

# Bases de données

*Prof. Giovanna Di Marzo Serugendo*

*Assistant: Mohammad Parhizkar*

[mohammad.parhizkar@unige.ch](mailto:mohammad.parhizkar@unige.ch)

Semestre: printemps 2019

session 7- 08/04/2019



UNIVERSITÉ  
DE GENÈVE

CENTRE UNIVERSITAIRE  
D'INFORMATIQUE

# Agenda

  
session 25 / 03 / 2019

- PHP Forms
- PHP Form Handling
- PHP Form Validation
- PHP Form Required
- PHP Form URL / e-mail
- TP6 (*deadline: 06/05/2019*)

**There is no TP session on 22/04/2019 and 29/04/2019!**

# Form Validation

Proper validation of form data is important to protect your form from hackers and spammers.

Example:

localhost:8888/Session6\_ImportedDataValid

## Session 6: PHP Form Validation

\* required field.

Name:  \* 1

E-mail:  \* 2

Website:  3

Comment:  4

Gender: ☐ Female ☒ Male \* 5

### Your Input:

Thierry Condé  
thierry.condé@hr.com  
www.thierry.com  
He was the head of IT department.  
male

In this example we have five `$_POST[...]` fields:

## Field

## Validation Rule

- | Field     | Validation Rule  |
|-----------|--|
| 1 Name    | Required. + Must only contain letters and whitespace<br>Name: <code>&lt;input type="text" name="name"&gt;</code>                         |
| 2 E-mail  | Required. + Must contain a valid email address (with @ and .)<br>E-mail: <code>&lt;input type="text" name="email"&gt;</code>             |
| 3 Website | Optional. If present, it must contain a valid URL<br>Website: <code>&lt;input type="text" name="website"&gt;</code>                      |
| 4 Comment | Optional. Multi-line input field (textarea)<br>Comment: <code>&lt;textarea name="comment" rows="5" cols="40"&gt;&lt;/textarea&gt;</code> |
| 5 Gender  | Required. Must select one  |

## Radio Buttons:

Gender:  
`<input type="radio" name="gender" value="female">Female`  
`<input type="radio" name="gender" value="male">Male`

# Form Validation

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

**test\_input()**

The **first** thing we will do is to create a function that will do all the checking for us:

- PHP **trim()** function: Remove unnecessary characters (extra space, tab, newline) from the user input data
- PHP **stripslashes()** function: Remove backslashes (\) from the user input data (with the)
- PHP **htmlspecialchars()** function: The last part is to pass all variables through php htmlspecialchars() function. This function converts some predefined characters to HTML entities.

# Form Validation: Required Fields

```
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }

    if (empty($_POST["website"])) {
        $website = "";
    } else {
        $website = test_input($_POST["website"]);
    }

    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
        $comment = test_input($_POST["comment"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>
```

## New variables:

\$nameErr, \$emailErr, \$genderErr, and \$websiteErr.

These **error variables** will hold error messages for the required fields.

Let's add an **if else** statement for each \$\_POST...

## PHP empty() function

- To check if the \$\_POST[...] is empty or not.
- If it is empty an error message will be saved in the new variable.
- If it is not empty, it will be sent to **test\_input()** function.

# Form Validation: Required Fields

## HTML part:

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

Name: <input type="text" name="name">

```
<span class="error">* <?php echo $nameErr;?></span>  
<br><br>
```

E-mail:

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

```
<span class="error">* <?php echo $emailErr;?></span>  
<br><br>
```

Website:

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

```
<span class="error"><?php echo $websiteErr;?></span>  
<br><br>
```

```
Comment: <textarea name="comment" rows="5" cols="40"></textarea>  
<br><br>
```

Gender:

```
<input type="radio" name="gender" value="female">Female
```

```
<input type="radio" name="gender" value="male">Male
```

```
<span class="error">* <?php echo $genderErr;?></span>  
<br><br>
```

```
<input type="submit" name="submit" value="Submit">
```

```
</form>
```

We add the error message of each field.

A <span> tag is used to change the text style (e.g. color).

# Form: Validate E-mail and URL

Let's see how we can validate the input data: names, e-mails, and URLs...

## We need to validate:

**Name:** The Name field should contain only letters and whitespace.

**E-mail:** The E-mail field should contain a valid e-mail address syntax.

**Website URL:** The Website field should contain a valid URL (if filled out).

# Form: Validate E-mail and URL

```
$name = test_input($_POST["name"]);  
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

The **preg\_match()** function searches a string for pattern, returning true if the pattern exists, and false otherwise.

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

**filter\_var()** function: Returns the filtered data, or **FALSE** if the filter fails.

```
$website = test_input($_POST["website"]);  
if (!filter_var($website, FILTER_VALIDATE_URL)) {  
    $websiteErr = "Invalid URL";  
}
```

example:

```
<?php  
var_dump(filter_var('bob@example.com', FILTER_VALIDATE_EMAIL));  
var_dump(filter_var('bob@example_com', FILTER_VALIDATE_EMAIL));  
?>
```

print

string(15) "bob@example.com"  
bool(false)





# TP6 - Hint

deadline: 06.05.2019

# TP6-Practical Information

We want to insert some data through a website to the Employees table.

