



IPL

escola superior
de tecnologia e gestão
instituto politécnico
de leiria

PHP

Forms

Marco Monteiro



Contributors

2

► Author(s):

- Marco Monteiro (marco.monteiro@ipleiria.pt)

► Contributor(s):

- Vitor Carreira (vitor.carreira@ipleiria.pt)
- Fernando Silva(fernando.silva@ipleiria.pt)



Summary

3

1. User interaction
2. Forms
3. Form Controls
4. References

1 – USER INTERACTION



Input / Output

5

- ▶ With server-side programming
(without considering client programming with Javascript)
- ▶ **Output** – the content of the HTTP response – usually HTML – produced by the server and “viewed” by the user in the browser.
- ▶ **Input** – It’s not possible to read directly, data introduced by the user, because the server has no direct access to the client – it just receives HTTP requests.

All users' input will eventually correspond to an HTTP Request.



User interaction

6

- ▶ User is directly responsible for sending HTTP request only on these 3 situations:
 1. When it requests a page explicitly through the browser, using URL address bar, favorites, history, etc.
 2. When it clicks on an **hyperlink** on the page
 3. When it submits data from a **form**



User interaction

7

- ▶ User may send HTTP request indirectly, through Javascript code executed on the client (*outside the scope of this course*)
- ▶ Browser detects user interaction events, such as keyboard pressed, mouse clicks or movements, etc.
- ▶ Browser events are passed on, and handled by, Javascript event handlers
- ▶ Javascript code creates and send HTTP Requests to the server and receives HTTP Responses from the server



User interaction on HTML

8

- ▶ Without Javascript, there are only 2 ways for an HTML page to offer users the possibility to interact with the Web Application (send HTTP requests):
 - ▶ **Hyperlinks**
 - ▶ **GET** method
 - ▶ **Forms**
 - ▶ **GET or POST** method



User interaction on HTML

9

▶ HTTP Request – Method **GET**

- ▶ Hyperlink on the page
- ▶ Form with method get
- ▶ *Javascript code*

```
<a href="a.php?id=1">...</a>
```

```
<form method="GET" . . . >  
. . .  
</form>
```

▶ HTTP Request – Method **POST**

- ▶ Form with method post
- ▶ *Javascript code*

```
<form method="POST" . . . >  
. . .  
</form>
```



10

2 –FORMS



Forms

11

- ▶ A form is a section of a Web Page with input fields that allow users to enter information and send it to the Web Server
- ▶ When the form is submitted, information in the input fields is packaged up and sent off to a Web server to be processed by a Web application
- ▶ One Web Page may have several Forms – each Form should have a submit button (otherwise, Javascript is required to send data to the server)



Forms

12

- ▶ Forms have 2 fundamental attributes:
 - ▶ **action** – URL (absolute or relative) that defines where the data is submitted to
 - ▶ **method** – the HTTP method used on the HTTP request that sends the data
 - ▶ Possible values: **get** or **post** (default is get)



Inputs

13

- ▶ There are several types of input fields (text, checkbox, radiobutton, password, etc...)
- ▶ Each input field included on the form will translate to a parameter sent to the server.
- ▶ Input field **name** will be the same as the parameter name submitted to the Web Server
 - ▶ name ≠ id
- ▶ Input field **value**, will translate to the parameter value submitted to the Web Server



Form Example 1 (get method)

14

```
<form action="process.php" method="get">
    First Name: <input type="text" name="firstname">
    Age:          <input type="text" name="age">
                  <input type="submit">
</form>
```



First Name:

Age:



Form Example 1 (get method)

15

Where data is sent to

HTTP Method to use

Data to send

```
<form action="process.php" method="get">  
    First Name: <input type="text" name="firstname">  
    Age: <input type="text" name="age">  
    <input type="submit">  
</form>
```

First Name: John

Age: 25

Submeter

Button to submit data

When user clicks on
submit button

action

Name of input fields

http://www.site.com/process.php?firstname=John&age=25

Values introduced by the User



Get Method

16

- ▶ When the user submits the form, every field will be part of the URL (in the **Query String**). e.g.:

