HTML Form Processing with PHP

Tips, Tricks, and Bad Ideas

Who is this guy?

Joe Ferguson - joe@joeferguson.me

Professionally

- Web Developer at RocketFuel
- PHP / LAMP Focused

Semi Professional

- Co-Organizer for MemphisPHP.org
- MidsouthMakers - Hackerspace Leader
- HACKmemphis.com Organizer

Types of Forms

- Login Form
- Contact Form
- Questionnaire
- Sign Up Form
- Order Form

Every application that interacts with the user uses forms in some manner.

The Problem (Use Case)

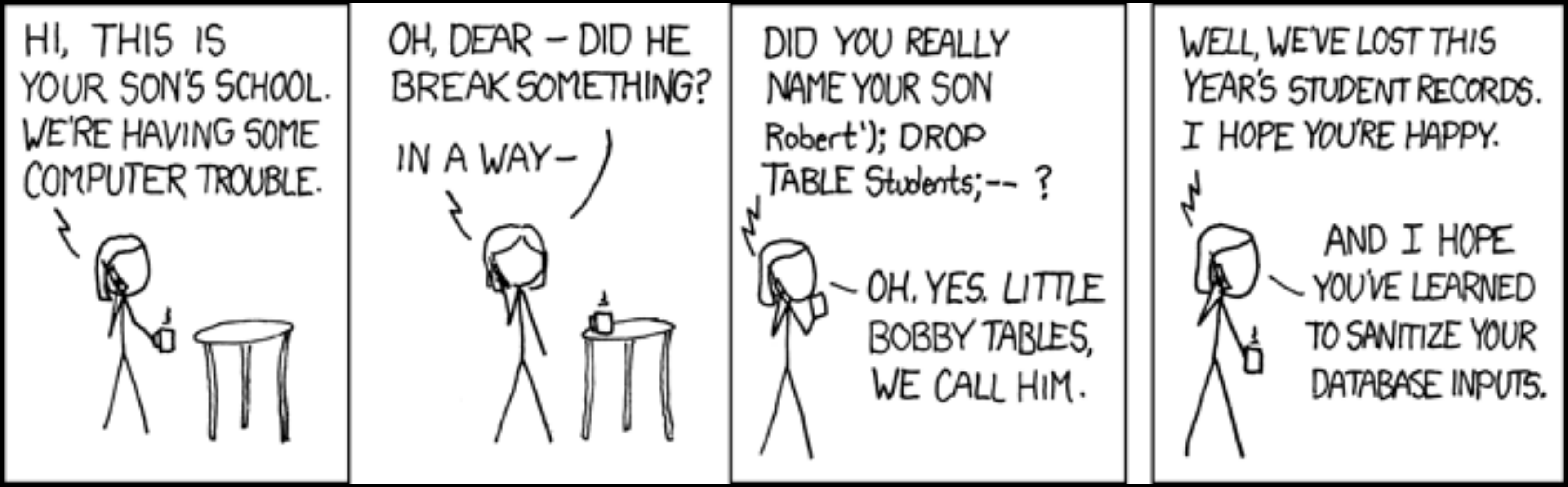
I have a PHP Application that needs input:

How do I…

- **safely**
- **securely**
- **reliably**

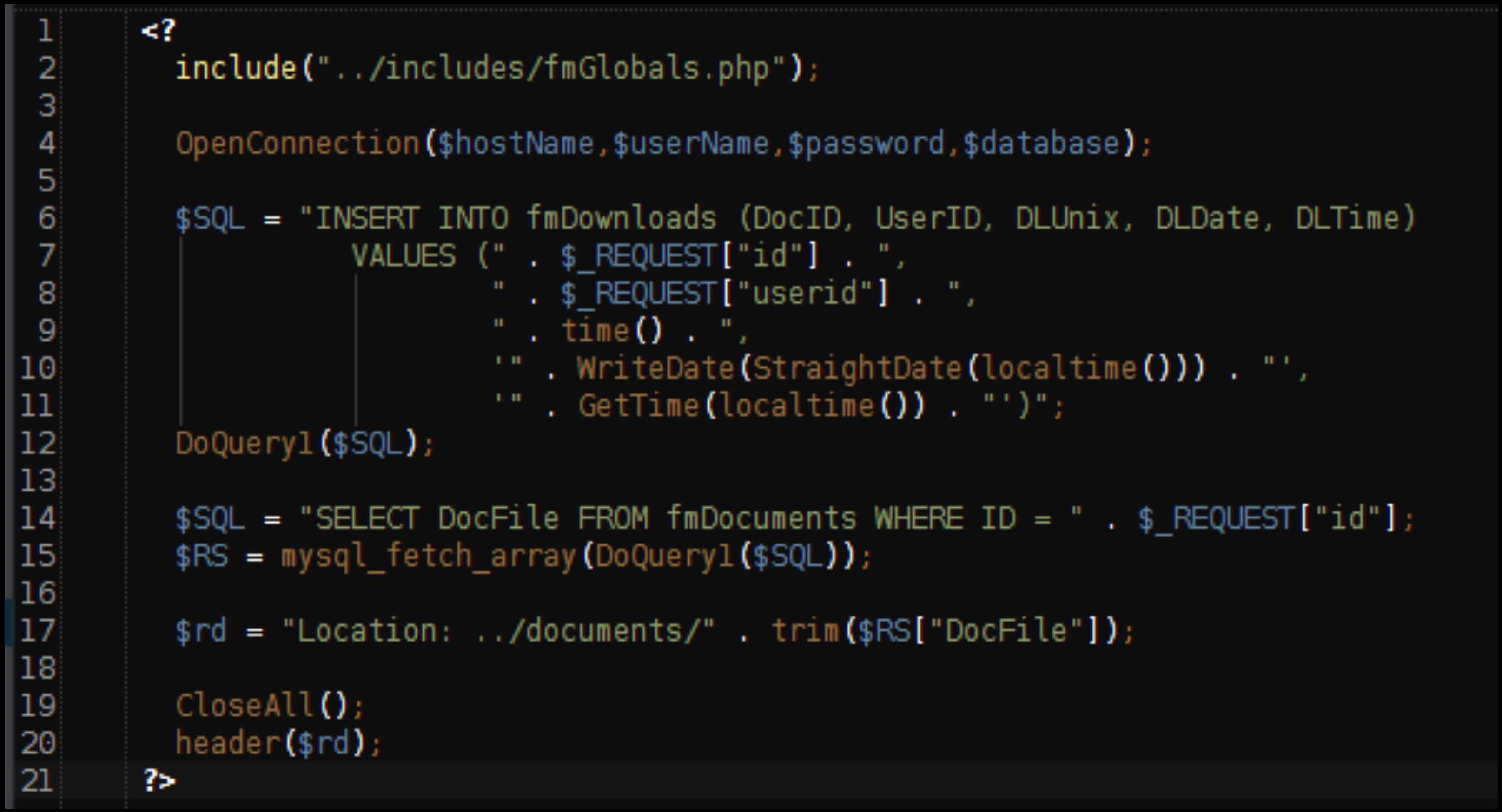
... get that input or other data from my users?

Little Bobby Tables...


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://xkcd.com/327/

Bad Ideas:

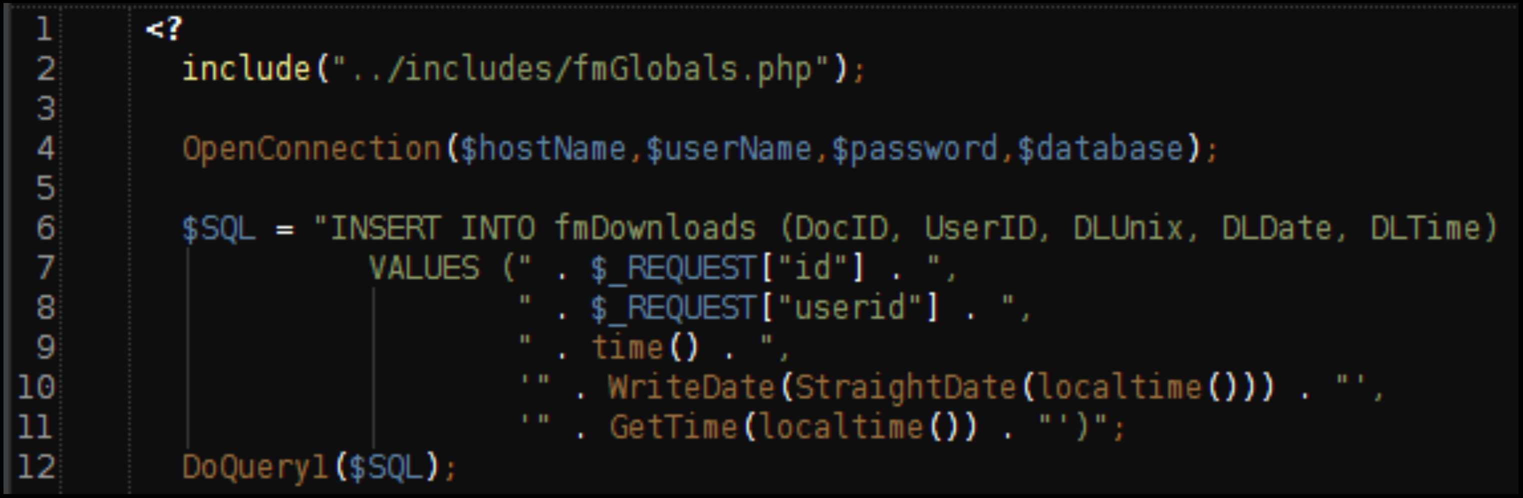


```
<?
  include("../includes/fmGlobals.php");

  OpenConnection($hostName,$userName,$password,$database);

  $SQL = "INSERT INTO fmDownloads (DocID, UserID, DLUnix, DLDate, DLTime)
            VALUES (" . $_REQUEST["id"] . ",
                    " . $_REQUEST["userid"] . ",
                    " . time() . ",
                    '" . WriteDate(StraightDate(localtime())) . "',
                    '" . GetTime(localtime()) . "')";
  DoQuery1($SQL);

  $SQL = "SELECT DocFile FROM fmDocuments WHERE ID = " . $_REQUEST["id"];
  $RS = mysql_fetch_array(DoQuery1($SQL));

  $rd = "Location: ../documents/" . trim($RS["DocFile"]);

  CloseAll();
  header($rd);
?>
```

Code stolen from someone I follow on twitter that was pointing out bad code…

Whiskey...

What is this code even doing?!



```php
<?
  include("../includes/fmGlobals.php");

  OpenConnection($hostName,$userName,$password,$database);

  $SQL = "INSERT INTO fmDownloads (DocID, UserID, DLUnix, DLDate, DLTime)
             VALUES (" . $_REQUEST["id"] . ",
                     " . $_REQUEST["userid"] . ",
                     " . time() . ",
                     '" . WriteDate(StraightDate(localtime())) . "',
                     '" . GetTime(localtime()) . "')";
  DoQuery1($SQL);
```

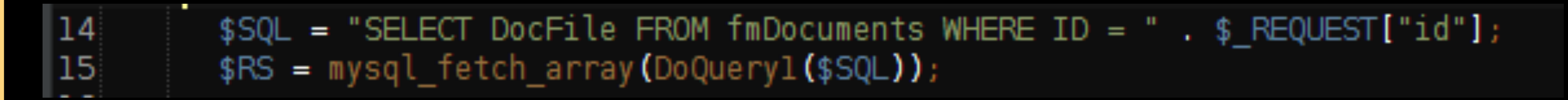
open a database connection...

insert some data...

run the insert query….

Tango...

What is this code even doing?!



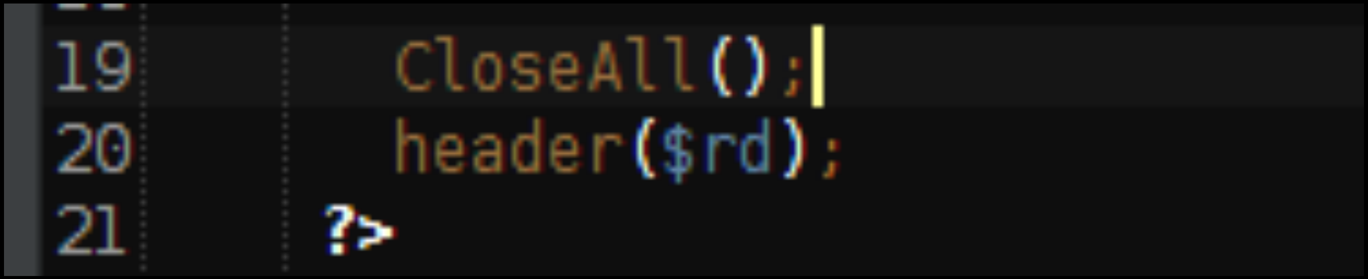
```
    $SQL = "SELECT DocFile FROM fmDocuments WHERE ID = " . $_REQUEST["id"];
    $RS = mysql_fetch_array(DoQuery1($SQL));
```

create a new query…

run the new query and return data...

Foxtrot...

What is this code even doing?!



```
    CloseAll();
    header($rd);
?>
```

close all open connections…

get us the heck out of here...

Why is the code bad?

- Using PHP short tags
  - Must be enabled in php.ini use sparingly if at all
- No data sanitization
  - Security?! never important
- No data validation
  - who cares! he trusts his users
- Directly saving user input into a database
  - Begging for a little bobby tables incident
- Not using prepared statements
  - PDO - it's the wave of the future!

How do I safely get form data?

Sanitize it!

How do I securely get form data?

HTTPS

(SSL)

How do I reliably get form data?

Validate it!

