



REGULAR EXPRESSION IN PHP

نویسنده : رضا علی

Alie6798@gmail.com

مفهوم عبارات منظم:

از سایت ویکی پدیا:

در علوم رایانه، عبارت باقاعده، که تحت عنوان **regex** (یا **regexp** مخفف عبارت انگلیسی **regular expression**) نیز نامیده می‌شود به معنی تطبیق رشته در متن است، که از قبیل نویسه‌های خاص، کلمات و الگوهای از نویسه‌ها می‌باشد. یک عبارت باقاعده با زبان معمولی نوشته می‌شود که می‌تواند توسط یک پردازشگر عبارت باقاعده، یا یک برنامه که به عنوان تولیدکننده **مترجم** یا بررسی‌کننده متن و تشخیص قسمت‌هایی از آن به وسیله مشخصات استفاده شود.

این نمونه‌ها می‌توانید قابلیت‌ها محدودی که عبارت با قاعده می‌تواند انجام دهد را نشان دهد:

- دنباله‌ای از نویسه‌های «car» در هر متن، از قبیل «car»، «cartoon» یا «bicarbonate»
- لغت «car» در زمانی که به صورت جداگانه استفاده شود
- لغت «car» وقتی که قبل از «blue» یا «red» آمده باشد
- یک نویسه «\$» که پس از آن یک یا چند رقم بیاید و پس از آن به صورت اختیاری یک ممیز بیاید و پس از ممیز دقیقاً دو رقم اضافه قرار داشته باشد (مانند «\$» ۱۰ «یا» ۲۴۵,۹۹ «»)

عبارت‌های باقاعده می‌توانند خیلی پیچیده‌تر از این مثال‌ها باشند.

نحوه نوشتن عبارات منظم در: php

دستورات در این روش از طریق توالی علامت‌ها، کلمات، حروف یا کاراکترهای خاصی (مانند `\w`، `|`، `*`، `+`، `()` و `0-9 a-z`...) که هر کدامشون مفهوم خاصی دارن، تعریف می‌شن، در واقع از کنار هم قرار گرفتن این علامت‌ها، دستور ویژه‌ای شکل گرفته و قسمت خاصی از رشته مورد نظر استخراج یا بررسی می‌شه.

مثال:

به عنوان مثال ما می‌خواهیم به دنبال `reza` در عبارت `reza_ali in tehran` بگردیم ببینیم اصلاً عبارت `reza` در

`reza_ali in tehran` وجود دارد یا نه؟

برای این کار ما از تابع `preg match` استفاده می‌کنیم:

کد:

```
<?php
$str = "reza_ali in tehran ";
```

```
echo preg_match('/reza/', $str); // output=1
?>
```

نکته: رشته ای که به دنبال آن می گردیم را در بین // قرار می دهیم

نکته ۲: این تابع حساس به حروف کوچک و بزرگ است

اگر بخواهیم تابع حساس به حروف کوچک و بزرگ نباشد باید به صورت زیر بنویسیم:

```
<?php
$str = "reza_ali in tehran";
echo preg_match('/REza/i', $str); // output=1
?>
```

برای تفهیم بیشتر به مثال زیر توجه کنید:

```
<?php
$string = "Reza_ali in http://tehar.org";
echo preg_match('/Reza/', $string); //result is 0
echo preg_match('/Reza/i', $string); //result is 1
?>
```

حال به نظر شما اگر بخواهیم عدد و حروف رو کنار هم استفاده کنیم چی کار بکنیم؟

خوب کاری نداره که به صورت زیر عمل می کنیم:

```
<?php
$string = "1234reza556";
$check = preg_match('/reza556/', $string); // Result is 1
echo $check;
?>
```

خروجی ۱ می باشد

حال به مثال زیر توجه کنید:

```
<?php
$string = "1234Reza556";
$check = preg_match('/reza556/', $string);
echo $check;
?>
```