`http://myserver.com/process.php?firstname=John&age=25`

- ▶ 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>



Form Example 2 (post method)

17

```
<form action="process.php" method="post">
    First Name: <input type="text" name="firstname">
    Age:          <input type="text" name="age">
                  <input type="submit">
</form>
```



First Name:

Age:



Form Example 2 (post method)

18

Where data is sent to

HTTP Method to use

Data to send

```
<form action="process.php" method="post">
    First Name: <input type="text" name="firstname">
    Age: <input type="text" name="age">
    <input type="submit">
</form>
```

First Name: John

Age: 25

Submeter

Button to submit data

When user clicks on
submit button

POST /process.php

content-type: application/x-www-form-urlencoded

...

firstName=Johnage=25



Post Method

19

- ▶ When the user submits the form, none of the fields will be part of the URL. e.g.:

`http://myserver.com/process.php`

- ▶ Instead, data will be within the HTTP Request
 - ▶ Better option for sending sensitive data (passwords, uids, etc).
 - ▶ No size limit (on the client) – it is limited by the server to prevent a type of DoS (Denial of Service) attack



Forms – Read data on Web Server

20

- ▶ When the Web Server receives an HTTP request it can access the form data (fields values) using one of the following built-in variables:
- ▶ **`$_GET`** – an associative array with all data from the Query String (GET Method)
- ▶ **`$_POST`** – an associative array with all data passed through a POST Method request (on the HTTP request body)



\$_GET & \$_POST usage

21

```
.../process.php ?firstname=John+Doe&age=25
```



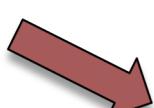
Array \$_GET	
key	value
firstname	"John Doe"
Age	"25"

POST /process.php

content-type: application/x-www-form-urlencoded

...

```
firstName=John+Doeage=25
```



Array \$_POST	
key	value
firstname	"John Doe"
Age	"25"



$\$_GET$ & $\$_POST$ usage

22

- ▶ Read GET data on the server

```
<?php  
    echo "First Name is " .  $\$_GET["firstname"]$ ;  
    . . .
```

Name of Parameter = Name of input field

- ▶ Read POST data on the server

```
<?php  
    echo "First Name is " .  $\$_POST["firstname"]$ ;  
    . . .
```



\$_GET & \$_POST usage

23

```
<form action="process.php?id=378" method="post">
    First Name: <input type="text" name="firstname">
    Age:         <input type="text" name="age">
    <input type="submit">
</form>
```

- ▶ Read data (GET & POST) on the server



```
<?php
    $id=    $_GET["id"];
    $name=  $_POST["firstname"];
    $age=   $_POST["age"];
    . . .
```



Empty \$_GET & \$_POST

24

- ▶ When the form is not yet submitted, \$_GET or \$_POST are still empty (arrays with 0 elements)
- ▶ When data is submitted to the same page where the form is, there are 2 moments to consider:
 - ▶ First time the page is opened
(form not submitted yet)
 - ▶ After the page is submitted

```
if (!empty($_POST)) { //Form was submitted
    $value = $_POST["fieldName"];
}
else { // Form was not submitted
    ...
}
```



GET versus POST

25

- ▶ GET main disadvantages
 - ▶ Data size is limited by the browser
 - ▶ Does not allow file uploads
 - ▶ Data is sent to the server in the URL
 - ▶ Inappropriate to sensitive data
- ▶ GET main advantages
 - ▶ Data can be persistent on a URL, which means that:
 - ▶ Requests can be bookmarked
 - ▶ Hyperlinks can include GET data:

```
<a href='productdetail.php?id=22'>...</a>
<a href='productdetail.php?id=41'>...</a>
<a href='productdetail.php?id=96'>...</a>
```