Assumptions for our examples

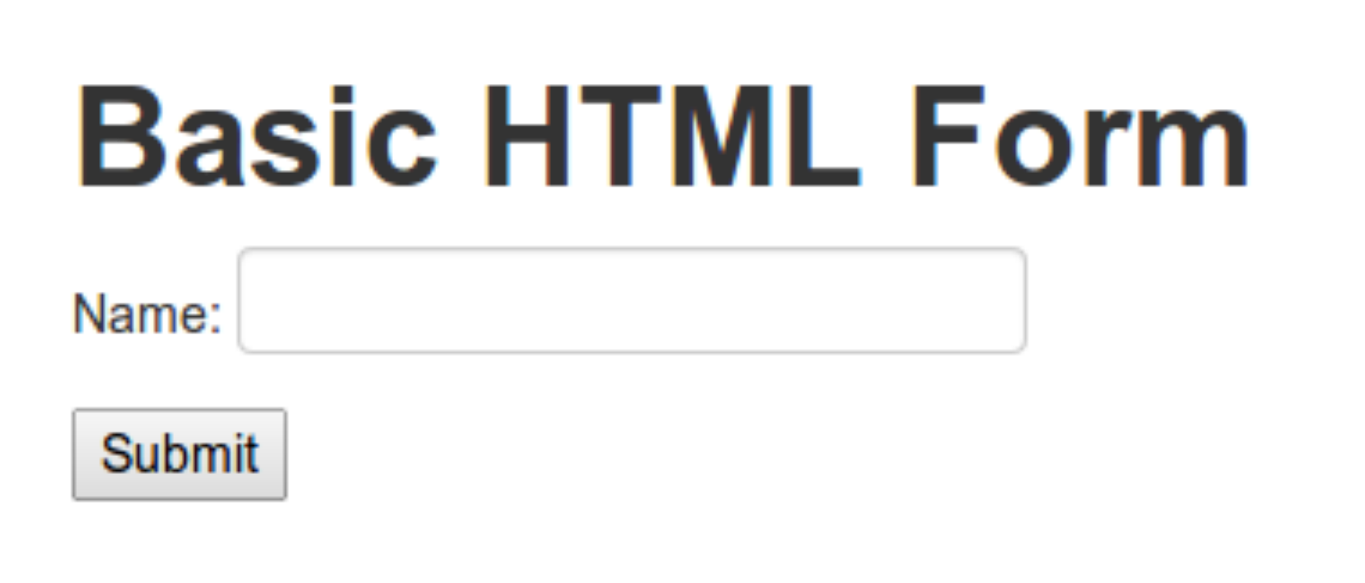
- Existing JavaScript validation that input exists.
- You are using a POST or GET method
- You’re using SSL

You can find these slides and examples:

https://github.com/Svpernova09

Oh PHP, you so easy….

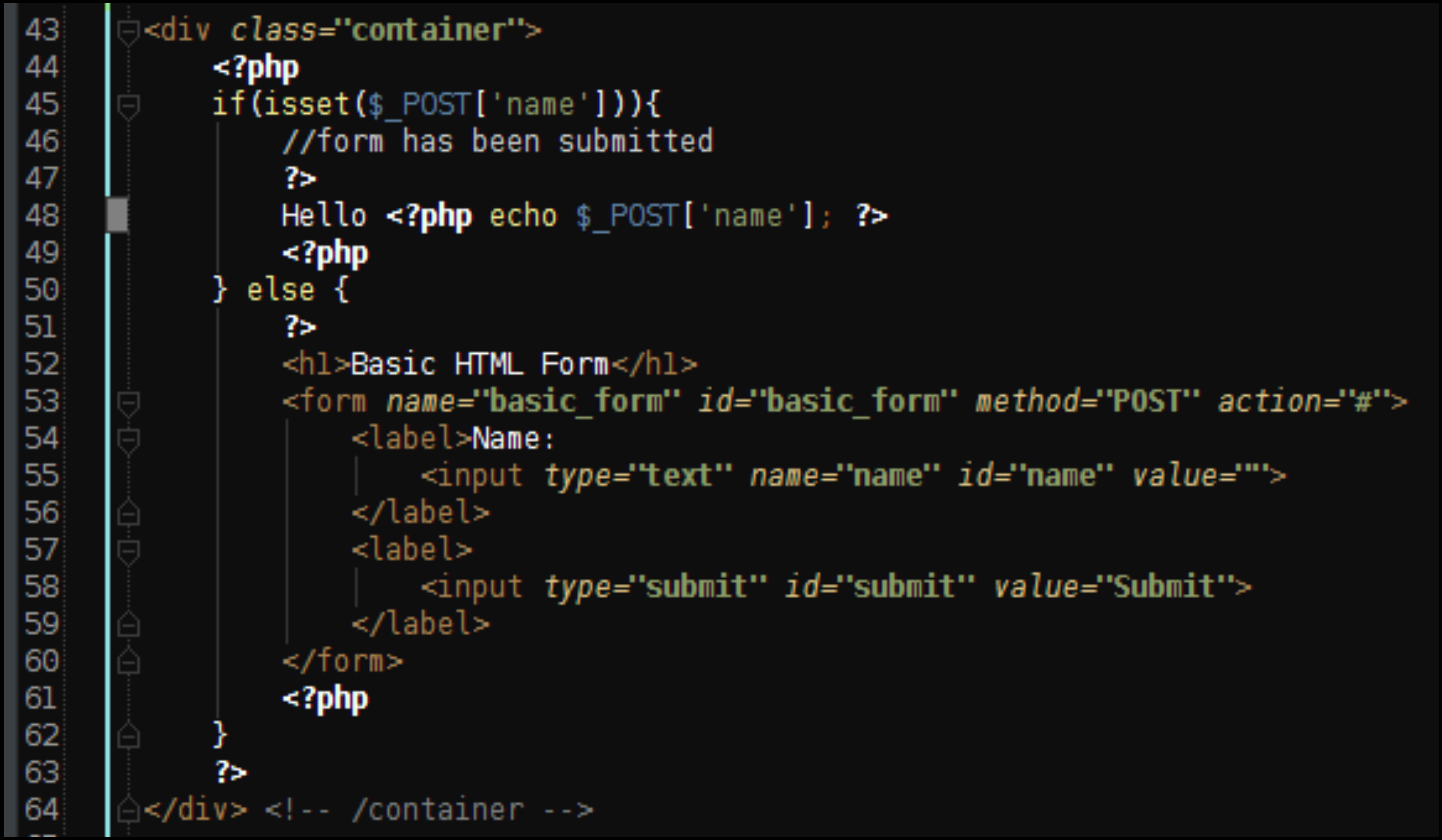
Create our form:



Basic HTML Form

Oh PHP, you so easy….

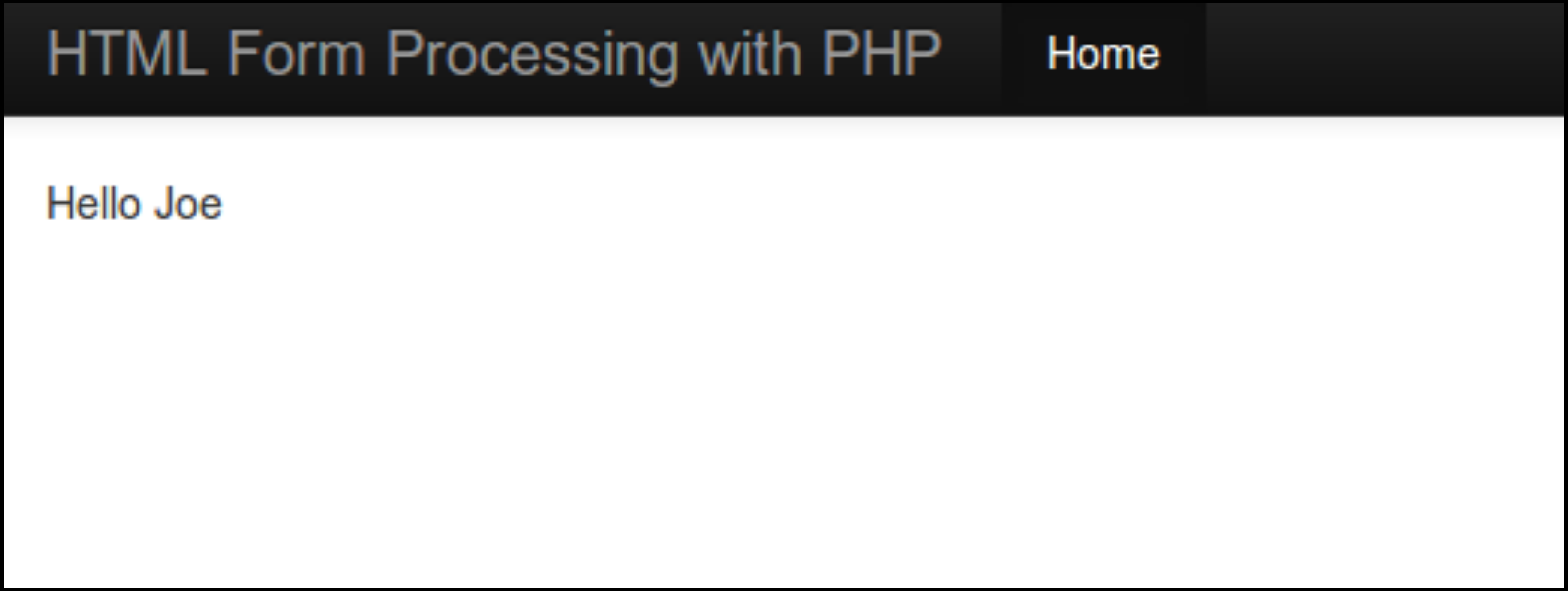
Add our form processing:



```
<div class="container">
    <?php
    if(isset($_POST['name'])){
        //form has been submitted
        ?>
        Hello <?php echo $_POST['name']; ?>
        <?php
    } else {
        ?>
        <h1>Basic HTML Form</h1>
        <form name="basic_form" id="basic_form" method="POST" action="#">
            <label>Name:
                <input type="text" name="name" id="name" value="">
            </label>
            <label>
                <input type="submit" id="submit" value="Submit">
            </label>
        </form>
        <?php
    }
    ?>
</div> <!-- /container -->
```

I’m REALLY good at this PHP thing

Output when the form has been submitted:



Hello Joe

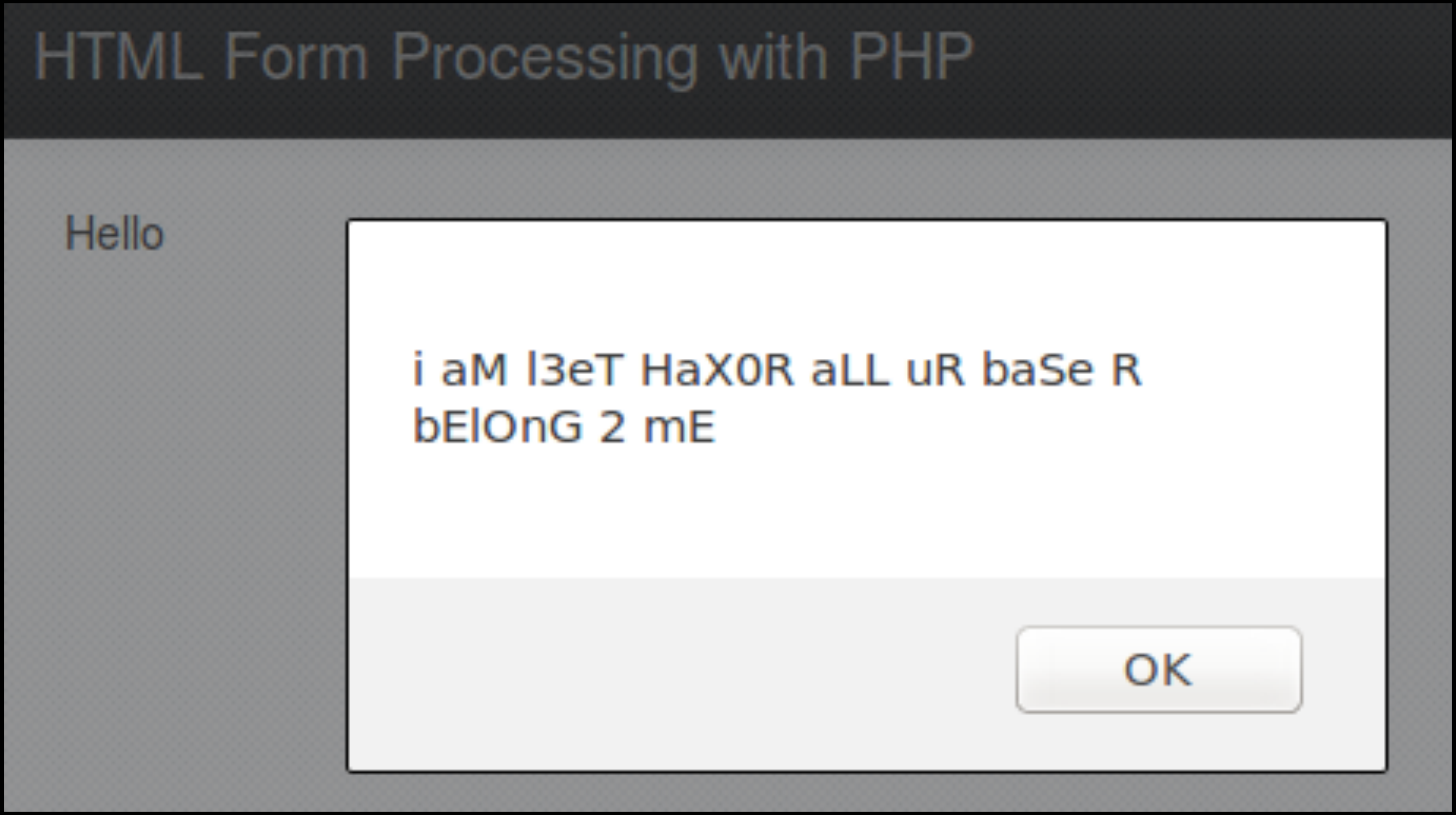
What if ....

...someone clever comes along?

...someone malicious comes along?

Wait a minute….

HOW DID THIS HAPPEN!!!111>>!??



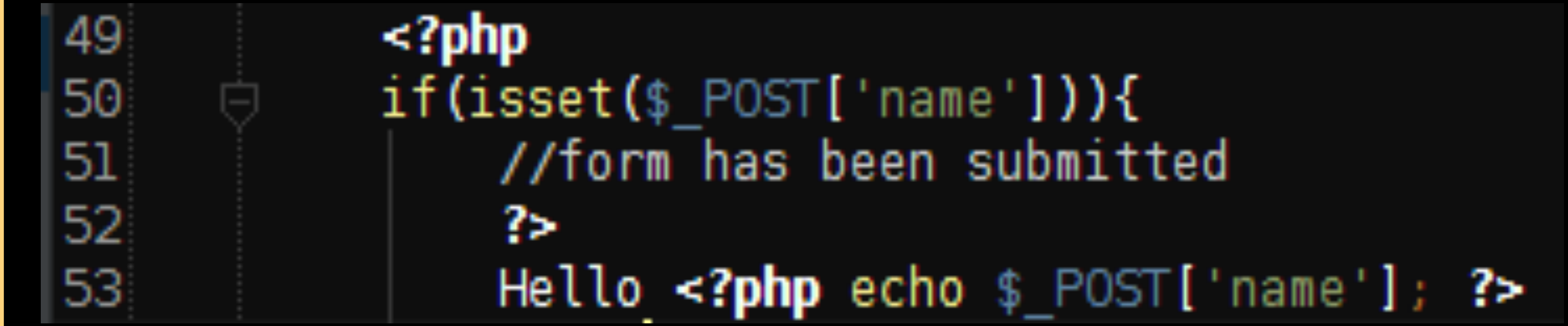
HTML Form Processing with PHP

Hello

i aM l3eT HaX0R aLL uR baSe R bElOnG 2 mE

It was all going so beautifully...

