

---

# PHP Forms

---

**Aplicações para a Internet**  
Engenharia Informática – 2014/2015



# Copyright

---

## **Based on contents originally created by:**

- **Vítor Carreira** (vitor.carreira@ipleiria.pt)
- **Marco Monteiro** (marco.monteiro@ipleiria.pt)

## **Contributors:**

- **Norberto Henriques** (norberto.henriques@ipleiria.pt)
- **Carlos Urbano** (carlos.urbano@ipleiria.pt)
- **Fernando Silva** (fernando.silva@ipleiria.pt)
- **Alexandrino Gonçalves** (alex@ipleiria.pt)

## **Revised on: March, 2015**

- **Fernando Silva** (fernando.silva@ipleiria.pt)

# Forms

---

- ▶ A form is basically a Web page with input fields that allows you to enter information. When the form is submitted, that information is packaged up and sent off to a Web server to be processed by a Web application

```
<form action="http://www.headfirstlabs.com/contest.php" method="post">
```

```
  <p>Just type in your name (and click Submit) to enter the contest: <br />
```

```
  First name: <input type="text" name="firstname" value="" /> <br />
```

```
  Last name: <input type="text" name="lastname" value="" /> <br />
```

```
  <input type="submit" />
```

```
</p>
```

```
</form>
```

- ▶ `action` – URL that defines where to send the data when the submit button is pushed
- ▶ `method` – the HTTP method for sending data to the action URL. Default is `get`. More on this later...

# Form controls

---

## ▶ Text Input

```
<input type="text" name="fullname" />
```

```
<input type="text" name="fullname" value="John Doe" />
```

## ▶ Submit input

```
<input type="submit" />
```

## ▶ Checkbox input

```
<input type="checkbox" name="spice[]" value="salt" />
```

```
<input type="checkbox" name="spice[]" value="pepper"  
checked="checked" />
```

## ▶ Radio input

```
<input type="radio" name="hotornot" value="hot" />
```

```
<input type="radio" name="hotornot" value="not hot"  
checked="checked" />
```

# Form controls

---

## ▶ Select

```
<select name="characters">  
  <option value="Buckaroo">Buckaroo Banzai</option>  
  <option value="Tommy">Perfect Tommy</option>  
  <option value="Penny">Penny Priddy</option>  
  <option value="Jersey">New Jersey</option>  
  <option value="John">John Parker</option>  
</select>
```

## ▶ // Multiple selection

```
<select name="characters" multiple="multiple">
```

## ▶ Textarea

```
<textarea name="comments" rows="10" cols="48">  
  </textarea>
```

# Form controls

---

## ► Fieldsets - groups related data in a form

```
<fieldset>
```

```
  <legend>Condiments</legend>
```

```
  <input type="checkbox" name="spice" value="salt" />
```

```
  Salt <br />
```

```
  <input type="checkbox" name="spice" value="pepper" />
```

```
  Pepper <br />
```

```
  <input type="checkbox" name="spice" value="garlic" />
```

```
  Garlic
```

```
</fieldset>
```

## ► Labels

```
<input type="radio" name="hotornot" value="hot" id="hot" />
```

```
<label for="hot">hot</label>
```

# Form controls

---

## ▶ Passwords

```
<input type="password" name="secret" />
```

## ▶ File input

```
<input type="file" name="doc" />
```

## ▶ Other types of inputs

```
<input type="button" name="action1" value="ButtonText"/><br/>
```

```
<input type="reset" name="reset" value="ResetForm"/><br/>
```

```
<input type="hidden" name="field1" value="fieldvalue"/><br/>
```

```
<input type="image" name="image" src="send.gif"/><br/>
```

# Form controls – HTML Input Attributes

---

- ▶ Some attributes common to several input types
  - ▶ **name:** name of the input field
  - ▶ **value:** specifies the initial value for an input field
  - ▶ **readonly:** specifies that is read only (cannot be changed)

```
<input type="text" name="firstName" value="Joana" readonly>
```

- ▶ **disabled:** specifies that the input field is disabled

```
<input type="text" name="firstName" value="Joana" disabled>
```

