

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 5: PHP Regular Expressions– PHP Shprehjet e rregullta

- Rregulla të përgjithshme [regular expressions php.txt](#)

Regular Expression	Will match...
foo	The string "foo"
^foo	"foo" at the start of a string
foo\$	"foo" at the end of a string
^foo\$	"foo" when it is alone on a string
[abc]	a, b, or c
[a-z]	Any lowercase letter
[^A-Z]	Any character that is not a uppercase letter
(gif jpg)	Matches either "gif" or "jpeg"
[a-z]+	One or more lowercase letters
[0-9\.\-]	Any number, dot, or minus sign
^[a-zA-Z0-9_]{1,}\$	Any word of at least one letter, number or _
((wx))((yz))	wy, wz, xy, or xz
[^A-Za-z0-9]	Any symbol (not a number or a letter)
(([A-Z]{3}) ([0-9]{4}))	Matches three letters or four numbers

Ushtrimi 1: Të ipet shembulli ku kontrollohet dhe validohet përmbajtja e shkrimit të një email adrese, përmes shprehjeve të rregullta në PHP.

```
<?php
$email1 = "test@hotmail.com";
print "Email-i testues: '$email1'";
print "<br>";
if (preg_match('/^[a-zA-Z0-9 _\-\.] +@[a-zA-Z0-9\-\.] +\.[a-zA-Z0-9\-\.] +$/ ',
$email1))
{
    echo "Email adresa1 është valide.";
    echo "<br>";
    echo "<br>";
}
```

```
}
else
{
    echo "Email adresa1 nuk është valide.";
    echo "<br>";
    echo "<br>";
}
?>

<?php
$email2 = "emri.mbiemri@aaa.bbb";
print "Email-i testues: '$email2'";
echo "<br>";

$regexp = "/^[^0-9][A-z0-9_]+([.][A-z0-9_]+)*[@][A-z0-9_]+([.][A-z0-9_]+)*[.][A-
z]{2,4}$/";

if (preg_match($regexp, $email2)) {
    echo "Email adresa2 është valide.";
    echo "<br>";
    echo "<br>";
} else {
    echo "Email adresa2 <u>nuk</u> është valide.";
    echo "<br>";
    echo "<br>";
}
?>
```

Ushtrimi 2: Të shkruhet një RegEx që kontrollon për format të caktuar të datës.

```
<?php

$date = "2014-04-13";
print $date;
print "<br>";
if (preg_match ('/([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})/', $date, $regs)) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format invalid i dates: $date";
}
?>
```

Ushtrimi 3: Të shkruhen PHP RegEx që manipulojnë dhe kontrollojnë shkrim të caktuar të numrave si p.sh. IP adresë, ID numër.

```
<?php
$var = "127.0.0.1";
if (preg_match('/^([0-9]{1,3}\.){3}[0-9]{1,3}$/',$var))
{
    print "IP adresa '$var' është valide.";
}
else
{
    print "IP adresa '$var' nuk është valide.";
}
?>
```

```
-----
<?php

$var = "213-25-5674";

if (preg_match('/^([0-9]{3}\-[0-9]{2}\-[0-9]{4})$|(^([0-9]{9})$)/',$var))
{
    print "SSN '$var' është valid.";
}
else
{
    print "SSN '$var' nuk është valid.";
}
?>
```

Ushtrimi 4: Të shkruhet një PHP RegEx, e cila kontrollon varësisht nga kushtet e RegEx, nëse plotësohen apo jo.

```
<?php
$var = "#FFFFFFE";

if(preg_match('/^#?([a-f]|[A-F]|[0-9]){3}(((a-f)|[A-F]|[0-9]){3})?$/',$var))
{
    print "'$var' është valide.";
}
else
{
    print "'$var' nuk është valide.";
}
?>
```

```
<?php
$var = "filan123";

if (preg_match('/^[a-zA-Z0-9_]{3,16}$/', $var))
{
    print "User-i '$var' është valid.";
}
else
{
    print "User-i '$var' nuk është valid.";
}
?>
```

