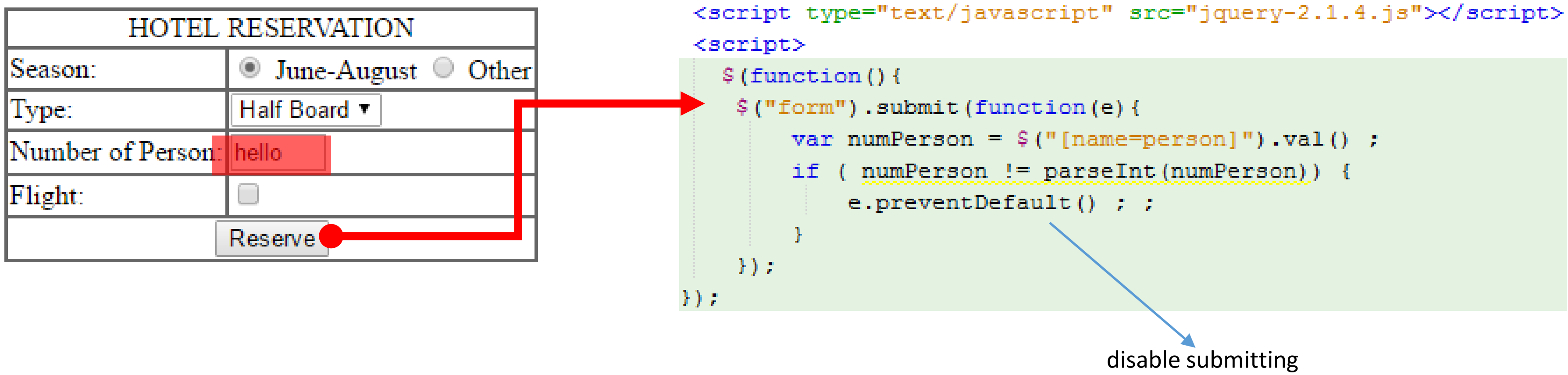
CTIS 256 Web Technologies II

Week # 5

Serkan GENÇ

Form Validation – (client-side)

- For security and proper execution of the application, the data acquired with any methods such as forms or query strings. They must be validated.
- If you obtain a data from a form field, you have to check its presence, type, format, valid range, white lists and so on. Then, use that data in your processing.
- Mainly, there are **two places** to validate data; client-side validation (*before sending data*) or server-side validation (*after sending data*).
- **Client-side validation** is done by using **Javascript**.
- Server-side validation is performed by server-side language, PHP, in our case.
- Client-side validation is **optional/recommended** but Server-side validation is **mandatory**.
- **Javascript** can be turned off from browser settings. This is why client-side validation is **not reliable**.
- Client-side validation increases **performance** since it prevents network round trip.
- **jQuery validate plugin** makes client-side validation easier.



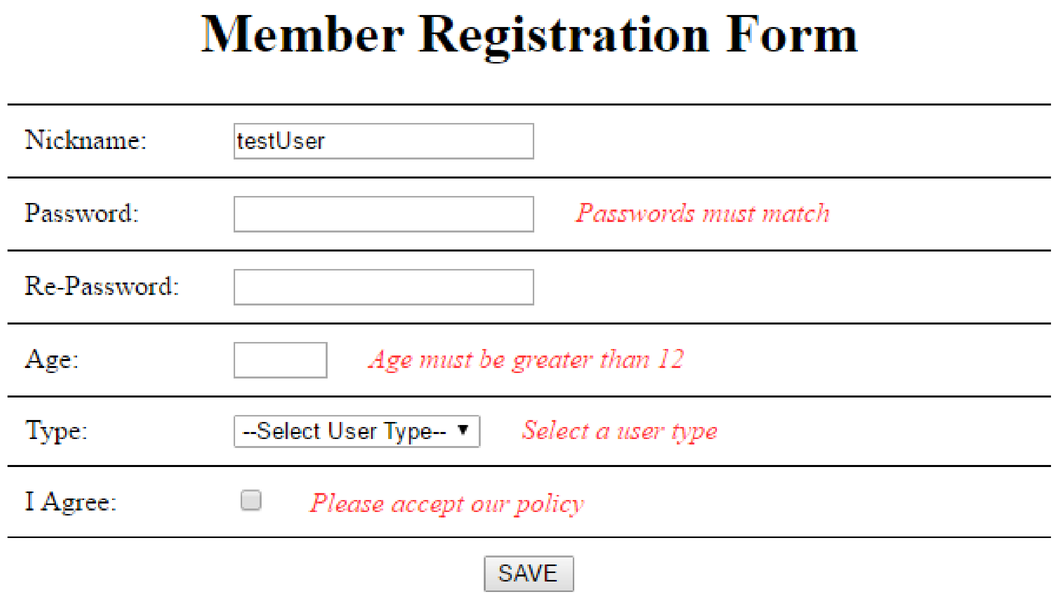
HOTEL RESERVATION	
Season:	● June-August ○ Other
Type:	Half Board ▾
Number of Person:	hello
Flight:	□
	Reserve

```
<script type="text/javascript" src="jquery-2.1.4.js"></script>
<script>
 $(function(){
  $("form").submit(function(e){
    var numPerson = $("[name=person]").val() ;
    if ( numPerson != parseInt(numPerson)) {
        e.preventDefault() ; ;
    }
  });
});
```