- ▶ **size:** specifies the size (in characters) for the input field

```
<input type="text" name="firstName" value="Joana" size="40">
```

- ▶ **maxlength:** maximum allowed length

```
<input type="text" name="firstName" value="Joana" maxlength="20">
```

More info on Input Attributes:

[http://www.w3schools.com/html/html\\_form\\_attributes.asp](http://www.w3schools.com/html/html_form_attributes.asp)



# Form controls – HTML 5 Compatibility

---

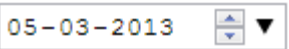




- ▶ HTML5 is not fully supported in all of the major browsers
  - ▶ Check compatibility on:
    - ▶ <https://html5test.com/>
  - ▶ You can also check for different controls, attributes, elements, etc..., on:
    - ▶ <http://caniuse.com/>

# Form controls - HTML5 Input Types

---

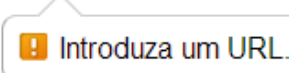
- ▶ New <input> control types on HTML5
  - ▶ Remember: **No universal support**

## Type

date	Used for input fields that should contain a date	
search	Used for search fields	
email	E-mail address (checks format)	
color	Used for input fields that should contain a color	
number	Input fields that should contain a numeric value	
range	For input fields that should contain a value within a range	

# Form controls - HTML5 Input Types

## Type

url	Input fields that should contain a URL address	<input type="url" value="www.meusite.com"/> 
tel	Input fields that should contain a telephone number	
month	Select a month and year	<input type="month" value="Abril de 2008"/>
week	Select a week and year	<input type="week" value="Semana 11, de 2013"/>
time	Select a time (no time zone)	<input type="time" value="14:25"/>
datetime	Select a date and time (with time zone)	
datetime-local	Select a date and time (no time zone)	<input type="datetime-local" value="07-03-2013 14:25"/>

More info on Input Types:

[http://www.w3schools.com/html/html\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html_form_input_types.asp)

# Form controls – new on HTML 5

---

## ▶ <datalist>

- ▶ The **<datalist>** element specifies a list of pre-defined options for an `<input>` element
- ▶ Users will see a drop-down list of pre-defined options as they input data
- ▶ The **list** attribute of the `<input>` element, must refer to the **id** attribute of the `<datalist>` element

```
<form action="action_page.php">
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
</form>
```

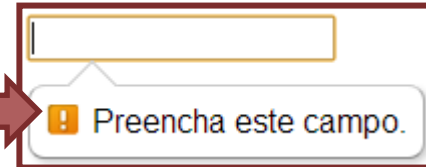
**Attention!!!**  
This control is not supported by all of the major browsers!  
e.g. IE11, Safari, Operamini,...

# Form controls – HTML5 Input Attributes

- ▶ **required:** mandatory field

- ▶ must be filled out before submitting the form

```
<input type="text" required>
```



- ▶ **placeholder:** hint that describes the expected value

- ▶ It is displayed in the input field before the user enters a value

```
<input type="email" placeholder="email.name@mail.com">
```

A diagram illustrating the 'placeholder' attribute. It shows an email input field with the placeholder text 'email.name@mail.com' displayed inside the field, providing a hint for the expected value.

- ▶ **autocomplete:** specifies whether a form or input field should have autocomplete on or off

```
<form action="action_page.php" autocomplete="on">  
  <input type="email" autocomplete="off">
```

More info on HTML5 Input Attributes:

[http://www.w3schools.com/html/html\\_form\\_attributes.asp](http://www.w3schools.com/html/html_form_attributes.asp)

# Forms: Method Attribute

---

## ▶ GET

- ▶ GET has a limit of the number of characters (that depends on the browser and the server configuration) ~ 4000
- ▶ Data is sent to server in the URL
- ▶ Requests can be bookmarked

## ▶ POST

- ▶ Limit depends only on server configuration
- ▶ Sent data is invisible (is it more secure?)