Where did we go wrong?



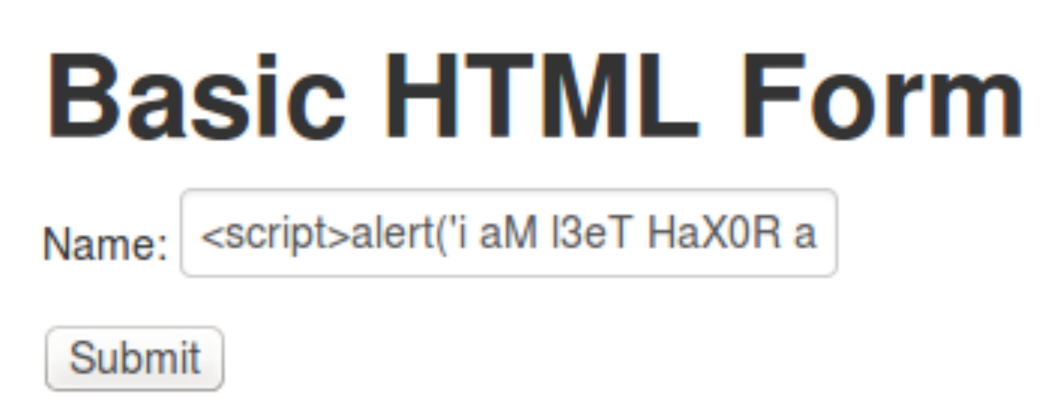
```
49  <?php
50  if(isset($_POST['name'])){
51      //form has been submitted
52      ?>
53      Hello <?php echo $_POST['name']; ?>
```

Line 53: We used the raw input from the user.

This leaves us WIDE open to many attacks.

WTF! But PHP is so EASY!

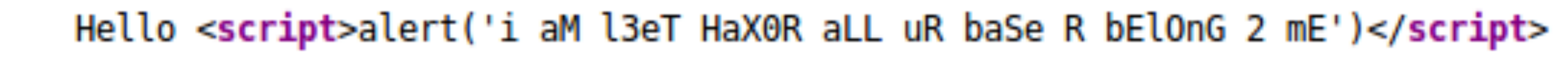
The user put JavaScript in our field:



Basic HTML Form

Name: <script>alert('i aM l3eT HaX0R a

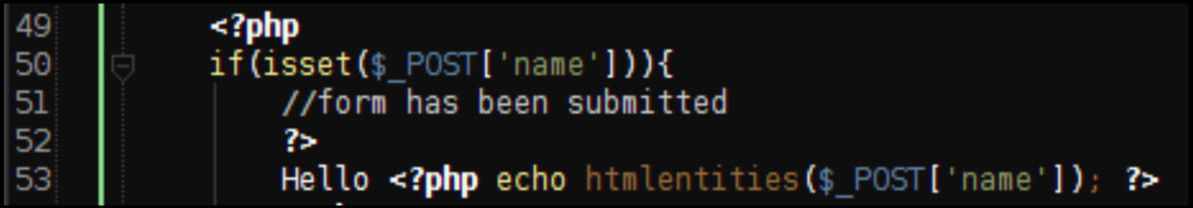
Since we just echoed the input, we injected the user's JavaScript directly into our page:



```
Hello <script>alert('i aM l3eT HaX0R aLL uR baSe R bElOnG 2 mE')</script>
```

How do we prevent such attacks?

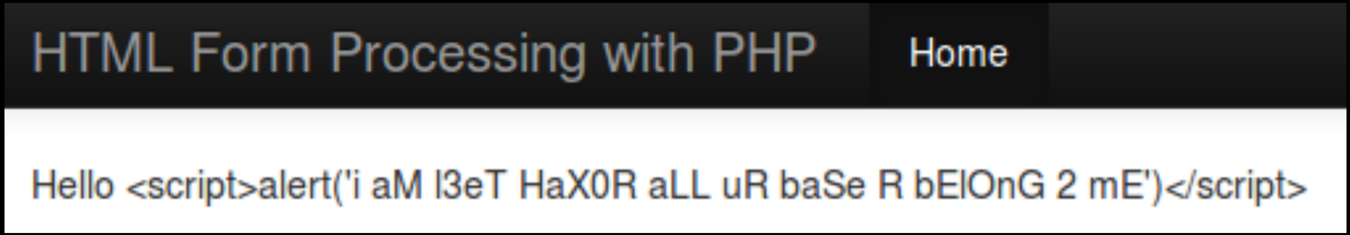
We must go Back... to line 53!



```
49      <?php
50      if(isset($_POST['name'])){
51          //form has been submitted
52          ?>
53          Hello <?php echo htmlentities($_POST['name']); ?>
```

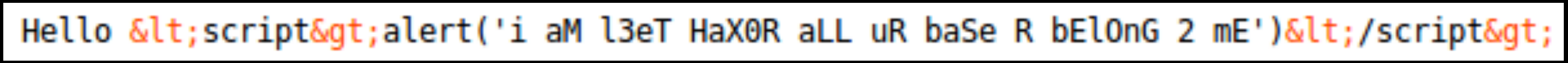
htmlentities() = don't parse as HTML

Browser:



Hello <script>alert('i aM l3eT HaX0R aLL uR baSe R bElOnG 2 mE')</script>

Source:



```
Hello &lt;script&gt;alert('i aM l3eT HaX0R aLL uR baSe R bEl0nG 2 mE')&lt;/script&gt;
```

Different ways to sanitize data

Data Filters

- Validate Filters
    - FILTER_VALIDATE_EMAIL
    - FILTER_VALIDATE_INT
- Sanitize Filters
    - FILTER_SANITIZE_STRING
    - FILTER_SANITIZE_NUMBER_INT
- htmlspecialchars()
- htmlentities()

htmlentities OR htmlspecialchars ?

- **htmlspecialchars()** will encode only characters that have special significance in HTML
  - Example:
    - echo **htmlspecialchars**('<Il était une fois un être>.');
    - // Outputs: &lt;Il était une fois un être&gt;.
- **htmlentities()** all characters which have HTML character entity equivalents are translated into these entities
  - Example:
    - echo **htmlentities**('<Il était une fois un être>.');
    - // Outputs: &lt;Il &eacute;tait une fois un &ecirc;tre&gt;

GREAT Explanation: http://stackoverflow.com/questions/46483/htmlentities-vs-htmlspecialchars

filter_var or html* ?

Data Filters are newer than html* functions

They are the “better” way.

They also depend on newer versions of PHP

But I’m a small shop, & trust users

You can’t trust input from ANYONE!

- You
- Your Parents
- Your Co Workers
- Your Boss
- Your App’s Users
- Other Apps
- The Internet

Should I sanitize or validate?

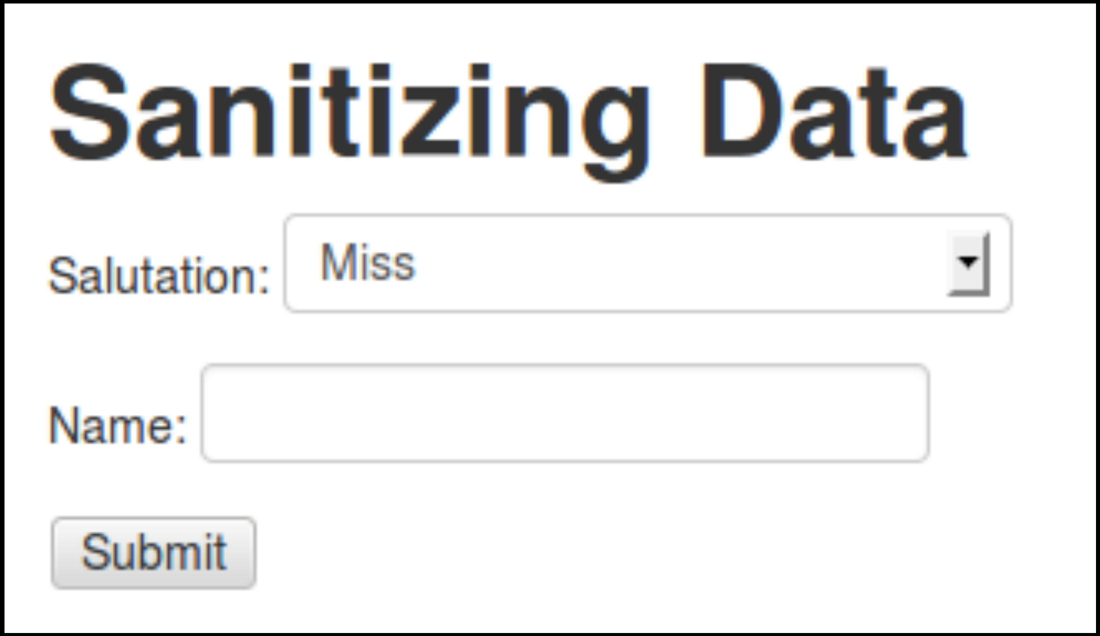
BOTH!

SANITIZING your data is removing (or encoding) a specific set of characters from your data.

VALIDATING your data is making sure the data is in the format you expect to be in.

Sanitizing Data

Our new form (Example: 02 - Sanitizing Data)



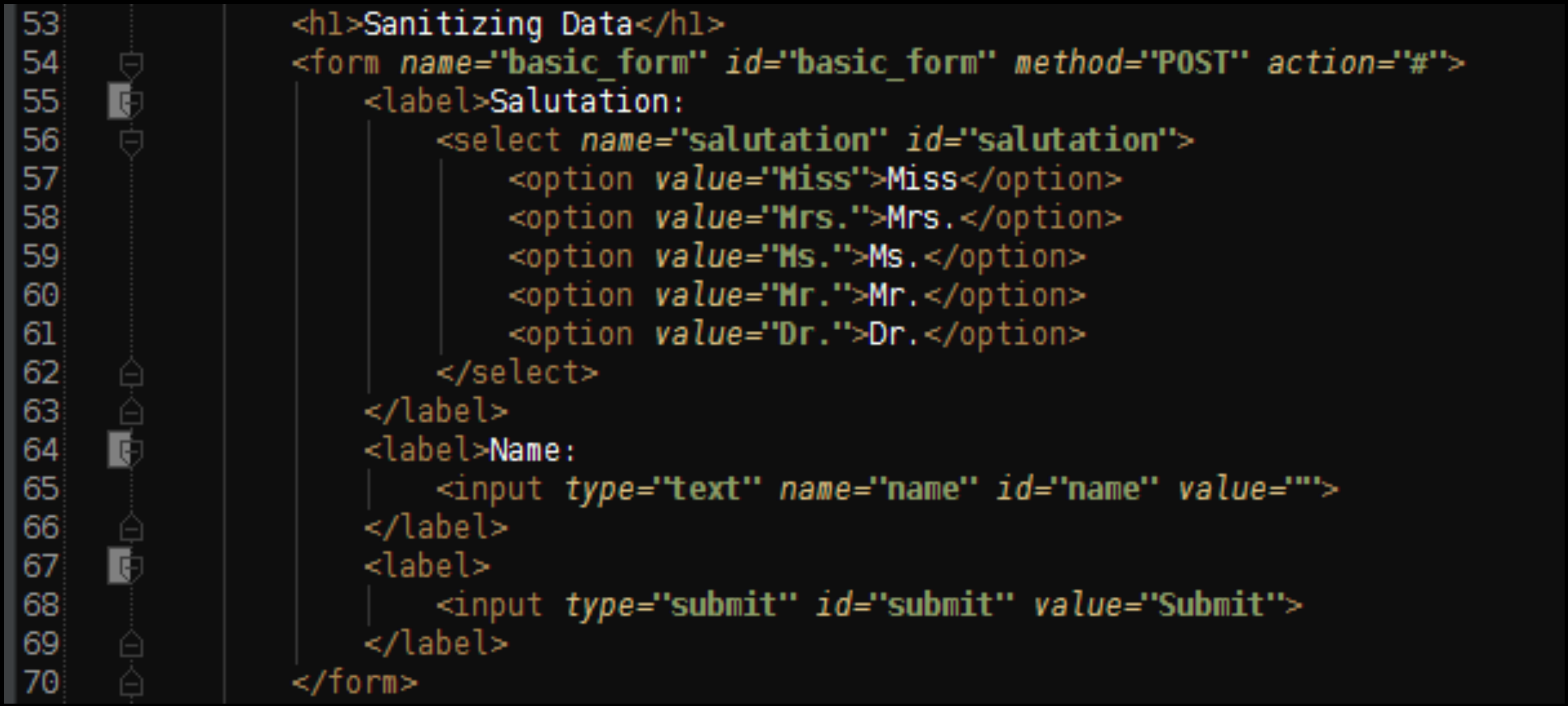
Sanitizing Data

Salutation: Miss

Name:

Sanitizing Data

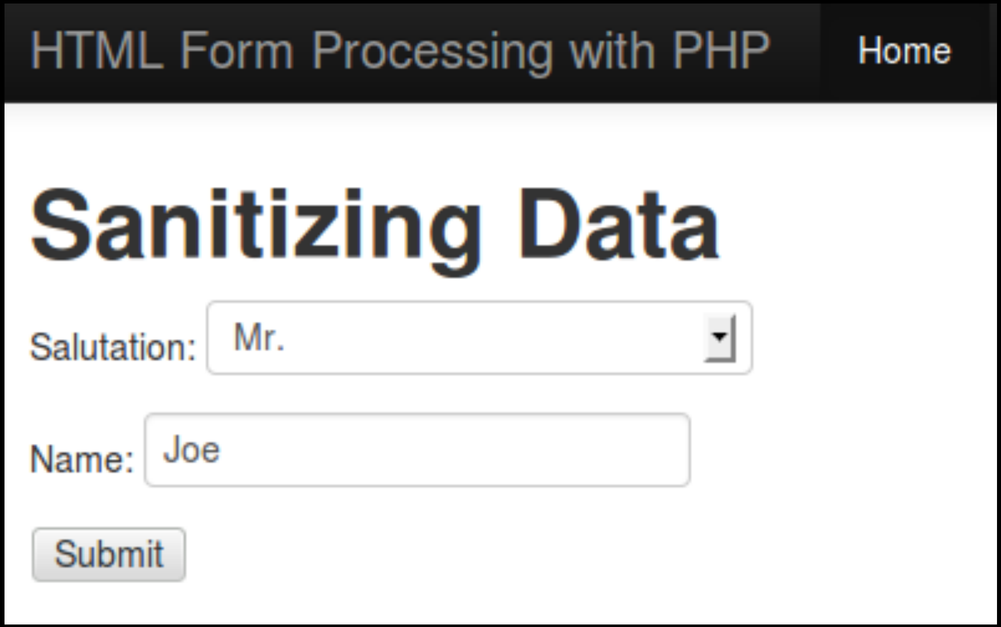
Form Code View (Example: 02 - Sanitizing Data)



```
<h1>Sanitizing Data</h1>
<form name="basic_form" id="basic_form" method="POST" action="#">
    <label>Salutation:
        <select name="salutation" id="salutation">
            <option value="Miss">Miss</option>
            <option value="Mrs.">Mrs.</option>
            <option value="Ms.">Ms.</option>
            <option value="Mr.">Mr.</option>
            <option value="Dr.">Dr.</option>
        </select>
    </label>
    <label>Name:
        <input type="text" name="name" id="name" value="">
    </label>
    <label>
        <input type="submit" id="submit" value="Submit">
    </label>
</form>
```

Sanitizing Data - Data Entry

This is the type of input we want...



