

# Regular Expressions

Regular Expressions janë algoritme të përputhjes së modeleve që mund të kryhen në një shprehje të vetme. Ato përdorin operatorë aritmetikorë të tillë si (+, -, ^) për të krijuar shprehje komplekse. Regular expressions na ndihmojnë të kryejmë detyra si validimi i email adresave, IP adresave, etj.

Pse të përdorim shprehjet e rregullta

Shprehjet e rregullta thjeshtojnë identifikimin e modeleve në stringje duke thirrur një funksion të vetëm. Kjo na kursen kohën e kodimit. I përdorim kur validojmë të dhënat hyrëse të përdoruesit të tilla si email adresat, numrat e telefonave, IP adresat.

Shprehjet e rregullta në PHP

PHP ka funksione që lejojnë të punojmë me shprehjet e rregullta.

`preg_match` – ky funksion përdoret për të kryer një përputhje modeli në një string. Kthen true nëse gjendet një përputhje dhe false nëse jo.

`preg_split` – ky funksion përdoret për të kryer një përputhje modeli në një string dhe pastaj të ndajë rezultatet në një varg numerik.

`preg_replace` – ky funksion përdoret për të kryer një përputhje modeli në një string dhe pastaj zëvendëson përputhjen me tekstin e specifikuar.

[abc]    A single character: a, b or c  
[^abc]    Any single character but a, b, or c  
[a-z]    Any single character in the range a-z  
[a-zA-Z]    Any single character in the range a-z or A-Z  
^    Start of line  
\$    End of line  
\A    Start of string  
\Z    End of string  
.    Any single character  
\s    Any whitespace character  
\S    Any non-whitespace character  
\d    Any digit  
\D    Any non-digit  
\w    Any word character (letter, number, underscore)  
\W    Any non-word character

\b Any word boundary character  
 (...) Capture everything enclosed  
 (a|b) a or b a?  
 Zero or one of a a\*  
 Zero or more of a a+  
 One or more of a a{3}  
 Exactly 3 of a a{3,}  
 3 or more of a  
 a{3,6} Between 3 and 6 of a

## Options

i case insensitive m treat as multi-  
 line string s dot matches newline x  
 ignore whitespace in regex A  
 matches only at the start of string  
 D matches only at the end of string  
 U non-greedy matching by default

Metacharacter	Description	Example
.	Matches any single character except a new line	/./ matches anything that has a single character
^	Matches the beginning of or string / excludes characters	/^PH/ matches any string that starts with PH
\$	Matches pattern at the end of the string	/com\$/ matches guru99.com,yahoo.com Etc.
*	Matches any zero (0) or more characters	/com*/ matches computer, communication etc.
+	Requires preceding character(s) appear at least once	/yah+oo/ matches yahoo
\	Used to escape meta characters	/yahoo+\.com/ treats the dot as a literal value
[...]	Character class	/[abc]/ matches abc
a-z	Matches lower case letters	/a-z/ matches cool, happy etc.
A-Z	Matches upper case letters	/A-Z/ matches WHAT, HOW, WHY etc.
0-9	Matches any number between 0 and 9	/0-4/ matches 0,1,2,3,4

## Si të validojmë emrin dhe mbiemrin duke përdorur shprehjet e rregullta në PHP

Kur lejojmë përdoruesin të dërgojë të dhënat në server, është ide e mirë të validojmë të dhënat para një procesimi të mëtejshëm. Fusha për shënimin e emrit është një fushë e zakonshme e formës për një web aplikacion. Në vazhdim tregohet se si të validojmë fushën e emrit për të kontrolluar nëse përdoruesi shënon emrin dhe mbiemrin e tij.

Regular Expression (REGEX) është mënyra më e lehtë për të validuar formatin e emrit të plotë në PHP. Përdorim funksionin `preg_match()` dhe një shprehje të rregullt për validimin e emrit në PHP.

```
preg_match("/^([a-zA-Z' ]+)$/","Given_Name");
Usage: <?php
if(preg_match("/^([a-zA-Z' ]+)$/",$givenName)){
echo 'Valid name given.';
}
else{
    echo 'Invalid name given.';
}
```

## Si të validojmë fjalëkalimet e fuqishme në PHP

Kur përdoruesi jep fjalëkalimin e llogarisë së tyre, gjithmonë rekomandohet të validojmë hyrjen e tyre. Validimi i fortësisë së fjalëkalimit është e dobishme për të kontrolluar nëse fjalëkalimi është i fuqishëm. Fjalëkalimet e fuqishme bëjnë më të sigurtë llogarinë e përdoruesit dhe ndihmon për të parandaluar thyerjen e llogarisë.

Duke përdorur shprehjet e rregullta, mund të validojmë fortësinë e fjalëkalimeve në PHP. Kodi në vazhdim, validon fjalëkalimin duke përdorur funksionin `preg_match()` në PHP me shprehjen e rregullt, për të kontrolluar nëse fjalëkalimi është i fuqishëm dhe i vështirë të supozohet.

- Fjalëkalimi duhet të jetë i gjatë të paktën 8 karaktere në gjatësi.
- Fjalëkalimi duhet të përfshijë të paktën një shkronjë të madhe.
- Fjalëkalimi duhet të përfshijë të paktën një numër.
- Fjalëkalimi duhet të përfshijë të paktën një karakter special.

```
// Given password
$password = 'user-input-pass';

// Validate password strength
$uppercase = preg_match('@[A-Z]@', $password);
$lowercase = preg_match('@[a-z]@', $password);
```

```

$number      = preg_match('@[0-9]@', $password);
$specialChars = preg_match('@[^\w]@', $password);

if(!$uppercase || !$lowercase || !$number || !$specialChars ||
strlen($password) < 8) {
    echo 'Password should be at least 8 characters in length and should
include at least one upper case letter, one number, and one special
character.';
}
else {
    echo 'Strong password.';
}

```

Si të validojmë email adresat në PHP

Në vazhdim jepet një shembull që kontrollon validitetin e një email adrese.

```

<?php
$my_email = "name@company.com
    ";
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "$my_email is a valid email address";
}
else {
    echo "$my_email is NOT a valid email address";
}
?>

```

ose

```

<?php
$my_email = "name@company.com
    ";
if (preg_match("/^[a-zA-Z0-9._-]+@[a-zA-Z0-9-]+\.[a-zA-Z.]{2,5}$/",
$my_email)) {
    echo "$my_email is a valid email address";
}
else {
    echo "$my_email is NOT a valid email address";
}
?>

```