

Хайлт

Лекц 6

ХАЙЛТ

pattern- илэрхийлэл

Preg_match

```
$result = preg_match($pattern, $subject, $matches);
```

\$pattern-д зааж өгсөн илэрхийллийг хангасан
\$subject-н утга нь \$matches массивт орно.

```
int preg_match ( string $pattern , string $subject )
```

**Тэмдэгт мөрд pattern-д заасан тэмдэгт байгаа
эсэхийг шалгана.**

^ - эхэндээ ямар нэг тэмдэгтгүй

\$- төгсгөлдөө ямар нэгэн тэмдэгтгүй.

```
Preg_match ( '/boy/' , 'cowboys' );  
Preg_match ( '/^boy/' , 'cowboys' );  
Preg_match ( '/^cow/' , 'cowboys' );  
Preg_match ( '/cow$/' , 'cowboys' );  
Preg_match ( '/boys$/' , 'cowboys' );
```

```
true  
false  
true  
false  
true
```

Meta characters

Copyright ©2018 MNUE Ц.Гэрэлтуяа

\$pattern- илэрхийллийг бичихэд хэрэглэгдэх тусгай
тэмдэгтүүд

1. \ ташуу зураас, @ тэмдэг, |-босоо зураас
Pattern бичихэд хэрэглэгдэх зааглах тэмдэгт.
Дээрх 3-н алийг нь ч хэрэглэсэн болно.
2. Цэг
 - Зөвхөн нэг тэмдэгтийг илэрхийлдэг.
 - preg_match('/./', 'PHP 5', \$matches);
 - \$matches массивийн 0-р элементэд “Р” утга очсон байгаа. Учир нь цэг нь зөвхөн нэг тэмдэгтийг илэрхийлэх бөгөөд хамгийн эхний Р үсгийг зөв хариу гэж үр дүнгийн массивт олгож байна.

Meta Characters for a Regular Expression

Copyright ©2018-МННБ Ц.Гэрэлтуяа

Дараах тэмдэгтүүд PREG_match функцд ашиглагдана.

- ^** – эхэндээ ямар нэгэн тэмдэгтгүй
- \$** – тухайн тэмдэгтээр төгсөж байгааг
- .** – ямар нэгэн тэмдэгт
- []** – тэмдэгтийн олонлог
- ()** – бүлэг
- *** – хэдэн ч удаа орсон байж болно
- +** – хамгийн багадаа ганц орсон
- ?** – яг нэг тэмдэгт
- {n, m}** : $m \rightarrow n$ удаа давтагдсан бол

3. ?-Асуултын тэмдэг

- Нэг тэмдэгт байж болохыг илэрхийлнэ.
- `preg_match('/PHP.?5/', 'PHP 5', $matches);`
- Энэ тохиолдолд PHP5 болон PHP 5 гэж бичсэн аль ч тохиолдолд зөв гэж авна. Учир нь асуултын тэмдэг нь түүний оронд дурын нэг тэмдэгт байж болно (байхгүй байсан ч болно)гэсэн утгыг илтгэнэ. Энэ илэрхийлэлд цэг нь тусгаарлах үүрэгтэй оролцоно.

4. + - НЭМЭХ ТЭМДЭГ

- Нэгээс олон тэмдэгт байж болохыг илтгэнэ.
- `/a+b/` гэж шалгавал `'ab'`, `'aab'`, `'aaaaaaab'`, бүгд зөв гэж гарна. `'b'` гэж байвал худал утга авна.
- `preg_match('/a+b/', 'caaaabc', $matches);`
- тохиолдолд `$matches` массив `aaab` утгыг авна. `$matches[0]=aaab`

5. Үржих тэмдэг буюу од *

- * тэмдэг нь тухайн тэмдэгт байхгүй байж болно эсвэл хэдэн ч удаа давтагдсан байж болно гэсэн утгыг илэрхийлнэ.
- '/de*f/' гэж зааж өгвөл 'df', 'def', 'deeeef' утгууд бүгд таарна.