GET versus POST

26

- ▶ POST main disadvantages
 - ▶ Cannot be bookmarked
 - ▶ Cannot be associated to an hyperlink
- ▶ POST main advantages
 - ▶ Data size is limited by server only
 - ▶ Allows file upload
 - ▶ Data is invisible (inside the body of the message)
 - ▶ This does not means that it is secure!
 - ▶ Data is still available for HTTP Sniffers
 - ▶ Only an encrypted HTTP protocol (HTTPS) guarantees that data is secure



GET versus POST

27

- ▶ When to use the **get** method?
 - ▶ When form is not used to save (insert, update, delete) any data, but instead it is used to select, search or filter data (read data)
- ▶ When to use the **post** method?
 - ▶ When form is used to save (insert, update, delete) data
 - ▶ When form is used to upload files
 - ▶ Whenever is necessary to hide information (e.g. authentication form)



File upload

28

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

```
<form action="upload.php" method="post"  
      enctype="multipart/form-data">  
    <input type="text" name="description"> ...  
    <input type="file" name="ficheiro"> ...  
    <input type="submit"> ...  
</form>
```

Input for file upload.
Where user defines which file to upload

File Upload Escolher ficheiro



File upload – access file on the server

29

- ▶ **`$_FILES`** – an associative array with all data relative to files uploaded on the HTTP request

```
<h3>$_FILES</h3>
<?php var_dump($_FILES); ?>
<h3>$_POST</h3>
<?php var_dump($_POST)
```

\$_FILES

```
/home/vagrant/ainet/site1/demos/04/upload.php:12:
array (size=1)
  'filefromclient' =>
    array (size=5)
      'name' => string '__document.pdf' (length=15)
      'type' => string 'application/pdf' (length=15)
      'tmp_name' => string '/tmp/php1c3XDN' (length=14)
      'error' => int 0
      'size' => int 121396
```

\$_POST

```
/home/vagrant/ainet/site1/demos/04/upload.php:14:
array (size=1)
  'description' => string 'Description Text' (length=16)
```



File upload – access file on the server

30

- ▶ When Web Server receives the HTTP request with an uploaded file, it creates a temporary copy of that file
 - ▶ The name of that file is: `$_FILES["field"]["tmp_name"]`
- ▶ Typical flow to handle upload files on the server, involves moving the temporary file to a known location with a known file name (or to a database). E.g.:

```
move_uploaded_file($_FILES["field"]["tmp_name"],  
                  "myfolder/files/file$someID");
```

- ▶ The original file name on the client (`$_FILES["f"]["name"]`) is available just for information purposes – it has nothing to do with the file name on the server



Superglobal Variables

31

- ▶ Superglobal variables are always available
 - ▶ **`$_GET`** - **`$_POST`** - **`$_FILES`**
 - ▶ **`$_COOKIE`** - HTTP Cookies
 - ▶ **`$_REQUEST`** - `$_GET` + `$_POST` +. `$_COOKIE`
 - ▶ **`$_FILES`** -- files uploaded on the HTTP Request
 - ▶ **`$_ENV`** – Information about the environment where PHP is running from
 - ▶ **`$_SERVER`** - headers, paths, and script locations – information provided by the Web Server. Example:

```
<form action="<?=$_SERVER["PHP_SELF"] ?>" method="post">
```



filename of the currently executing script.



Form - Security

32

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

```
$value = htmlspecialchars($_GET["fieldname"]);
```

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 (\)



Form – Server Validation

33

- ▶ 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()

34

▶ Using preg_match() to validate input format

\$firstName can only have letters and white spaces

```
if (empty($firstName)) {  
    $firstNameErr = "First name is required";  
} elseif (!preg_match("/^[\w ]*$/", $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()

35

- ▶ Using preg_match() to obtain text with the matched pattern

```
// get host name from URL
preg_match('@^(?:http://)?([^\/]*)@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()