## ▶ Attribute enctype:

- ▶ Specifies the encoding used when sending form data. Possible values:
  - ▶ application/x-www-form-urlencoded - Default. All characters are encoded before sent
  - ▶ multipart/form-data - No characters are encoded. Form data is sent as a MIME document. **This parameter is required when using file upload**
  - ▶ text/plain - Spaces are converted to "+", but no special character encoding

# PHP and Forms

---


- ▶ PHP provides 5 built-in **superglobal** variables for Form processing:
  - ▶ **\$\_GET** - an associative array of variables passed to the current script via the URL parameters (HTTP GET request)
  - ▶ **\$\_POST** - an associative array of variables passed to the current script via the HTTP POST method
  - ▶ **\$\_COOKIE** - an associative array of variables passed to the current script via HTTP Cookies
  - ▶ **\$\_REQUEST** - an associative array that by default contains the contents of **\$\_GET**, **\$\_POST**, **\$\_COOKIE**
  - ▶ **\$\_FILES** - An associative array of files uploaded to the current script via the HTTP POST method
- ▶ The key used to fetch the value is the name of the control (attribute **name**)

# GET method 1/3

---

## ► File “form\_get.html”

```
<form action="process_form_get.php" method="get">
<div>
  <label for="firstName">First Name:</label>
  <input type="text" name="firstName" id="firstName" />
</div>
<div>
  <label>Age: <input type="text" name="age" /></label>
</div>
<div>
  <input type="submit" />
</div>
</form>
```



Alternative way of  
using <label>, without  
the *for* attribute



# GET method 2/3

---

## ► File “process\_form\_get.php”

```
<body>
<h1>Welcome</h1>
<?php
    echo "<p>First name: " . $_GET["firstName"] . "</p>\n";
    echo "<p>Age: " . $_GET["age"] . "</p>\n";
?>
</body>
```

# GET method 3/3

---

- ▶ When the user submits the form, every field will be part of the URL. e.g.:

- ▶ [http://10.10.1.101/Aula03/process\\_form\\_get.php?firstName=Ana&age=27](http://10.10.1.101/Aula03/process_form_get.php?firstName=Ana&age=27)

- ▶ This is not a good option for sending sensitive data (passwords, uids, etc).

- ▶ Although the specification of the HTTP protocol does not specify any maximum length, practical limits are imposed by web browser and server software.

- ▶ <http://www.boutell.com/newfaq/misc/urllength.html>

# POST method 1/3

---

## ► File “form\_post.html”

```
<form action="process_form_post.php" method="post">
<div>
  <label>First Name: <input type="text" name="firstName" />
</label>
</div>
<div>
  <label>Age: <input type="text" name="age" /></label>
</div>
<div>
  <input type="submit" />
</div>
</form>
```

# POST method 2/3

---

## ► File “process\_form\_post.php”

```
<body>
<h1>Welcome</h1>
<?php
    echo "<p>First name: " . $_POST["firstName"] . "</p>\n";
    echo "<p>Age: " . $_POST["age"] . "</p>\n";
?>
</body>
```

# POST method 3/3

---

- ▶ When the user submits the form, none of the fields will be part of the URL. E.g.:
  - ▶ [http://10.10.1.101/Aula03/process\\_form\\_post.php](http://10.10.1.101/Aula03/process_form_post.php)
- ▶ There is no limit (client side) for the size of the request
- ▶ The content of a request (POST) is normally limited by the server on a byte size basis in order to prevent a type of DoS attack

# Uploading files1/2

---

## ► File “form\_file.html”

```
<form action="upload.php" method="post"
enctype="multipart/form-data">
<div>
  <label for="description">Image description:</label>
  <input type="text" id="description" name="description" />
</div>
<div>
  <label for="image">Image:</label>
  <input type="file" name="image" id="image"/>
</div>
<div>
  <input type="submit" value="Send Image"/>
</div>
</form>
```

Specifies the encoding of the submitted data.  
**Required when using file upload!**

# Uploading files2/2

---

## ► File “upload.php”

```
<?php
echo '<h1>$_FILES</h1>';
echo '<pre>';
var_dump($_FILES);
echo '</pre>';
echo '<h1>$_POST</h1>';
echo '<pre>';
var_dump($_POST);
echo '</pre>';
?>
```