disable submitting

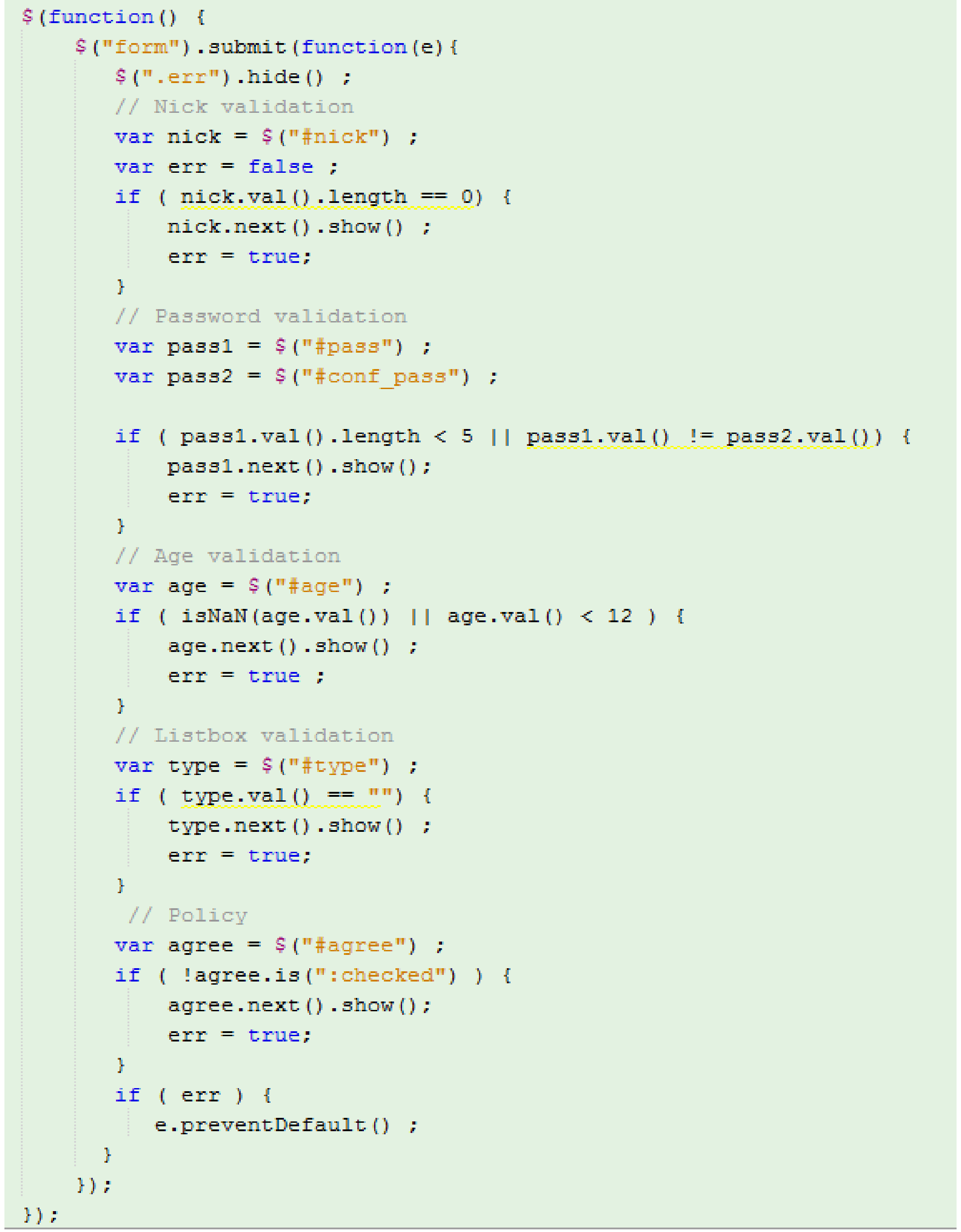
client-side validation

Validation With JQuery



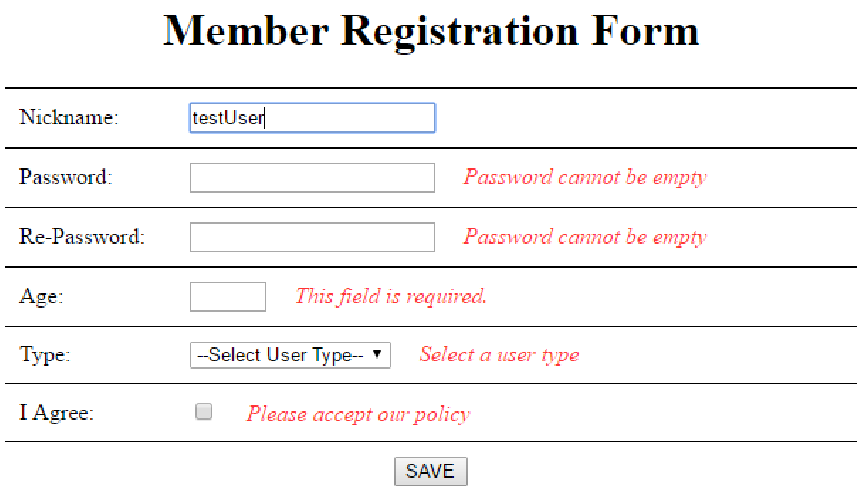
Member Registration Form

Nickname:	testUser	
Password:		*Passwords must match*
Re-Password:		
Age:		*Age must be greater than 12*
Type:	--Select User Type-- ▼	*Select a user type*
I Agree:	□	*Please accept our policy*