HTML Form Processing with PHP

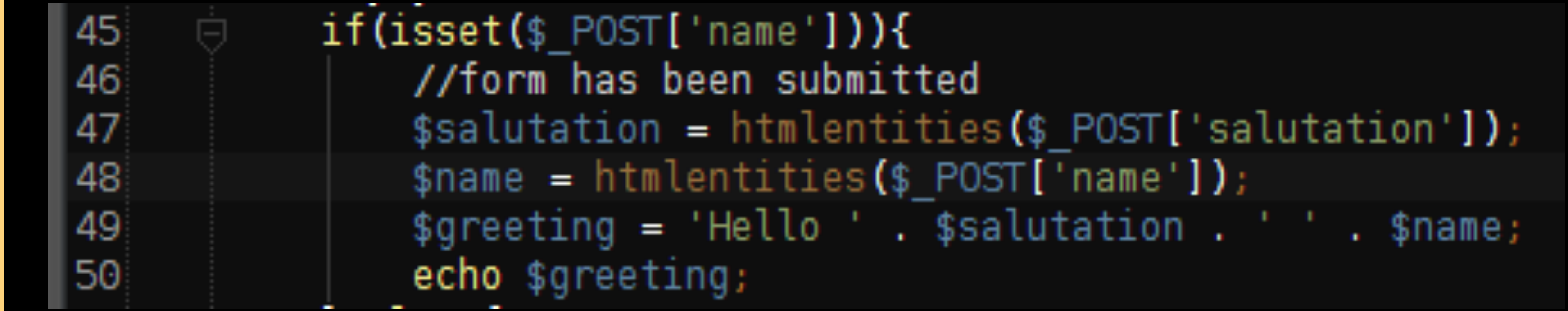
Sanitizing Data

Salutation: Mr.

Name: Joe

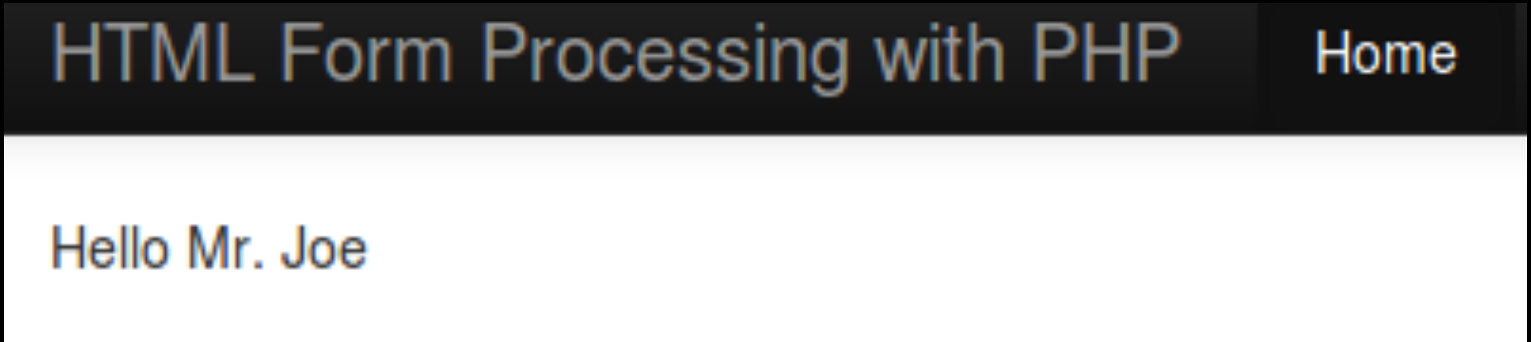
Sanitizing Data - Data Entry

Make sure no malicious code is in the data...



```
if(isset($_POST['name'])){
    //form has been submitted
    $salutation = htmlentities($_POST['salutation']);
    $name = htmlentities($_POST['name']);
    $greeting = 'Hello ' . $salutation . ' ' . $name;
    echo $greeting;
```

Output:

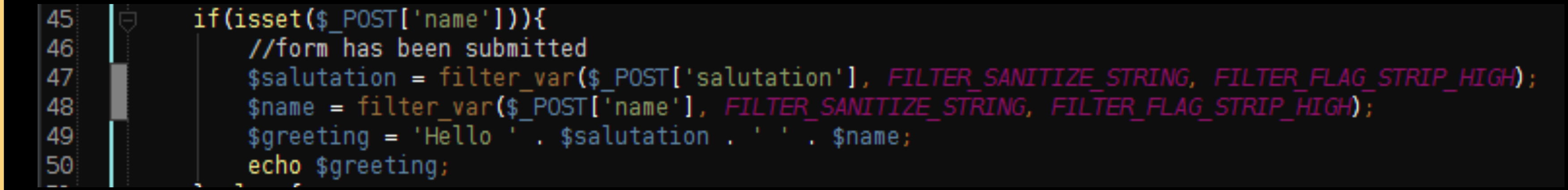


HTML Form Processing with PHP

Hello Mr. Joe

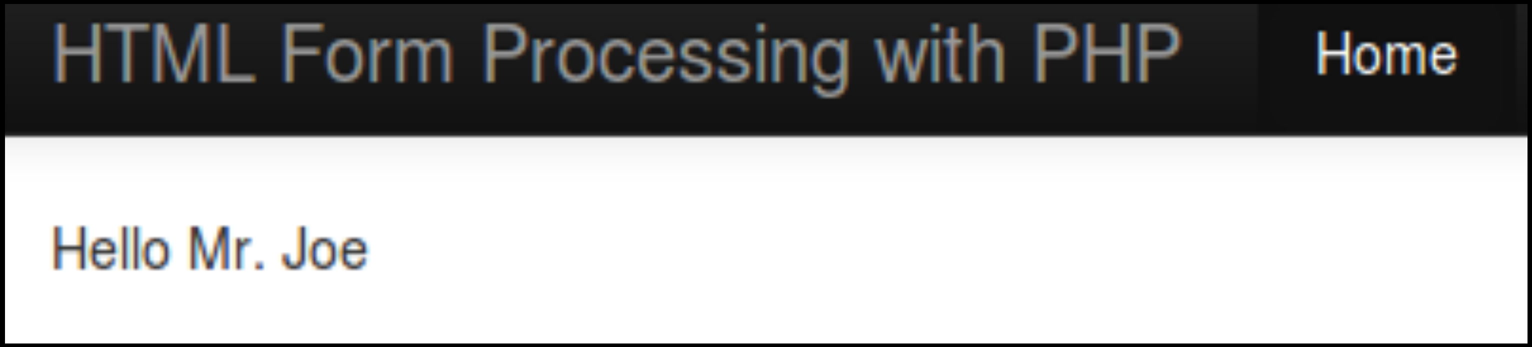
Sanitizing Data - BETTER way

The more modern way: filter_var with flags:



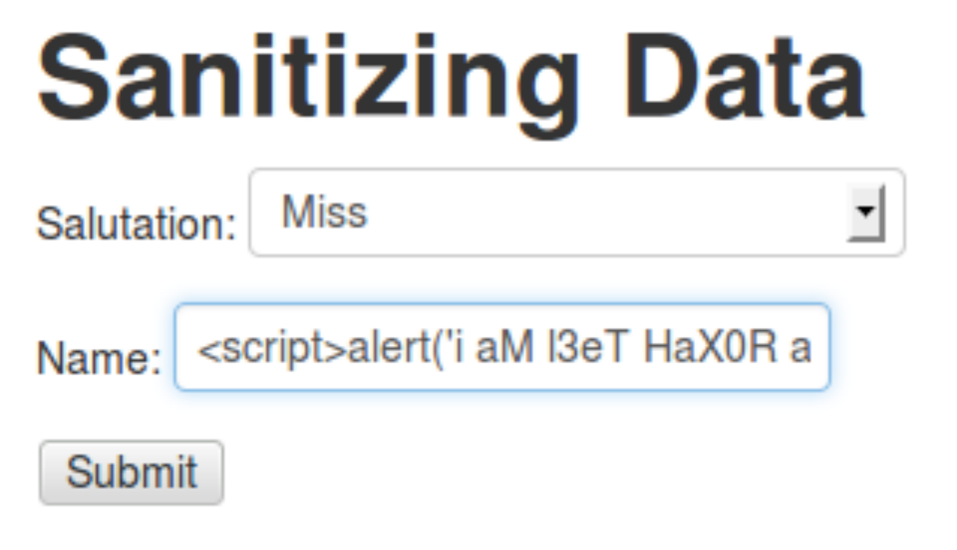
```
if(isset($_POST['name'])){
    //form has been submitted
    $salutation = filter_var($_POST['salutation'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $name = filter_var($_POST['name'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $greeting = 'Hello ' . $salutation . ' ' . $name;
    echo $greeting;
```

Output is the same:



```
Hello Mr. Joe
```

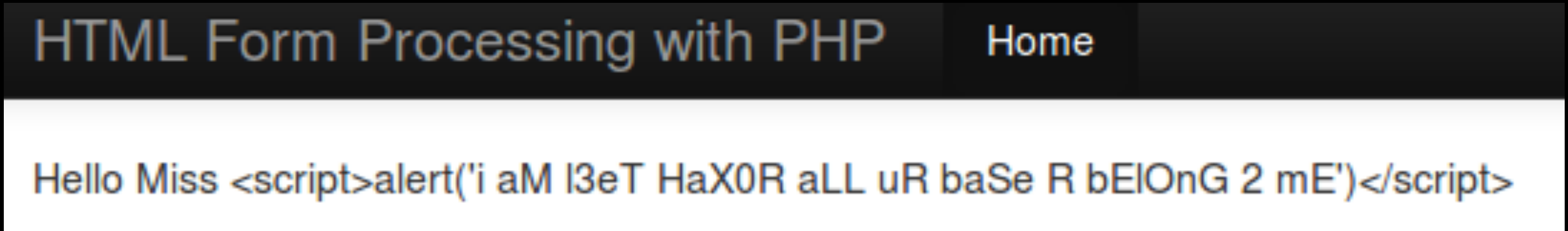
Testing our Data Sanitization



Sanitizing Data

Salutation: Miss

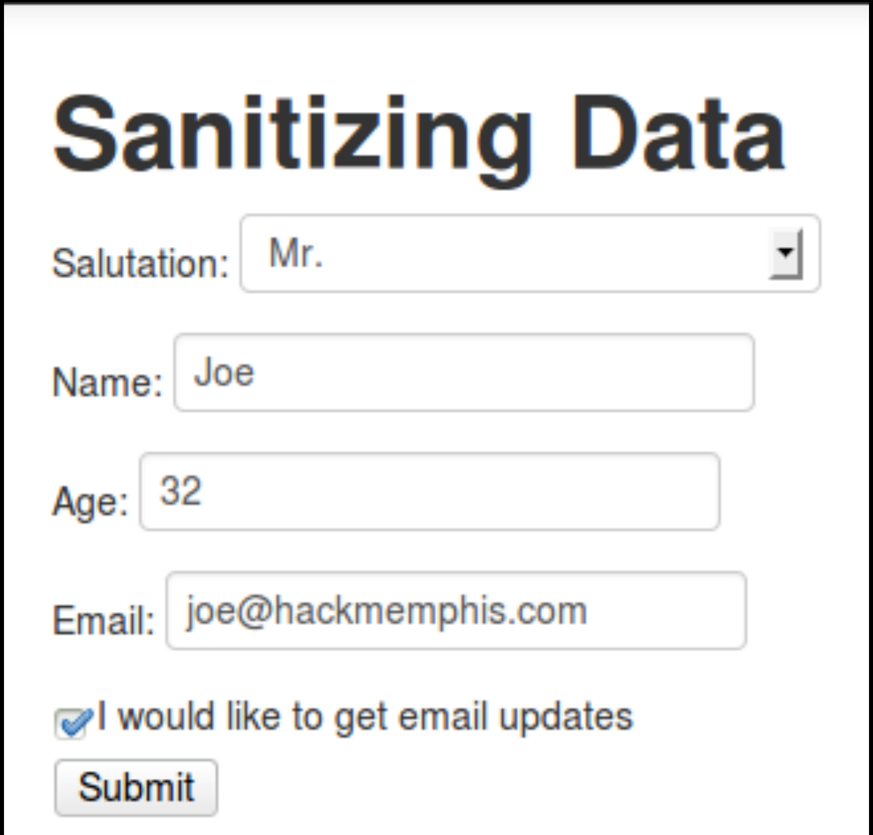
Name: <script>alert('i aM l3eT HaX0R a



Hello Miss <script>alert('i aM l3eT HaX0R aLL uR baSe R bElOnG 2 mE')</script>

What about check boxes?

Let's add a check box to our form



Sanitizing Data

Salutation: Mr.

Name: Joe

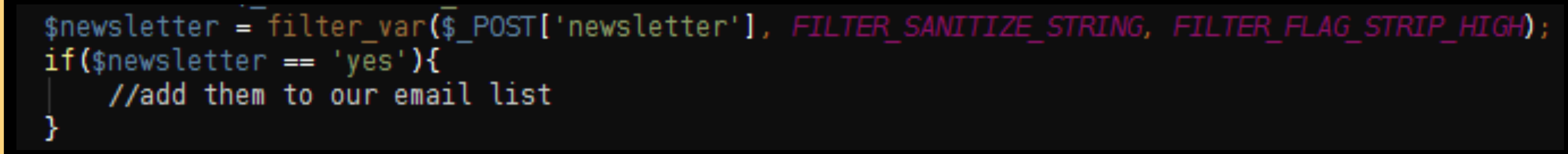
Age: 32

Email: joe@hackmemphis.com

☑ I would like to get email updates

Sanitize everything!

