

Working With Forms

Christian Wenz
@chwenz



pluralsight 
hardcore dev and IT training

Agenda

- Form Basics
- Processing Form Data
- Validating and Prefilling the Form

HTML Forms

```
<form method="post" action="">
```

Method **GET**

- Data appended to the URL
- Size limit (~500 to 2000 characters)
- PHP array \$_GET

Method **POST**

- Data appended to the HTTP request
- No size limits, file uploads possible
- PHP array \$_POST

A Look at HTTP

```
GET http://localhost/html5/communication/form.php?txt=Text+field&pwd=Password&area=Multi-
Accept: text/html, application/xhtml+xml, */*
Referer: http://localhost/html5/communication/form.php
Accept-Language: de-DE,de;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
DNT: 1
Host: localhost
```

```
POST http://localhost/html5/communication/form.php HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: http://localhost/html5/communication/form.php
Accept-Language: de-DE,de;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Content-Length: 140
DNT: 1
Host: localhost
Pragma: no-cache

txt=Text+field&pwd=Password&area=Multi-line%0D%0Atext+field&chb=on&group=option2&ddl=item2,
```

Form Elements

```
<input type="text" name="txt">
```

```
<input type="password" name="pwd">
```

```
<textarea name="area"></textarea>
```

Text field

••••••••

Multi-line
text field

Text fields

```
<input type="checkbox" name="chb" value="on">
```

```
<input type="radio" name="group" value="option1">1
```

```
<input type="radio" name="group" value="option2">2
```



☐ 1 ☒ 2

Radio buttons /
checkboxes

```
<select name="ddl">
```

```
<option value="item1">one</option>
```

```
<option value="item2">two</option>
```

```
<option value="item3">three</option>
```

```
</select>
```

one ▼

one
two
three

Selection lists

Processing Form Fields

Data is put in `$_GET/$_POST` arrays

Element's name is array key

Array value is form input

Valid for text fields and buttons

```
<input type="text" name="txt">
```

```
<input type="password" name="pwd">
```

```
<textarea name="area"></textarea>
```

Multi-line
text field

⬆
⬇

Processing Radio Buttons and Checkboxes

When a checkbox is activated,
\$_GET/\$_POST contains its
value

```
<input type="checkbox" name="chb" value="on">
```

```
<input type="radio" name="group" value="option1">1
```

```
<input type="radio" name="group" value="option2">2
```

When a radio button is
activated, \$_GET/\$_POST
contains its value



☐ 1 ☒ 2

For radio button, the group
name is the array key

Processing Lists

For regular lists,
\$_GET/\$_POST contains the
selected element's value (or
caption, if no value set)

For multi-select lists,
\$_GET/\$_POST contains an
array of values

```
<select multiple name="list[]" size="3">  
  <option value="item1">one</option>  
  <option value="item2">two</option>  
  <option value="item3">three</option>  
</select>
```

Multi-select list name must
end with []

Escaping Form Output

User input is not trustworthy

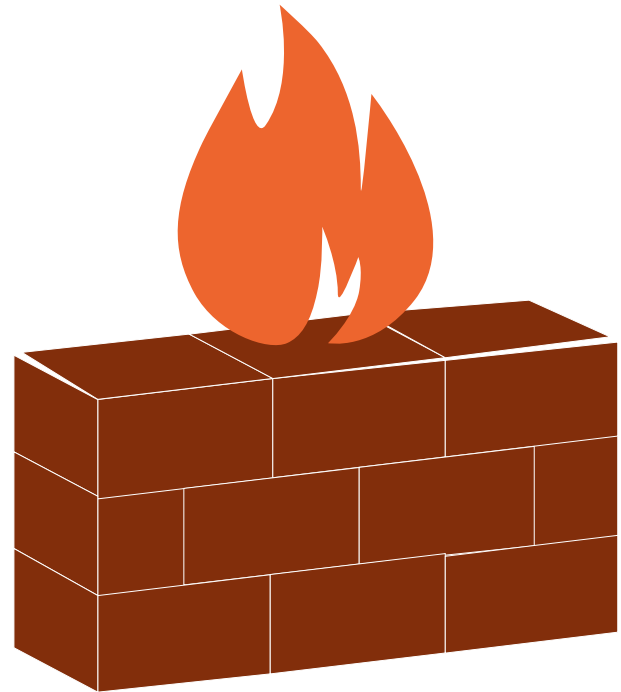
May contain HTML

Use `htmlspecialchars()` to escape special characters

This will take care of:

< > " &

Option `ENT_QUOTES` escapes
' character



Validating Form Data



Check for non-empty value for text fields, radio buttons, and checkboxes

Special treatment for lists

Consider using JavaScript as an additional feature (not replacement!)

Form Validation With Regular Expressions

```
$regex = '/^\w+@(\w+\.)+\w+$/'; //regular expr.  
  
if (!preg_match($regex, $_POST['key'])) {  
    // input does not match regular expression  
    echo 'Format error!';  
}
```

Prefilling Form Fields

Text field

••••••••

Multi-line
text field



☐ 1 ☒ 2

one ▼

one
two
three

Set value or content
appropriately

Pre-activate item if
applicable

Pre-select list item(s)

Summary

- PHP puts form input in `$_GET` and `$_POST` arrays, depending on the HTTP method used
- Validating forms is rather easy, prefilling them upon error requires a bit more effort
- Take care to avoid Cross-Site Scripting (XSS) when processing form data