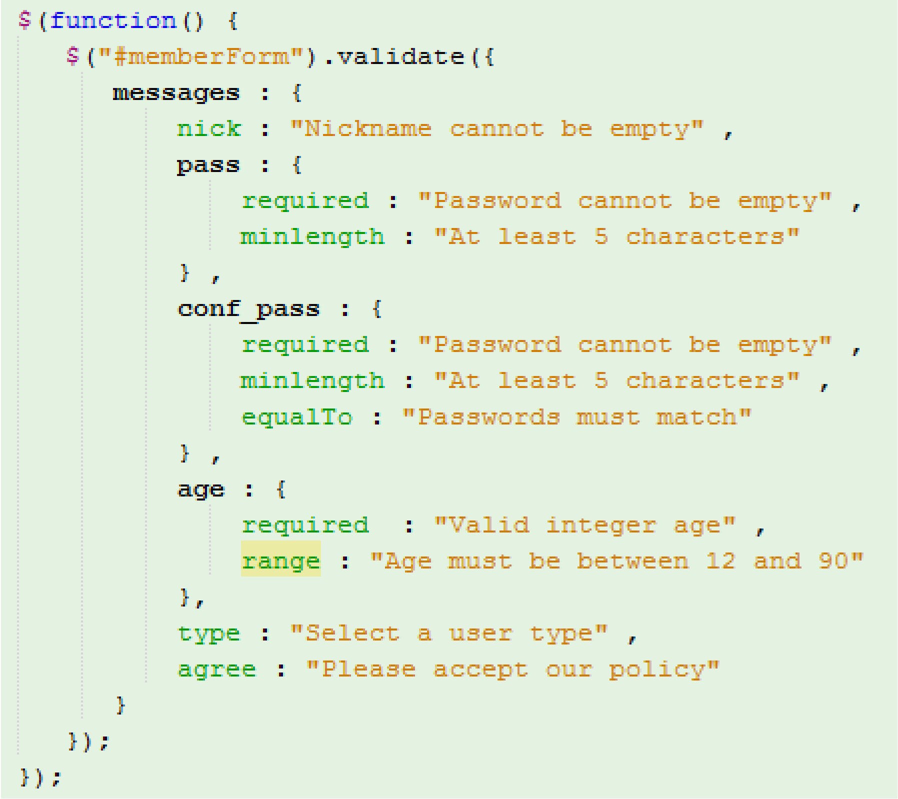
```
$(function() {
    $("form").submit(function(e){
        $(".err").hide() ;
        // Nick validation
        var nick = $("#nick") ;
        var err = false ;
        if ( nick.val().length == 0) {
            nick.next().show() ;
            err = true;
        }
        // Password validation
        var pass1 = $("#pass") ;
        var pass2 = $("#conf_pass") ;

        if ( pass1.val().length < 5 || pass1.val()  != pass2.val()) {
            pass1.next().show();
            err = true;
        }
        // Age validation
        var age = $("#age") ;
        if ( isNaN(age.val()) || age.val() < 12 ) {
            age.next().show() ;
            err = true ;
        }
        // Listbox validation
        var type = $("#type") ;
        if ( type.val() == "") {
            type.next().show() ;
            err = true;
        }
         // Policy
        var agree = $("#agree") ;
        if ( !agree.is(":checked") ) {
            agree.next().show();
            err = true;
        }
        if ( err ) {
           e.preventDefault() ;
       }
    });
});
```

Validation With **JQuery Validate Plugin**



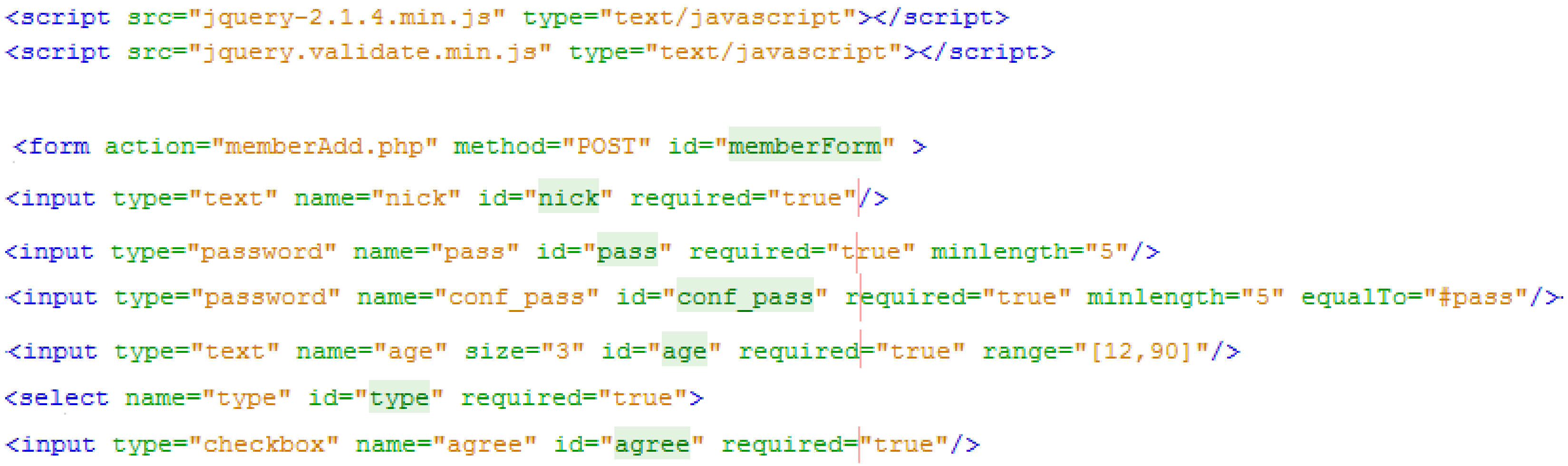
Member Registration Form

Nickname:	testUser	
Password:		*Password cannot be empty*
Re-Password:		*Password cannot be empty*
Age:		*This field is required.*
Type:	--Select User Type-- ▾	*Select a user type*
I Agree:	☐	*Please accept our policy*



```
$(function() {
    $("#memberForm").validate({
        messages : {
            nick : "Nickname cannot be empty" ,
            pass : {
                required : "Password cannot be empty" ,
                minlength : "At least 5 characters"
            } ,
            conf_pass : {
                required : "Password cannot be empty" ,
                minlength : "At least 5 characters" ,
                equalTo : "Passwords must match"
            } ,
            age : {
                required  : "Valid integer age" ,
                range : "Age must be between 12 and 90"
            },
            type : "Select a user type" ,
            agree : "Please accept our policy"
        }
    });
});
```