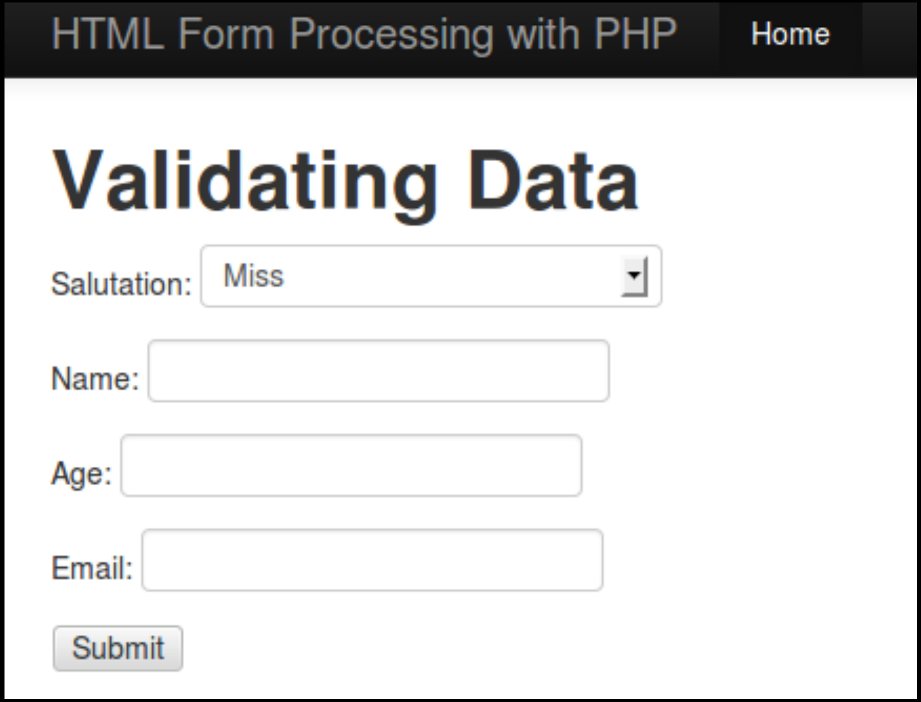
Even check boxes should be sanitized



```
$newsletter = filter_var($_POST['newsletter'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
if($newsletter == 'yes'){
    //add them to our email list
}
```

Validating User Input

Our new form (Example: 03 - Validating Data)



Validating Data

Salutation: Miss

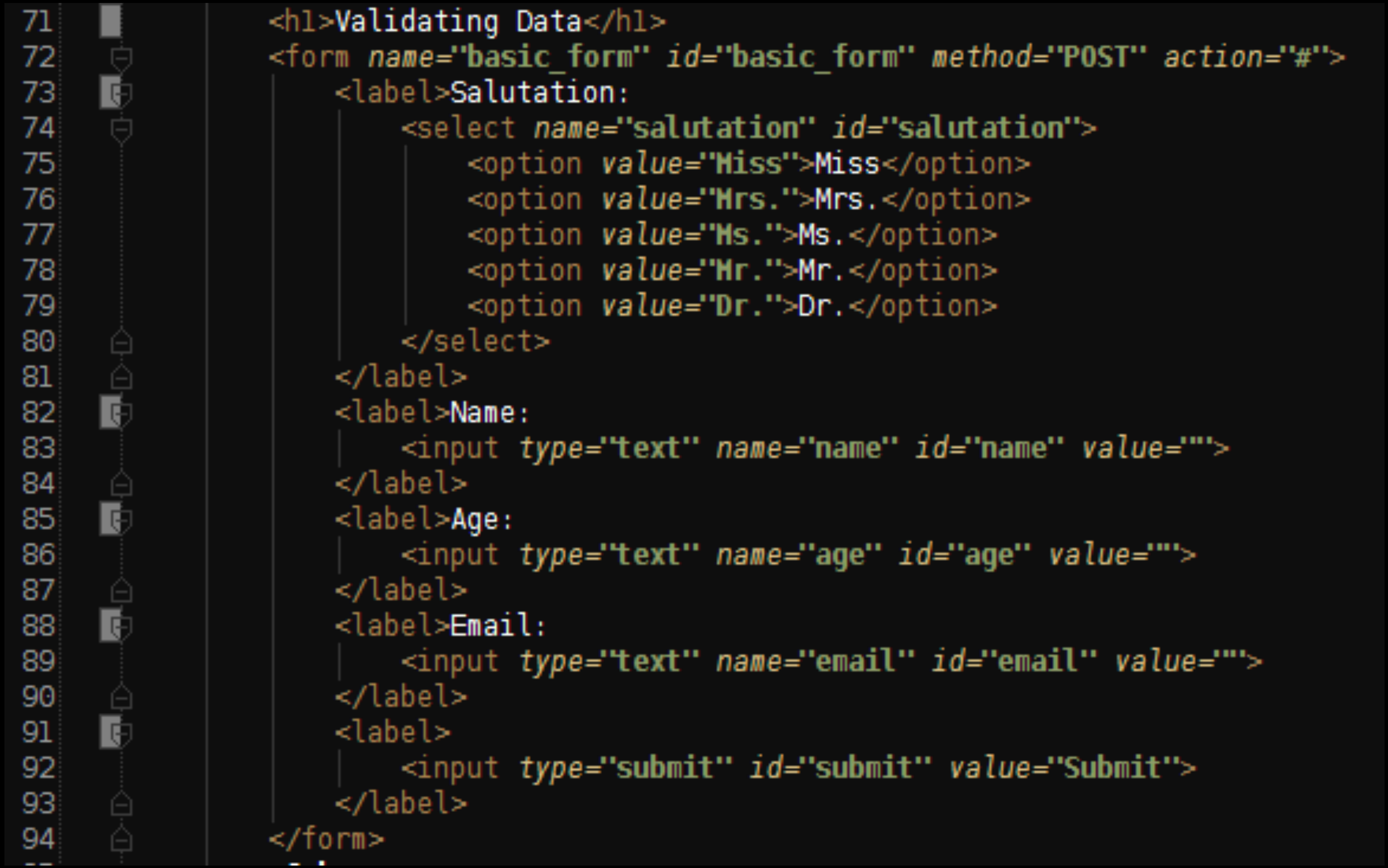
Name:

Age:

Email:

Validating User Input

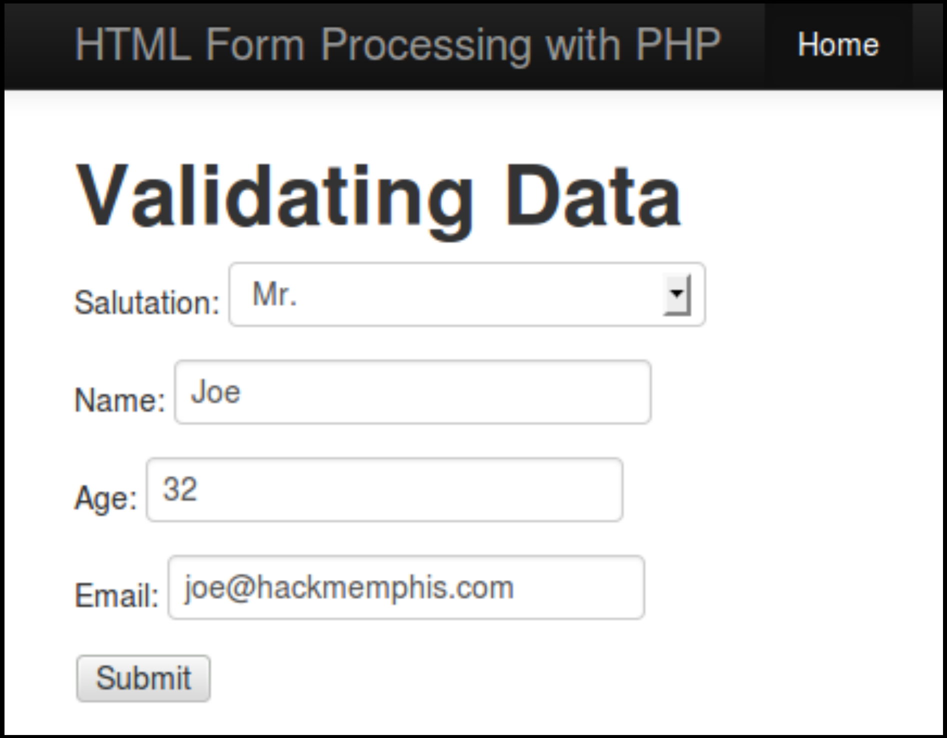
Form Code View (Example: 03 - Validating Data)



```
<h1>Validating Data</h1>
<form name="basic_form" id="basic_form" method="POST" action="#">
    <label>Salutation:
        <select name="salutation" id="salutation">
            <option value="Miss">Miss</option>
            <option value="Mrs.">Mrs.</option>
            <option value="Ms.">Ms.</option>
            <option value="Mr.">Mr.</option>
            <option value="Dr.">Dr.</option>
        </select>
    </label>
    <label>Name:
        <input type="text" name="name" id="name" value="">
    </label>
    <label>Age:
        <input type="text" name="age" id="age" value="">
    </label>
    <label>Email:
        <input type="text" name="email" id="email" value="">
    </label>
    <label>
        <input type="submit" id="submit" value="Submit">
    </label>
</form>
```

Validating User Input - Data Entry

This is the type of input we want...



Validating Data

Salutation: Mr.

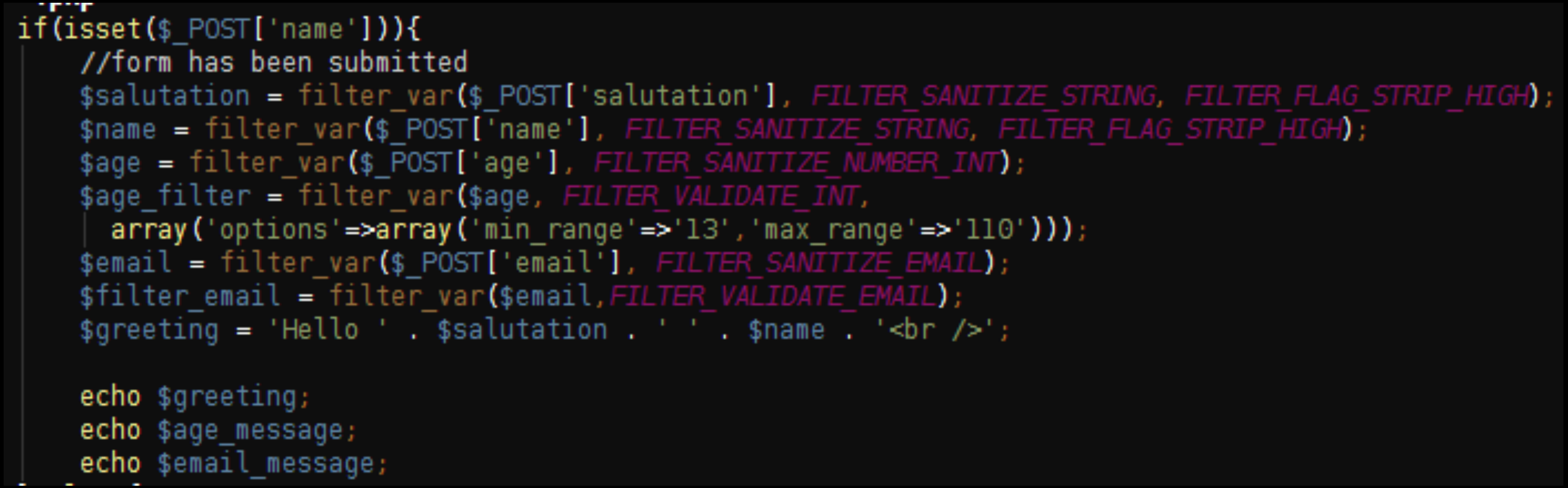
Name: Joe

Age: 32

Email: joe@hackmemphis.com

Validating Data - Data Entry

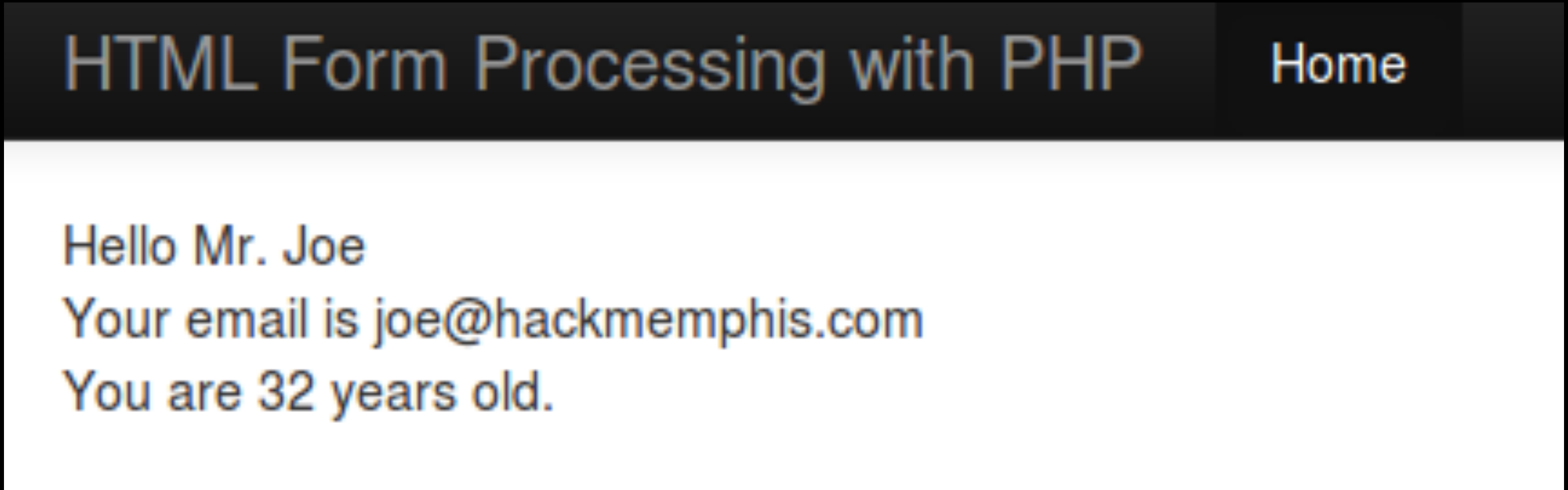
Make sure the data conforms to expectations



```
if(isset($_POST['name'])){
    //form has been submitted
    $salutation = filter_var($_POST['salutation'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $name = filter_var($_POST['name'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $age = filter_var($_POST['age'], FILTER_SANITIZE_NUMBER_INT);
    $age_filter = filter_var($age, FILTER_VALIDATE_INT,
      array('options'=>array('min_range'=>'13','max_range'=>'110')));
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    $filter_email = filter_var($email,FILTER_VALIDATE_EMAIL);
    $greeting = 'Hello ' . $salutation . ' ' . $name . '<br />';

    echo $greeting;
    echo $age_message;
    echo $email_message;
```