36

► 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($firstName,  
    FILTER_VALIDATE_REGEXP,  
    ['options' => ['regexp' => '/^[\w\W ]+$/']]))) {  
    $err= 'Only letters and whitespaces are allowed';  
}
```

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



3 – FORM CONTROLS



Types of Form Controls

38

- ▶ All fields values are submitted to the server as strings.
- ▶ However, on the web page HTML, fields may have different types of controls to represent different types of data.
 - ▶ For example, a field that represents a date may be associated to a calendar for date input. When the form is submitted to the server, the date will be passed on the server as a string



Text & Password <input>

39

▶ Text

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

This will be the **default value**
Initial value when the page opens

▶ Password

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



Radio <input>

40

Male: Female:

```
<label for="input_sex_male_id">Male:</label>
<input type="radio" id="input_sex_male_id"
       name="input_sex" value="male" checked>

<label for="input_sex_female_id">Female:</label>
<input type="radio" id="input_sex_female_id"
       name="input_sex" value="female">
```

- ▶ All “radio” controls with the same name work together
- ▶ Each radio control should have a distinct value
- ▶ Attribute checked defines which is selected when the web page is opened – it defines the value



Checkbox <input>

41

Is Active:

```
<br><label for="id_active">Is Active:</label>
<input type="checkbox" id="id_active"
       name="active" value="yes" checked>
```

- ▶ Attribute **checked** defines if the control has a checkbox or not, when the web page is opened.
- ▶ Value is passed on to the server, if the control is checked.
- ▶ If the control is not checked, **nothing** is passed on to the server (field is not available on \$_GET or \$_POST)
- ▶ Example of code on the server:

```
Is Active = <?= $_POST['active'] ?? 'no' ?>
```



Checkbox <input> (Multiple Values)

42

```
<input type="checkbox" name="courses[]"  
<input type="checkbox" name="courses[]"  
<input type="checkbox" name="courses[]"
```



```
$coursesValues = $_POST["disc"]; // ARRAY  
echo "First course = " . $coursesValues[0];
```

- ▶ Multiple “checkbox” controls with the same name work together
- ▶ Name suffix should include [] → value will be available on the server as an array
- ▶ Each checkbox control should have a distinct value
- ▶ Attribute checked defines which controls are selected when the web page is opened
- ▶ If no checkbox is selected, field is not passed on to the server



Checkbox <input> (Multiple Values)

43

Tecnologias de Internet:

Aplicações para a Internet:

Desenvolvimento Aplicações Distribuídas:

```
<br><label for="id_ti">Tecnologias de Internet:</label>
<input type="checkbox" id="id_ti" name="courses[]" value="TI" checked>

<br><label for="id_ainet">Aplicações para a Internet:</label>
<input type="checkbox" id="id_ainet" name="courses[]" value="AInet">

<br><label for="id_dad">Desenvolvimento Aplicações Distribuídas:</label>
<input type="checkbox" id="id_dad" name="courses[]" value="DAD" checked>
```



<input> controls

44

Type

text

Text field

Texto

password

Field to input passwords

....

Same as texto field, but hides the value

checkbox

Logical values (or set of logical values)

Valor 1

radio

Exclusive multiple values (select only one)

Radio 1 Radio 2

submit

Button to submit form

Submeter

reset

Button to clean form data

Repor

button

Button (with no semantic)

Botão 1

Requires Javascript to execute something when clicked



<input> controls

45

Type

Hidden field

hidden

Not visible on the Web page, but the value is passed on to the client and back on to the server

file

File upload

[File Upload](#) [Escolher ficheiro](#)

image

Image that can be used to submit the form

Also sends the coordinates (x,y) from the position where it is clicked



<input> controls

46

- ▶ New <input> controls on **HTML 5**
 - ▶ Support is not Universal
 - ▶ Check compatibility on: <http://html5test.com/>

Tipo

date	Date (day)	<input type="date" value="05-03-2013"/>
search	To search	<input type="search" value="Pesq"/>
email	E-mail (validates format)	
color	Pick a color	<input type="color"/>
number	Numeric field	<input type="number" value="3,2"/>
range	Numeric field with range selection	<input type="range"/>



<input> controls

47

► New <input> controls on HTML 5

Type

url

URL (validates format)

www.meusite.com

! Introduza um URL.

tel

Telephone number

month

Month (includes year)

Abril de 2008



week

Week (includes year)

Semana 11, de 2013



time

Time (hour)

14:25

datetime-local

Date and time

07-03-2013 14:25



Wih no timezone information



<select> control

48

- ▶ To choose one value from a list of values in a dropdownlist