ADD PLUGIN



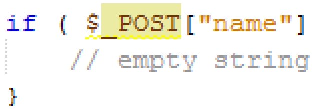
```
<script src="jquery-2.1.4.min.js" type="text/javascript"></script>
<script src="jquery.validate.min.js" type="text/javascript"></script>


 <form action="memberAdd.php" method="POST" id="memberForm" >
<input type="text" name="nick" id="nick" required="true"/>
<input type="password" name="pass" id="pass" required="true" minlength="5"/>
<input type="password" name="conf_pass" id="conf_pass" required="true" minlength="5" equalTo="#pass"/>
<input type="text" name="age" size="3" id="age" required="true" range="[12,90]"/>
<select name="type" id="type" required="true">
<input type="checkbox" name="agree" id="agree" required="true"/>
```

PHP Form Validation – (server-side)

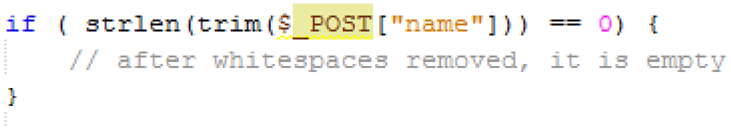
All data from $_POST/$_GET are in string format

1. Check if data is empty string:

- trim(str) : removes whitespaces around the string
- strlen(str) : return the number of chars

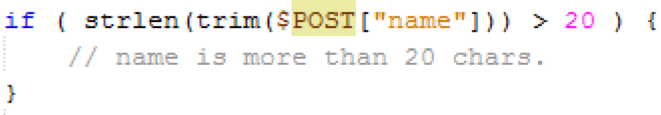


```
if ( $_POST["name"] == "") {
    // empty string
}
```



```
if ( strlen(trim($_POST["name"])) == 0) {
    // after whitespaces removed, it is empty
}
```

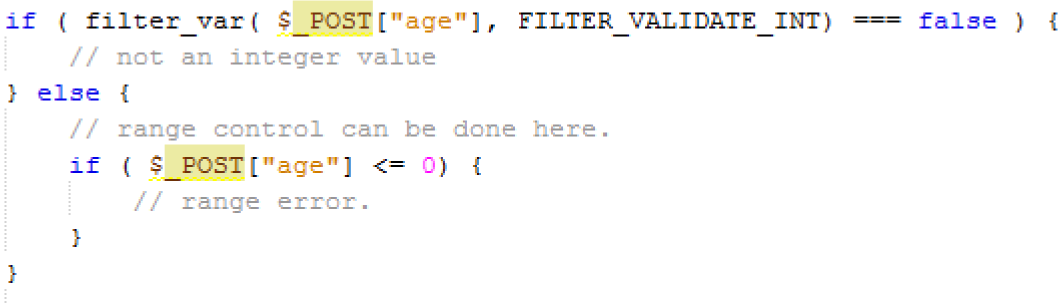
2. Data size:



```
if ( strlen(trim($POST["name"])) > 20 ) {
    // name is more than 20 chars.
}
```

3. Integer/Float:

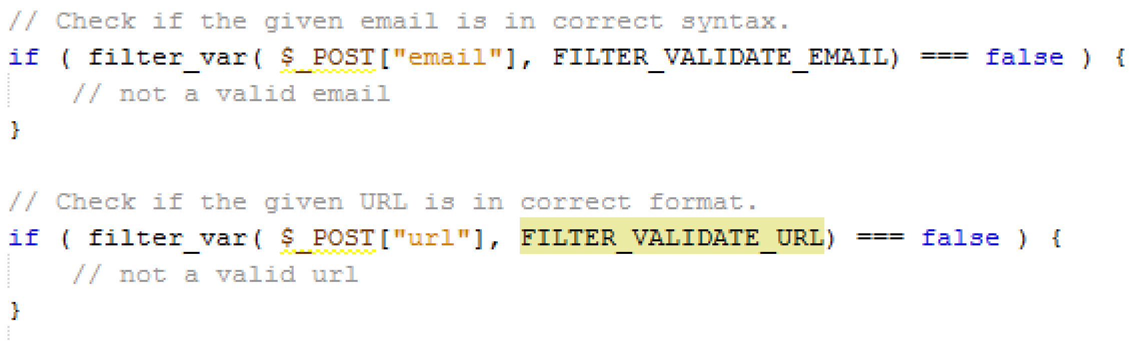
- filter_var() is used to validate integer, float, boolean
- FILTER_VALIDATE_INT, FILTER_VALIDATE_FLOAT
- *is_numeric()* also checks if the given string is integer **or** float.



```
if ( filter_var( $_POST["age"], FILTER_VALIDATE_INT) === false ) {
    // not an integer value
} else {
    // range control can be done here.
    if ( $_POST["age"] <= 0) {
        // range error.
    }
}
```

