

# PHP and MySQL

Code: `php-mysql`

## Author and version

- Daniel K. Schneider
- Email: [Daniel.Schneider@tecfa.unige.ch](mailto:Daniel.Schneider@tecfa.unige.ch)
- Version: 0.1 (modified 20/11/07 by DKS)

## Prerequisites

- HTML



## Objectives

- Learn some PHP
- Understand how php can deal with HTML forms
- Understand how php can interact with MySQL

**... sorry, there is some french left in some examples**

# 1. Table of Contents

1. Table of Contents	2
2. Introduction	5
2.1 PHP features	6
2.2 HTML and PHP integration	7
2.3 File inclusion	7
Example 2-1:File inclusion	8
3. The programming language PHP	9
3.1 Elements of programming	9
3.2 PHP syntax	10
3.3 Variables and assignments	10
A.Simple variables	10
Example 3-1:Display variable contents	11
B.Simple arrays	12
Example 3-2:Simple variables and some HTML generation	13
C.Associative arrays and multi-dimensional tables	14
D.Summary - variables	14
3.4 Constants.	15
A.Definition	15
3.5 Simple expressions and operators	16
A.Arithmetic operators	16
Example 3-3:Simple arithmetic	16
B.Operators for strings	17
C.Logical operators	17
D.comparison	18
Example 3-4:Simple comparisons	18
3.6 Selection (Conditions and tests)	19
Example 3-5:Simple "if" (comparison)	20
3.7 PHP functions	22
Example 3-6:Color mixing for paint	23
Example 3-7:HTML generation with functions	24
3.8 Loops (iterations)	25

Example 3-8:Love generation	25
Example 3-9:Generation of html tables	26
<b>4. Practical advice</b>	<b>27</b>
<b>4.1 Debugging</b>	<b>27</b>
<b>4.2 Portals</b>	<b>27</b>
<b>5. HTML forms processing with PHP</b>	<b>28</b>
<b>5.1 Forms processing with PHP I (Calcul)</b>	<b>28</b>
Example 5-1:Simple quiz and POST to a php file	28
A.Retrieve values of an HTML form	30
B.Computing and display of results	31
C.Inhibit direct access to PHP (without data)	31
<b>5.2 Forms processing with PHP II</b>	<b>32</b>
Example 5-2:Checkboxes with PHP - arrays	32
<b>5.3 All in one solution ?</b>	<b>34</b>
<b>5.4 Polishing: Test if we have all the POST/GET variables</b>	<b>35</b>
<b>5.5 Session management</b>	<b>36</b>
Example 5-3:Restrict repetitive access to a page	36
<b>6. On-line surveys and file-based storage</b>	<b>37</b>
Example 6-1:Survey	37
A.The HTML form	37
B.Dump contents of a file	40
<b>7. Other formats than HTML</b>	<b>41</b>
Example 7-1:Generate some simple XML	41
<b>8. PHP and MYSQL</b>	<b>42</b>
<b>8.1 A complete documented example:</b>	<b>42</b>
<b>8.2 Functions PHP - MySQL</b>	<b>43</b>
A.Connect to a database	43
B.Select a database	43
C.Processing of results	44
D.Error handling	45
Example 8-1:simple error handling:	45
Example 8-2:Generation of an HTML table with the results	46

E.Processing a result and using field names	47
Example 8-3:HTML generation using field names	47
<b>9. Workthrough example: Guestbook.</b>	<b>49</b>
<b>9.1 Example files</b>	<b>49</b>
<b>9.2 Structure of the comments table</b>	<b>49</b>
<b>9.3 Task.:</b>	<b>50</b>

## 2. Introduction

- "PHP" = "Hypertext Preprocessor"

History:

- "Personal Home Page Generator" (Php2/FI) in the mid-nineties
- PHP 3 since 1997,
- PHP 4 since 1999,
- PHP 5 since 2004/2005

PHP 3.0

- PHP Version 3.0 is an ***HTML-embedded scripting language***. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

Principle:

- Analogy with JavaScript: PHP code can be mixed with HTML
- BUT: The server reads the files and computes it (and end-user never can see the code)
- Servers are configured to read \*.php files as PHP (sometimes also \*.php3, \*.php4 etc.)

