

PHP - Form Email/URL

PHP provides two alternatives for validating the form data items which are strings but are expected to be a representation of Email ID or a URL. One way to check the form element contains email/URL is with the use of **RegEx (regular expressions)**, and the other, more convenient approach is to use `filter_var()` function. Let us apply both these methods and validate email and URL submitted by a form to a PHP script.

The HTML Form used for this chapter is as follows –

</>

Open Compiler

```
<h1>Email and URL Validation</h1>
<form action="hello.php" method="POST">
  <p><label for="email">Enter your email:</label>
  <input type="text" id="email" name="email"></p>
  <p><label for="URL">Enter your website</label>
  <input type = "text" id="URL" name="url"></p>
  <input type="submit">
</form>
```

Validation with Regex

PHP's built-in function library includes the **preg_match() function** that performs a regular expression match.

```
preg_match(
    string $pattern,
    string $subject,
    array &$matches = null,
    int $flags = 0,
    int $offset = 0
): int|false
```

This function searches subject for a match to the regular expression given in pattern. **preg_match()** returns 1 if the pattern matches given subject, 0 if it does not, or **false** on failure.

A valid email ID should satisfy the following regular expression –

```
"/^([a-z0-9\+\_\-]+)(\.[a-z0-9\+\_\-]+)*@([a-z0-9\-\-]+\.[a-z]{2,6})$/ix"
```

Similarly, a valid URL should satisfy the following regular expression –

```
"/\b(?:(:https?|ftp):\\\/|www\.)[-a-z0-9+&@#\/%?~_!|:,.;]*[-a-z0-9+&@#\/%~_!|]/i"
```

The following function returns "1" or "0" if the string is a valid email ID.

```
function checkemail($str) {
    return (!preg_match("/^([a-z0-9\+\_\-]+)(\.[a-z0-9\+\_\-]+)*@([a-z0-9\-\-]+\.[a-z]{2,6})$/ix", $str)) ? 0 : 1;
}
```

Example

Let us use the **checkmail() function** to check whether the email field in the above HTML is valid or not, with the help of following PHP code –

```
<?php
function checkemail($str) {
    return (!preg_match("/^([a-z0-9\+\_\-]+)(\.[a-z0-9\+\_\-]+)*@([a-z0-9\-\-]+\.[a-z]{2,6})$/ix", $str)) ? FALSE : TRUE;
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    if(!checkemail($email)){
        echo "Invalid email address.";
    } else {
        echo "Valid email address.";
    }
}
?>
```

The HTML form is rendered as below –

Email and URL Validation

Enter your email:

Enter your website

Test the PHP code by entering valid/invalid email string in the email field.

The following **checkURL()** function checks if a string represents a valid or invalid URL, and returns "1" or "0".

```
function checkURL($str) {  
    return (!preg_match("/\b(?:(:https?|ftp):\\\/\\\/|www\\.)  
        [-a-z0-9+&@#\\\/%?~_!|:,.;]*[-a-z0-9+&@#\\\/%?~_]/i", $str)) ? FALSE : TRUE;  
}
```

Example

The URL field extracted from the \$_POST array is given as argument to the above function.

```
<?php  
function checkURL($str) {  
    return (!preg_match("/\b(?:(:https?|ftp):\\\/\\\/|www\\.)  
        [-a-z0-9+&@#\\\/%?~_!|:,.  
        *[-a-z0-9+&@#\\\/%?~_]/i", $str)) ? FALSE : TRUE;  
}  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $url = $_POST['url'];  
    if(!checkURL($url)){  
        echo "Invalid URL."  
    } else {  
        echo "Valid URL."  
    }  
}  
?>
```

You can test the above code by entering URL string in the URL field of the above form.

Using filter_var() function

The built-in filter_var() function filters a variable with a specified filter.

```
filter_var(mixed $value, int $filter = FILTER_DEFAULT, array|int $options = 0): mixed
```

Depending on the enumerated filter ID as the value of **\$filter** parameter, the **\$value** parameter is checked and the function returns the filtered data, or false if the filter fails.

There are various **predefined filter ID constants** available –

Sr.No	ID & Description
1	FILTER_VALIDATE_BOOL Returns true for "1", "true", "on" and "yes". Returns false otherwise.
2	FILTER_VALIDATE_DOMAIN Validates whether the domain name label lengths are valid.
3	FILTER_VALIDATE_EMAIL Validates whether the value is a valid e-mail address.
4	FILTER_VALIDATE_IP Validates value as IP address
5	FILTER_VALIDATE_URL Validates value as URL

Example

The following PHP script validates the email and URL data submitted by the HTML for above –

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    $url = $_POST['url'];

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Invalid email format and please re-enter valid email\n";
    }
    else
        echo "Email entered is in valid format\n";
}
```

```
if (!filter_var($url, FILTER_VALIDATE_URL)) {  
    echo "Invalid URL format and please re-enter valid URL\n";  
}  
else  
    echo "URL entered is in valid format\n";  
}  
?>
```

You can test the performance of the above script by entering valid/invalid email/URL.