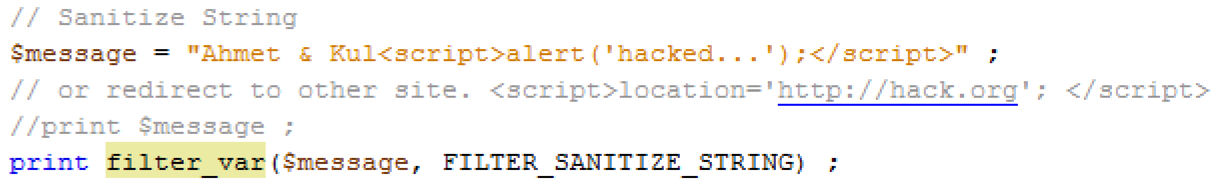
PHP Form Validation/Sanitize

4. Email, URL, IP validation :
   - FILTER_VALIDATE_EMAIL, FILTER_VALIDATE_URL, FILTER_VALIDATE_IP
   - For custom formats/patterns, use "regular expression"



```
// Check if the given email is in correct syntax.
if ( filter_var( $_POST["email"], FILTER_VALIDATE_EMAIL) === false ) {
    // not a valid email
}

// Check if the given URL is in correct format.
if ( filter_var( $_POST["url"], FILTER_VALIDATE_URL) === false ) {
    // not a valid url
}
```

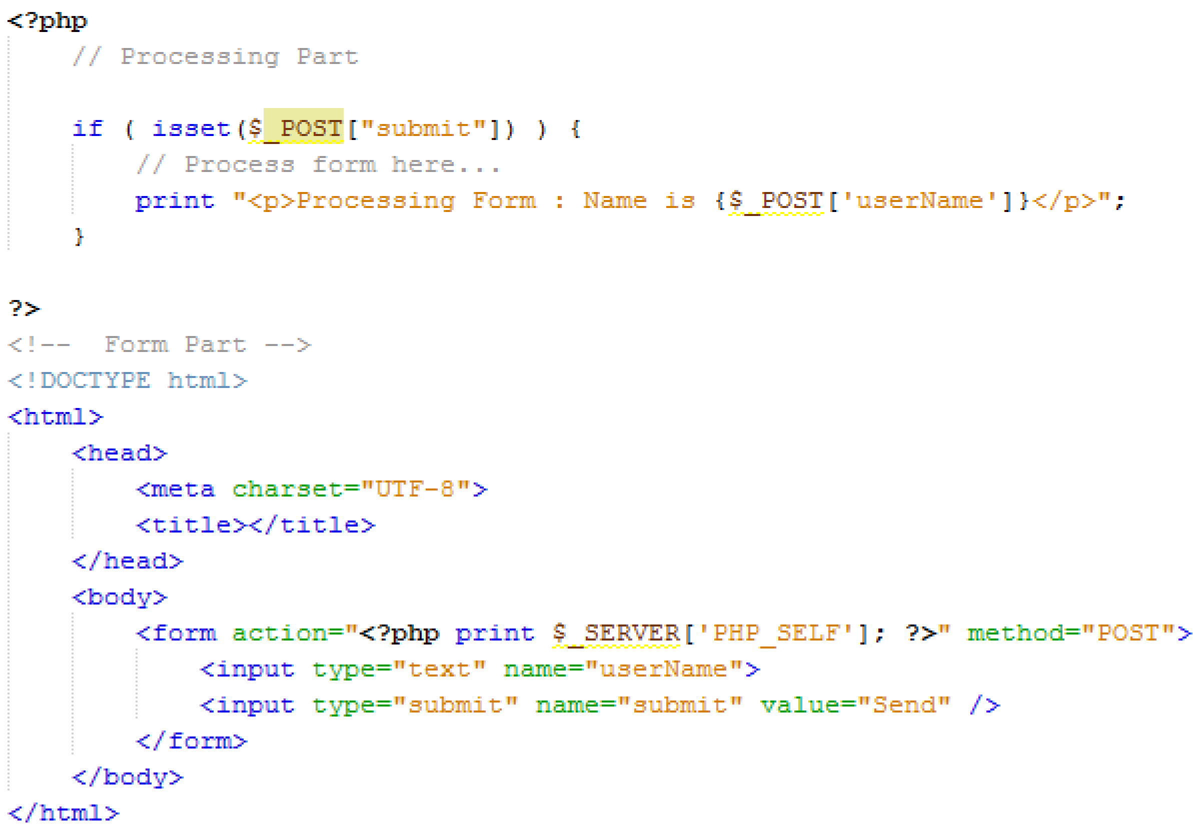
5. **Sanitize** : remove or change dangerous characters, tags from the string.
   - **strip_tags()** and FILTER_SANITIZE_STRING : remove tags.
   - **htmlspecialchars()** and FILTER_SANITIZE_FULL_SPECIAL_CHARS
   - Sanitize may change the orginal string.



```
// Sanitize String
$message = "Ahmet & Kul<script>alert('hacked...');</script>" ;
// or redirect to other site. <script>location='http://hack.org'; </script>
//print $message ;
print filter_var($message, FILTER_SANITIZE_STRING) ;
```