```
<select name="active_course">
    <option value="TI">
        Tecnologias de Internet
    </option>
    <option value="AInet" selected>
        Aplicações para a Internet
    </option>
    <option value="DAD">
        Desenvolvimento Aplicações Distribuídas
    </option>
</select>
```



<select> control

49

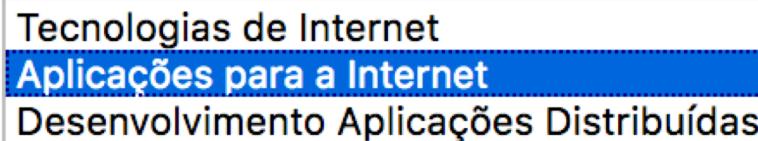
- ▶ Each <option> element has a value and a text description
 - ▶ Value – what is submitted to the server
 - ▶ Text description – what the user “sees”
- ▶ “selected” attribute is applied to the option element that is selected – it defines the value of the <select> control
- ▶ If “value” attribute is not specified, value will be the same as the text description



<select> control – multiple selection

50

- To choose multiple value from a list of values, just add the attribute “multiple” to the select control



```
<select name="multiple_courses" multiple>
    <option value="TI">
        Tecnologias de Internet
    </option>
    . . .
</select>
```



<textarea> control

51

- ▶ To input text with several lines

Initial text on the box

```
<textarea name="textarea_field" rows="5" cols="48">  
Initial text on the box  
</textarea>
```

- ▶ Initial value of the textarea field is defined by the content of <textarea> element, not by any attribute



<fieldset>

52

- ▶ Groups (only visually) related fields on the form
- ▶ Can include a <legend> with group description

Condiments

Salt
 Pepper
 Garlic

```
<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<br>
</fieldset>
```



Common attributes

53

- ▶ Some attributes that are common to several input types:
 - ▶ **name**: name of the input field
 - ▶ **value**: specifies the initial value for an input field
 - ▶ **readonly**: specifies that the field is read only (cannot be changed)
 - ▶ **disabled**: specifies that the input field is disabled
 - ▶ **size**: specifies the size (in characters) for the input field
 - ▶ **maxlength**: maximum allowed length



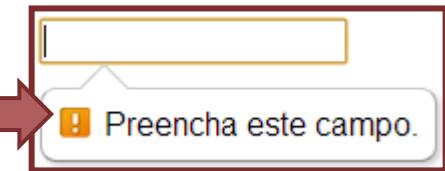
Common attributes – HTML 5

54

▶ **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">
```



▶ **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">
```



HTML 5 compatibility

55

- ▶ 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/>



9 – REFERENCES



References

57

▶ HTML 5 Forms:

- ▶ <http://www.w3schools.com/html/default.asp>
- ▶ http://www.w3schools.com/html/html_forms.asp

▶ 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

▶ Regex

- ▶ <http://php.net/manual/en/pcre.pattern.php>
- ▶ <http://www.phpliveregex.com>



References

58

- ▶ Official (PHP)
 - ▶ <http://www.php.net/>
 - ▶ <http://php.net/manual/en/>
 - ▶ http://php.net/manual/pt_BR/
- ▶ Other documentation:
 - ▶ Luke Welling and Laura Thomson, “PHP and MySQL Web Development (4th Edition)”, Addison-Wesley 2009
 - ▶ <http://www.w3schools.com/php/default.asp>
 - ▶ http://commons.oreilly.com/wiki/index.php/PHP_Cookbook
 - ▶ <http://www.tuxradar.com/practicalphp>
 - ▶ <http://www.tizag.com/phpT/>
- ▶ HTML 5
 - ▶ <http://www.ibm.com/developerworks/library/wa-html5structuraltags/index.html>
 - ▶ <http://html5doctor.com>