
Ch3 - Lesson 2: PHP Form Validation

FORM VALIDATION

- Data should be validated before submitted to the server to protect garbage data from stored into your system
- Two type of validation:
 - **Browser validation (client side validation)**
 - Via **JavaScript**, performed before submission to the server
 - **Server validation**
 - Via **server side scripts**: performed after user clicks the submit button and checked before data sent to the server or stored into the database

FORM VALIDATION

- A good way to validate a form on the server is to post the form to itself, instead of jumping to a different page.
- The user will then get the error messages on the same page as the form. This makes it easier to discover the error.

FORM VALIDATION

- Common validations:

- Presence Validation
- String Length Validation
- Type Validation
- Inclusion in set Validation
- Uniqueness Validation
- Format Validation

FORM VALIDATION

- Presence validation:

```
    $value="";  
    if(!isset($value) || empty($value))  
        die("Validation failed");
```

Student Registration Form

ID	<input type="text"/>
Name	<input type="text"/>



```
    $id=$_POST['id'];  
    $name=$_POST['name'];  
    if(!isset($_POST['submit_btn']) || empty($id) ||  
        empty($name))  
        die("Please fill all the required fields");
```

FORM VALIDATION

- String length validation:

```
        $password="myPass1234";  
        $min=6;  
        $max=10;  
if (strlen($password) < $min || strlen($password) > $max)  
    die("Password doesn't fulfill the requirement");
```

FORM VALIDATION

- Type validation:

```
    $value=5;  
    if(!is_numeric($value))  
die("Please enter only numeric values");
```

FORM VALIDATION

- Inclusion in set validation:

```
$value=$_POST['value'];  
    $set=array("M","F");  
    if(!in_array($value,$set))  
die("Item is not in the list");
```

FORM VALIDATION

- Uniqueness validation:

```
$username=$_POST['value'];  
$usernamefromdb=array();  
if(in_array($username,$usernamefromdb))  
die("User Name is already taken by other  
user");
```

FORM VALIDATION

- **Format validation:** used to validate whether the value has the right format.
- i.e. Email with @ symbol, currency \$ symbol, DateTime with **AM** or **PM**, acceptable telephone number format
- Uses regular expressions on the given string

```
preg_match($regexp, $value)
```

FORM VALIDATION

- Format validation

```
if (!preg_match("/PHP/", "PHP is SSS"))  
    die("match not found");  
else  
    echo "Match Found!";
```

FORM VALIDATION

- **Format validation: numbers only**

```
if (!preg_match("/^[.0-9]*$/", $value))  
    die("Please enter numbers only for Age");
```

FORM VALIDATION

- **Format validation: letters a-z and A-Z only**

```
if (!preg_match("/^[a-zA-Z ]*$/", $value)) {  
    die("Only letters and white space allowed");  
}
```

FORM VALIDATION

- **Format validation: validate email address**

```
if (!filter_var($value, FILTER_VALIDATE_EMAIL))  
    die("Invalid email format");
```

- OR

- We can use user defined regular expressions to validate email

FORM VALIDATION

More on regular expressions

https://www.tutorialspoint.com/php/php_regular_expression.htm



thankYou();