thay thế. |
| 4 | [**preg\_split ()**](https://www.tutorialspoint.com/php/php_preg_split.htm)  Các preg\_split () chức năng hoạt động chính xác như split (), ngoại trừ biểu thức thông thường được chấp nhận như là các thông số đầu vào cho mô hình. |
| 5 | [**preg\_grep ()**](https://www.tutorialspoint.com/php/php_preg_grep.htm)  Các preg\_grep () chức năng tìm kiếm tất cả các yếu tố của input\_array, trả lại tất cả các yếu tố phù hợp với mô hình regexp. |
| 6 | [**quote preg\_ ()**](https://www.tutorialspoint.com/php/php_preg_quote.htm)  Trích dẫn ký tự biểu hiện thường xuyên |