### **\$\_FILES**

```
array (size=1)
  'image' =>
    array (size=5)
      'name' => string 'Penguins.jpg' (length=12)
      'type' => string 'image/jpeg' (length=10)
      'tmp_name' => string '/tmp/phpV1WCEH' (length=14)
      'error' => int 0
      'size' => int 777835
```

### **\$\_POST**

```
array (size=1)
  'description' => string 'Imagem bonita' (length=13)
```

# PHP Forms – Some Security Issues

---

Use PHP to validate form data

- ▶ **htmlspecialchars():** converts special characters to HTML entities
  - ▶ Replaces HTML characters like < and > with &lt; and &gt;
  - ▶ Prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks - XSS) in forms

```
<form method="post" action="<?= htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```



**What is the `$_SERVER["PHP_SELF"]` variable?**

The `$_SERVER["PHP_SELF"]` is a super global variable that returns the filename of the currently executing script.

According with the context, it may also be useful to know the following functions:

- ✓ **trim():** Strip unnecessary characters (extra space, tab, newline)
- ✓ **stripslashes():** Remove backslashes (\)



# PHP – Server Validation

---

Although some HTML controls, as well as other technologies such as javascript, allow validating some of the user input on the client side, **it is always necessary to perform proper validation at the server side**

- ▶ In PHP there are some functions that helps on validating user input in the server side
  - ▶ **preg\_match()**: Perform a regular expression match
  - ▶ **filter\_var()**: Filters a variable with a specified filter

# PHP: preg\_match()

---

- Using preg\_match() to validate input format

`$firstName` can only have letters and white spaces

```
<?php
    if (empty($firstName)) {
        $firstNameErr = "First name is required";
    } elseif (!preg_match("/^[a-zA-Z ]*$/", $firstName)) {
        $firstNameErr = "Only letters and whitespaces allowed";
    }
?>
```

`$dateOfBirth` must be a date in the format yyyy-mm-dd

```
if (!preg_match('/^\d{4}[-]\d{1,2}[-]\d{1,2}+$/', $dateOfBirth)) {
    $dateOfBirthErr = 'Invalid date format (yyyy-mm-dd)';
}
```

# PHP: preg\_match()

---

- ▶ Using preg\_match() to obtain text with the matched pattern

```
<?php
    // get host name from URL
    preg_match('@^(?:http://)?([^\s/]+)@i',
        "http://www.php.net/index.html", $matches);
    $host = $matches[1];
    echo "host name is: {$matches[1]}\n";
?>
```

Output:

host name is: www.php.net

- **\$matches[0]** will contain the text that matched the full pattern;
- **\$matches[1]** will have the text that matched the first captured parenthesized sub pattern;
- and so on...

<http://php.net/manual/en/function.preg-match.php>

<http://php.net/manual/en/pcre.pattern.php>

# PHP: filter\_var()

---

- ▶ Using filter\_var() to validate e-mail format

```
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    $emailErr = "Invalid email format";  
}
```

- ▶ Using filter\_var() to validate regular expressions

```
if (!filter_var($firstName, FILTER_VALIDATE_REGEXP, ['options' =>  
    ['regexp' => '/^[a-zA-Z ]+$/']])) {  
    $firstNameErr = 'Only letters and whitespaces are allowed';  
}
```

<http://php.net/manual/en/function.filter-var.php>  
<http://php.net/manual/en/filter.filters.php>

# References

---

- ▶ HTML 5 Forms:
  - ▶ <http://www.w3schools.com/html/default.asp>
  - ▶ [http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)
- ▶ PHP and MySQL Web Development (4th Edition)
  - ▶ Luke Welling and Laura Thomson, Addison-Wesley 2009
- ▶ PHP Documentation
  - ▶ <http://www.php.net/manual/en/language.variables.superglobals.php>
  - ▶ <http://php.net/manual/en/function.preg-match.php>
  - ▶ <http://php.net/manual/en/function.filter-var.php>
  - ▶ <http://php.net/manual/en/filter.filters.php>
  - ▶ [http://www.w3schools.com/php/php\\_forms.asp](http://www.w3schools.com/php/php_forms.asp)
- ▶ Regex
  - ▶ <http://php.net/manual/en/pcre.pattern.php>
  - ▶ <http://www.phpliveregex.com>