Purpose:

- Dynamic web pages (small applications) or larger web applications

Links:

url: [http://edutechwiki.unige.ch/en/PHP\\_links](http://edutechwiki.unige.ch/en/PHP_links)

url: <http://www.php.net/docs.php>

## 2.1 PHP features

### Availability

- Free and open source (GPL)
- cross-platform (Unix, Linux, BSD, MacOS X, Win32, etc.)

### Installation

- can run as CGI program (external to a web server)
- can run as web server module (this is the standard case, e.g. with the Apache server)
- can be used as command-line scripting engine

### Highlights

- good database support (Oracle, Sybase, Microsoft, MySQL, Postgres, ODBC, etc.)
- good system integration (files)
- complete programming language, including OO support
- easy to learn
- made for internet application (cookies, authentication, sessions, redirection...)
- dozens of integrated libraries (LDAP, PDF, XML, GIF,...)

### Alternatives

- ASP (Microsoft)
- JSP (Java)
- Cold Fusion (Adobe)

## 2.2 HTML and PHP integration

- PHP code is defined within an XML processing instruction

```
<?php ..... ?>
```

```
    <?php echo("if you want to serve XML documents, do like this\n"); ?>
```

## 2.3 File inclusion

- PHP code can be spread of several files

### **Include**

inserts content of file when this expression is evaluated

Syntax: `include ("file name") ;`

Example: `include("style.php") ;`

### **Require**

inserts content of file when the php file is loaded

Syntax: `require ("file name") ;`

Example: `require("my_functions.inc") ;`

Variant (recommended).

`include_once()` and `require_once()`.

only include once, will make your application faster.

## Example 2-1: File inclusion

url: <http://tecfa.unige.ch/guides/php/examples/includes/>

```
<HTML>
  <HEAD>
    <TITLE>Simple Include Demo (21-Apr-1998)</TITLE>
<?php include("style.text"); ?>
  </HEAD>
  <BODY>
    <H1>Simple Include Demo</H1>
```

In this file we include a [style sheet](#) and a [footer](#).

[Look at the formatted source](#)

or the [unformatted one](#)

if you want to know how this is done.