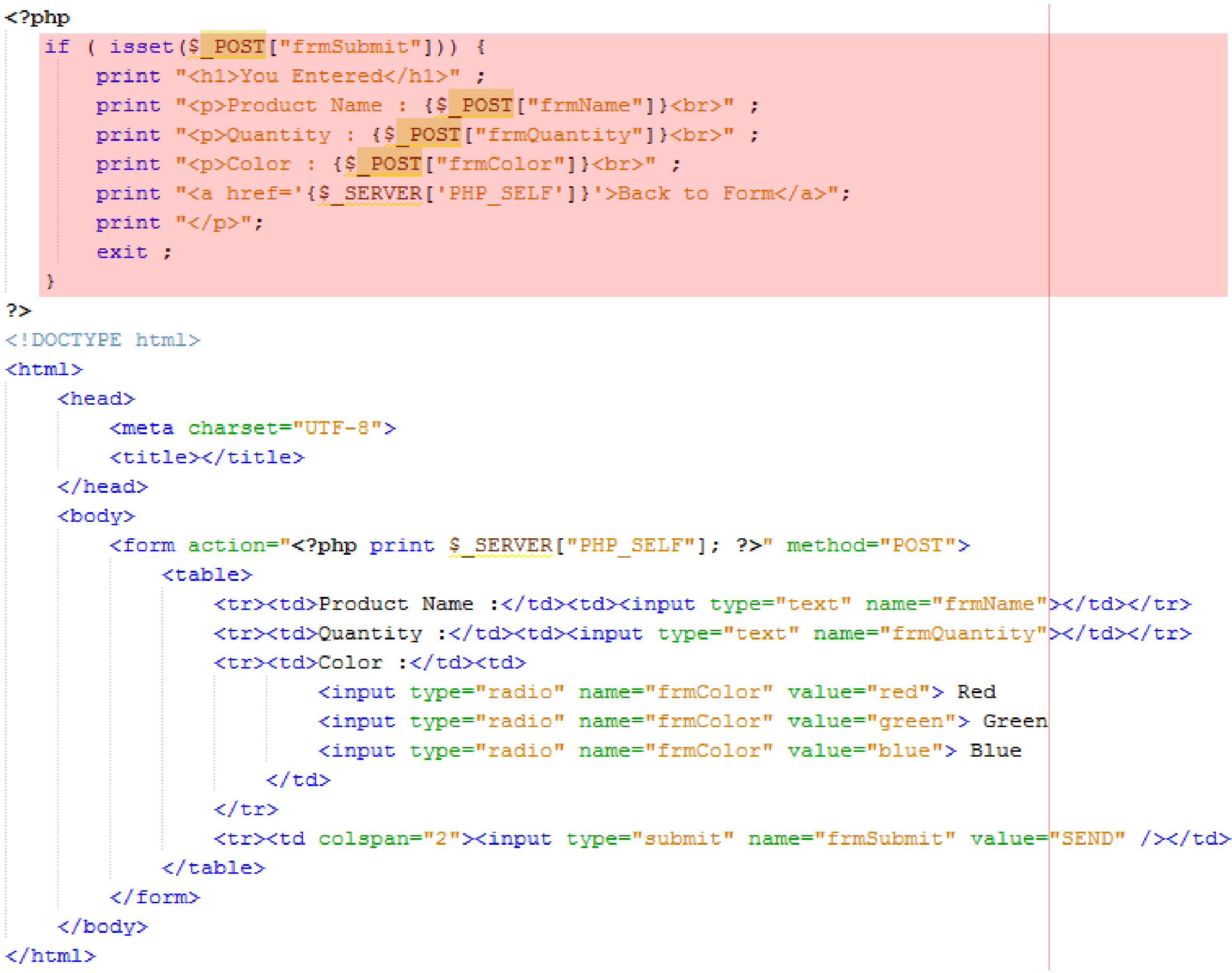
Single File Forms

- Form and processing file can be merge in the same file.
- **$_SERVER** is an array that shows the http headers, paths, and script locations.
- 'REQUEST_METHOD' : GET, POST, HEAD, PUT
- 'REMOTE_ADDR' : the IP address of the web client.

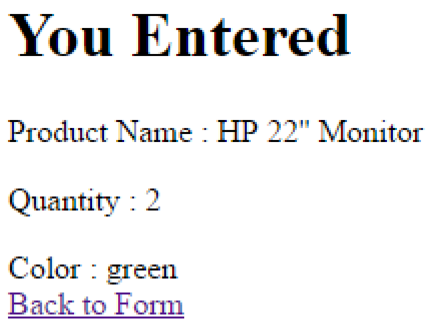


```
<?php
    // Processing Part

    if ( isset($_POST["submit"]) ) {
        // Process form here...
        print "<p>Processing Form : Name is {$_POST['userName']}</p>";
    }

?>
<!--  Form Part -->
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <form action="<?php print $_SERVER['PHP_SELF']; ?>" method="POST">
            <input type="text" name="userName">
            <input type="submit" name="submit" value="Send" />
        </form>
    </body>
</html>
```

An Example



```
<?php
  if ( isset($_POST["frmSubmit"])) {
      print "<h1>You Entered</h1>" ;
      print "<p>Product Name : {$_POST["frmName"]}<br>" ;
      print "<p>Quantity : {$_POST["frmQuantity"]}<br>" ;
      print "<p>Color : {$_POST["frmColor"]}<br>" ;
      print "<a href='{$_SERVER['PHP_SELF']}'>Back to Form</a>";
      print "</p>";
      exit ;
  }
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <form action="<?php print $_SERVER["PHP_SELF"]; ?>" method="POST">
            <table>
                <tr><td>Product Name :</td><td><input type="text" name="frmName"></td></tr>
                <tr><td>Quantity :</td><td><input type="text" name="frmQuantity"></td></tr>
                <tr><td>Color :</td><td>
                        <input type="radio" name="frmColor" value="red"> Red
                        <input type="radio" name="frmColor" value="green"> Green
                        <input type="radio" name="frmColor" value="blue"> Blue
                    </td>
                </tr>
                <tr><td colspan="2"><input type="submit" name="frmSubmit" value="SEND" /></td></tr>
            </table>
        </form>
    </body>
</html>
```

