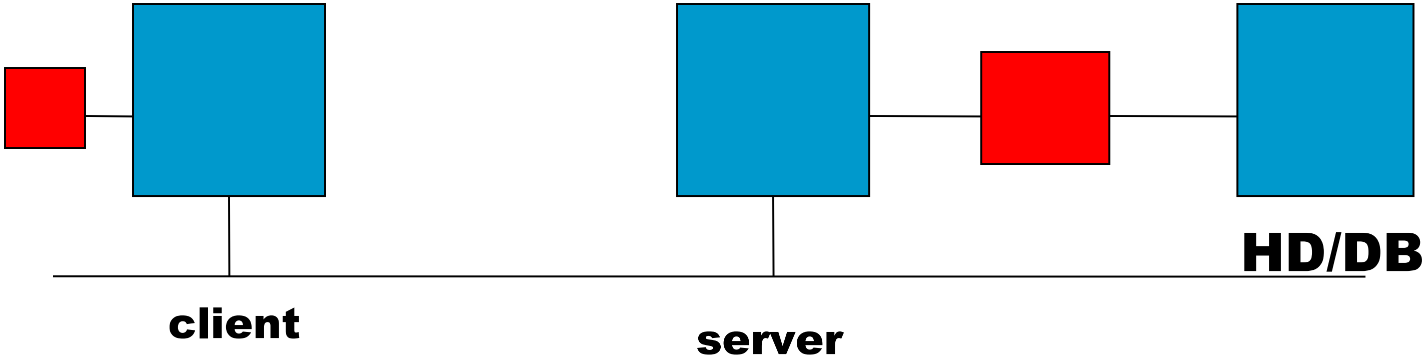
PHP: introduction

- Dynamic model



- Included with HTML code
- Execution server ide
- Better security...

Main lines

- Inclusion PHP: specific markup

```
<?php…
..........
?>
```

- File extension: .php instead of .html
- Syntax:similar to C/JavaScript/Java
- Execution server side ;-)
- To practice : WAMP or LAMP

1st example

```
<html>
<head>
  <title>test PHP</title>
</head>
<body>
<?php echo "Hello <strong>Everybody</strong>"
?>
</body>

</html>
```

- Exercise: center, bold… ;-)
- PHP = text generator

Include command

```
<html><head>
  <title>MODULARITE</title>
</head>
<body>
<?php  include("header.html");
       include("menu.html"); ?>
Blablabla HTML...(main content)..
<?php include("footer.html"); ?>
</body></html>
```

- Include external text files
- If missing file
  - `include` --> warning (more flexible)
  - `require` --> fatal error

Main template for a web page

```
<?php
include(“header”);
include(“menu”);
?>
Bla bla bla
<?php include(“footer”); ?>
```

- Where are the div? Within included files!
- Where is JavaScript? Within included files!
- Can be recursive : ex. include(“alert.php”)
- What is sent to the client? HTML only ;-)

With forms!

```
<body>
<form name="maforme
   "action="alert.php" method="POST" >
enter your surname:
   <input type="text" name="yoursurname"><br/>
enter your firstname:
   <input type="text" name="yourfirstname">
   <input type="submit" value="clic here">
</form>
</body>
</html>  Nothing really new...
```

- **To be written: alert.php**

Get the parameters

```
<?php
$name = $_POST['yourfirstname'];
if ($name=="") {echo "enter your name please !";}
else {
   $string= "<strong>Hello " . $name .
   "</strong>";
   echo $string;
   } ?>
```

- . : concatenation operator
- Boring syntax $_POST[‘...’] !
- Variable name with prefix $
- No type declaration (see JavaScript)
- Case sensitive PhP != pHp ;-)

Simple exercise

- Add an entry for the firstname and display «Hello firstname surname» with PHP
- Check the parameters with PHP.
- Mix JavaScript/PHP: checking client side with Javascript
  - Write a script `checkParam(…)`
  - Event OnSubmit to add as a form parameter
  - Action is executed iff OnSubmit returns true

```
<form action=”...php” method=”POST”
  OnSubmit=”return(checkParam(.))”>
```

Pro/cons

- **Easy integration with HTML**
- **Syntax similar to C/Java**
  - **OO philosophy**
  - **Modularity with functions, objects**
- **No compatibility issue with browsers**
- **A lot of web sites to learn**

But...

Security again ….!

Files with PHP

- Add a counter (web profiling) ?
- How ?
- Idea: (server side... of course!)
  - A file with the counter value inside
  - Read counter value in the file
  - Update counter value and write

» Need file handling functions !

File functions

- Functions (similar to C)
  - fopen(filename, mode (r w a)
  - fread/fgets(file handler, size data to be read)
  - fwrite/fputs(file handler, data)
  - fclose(file handler) //very important
  - include /require
  - etc...
- What about our counter now ?