خروجی صفر می باشد برای این که حساس به حروف کوچک و بزرگ می باشد برای این که به حروف کوچک و بزرگ حساس نباشد کاراکتر را بعد از / قرار می دهیم به صورت زیر:

```
<?php
$string = "1234Reza556";
$check = preg_match('/reza556/i', $string);
echo $check;
?>
```

کاراکتر های قلاب و کروشه:

قلاب، کروشه یا bracket کاراکتر دیگری هست که در دستورات با قاعده کاربرد زیادی دارد، این کاراکتر در نقش دستوری، به کاراکتر محدود ساز هست. و فقط کاراکترهای خودشو شامل می شه (در حالت عادی حساس به حروف بزرگ و کوچک هم هست)، فرض کنید [a] تنها شامل رشته ای می شه که دست کم حرف a در آن به کار رفته باشه.

-> برا تفهیم بیشتر چند نمونه میارم:

» [a]

شامل رشته ای می شه که حداقل a در آن به کار رفته باشه، مانند abcd , aadc , baac (به طور مثال شامل bnv, 232, cccz و... نمی شه).

» [znx]

شامل رشته ای می شه که دست کم z یا x یا n در آن به کار رفته باشه، مثله xzn, regex, zend و... (به طور مثال شامل ZxN, abc, 123 و... نمی شه).

» [a-z]

شامل رشته ای می شه که دست کم حروف a تا z الفبای انگلیسی با حروف کوچک (در اون به کار رفته باشه، مانند bBbb, Regex, Php و... (به طور مثال شامل ۱@۱۲۲, REGEX, 12223 و... نمی شه).

» [A-Z]

شامل رشته ای می شه که دست کم حروف A تا Z الفبای انگلیسی با حروف بزرگ (در اون به کار رفته باشه، مانند PHP، ReGex، BCBBB و... (به طور مثال شامل abbcc، 1223، aca و... نمی شود).

» [0-9]

شامل رشته ای می شه که دست کم اعداد ۰ تا 9 در اون به کار رفته باشه، مثله CC۳۵، 666، b122 و... (به طور مثال شامل abcd، p%*&، aa و... نمی شه)

» [abct-z]

شامل رشته ای می شه که حداقل حرف a یا b یا c یا t در اون به کار رفته باشه (دقت کنید که در اینجا چون توالی حروف انگلیسی بهم خورده، علامت - از حرف t به بعد را شامل می شه)، مانند You، bb، abc و... (به طور مثال شامل Yo، 142، de و... نمی شه)

» نکات

نکته ۱:

همون طور که در دستورات بالا دیدید، علامت - یک کاراکتر همراهی کننده در عبارات با قاعده (یعنی در کنار سایر علامت ها استفاده می شه) و به معنی "تا" هست .

نکته ۲:

علامت - در صورتی که در ابتدا یا انتهای دستور قرار بگیره، خاصیت دستوری خودشو از دست می ده، به طور مثال [az-] شامل a یا z یا علامت - می شه (همچنین [az-])

نکته ۳:

علامت [یا] چنانچه بلافاصله بعد از اولین قلاب قرار بگیره، حالت دستوری خودشو از دست می ده، به طور مثال [zxn] شامل z یا x یا n و یا علامت [می شه.

در این قسمت از آموزش کاراکتر [^] رو بررسی میکنیم.

در قسمت قبل به بررسی کاراکتر [] پرداختیم و گفتیم که این کاراکتر به نوع محدود کننده هست در نقش دستوری.

حال به بررسی کاراکتر [^] میپردازیم.

در کنار کاراکتر گروه (قلاب)، کاراکتر گروه و علامت ^ وجود داره که برعکس حالت قبل هست. یعنی هر کاراکتری جز آنچه درون قلاب تعریف شده است.