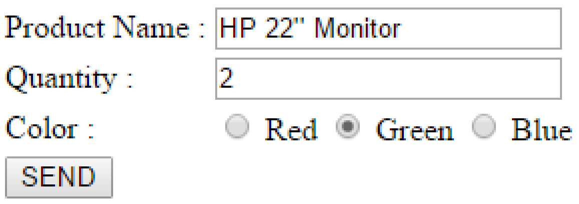


You Entered

Product Name : HP 22" Monitor

Quantity : 2

Color : green
Back to Form



Product Name : HP 22" Monitor

Quantity : 2

Color : ○ Red ● Green ○ Blue

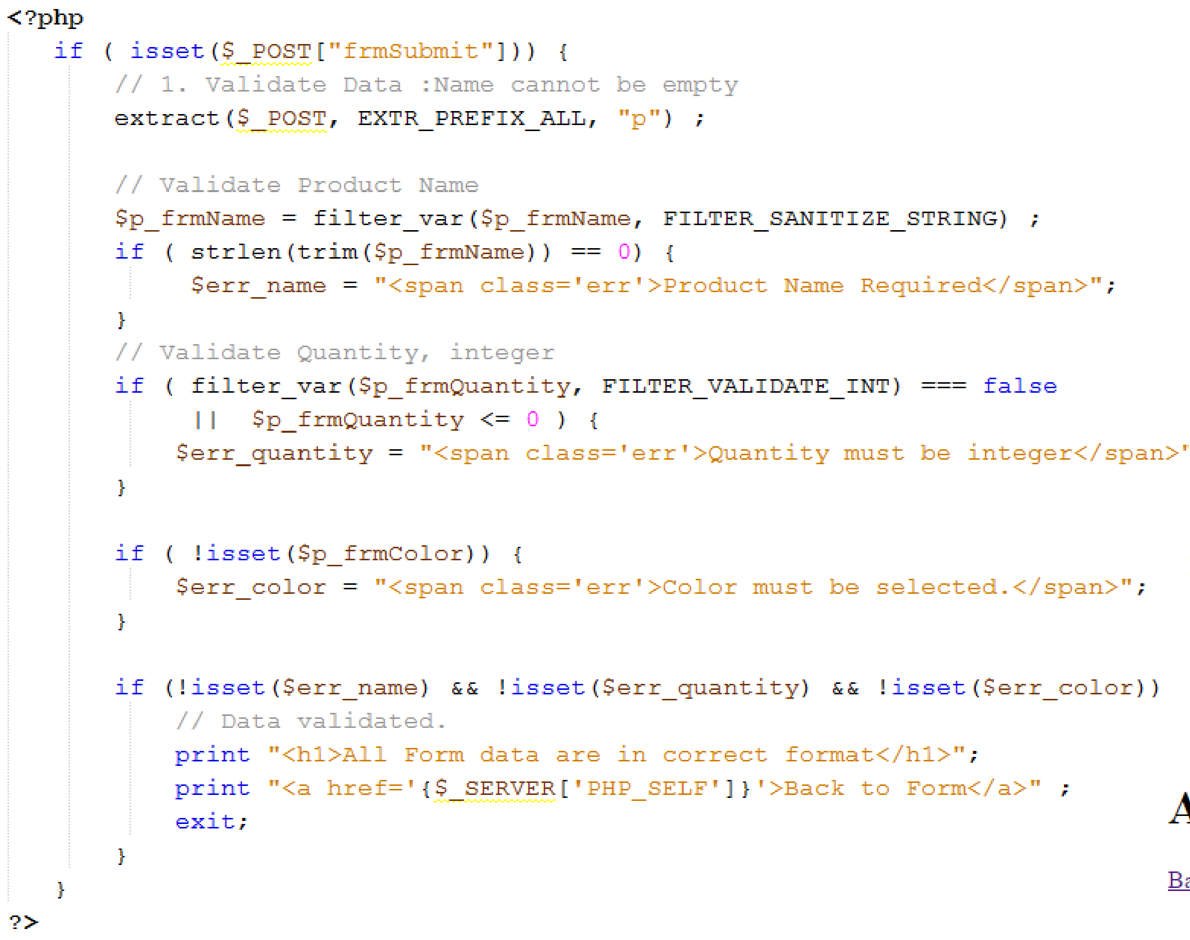
SEND

exit : to quit from the script execution.

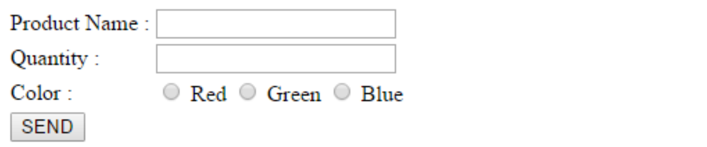
Errors and Sticky Forms

- It is a method to prevent retyping when an error occured in the form.

Processing Part

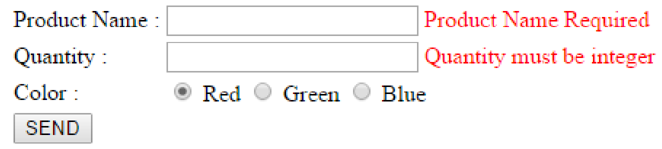


```php
<?php
    if ( isset($_POST["frmSubmit"])) {
        // 1. Validate Data :Name cannot be empty
        extract($_POST, EXTR_PREFIX_ALL, "p") ;

        // Validate Product Name
        $p_frmName = filter_var($p_frmName, FILTER_SANITIZE_STRING) ;
        if ( strlen(trim($p_frmName)) == 0) {
             $err_name = "<span class='err'>Product Name Required</span>";
        }
        // Validate Quantity, integer
        if ( filter_var($p_frmQuantity, FILTER_VALIDATE_INT) === false
             ||  $p_frmQuantity <= 0 ) {
            $err_quantity = "<span class='err'>Quantity must be integer</span>'
        }

        if ( !isset($p_frmColor)) {
            $err_color = "<span class='err'>Color must be selected.</span>";
        }

        if (!isset($err_name) && !isset($err_quantity) && !isset($err_color))
            // Data validated.
            print "<h1>All Form data are in correct format</h1>";
            print "<a href='{$_SERVER['PHP_SELF']}'>Back to Form</a>" ;
            exit;
        }
    }
?>
```