The code...

```
<?php function calc(){
$file = fopen("counter.txt", "r");        //we read
$counter = fread($file , filesize("counter.txt")) ;
fclose($file) ;
$file = fopen ("counter.txt" , "w");
$counter = $counter + 1 ;                 //we update
fwrite($file , $counter ) ;               //we write
fclose($file) ;
return $counter; }
  ?></head>
<body>
hello.<br>
You are the <?php  echo calc() ?> visitor. //we display
</body>
```

A simple exercise

- Add a « reset » button to reset the counter (Add a reset.php file)

  This is not a basic reset HTML button;-)

- Modify the code in such a way as the counter appears in the same page in a text area bottom of the page

Files handling functions

- feof($file): boolean end of file
- explode($delim, $string)
  - Return an array
  - Allow to analyze a file
- file_exists($file): boolean
- Etc... classic functions (see Web)

Upload files?

- A) Prepare the form

```
<form method=“POST” action=“get-file.php”
       enctype=“multipart/form-data” >
```

  - Special input type

```
       <input type=“file” name=“myfile ”>
```

  - then submit button as usual ;-)
- B) Write **`get-file.php`** !

get-file.php

- **How does it work server side?**
  - files stored in a temporary area
  - Global variable array 2x2 $_FILES server side
  - 1line per file characteristics of the file
  - access via **`$_FILES['myfile'][‘???']`**
  - Useful function: **`basename`**

```
<?php

$name=($_FILES['myfile']['name']);//name client side

$size=($_FILES['myfile']['size']);//size of the file

$tempname=($_FILES['myfile']['tmp_name']);//temp name
   of the file
```

- **Then move the file …asap ;-)**

move_uploaded_file

- **PHP function**

```
move_uploaded_file(what, where)
```

  - What: temporary file
  - Where: wherever we want to

- **Then...**

```
$name=basename($_FILES['myfile']['name']);
$datafile= "uploads/".$name;
move_uploaded_file($_FILES['myfile']['tmp_name'],
  $datafile)
```

- **Return a Boolean value: TRUE/FALSE**

And we get

```
$size=($_FILES[myfile']['size']);
if ($size==0) {
 echo "Your file is empty. Please choose a non empty
   file.";}
 else { $type=($_FILES[‘myfile']['type']);
        $name=basename($_FILES[‘myfile']['name']);
        $datafile="$workdir" . "$name"; //the new place
   and name where we move the client file
if (move_uploaded_file($_FILES[‘myfile']['tmp_name'],
   $datafile)){// we are happy ;-) we can work....
```

- **Then ... file content manipulation!**

File treatment

- **General scheme**

```
$file=fopen(“monfichier”,”r”);
while(!feof($file)) {
  $line = fgets($file);
  $tab=explode(“,”,$line);//get an
  array
 //we compute };
fclose($file);
```

That’s it !

It's up to you now ;-)

Redirection

- Pb:
  - Redirect from a missing page
  - Enhance the UX

```
header(‘Location:http://www…’)
```

- Should be at the beginning
- Useful for mobiles/tablet PC
- Pb: where does the query come from ?

Client info with PHP

- Pb: who is the client to provide adequate answer (OS and browser)
- Method 1:
  - Variable $HTTP_USER_AGENT
  - String analysis with `strpos` for instance
  - No standard till date
- Method 2
  - file browscap.ini database a
  - fonction `get_browser(a,b)`

```
if (get_cfg_var('browscap')){//do we have browscap.ini
$browser = get_browser(null, true); //we get data in an
   array }
else
{echo "<br/>browscap not set";die();}
```

Browscap analysis…

- We get…

```
Array
(
[browser_name_regex] => ^mozilla/5\.0 (windows; .; ....
[browser_name_pattern] => Mozilla/5.0 (Windows; ?; ....
[parent] => Firefox 0.9
[platform] => WinXP
[browser] => Firefox etc….
}
```

- Analysis and redirection (ex. : bbc, etc.)

Pro/cons

- Pb:
  - browscap.ini difficult to handle
  - browscap.ini not updated

- Idea: client side info
  - Redirection with
    - Light CSS (see mobile.lemonde.fr)
    - With JavaScript

Client Info with JavaScript

Object : navigator

- Browser Code : navigator.appCodeName
- Browser Name: navigator.appName
- Browser Version: navigator.appVersion
- Cookies Enabled: navigator.cookieEnabled
- OS: navigator.platform
- User-agent header:navigator.userAgent

analysis and redirection with Javascript

```
document.location.replace(URL)
```

Conclusion

- HTML/PHP interface
  - simple
  - powerful
- A lot of info (open source)
- Widely distributed and well known issues
- A lot of free packages with PHP(oscommerce, moodle, wordpress,etc.)
- LAMP/WAMP very stable