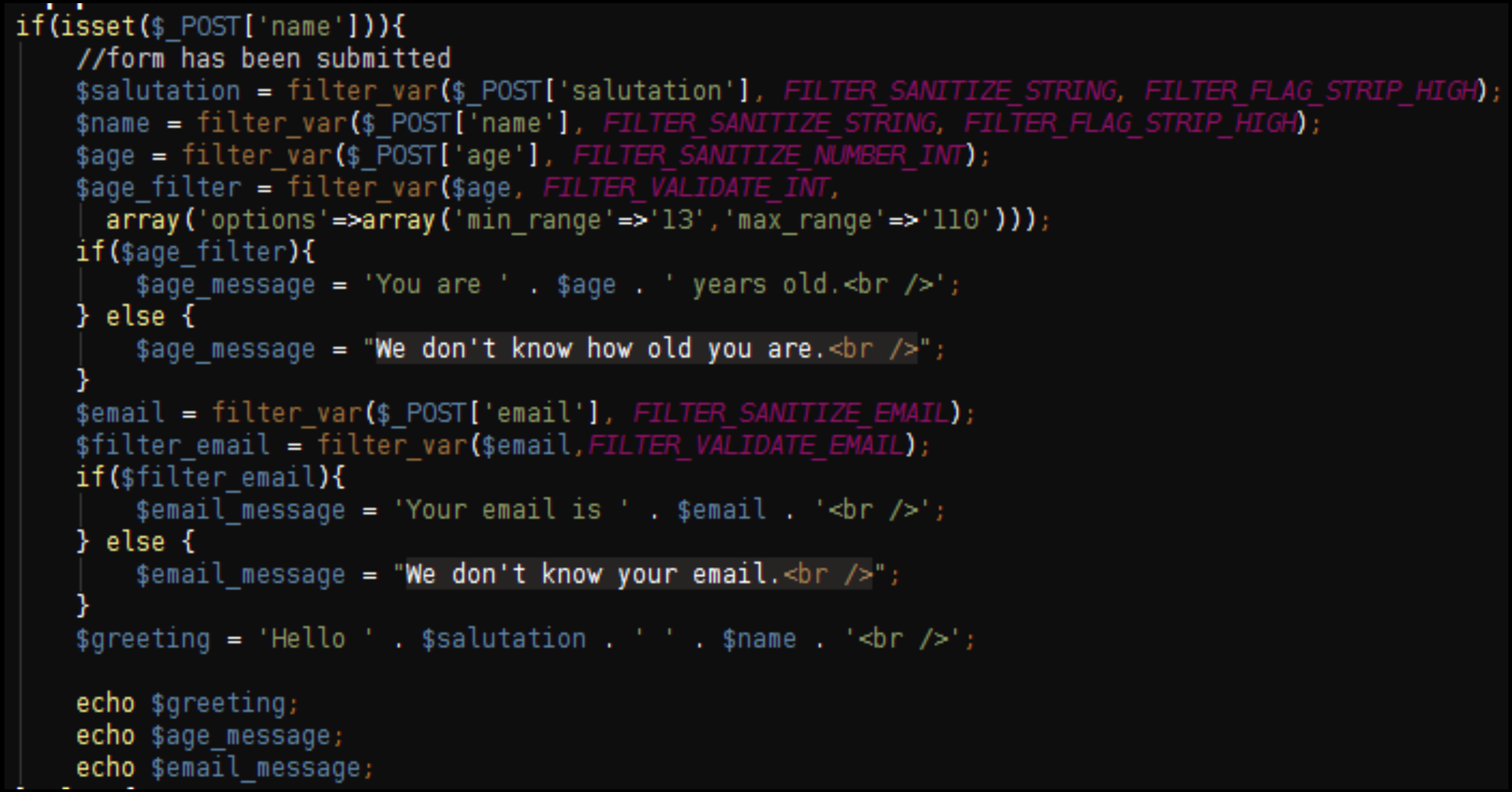
Validating Data - Data Entry

Output



```
Hello Mr. Joe
Your email is joe@hackmemphis.com
You are 32 years old.
```

Testing Our Validation

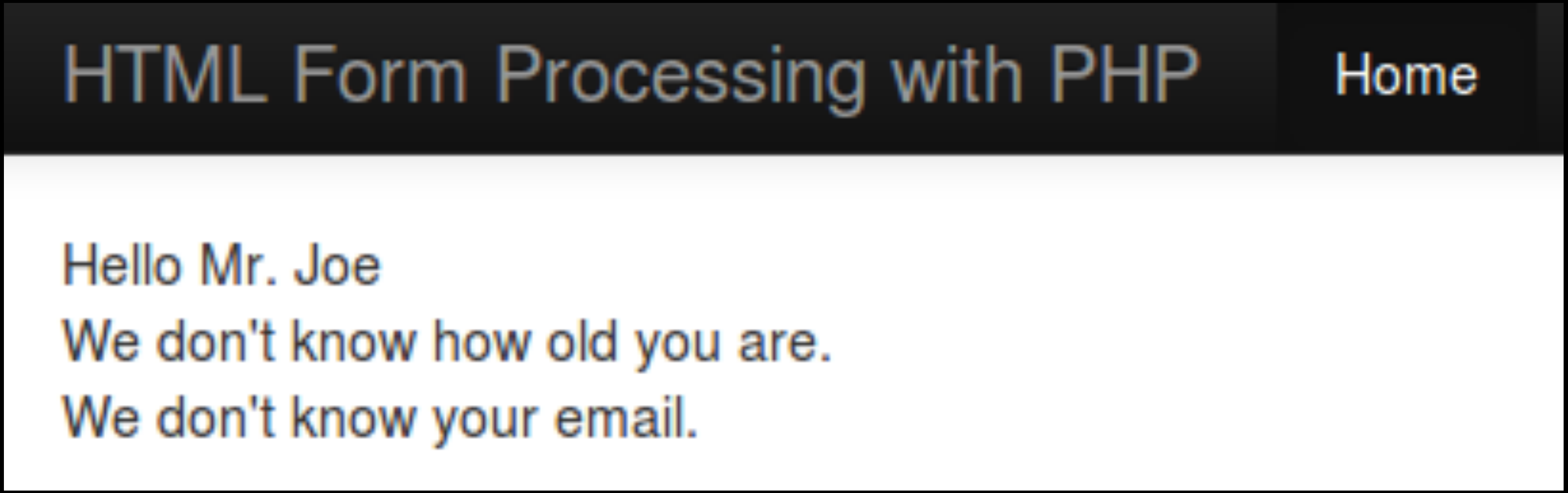


```
if(isset($_POST['name'])){
    //form has been submitted
    $salutation = filter_var($_POST['salutation'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $name = filter_var($_POST['name'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $age = filter_var($_POST['age'], FILTER_SANITIZE_NUMBER_INT);
    $age_filter = filter_var($age, FILTER_VALIDATE_INT,
     array('options'=>array('min_range'=>'13','max_range'=>'110')));
    if($age_filter){
        $age_message = 'You are ' . $age . ' years old.<br />';
    } else {
        $age_message = "We don't know how old you are.<br />";
    }
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    $filter_email = filter_var($email,FILTER_VALIDATE_EMAIL);
    if($filter_email){
        $email_message = 'Your email is ' . $email . '<br />';
    } else {
        $email_message = "We don't know your email.<br />";
    }
    $greeting = 'Hello ' . $salutation . ' ' . $name . '<br />';

    echo $greeting;
    echo $age_message;
    echo $email_message;
```

Testing Our Validation

If we don’t enter an age…

if we don’t enter our email...



Hello Mr. Joe

We don't know how old you are.

We don't know your email.

What do we do with the data?

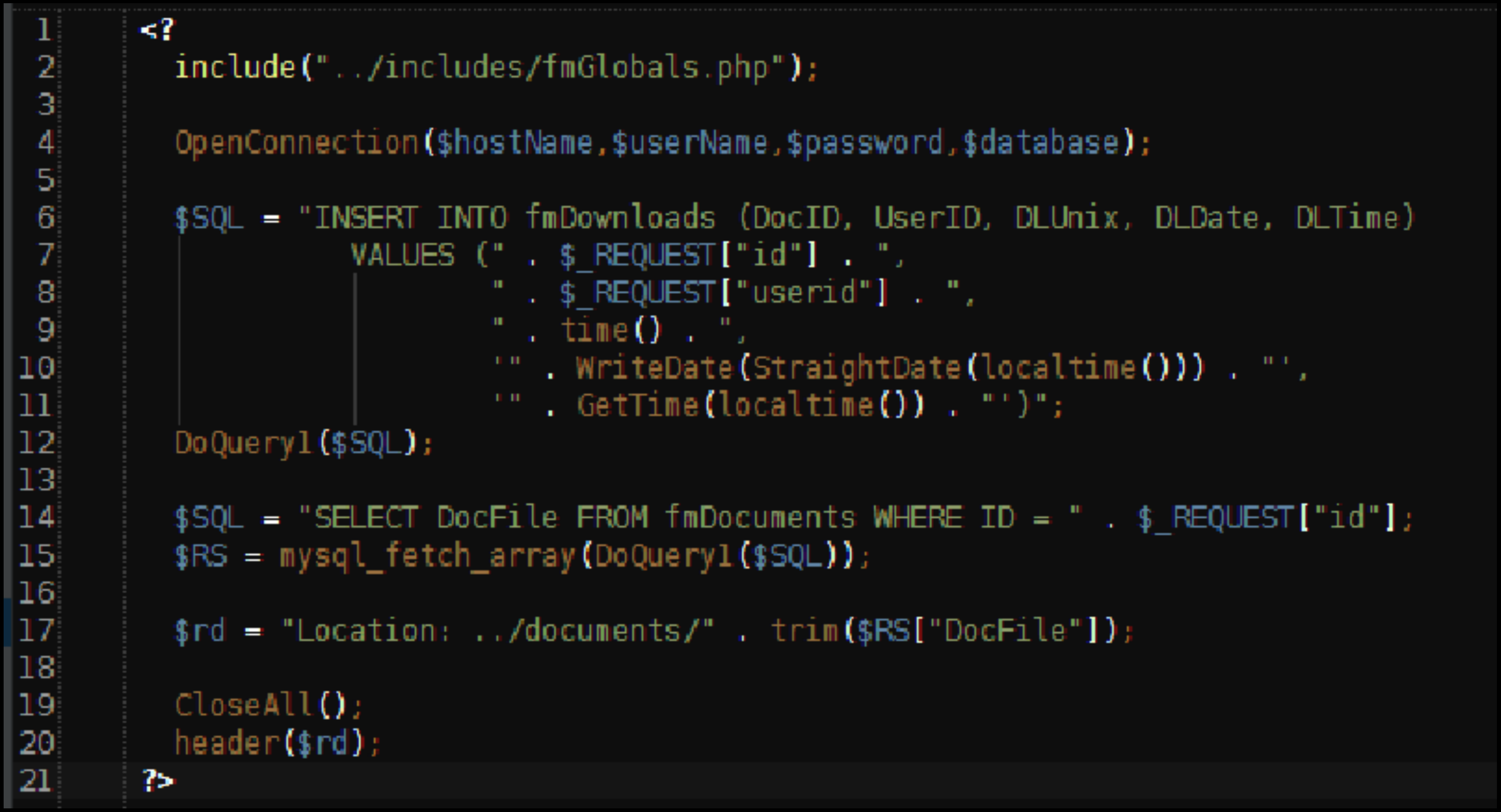
No matter what you do from here…

- the hard part is over.

Your data is now:

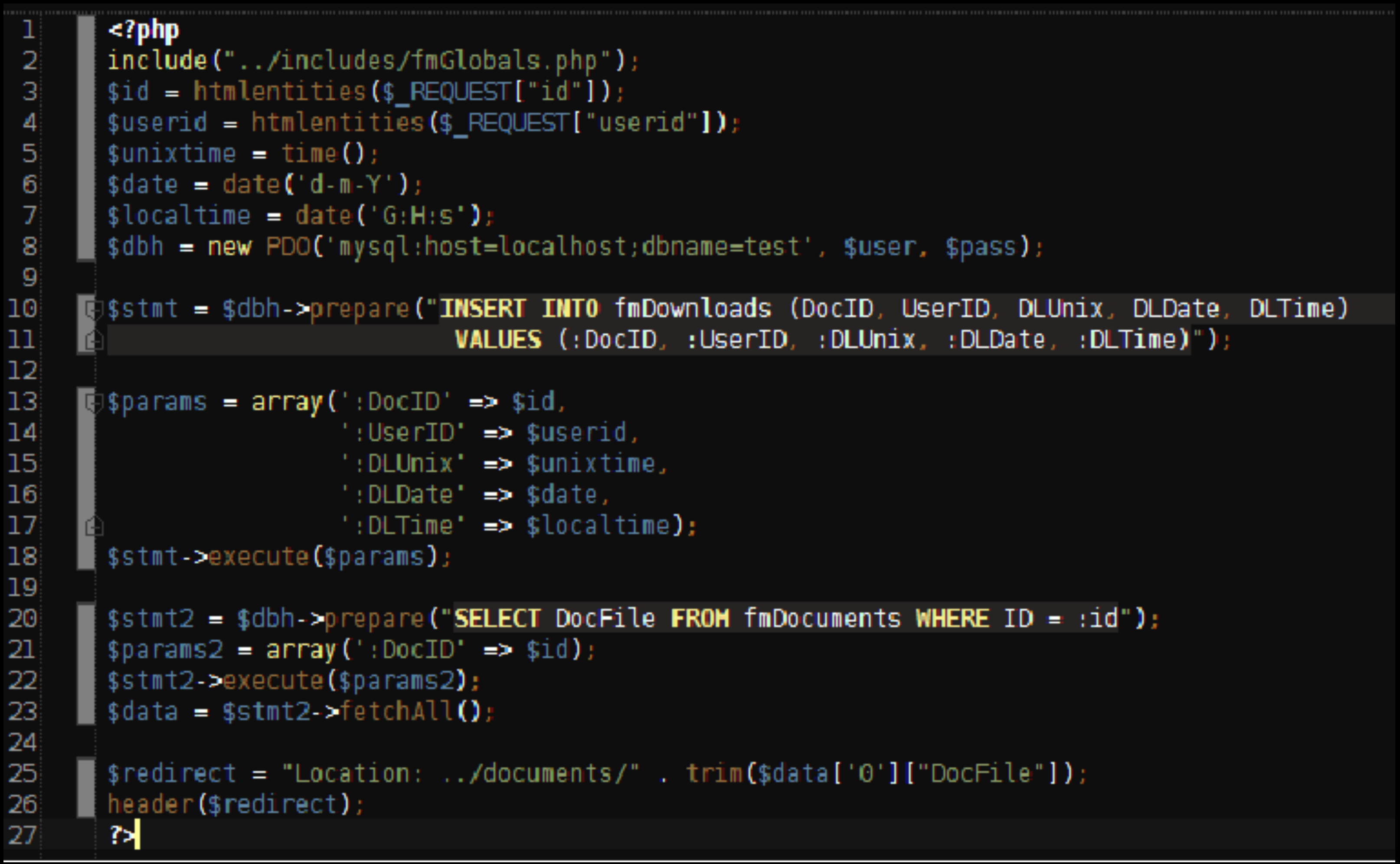
- **Validated!**
- **Sanitized!**

Remember that Bad Idea?



```
<?
  include("../includes/fmGlobals.php");

  OpenConnection($hostName,$userName,$password,$database);

  $SQL = "INSERT INTO fmDownloads (DocID, UserID, DLUnix, DLDate, DLTime)
          VALUES (" . $_REQUEST["id"] . ",
                  " . $_REQUEST["userid"] . ",
                  " . time() . ",
                  '" . WriteDate(StraightDate(localtime())) . "',
                  '" . GetTime(localtime()) . "')";
  DoQuery1($SQL);

  $SQL = "SELECT DocFile FROM fmDocuments WHERE ID = " . $_REQUEST["id"];
  $RS = mysql_fetch_array(DoQuery1($SQL));

  $rd = "Location: ../documents/" . trim($RS["DocFile"]);

  CloseAll();
  header($rd);
?>
```

Better idea...