```
CREATE TABLE employees (  
  employee_id INT  NOT NULL,  
  first_name  VARCHAR(14) NOT NULL,  
  last_name   VARCHAR(16) NOT NULL,  
  email       VARCHAR(20) NOT NULL,  
  phone_number VARCHAR(20),  
  hire_date   DATE NOT NULL,  
  job_id      VARCHAR(20) NOT NULL,  
  salary      INT,  
  commission_pct DOUBLE(3, 1),  
  manager_id  INT NOT NULL,  
  department_id INT NOT NULL,  
  PRIMARY KEY (employee_id)  
);
```

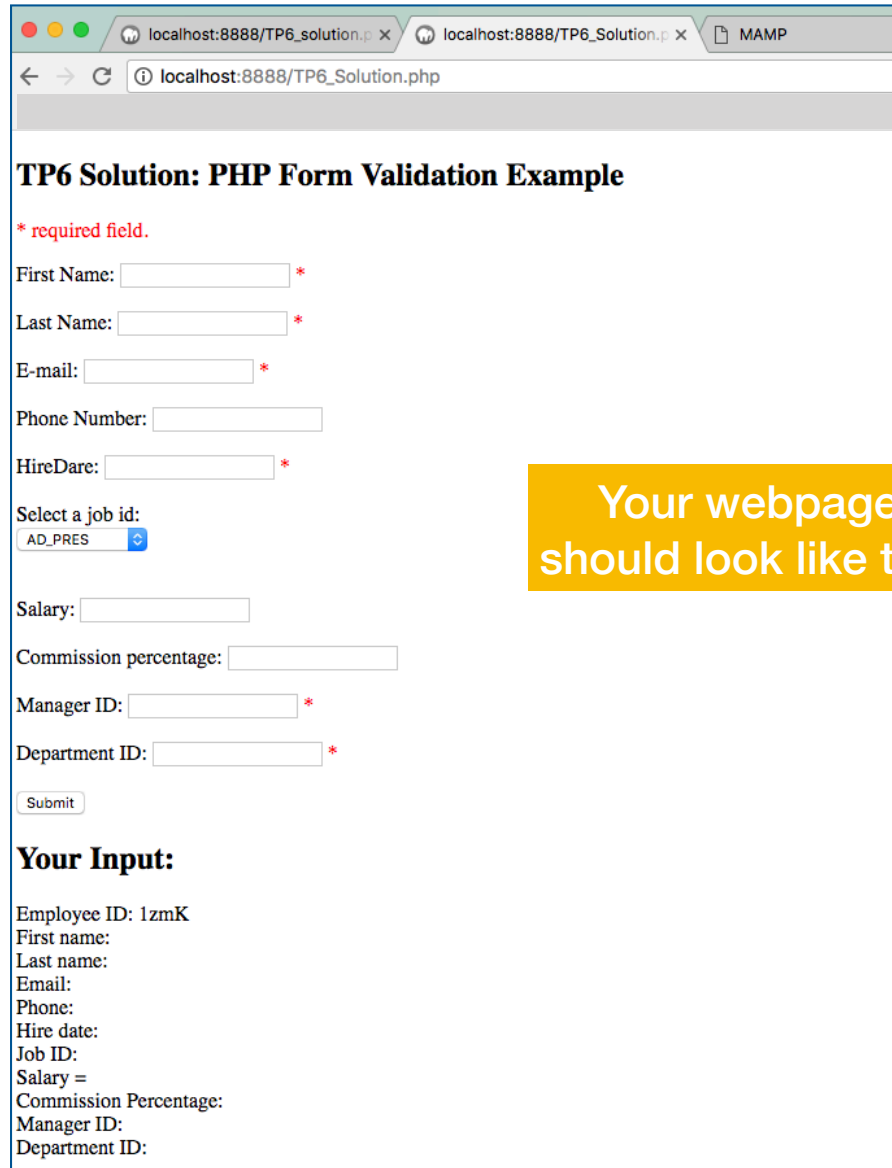
**VARCHAR(4)** 

**1**

**phpMyAdmin: you must make this table, in HR database**

# TP6-Practical Information

We want to insert data through a platform to the Employees table.



**TP6 Solution: PHP Form Validation Example**

\* required field.

First Name:  \*

Last Name:  \*

E-mail:  \*

Phone Number:

Hire Date:  \*

Select a job id:  
AD\_PRES

Salary:

Commission percentage:

Manager ID:  \*

Department ID:  \*

**Your Input:**

Employee ID: 1zmK  
First name:  
Last name:  
Email:  
Phone:  
Hire date:  
Job ID:  
Salary =  
Commission Percentage:  
Manager ID:  
Department ID:

These fields can not be '*null*':

- First Name
- Last Name
- Email
- Hire Date
- Manager ID
- Department ID

Your webpage  
should look like this

# TP6-Practical Information

- You need to make one .php file ('TP6\_yourname' .php).

✓ Also, uploading two .php files can be ok (if you could not put everything in one file) → submit as .zip file

```
<html>
<body>
  <?php
  ?>
  <form
  </form>
</body>
</html>
```

for each html field, we must have a variable to keep the value and another variable to keep the error related to it.