به فرض [m^] شامل رشته ای میشه که به جزء حرف m، کاراکتر دیگه ای توی اون باشه (اگر تنها حرف m باشد، نتیجه FALSE خواهد بود)، یا [acd^] شامل رشته ای می شه که جز حروف a و c و d، کاراکتر دیگه ای نیز داشته باشه.

چند مثال میزنم:

[a^] :

شامل رشته ای می شه که به جزء a، کاراکتر دیگه ای در اون به کار رفته باشه، مثله abcd، bagd، accab و... (به طور مثال شامل aaaa، a، aaa و... نمی شه).

[msn^] :

شامل رشته ای می شه که به جزء m و s و n، کاراکتر دیگه ای در اون به کار رفته شه، مانند MSN، MASEN، و... (به طور مثال شامل msn، snnn، mmm و... نمی شه).

[a-z^] :

شامل رشته ای می شه که به جزء حروف a تا Z الفبای انگلیسی با حروف کوچک (، کاراکتر دیگه ای در اون به کار رفته شه، مثله Tehran، rezaali، Php، ... (به طور مثال شامل php، rezaali، bbbb و... نمی شود).

[A-Z^] :

شامل رشته ای می شه که به جزء حروف A تا Z الفبای انگلیسی با حروف بزرگ (، کاراکتر دیگه ای در اون به کار رفته باشه، مانند Tehran، REZAALI، PHP، ... (به طور مثال شامل PHP، REZAALII، TEHRANI و... نمی

شه.)

: [0-9^]

شامل رشته ای می شه که به جز اعداد ۰ تا 9، کاراکتر دیگه ای توی اون به کار رفته باشه، مانند 22&، vC۴۴، m112و... (به طور مثال شامل ۲۲۲، ۵۵۵، ۱۲۳ و... نمی شه.)

: [abct-z^]

شامل رشته ای می شه که به جز حرف a و b و c و t (تا Z با حروف کوچک)، کاراکتر دیگه ای در اون به کار رفته باشه، مانند You، b&b، ab1c و... (به طور مثال شامل u، bb، abc و... نمی شه.)

نکته:

به حساس بودن دستورات به حروف بزرگ و کوچک دقت کنید.

در این قسمت از آموزش کاراکتر (hat) ^ رو بررسی میکنیم:

در قسمت قبل به بررسی کاراکتر [^] پرداختیم. در این قسمت کاراکتر (hat) ^ به تنهایی رو مورد بررسی قرار میدیم.

کاراکتر ^ (شروع)

علاوه بر استفاده از کاراکتر (hat) ^ در داخل گروه، این کاراکتر به تنهایی نیز در دستورات مربوط به عبارات با قاعده کاربرد فراوان داره.

علامت) ^ خارج از قلاب و در شروع دستور) به معنی نقطه شروع یک رشته هست، به طور مثال دستور mha^ به معنی هر عبارتی هست که از ابتدا با mha شروع شده باشه؛

بنا بر این رشته ای مته mha را شامل می شه، اما abcd رو خیر (چون عبارت اول دقیقا با mha شروع شده اما عبارت دوم با.abc)

مثال برای تفهیم بیشتر:

```
<?php
$string = 'Regex Is Easy!';
$check = preg_match('/^Regex/', $string);
echo $check;
?>
```

:شروع شده، اما اگه کد به این شکل بود php خواهد بود، چون عبارت دقیقا با TRUE خروجی دستور بالا برابر ۱ یا

```
<?php
$string = 'why Rereg Is Easy!';
$check = preg_match('/^Rereg/', $string);
echo $check;
?>
```

خروجی برابر ۰ یا FALSE هست، چون در دستور تنها رشته ای رو در نظر گرفتیم که از شروع با عبارت Rereg باشد، در حالی که شروع رشته ما با عبارت فرضی why است.

نکته :

همون طور که پیش از این ملاحظه کردید، کاربرد علامت ^ درون قلاب با حالت عادی متفاوت است.

موفق و پیروز باشید.

رضا علی

اردیبهشت ۱۳۹۴