6. [...] хаалт

- Бүлэг тэмдэгтийг энэ хаалтанд зааж өгч болно. Тухайн зааж өгсөн бүлэг доторх утгуудтай таарч байгааг нь `$matches` массивт олгоно.
- *Тухайлбал [0-3] гэж зааж өгсөн бол 0,1,2,3 гэсэн тоонуудаас орсон л бол зөв утга гэж авах болно.*

Жишээ :

- `preg_match('/[0-9]+/', 'PHP is released in 2005.', $matches);`
- `$matches` хувьсагч нь 2005 гэсэн утгыг авна.
- + *тэмдэг нь тоонууд нь хэд ч байж болно гэсэн утгыг илтгэж байна.*

6. Жишээ 2

- ^ тэмдгийг [] хаалтан дотор ашиглавал
”тухайн тэмдэгтүүд ороогүй бол” гэсэн утгыг
илэрхийлнэ.
- `preg_match('/[^0-9]+/', 'PHP is released in 2005.', $matches);`
 - `$matches` хувьсагчид “PHP is released in” утга очно.
 - *Энэ нь 0-9 тоонууд ороогүй хэсгийг авч байна гэсэн үг.*

7. (...) хаалт

Pattern-д агуулагдаж буй элементүүдийн групп, дэд pattern үүсэхэд хэрэглэгдэнэ. Жишээ нь, PHP in 2005 гэсэн тэмдэгт мөрөөс бид зуун болон жилийг салгаж \$matches массивт авна гэвэл дараах илэрхийллийг бичнэ.

`'/([1-2][0-9])([0-9]{2})/'` Энд байгаа `()` хаалтын дагуу хоёр дэд илэрхийллийн утгыг хангаж байх утгууд \$matches – массивт орно.

Эхний илэрхийлэл `([1-2][0-9])` нь **10-29** хоорондох бүх зууныг авна.
Эхний орон нь 1 эсвэл 2 дараагийн орон нь 0-9 хооронд байна.

Дараагийн илэрхийлэл `([0-9]{2})` жилийг авна. *0-9 хоорон дахь тоонууд 2 удаа давтагдсан бол гэж байгаа тул сүүлийн 2 оронг авна*

```
preg_match('/([12][0-9])([0-9]{2})/','PHP in
2005.',$matches);
```

\$matches хувьсагч нь одоо дараах утгатай болно.

Array()

- `[0]` => 2005 /0 индекстэй элемент нь бүх нөхцлийг зэрэг хангаж байгааг /
- `[1]` => 20 /дэд илэрхийллийн дагуу 1-р хаалтны нөхцлийг хангаж байгааг/
- `[2]` => 05 /2-р хаалтны нөхцлийг хангаж байгааг /

8. (?: ...)

Үр дүнгийн массивт орж болохгүй элементүүд бүхий дэд илэрхийллийг үүсгэдэг. Үүнийг дараах байдлаар ашиглаж болно. `preg_match('@([A-Za-z]+)(?:hans)@', 'Derick Rethans', $matches);`

- `$matches` дараах утгатай болно.
- `Array()`
- `[0]` => Derick Rethans
- `[1]` => Derick Ret /hans тэмдэгтийг үр дүнгээс хассан байна./
- `[A-Za-z]` –том жижиг үсгүүдийг авна. + тэмдэг нь хэд ч орсон байж болно.
- `@` - тэмдэг нь ташуу зураастай адил үүрэг гүйцэтгэнэ.

9. (?P<name>...)

- Нэр бүхий дэд илэрхийллийг үүсгэнэ. Энэ нь энгийн дэд илэрхийлэлтэй адил боловч гаралтын массивт нэмэлт элемент үүсгэдэг.
- `preg_match('/(?P<century>[12][0-9])(?P<year>[0-9]{2})/','PHP in 2005.', $matches);`
- `$matches` –н утга
- `Array()`
- `[0] => 2005`
- `[century] => 20`
- `[1] => 20`
- `[year] => 05`
- `[2] => 05`
- Энэ нь бүрэн pattern хийгээд дунд нь будилахгүй байх боломжийг олгодог.