```
$lastNameErr =$firstNameErr =$emailErr = $phoneNumberErr =
$hireDateErr=$job_idErr=$salaryErr 
=$commission_pctErr=$manager_idErr=$department_idErr=False;
```

```
$salary=$lastName =$firstName = $email = $phoneNumber =
$hireDate=$job_id=$commission_pct= $manager_id=$department_id = "";
```

# TP6-Practical Information

```
if(preg_match( '/^[+]?([\d]{0,3})?[\(\)\.\-\\s]?([\d]{3})[\(\)\.\-\\s]*([\d]{3})[\.\-\\s]?([\d]{4})$/', $phoneNumber ) ) {
```

*How to check a field related to phone number*

```
// specify your date's original format, in this example m/d/Y (e.g. 08/31/2013)
if (empty($_POST["hireDate"])) {
    $hireDateErr="Hire date is required.";
} else {
    $format = "m/d/Y";
    $hireDate = DateTime::createFromFormat($format, $_POST["hireDate"]);
    if(!$hireDate) {
        // createFromFormat returns false if the format is invalid;
        $hireDateErr = "Invalid date format";
    } else {
        //change it to any format you want with format() (e.g. 2013-08-31)
        $hireDate= $hireDate->format("Y-m-d");
    }
}
```

*How to check a field related to hire\_date*

# TP6-Practical Information

localhost:8888/TP6\_solution.p x localhost:8888/TP6\_Solution.p x MAMP

localhost:8888/TP6\_Solution.php

### TP6 Solution: PHP Form Validation Example

\* required field.

First Name:  \*

Last Name:  \*

E-mail:  \*

Phone Number:

HireDate:  \*

Select a job id:

Salary:

Commission percentage:

Manager ID:  \*

Department ID:  \*

### Your Input:

Employee ID: 1zmK  
First name:  
Last name:  
Email:  
Phone:  
Hire date:  
Job ID:  
Salary =  
Commission Percentage:  
Manager ID:  
Department ID:

in HTML part:

```
<label for="job_id">Select a job id:</label>
<div class="custom-select" style="width:200px;">
  <select name="job_id">
    <option value="AD_PRES">AD_PRES</option>
    <option value="AD_VP">AD_VP</option>
    <option value="IT_PROG">IT_PROG</option>
    <option value="FI_MGR">FI_MGR</option>
    <option value="FI_ACCOUNT">FI_ACCOUNT</option>
    <option value="PU_MAN">PU_MAN</option>
    <option value="PU_CLERK">PU_CLERK</option>
    <option value="ST_MAN">ST_MAN</option>
    <option value="ST_CLERK">ST_CLERK</option>
    <option value="SA_MAN">SA_MAN</option>
    <option value="SA_RAP">SA_RAP</option>
    <option value="SA_REP">SA_REP</option>
  </select>
</div>
<br><br>
```

# TP6-Practical Information

localhost:8888/TP6\_solution.p x localhost:8888/TP6\_Solution.p x MAMP

localhost:8888/TP6\_Solution.php

## TP6 Solution: PHP Form Validation Example

\* required field.

First Name:  \*

Last Name:  \*

E-mail:  \*

Phone Number:

HireDate:  \*

Select a job id:

Salary:

Commission percentage:

Manager ID:  \*

Department ID:  \*

### Your Input:

Employee ID: 1zmK

First name:

Last name:

Email:

Phone:

Hire date:

Job ID:

Salary =

Commission Percentage:

Manager ID:

Department ID:

```
//-----Generate unique ID-----  
//$employee_id=uniqid ( $prefix = "employee_" );  
//set the random id length  
$emp_id_length = 4;  
//generate a random id encrypt it and store it  
$emp_id = crypt(uniqid(rand(),1));  
//to remove any slashes that might have come  
$emp_id = strip_tags(stripslashes($emp_id));  
//Removing any . or / and reversing the string  
$emp_id = str_replace(".", "", $emp_id);  
$emp_id = strrev(str_replace("/", "", $emp_id));  
//finally I take the first 3 characters from the $rnd_id  
$emp_id = substr($emp_id,0,$emp_id_length);
```

# TP6-Practical Information

Server: localhost:8889 » Database: HR » Table: employees

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Run SQL query/queries on table HR.employees: ?

Columns: phone\_number, hire\_date, job\_id, salary, manager\_id

SELECT \* SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters ?

Delimiter ; ] ☒ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

1  
2  
3  
4  
5

A red arrow points to the large text area for entering SQL queries.

**last part:** transfer the data to the HR database

Example:

```
INSERT INTO `employees` ( ... ) VALUES ( ... );
```



# Thank you!