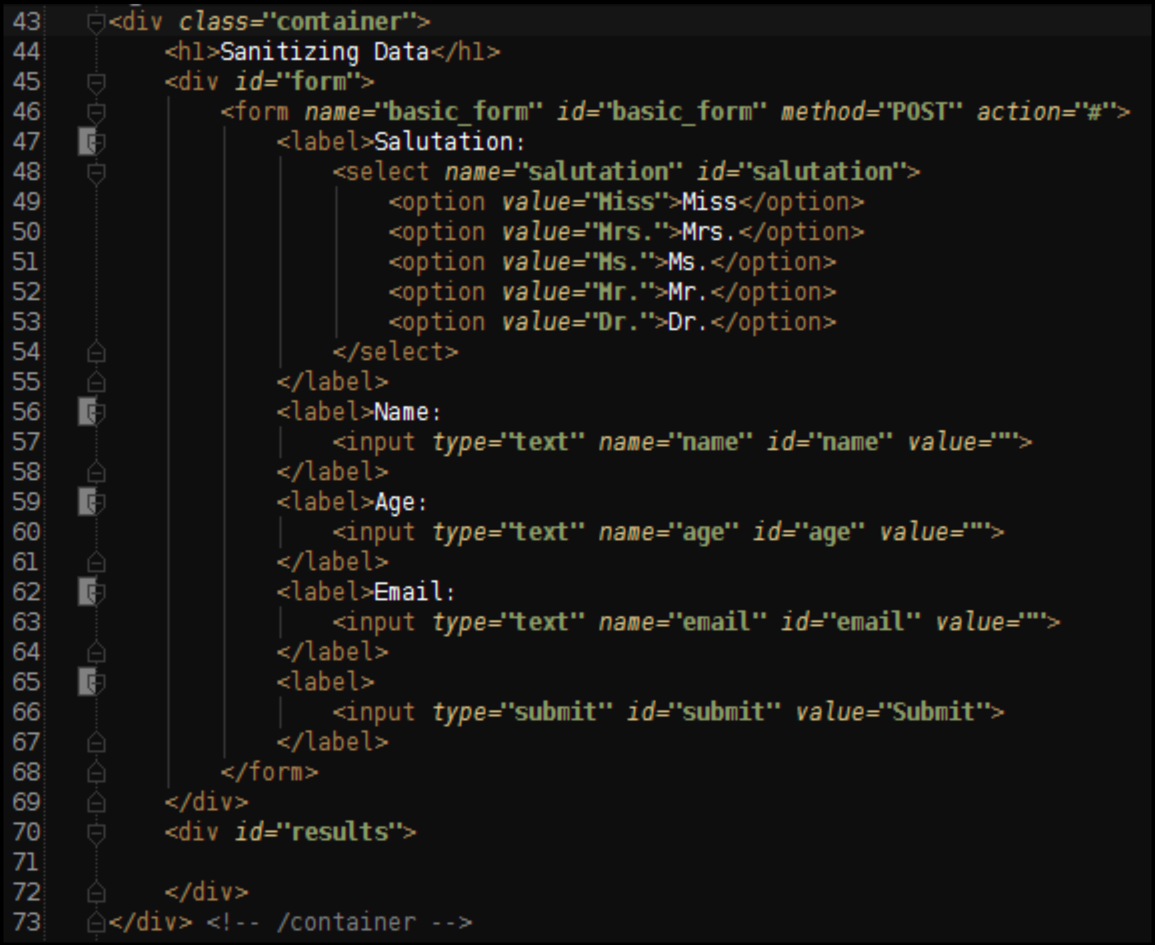
```php
<?php
include("../includes/fmGlobals.php");
$id = htmlentities($_REQUEST["id"]);
$userid = htmlentities($_REQUEST["userid"]);
$unixtime = time();
$date = date('d-m-Y');
$localtime = date('G:H:s');
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);

$stmt = $dbh->prepare("INSERT INTO fmDownloads (DocID, UserID, DLUnix, DLDate, DLTime)
                        VALUES (:DocID, :UserID, :DLUnix, :DLDate, :DLTime)");

$params = array(':DocID' => $id,
                ':UserID' => $userid,
                ':DLUnix' => $unixtime,
                ':DLDate' => $date,
                ':DLTime' => $localtime);
$stmt->execute($params);

$stmt2 = $dbh->prepare("SELECT DocFile FROM fmDocuments WHERE ID = :id");
$params2 = array(':DocID' => $id);
$stmt2->execute($params2);
$data = $stmt2->fetchAll();

$redirect = "Location: ../documents/" . trim($data['0']["DocFile"]);
header($redirect);
?>
```

Tips

- ALWAYS Sanitize your data.
  - Even if you don’t validate it.
  - Some data is harder to validate than others
- Use JavaScript to enforce requirements
  - JavaScript is great for checking data before it gets submitted. This makes PHP’s job easier
  - Don’t rely on JS to sanitize. Let PHP handle it.
- Offload your PHP processing via Ajax
  - Keeps your code cleaner by separating heavy lifting of data validation out of your form code.
- Show your users where they failed.
  - This also makes your own debugging easier

Tricks - Ajax Form Handling

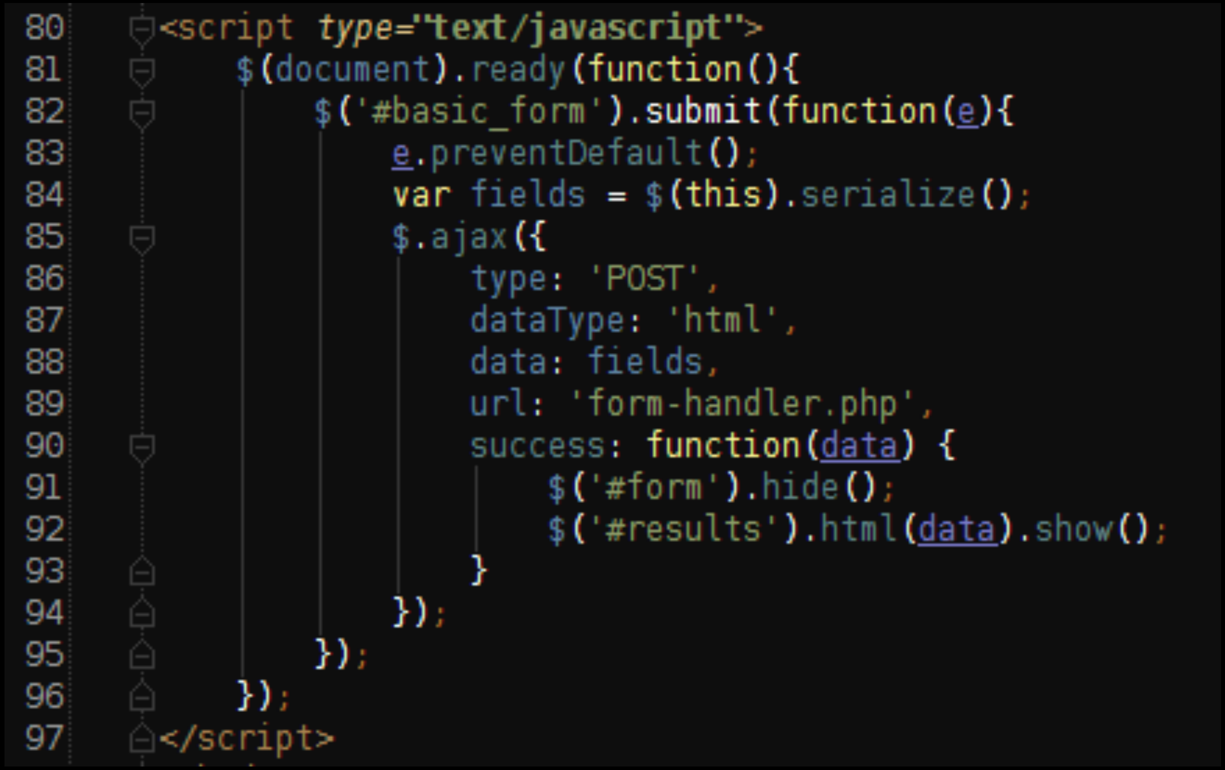
Form Code View (Example: 04 Ajax Form Handling)



```
<div class="container">
    <h1>Sanitizing Data</h1>
    <div id="form">
        <form name="basic_form" id="basic_form" method="POST" action="#">
            <label>Salutation:
                <select name="salutation" id="salutation">
                    <option value="Miss">Miss</option>
                    <option value="Mrs.">Mrs.</option>
                    <option value="Ms.">Ms.</option>
                    <option value="Mr.">Mr.</option>
                    <option value="Dr.">Dr.</option>
                </select>
            </label>
            <label>Name:
                <input type="text" name="name" id="name" value="">
            </label>
            <label>Age:
                <input type="text" name="age" id="age" value="">
            </label>
            <label>Email:
                <input type="text" name="email" id="email" value="">
            </label>
            <label>
                <input type="submit" id="submit" value="Submit">
            </label>
        </form>
    </div>
    <div id="results">

    </div>
</div> <!-- /container -->
```

Tricks Ajax Form Handling

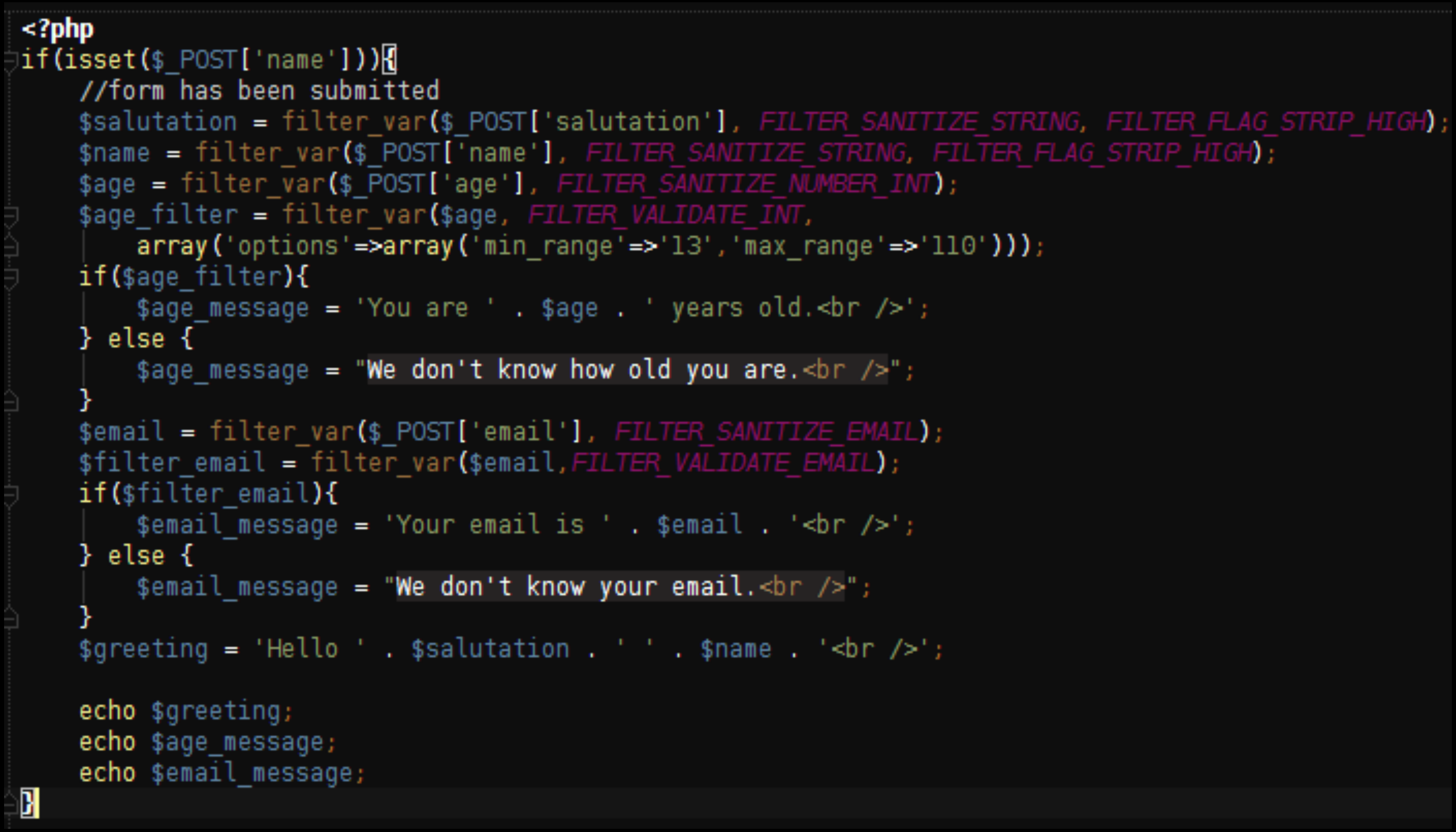
Ajax JavaScript to post the form



```
<script type="text/javascript">
    $(document).ready(function(){
        $('#basic_form').submit(function(e){
            e.preventDefault();
            var fields = $(this).serialize();
            $.ajax({
                type: 'POST',
                dataType: 'html',
                data: fields,
                url: 'form-handler.php',
                success: function(data) {
                    $('#form').hide();
                    $('#results').html(data).show();
                }
            });
        });
    });
</script>
```