Ushtrimi 5: Të shkruhen RegEx të cilat kontrollojnë për rregulla të caktuar për manipulim me numra, p.sh. nr. te tel etj.

```
<?php
$var = "1-608-234-1234";

if( !preg_match("/^([1-])?[0-9]{3}-[0-9]{3}-[0-9]{4}$/i", $var) ) {
    echo 'Ju lutem shkruani numrin e telefonit ne forme valide.';
}
else
{ echo 'Numri '. $var .' eshte shkruar ne forme valide!';}
?>
```

```
-----
<?php
$var = "-15.405";
if (preg_match("/^[+-]?[0-9]*\.[0-9]+$/", $var))
{
    print "Numri i dhënë '$var' është valid.";
}
else
{
    print "Numri i dhënë '$var' nuk është valid.";
}
?>
```

Ushtrimi 6: Të shkruhen PHP RegEx për të kontrolluar përmbajtje të caktuara të stringjeve.

```
<?php
$string = 'XYZ';
if (eregi('z', $string))
{
    echo "Stringu '$string' permban ndonje 'z' or 'Z'!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje 'z' or 'Z'!";
}
?>
```

```
<?php
$string = 'test';
if (eregi("[[:alpha:]]", $string))
{
    echo "Stringu '$string' permban ndonje shkronje!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje shkronje!";
}
?>
```

```
<?php
$string = '123';
if (eregi("[[:alnum:]]", $string))
{
    echo "Stringu '$string' permban ndonje numer ose shkronje!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje numer ose shkronje!";
}
?>
```

Ushtrimi 7: Shembujt në vazhdim tregojnë përdorimin e funksionit preg_replace()

```
<?php
$string = "Ky eshte nje test!";
print $string;
print "<br />";
echo str_replace(" eshte", " ishte", $string);
print "<br />";
echo preg_replace("/ eshte/", " ishtee", $string);
echo "<br>";
    echo "<br>";
?>
```

```
<?php
/* Nuk do të funksionoj, psh ne POSIX, por tani eshte ne rregull me Perl RegEx. */
$num = 4;
$string = "Ky string ka kater fjale.";
$string = preg_replace('/kater/', $num, $string);
echo $string;
print "<br />";
/* Kjo do të funksionoj. */
$num = '4';
$string = "Ky string ka kater fjale.";
$string = preg_replace('/kater/', $num, $string);
echo $string;
?>
```

Ushtrimi 8: Të tregohet një shembull se si punohet me RegEx në PHP

```
<?php
$str = "Sot eshte dite me diell.";

print( "Test stringu eshte: '$str'<br /><br />" );
// thirrja e funksionit preg_match te kerkimi pEr modelin 'Sot' nE variabelEn str
if (preg_match( "/Sot/", $str ) )
print( "Stringu 'Sot' eshte gjetur.<br />" );

// kerkimi pEr shprehjen 'Sot' nE fillim tE stringut
if ( preg_match( "/^Sot/", $str ) )
print( "Stringu 'Sot' eshte gjetur nE fillim te rreshtit.<br />" );

// kerkimi pEr modelin 'Sot' tek fundi i stringut
if (preg_match( "/Sot$/", $str ) )
print( "Stringu 'Sot' eshte ne fund te rreshtit.<br />" );

// marrim emrin e hostit prej URL
preg_match('@^(?:http://)?(?:[^/]+)@i',
"http://www.php.net/index.html", $matches);
$host = $matches[1];

// marrim 2 segmentet e fundit te host name
preg_match('/[^\.]+\.[^\.]+$/', $host, $matches);
echo "Emri i domain-it eshte: {$matches[0]}\n";
?>
```

Ushtrimi 9: Të shkruhet RegEx në PHP e cila kontrollon nëse një password është “strong” apo jo.

```
<?php
$string = "Fyfjk34sdfjfsjq7";

if (preg_match("/^.*(?=.{8,})(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).*$/", $string)) {
    echo "String eshte 'alfa'.";
} else {
    echo "String eshte 'beta'.";
}
?>
Shpjegimi:
```

The "?"=" combination means "the next text must be like this"

The first grouping is `(?=.*{8,})`. This checks if there are at least 8 characters in the string.

The next grouping `(?=.*[0-9])` means "any alphanumeric character can happen zero or more times, then any digit can happen"

The next groupings `(?=.*[a-z])` and `(?=.*[A-Z])` are looking for the lower case and upper case letter accordingly anywhere in the string.

9b) Përdorimi i funksionit `preg_split()`

```
<?php
$keywords = preg_split("/[\s,]+/", "PHP, shprehjet e rregullta!");
print_r( $keywords );
?>
```