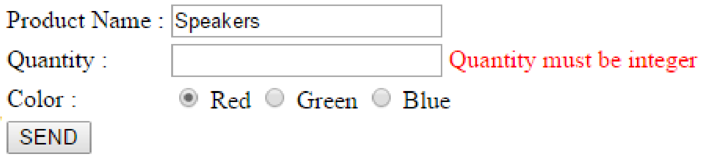
Product Name :	
Quantity :	
Color :	○ Red ○ Green ○ Blue

SEND



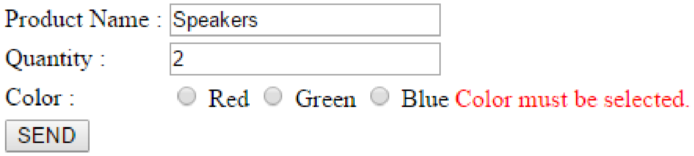
Product Name :		Product Name Required
Quantity :		Quantity must be integer
Color :	● Red ○ Green ○ Blue	

SEND



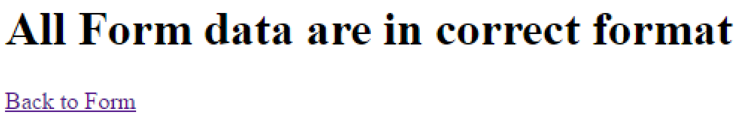
Product Name :	Speakers	
Quantity :		Quantity must be integer
Color :	● Red ○ Green ○ Blue	

SEND



Product Name :	Speakers
Quantity :	2
Color :	○ Red ○ Green ○ Blue Color must be selected.

SEND

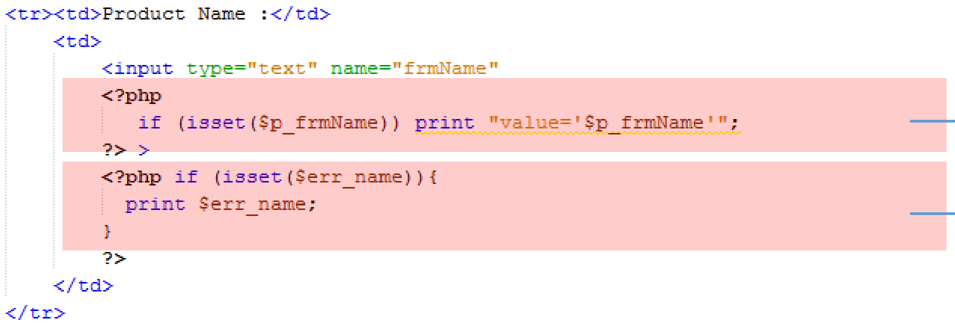


All Form data are in correct format

Back to Form

Errors and Sticky Forms

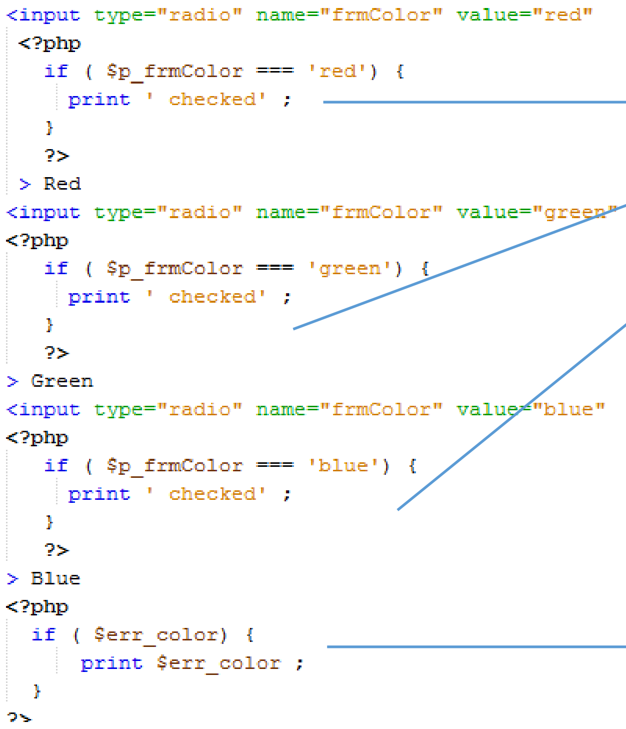
Form Part



```
<tr><td>Product Name :</td>
    <td>
        <input type="text" name="frmName"
        <?php
           if (isset($p_frmName)) print "value='$p_frmName'";
        ?> >
        <?php if (isset($err_name)){
          print $err_name;
        }
        ?>
    </td>
</tr>
```

remember the value entered before

display error message



```
<input type="radio" name="frmColor" value="red"
 <?php
   if ( $p_frmColor === 'red') {
     print ' checked' ;
   }
   ?>
 > Red
<input type="radio" name="frmColor" value="green"
<?php
   if ( $p_frmColor === 'green') {
     print ' checked' ;
   }
   ?>
> Green
<input type="radio" name="frmColor" value="blue"
<?php
   if ( $p_frmColor === 'blue') {
     print ' checked' ;
   }
   ?>
> Blue
<?php
  if ( $err_color) {
     print $err_color ;
  }
?>
```

remember which radio button is selected.

display error message for the radio buttons