Tricks Ajax Form Handling

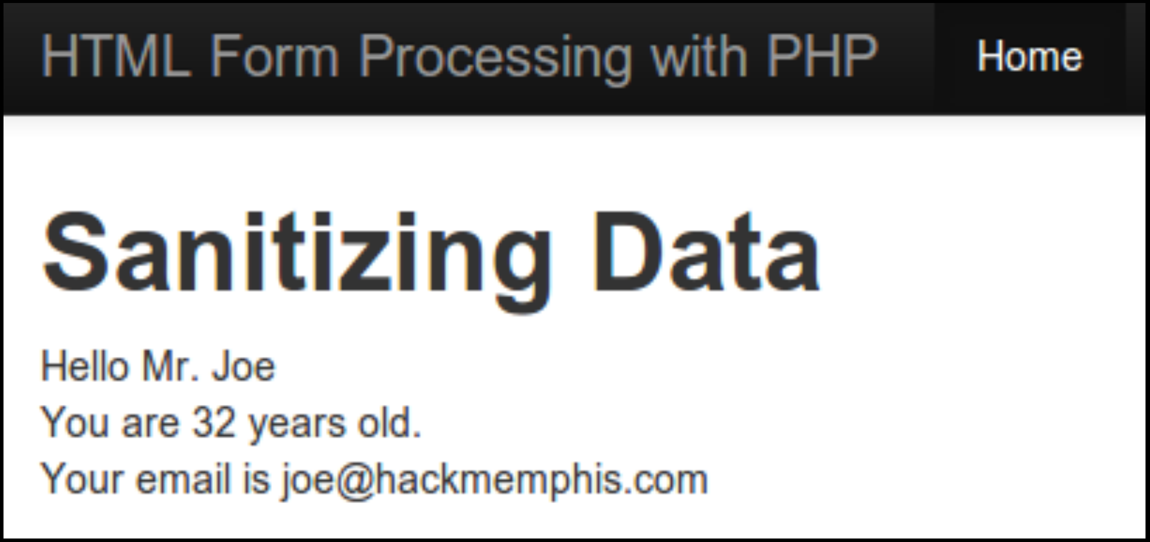
This is the file we’re posting the form data to



```
<?php
if(isset($_POST['name'])){
    //form has been submitted
    $salutation = filter_var($_POST['salutation'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $name = filter_var($_POST['name'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
    $age = filter_var($_POST['age'], FILTER_SANITIZE_NUMBER_INT);
    $age_filter = filter_var($age, FILTER_VALIDATE_INT,
        array('options'=>array('min_range'=>'13','max_range'=>'110')));
    if($age_filter){
        $age_message = 'You are ' . $age . ' years old.<br />';
    } else {
        $age_message = "We don't know how old you are.<br />";
    }
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    $filter_email = filter_var($email,FILTER_VALIDATE_EMAIL);
    if($filter_email){
        $email_message = 'Your email is ' . $email . '<br />';
    } else {
        $email_message = "We don't know your email.<br />";
    }
    $greeting = 'Hello ' . $salutation . ' ' . $name . '<br />';

    echo $greeting;
    echo $age_message;
    echo $email_message;
}
```

Tricks Ajax Form Handling Output

This allows you to separate logic away from the file that contains your form.



Sanitizing Data

Hello Mr. Joe
You are 32 years old.
Your email is joe@hackmemphis.com

Tricks - Some tricks won’t last...

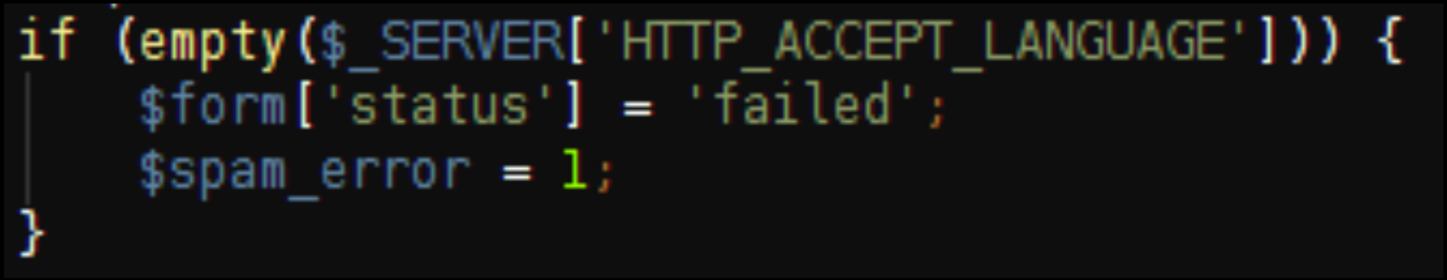
Issue: Application Form getting spammed.

Steps taken:

- We added reCAPTCHA

Issue slowed, after some time they continued

Further steps:

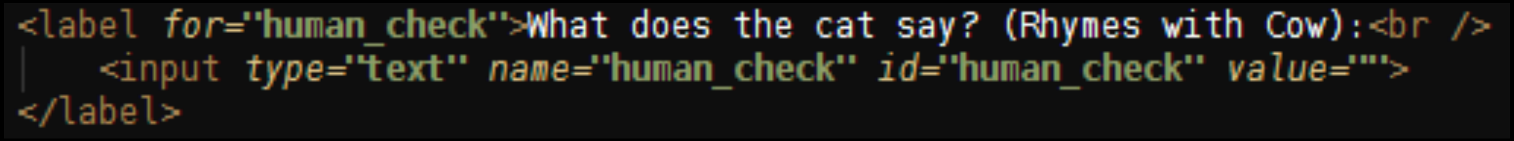


```
if (empty($_SERVER['HTTP_ACCEPT_LANGUAGE'])) {
    $form['status'] = 'failed';
    $spam_error = 1;
}
```

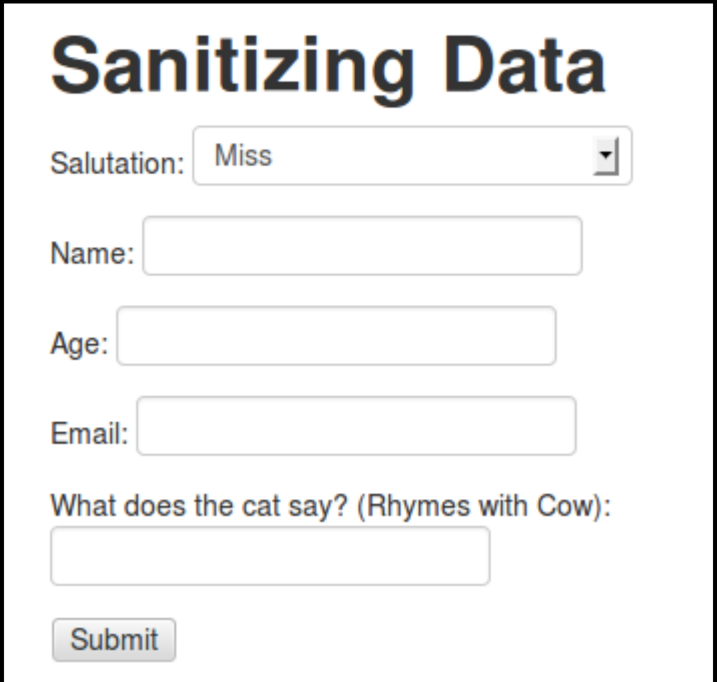
For some reason the spam stopped...

Tricks - Getting Creative

Add a question to your form:



```
<label for="human_check">What does the cat say? (Rhymes with Cow):<br />
    <input type="text" name="human_check" id="human_check" value="">
</label>
```



Sanitizing Data

Salutation: Miss

Name:

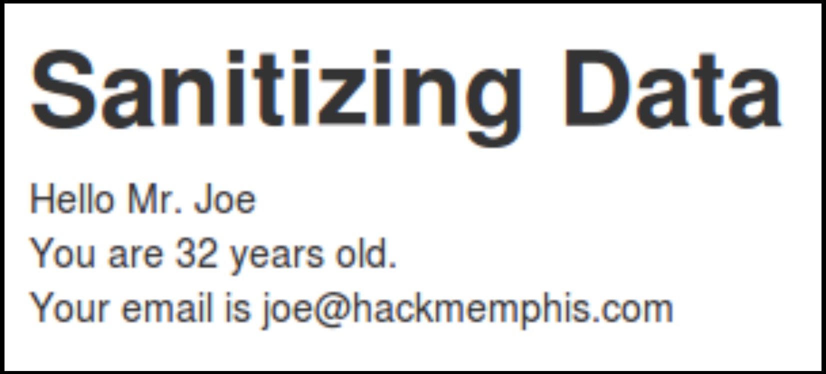
Age:

Email:

What does the cat say? (Rhymes with Cow):

Tricks - Getting Creative

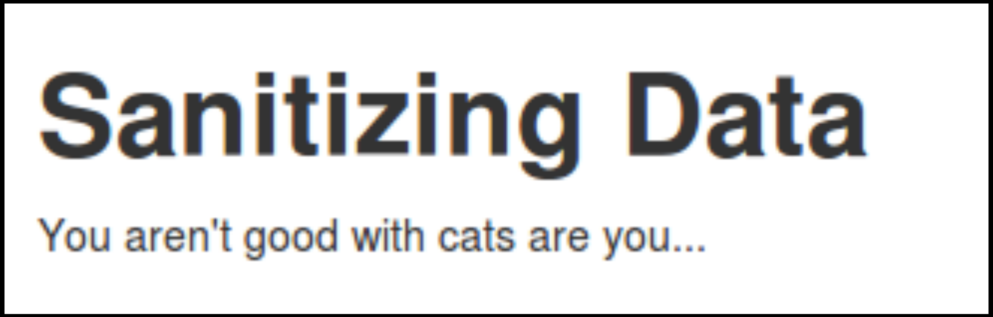
If the user answers correctly:



Sanitizing Data

Hello Mr. Joe
You are 32 years old.
Your email is joe@hackmemphis.com

If the user answers incorrectly:

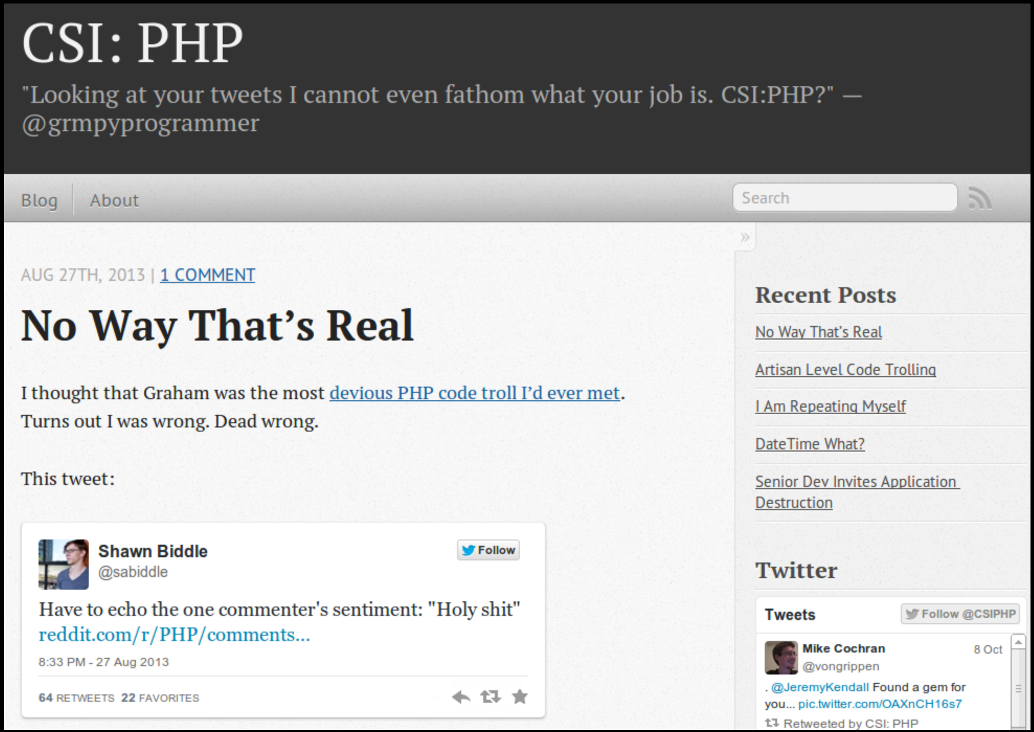


Sanitizing Data

You aren't good with cats are you...

Better Ideas: CSI:php - csiphp.com

CSI:PHP is a great site to see not only bad code, but WHY it’s bad and how to make it better.



CSI: PHP

"Looking at your tweets I cannot even fathom what your job is. CSI:PHP?" — @grmpyprogrammer

AUG 27TH, 2013 | 1 COMMENT

No Way That’s Real

I thought that Graham was the most devious PHP code troll I’d ever met. Turns out I was wrong. Dead wrong.

This tweet:

Shawn Biddle @sabiddle

Have to echo the one commenter's sentiment: "Holy shit" reddit.com/r/PHP/comments...

8:33 PM - 27 Aug 2013

64 RETWEETS 22 FAVORITES

Recent Posts

- No Way That’s Real
- Artisan Level Code Trolling
- I Am Repeating Myself
- DateTime What?
- Senior Dev Invites Application Destruction

Twitter

Tweets

Mike Cochran @vongrippen 8 Oct

. @JeremyKendall Found a gem for you... pic.twitter.com/OAXnCH16s7

Retweeted by CSI: PHP

Links - Q & A

- MemphisPHP.org
- phptherightway.com
- CSIPHP.com
- Slides & Code Samples:
  - https://github.com/svpernova09
- htmlentities() or htmlspecialchars()?
  - http://stackoverflow.com/questions/46483/htmlentities-vs-htmlspecialchars
- xkcd: http://xkcd.com/