Meta Characters

Copyright ©2018 MNUE Ц.Гэрэлтуяа

Илэрхийлэлд ашиглаж болох meta - тэмдэгтүүд

i – том жижиг үсгийг ялгаагүй гэж үзнэ.

```
<?php
if (preg_match("/php/i", "PHP бол вэб
програмчлалын хэл.")) {
    echo "олдлоо.";
} else { echo "олдсонгүй."; }
?>
```

\b – зөвхөн тухайн үг олдсон тохиолдолд үнэн, бусад тохиолдолд худал утга буцаана.

```
<?php
```

```
if (preg_match("/\bweb\b/i", "PHP бол вэб  
програмчлалын хэл.)) { echo "олдлоо.";  
} else { echo "олдсонгүй.";  
}
```

```
if (preg_match("/\bweb\b/i", "PHP бол вэбсайт  
програмчлалын хэл.)) { echo "олдлоо.";  
} else { echo "олдсонгүй.";  
}  
?>
```

\? \+ *\[\] \{\}

- Escape тэмдэгтүүдийн анхны хэрэглээ нь бусад метатэмдэгтүүдийн тусгай хэрэглээнээс (зарим тэмдэгтүүд нийлэх үедээ өөр өөр утга илэрхийлдэг.) зайлсхийх, жишээ нь, 4** гэдгийг өөрийн pattern-д хэрэглэнэ гэвэл дараахийг ашиглаж болно.
- `'/^4**$/'`
- Давхар хашилтыг хэрэглэхээс зайлсхийх хэрэгтэй учир нь RНР нь \-д хүртэл тусгай хэрэглээтэй байдаг. Дараах pattern нь дээрхтэй адил
- `"/^4**$/"`
- (Энэ тохиолдолд, `"/^4**$"` гэж бичсэн нь мөн ажиллана.

Жишээ:

- `"/([^<]+)<([a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\\.)+[a-zA-Z0-9_\-]+)>/"`
- Дараах формат бүхий мэйл хаягийг олоход хэрэглэгдэнэ.
- `'Derick Rethans <derick@php.net>'`

Жишээ

- `<?php`
- `$mon_date = date('Y-m-d',strtotime(date('Y') . 'W' . str_pad(date('W'), 2, '0',STR_PAD_LEFT) . '1'));`
- `echo $mon_date;`
- `$sun_date = date('Y-m-d',strtotime(date('Y') . 'W' . str_pad(date('W'), 2, '0',STR_PAD_LEFT).'7'));`
- `echo "
",$sun_date;`
- `?>`

URL хаягнаас домэйн нэрийг авах

Домайн нэр: php.net

- `<?php`
- `// get host name from URL`
- `preg_match('@^(:http://)?([^\s/]+)@i',`
- `"http://www.php.net/index.html", $matches);`
- `$host = $matches[1];`
- `// get last two segments of host name`
- `preg_match('/[^\s]+\.[^\s]+$/i', $host, $matches);`
- `echo "домэйн нэр нь: {$matches[0]}\n";`
- `?>`

Дасгал 1:

Мэйл хаяг шалгах илэрхийлэл бич.

- [a@b.c](#)
- а- буюу нэр нь зөвхөн үсгээр эхэлнэ. а дунд доогуур зураас, цэг орж болно. а нь тоо эсвэл үсгээр төгсөнө.
- Мэйл нь @ тэмдэг, цэгтэй байна.
- b-нь зөвхөн үсэг байна. Дунд нь элдэв тэмдэгт байхгүй.
- c- нь 2-3 тэмдэгтийн урттай байна. Зөвхөн үсэг байна. Дунд нь элдэв тэмдэгт байхгүй.

Дасгал 2: Функц зохиох

- “Мессеж – зар” дотор болохгүй үгнүүд байгаа эсэхийг шалгах функц зохиох
- <http://www.a.b> Гараас оруулсан вэбийн хаяг зөв эсэхийг шалгах функц зохиох
- Дасгал1-н дагуу мэйл хаяг шалгах функц зохиох