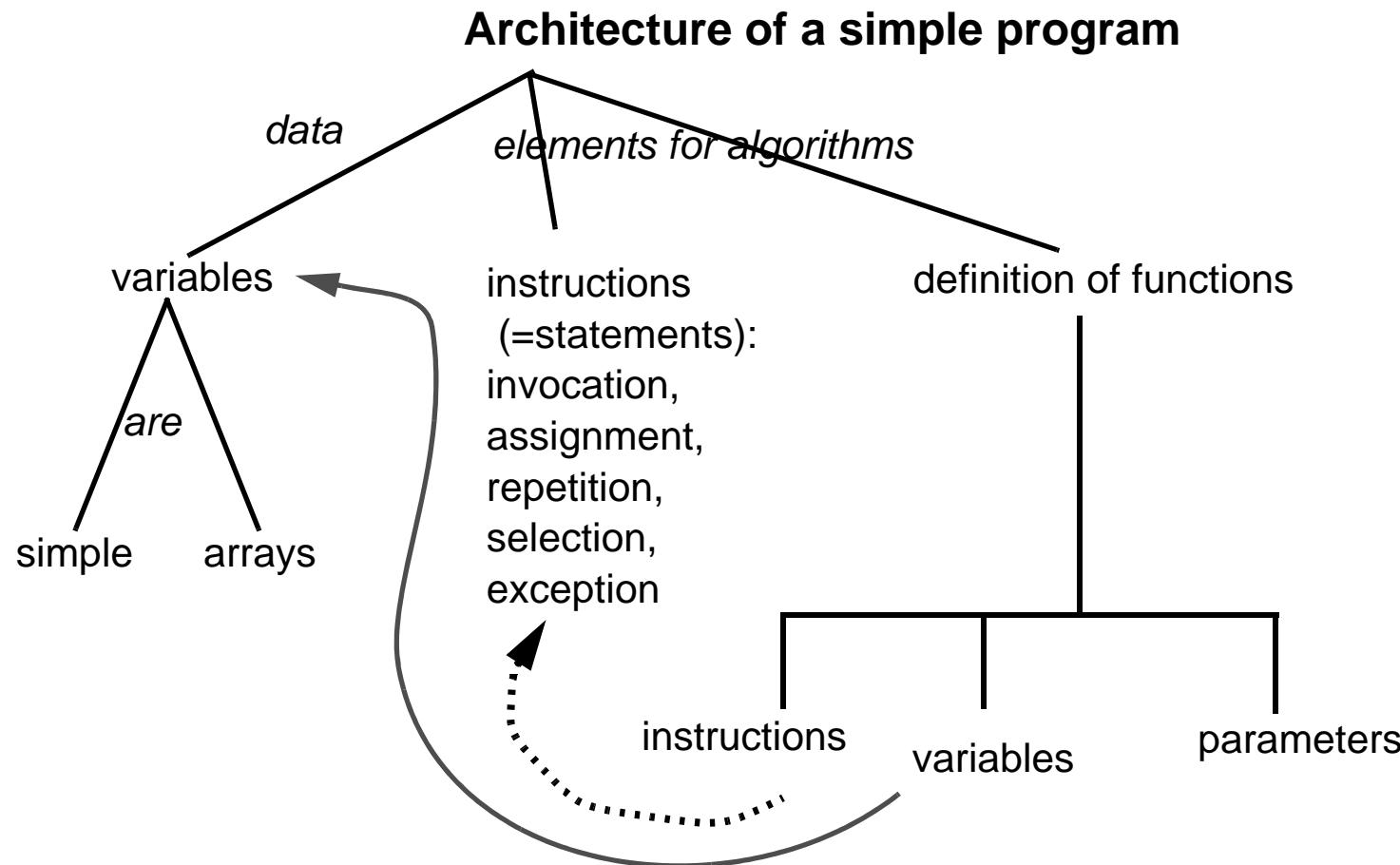
```
<H1>Yet another styled title</H1>
<UL>
  <LI> bullet item </LI>
  <LI> bullet item </LI>
</UL>

<?php
/* A footer */
include( "footer.text" );
?>
  </BODY>
</HTML>
```

# 3. The programming language PHP

## 3.1 Elements of programming

Figure 3-1: Program= algorithm + data structures



*Note: Objects are missing here !*

## 3.2 PHP syntax

- PHP looks like "C" (C, C++, Java, Perl, etc.)
- Each instruction is ended with a ";"
- Comments // or #, or included within /\* ... \*/

## 3.3 Variables and assignments

- Variables are “containers” for information.
- Each identifier with a \$ in front is a variable
- Variables don’t need to be declared
- Assignment operator: =

### A. Simple variables

Syntax: assignment

```
$variable = "content" ;
```

Illustrations:

```
$a = 10;  
$name = "Patrick Ott";  
$sum = 123.456;
```

## Example 3-1: Display variable contents

- Files:

Application	url: <a href="#">/guides/php/examples/simple/simple-echo.php</a>
Source (to see)	url: <a href="#">/guides/php/examples/simple/simple-echo.phps</a>
Source (to copy)	url: <a href="#">/guides/php/examples/simple/simple-echo.text</a>

```
<BODY>
  <H1>Simple Echo of variables with PHP</H1>
```

```
<?php
```

```
$a = 10;
$nom = "Patrick Ott";
$somme = 123.456;
```

```
echo "Le nommé $nom a $somme francs dans la poche, mais il voudrait $a fois
plus.";
```

```
?>
```

```
<p><hr>
</BODY>
```

- echo an “instruction” to display a string (chain of characters)
- Note: all the \$xxx are replaced by their contents !

## B. Simple arrays

- Arrays are a sort of list (several values within the same variable)

Method 1:

```
$numbers[ ] =1;  
$numbers[ ] =2;  
$numbers[ ] =3;  
$numbers[ ] =4;
```

Method 2:

```
$numbers = array (1, 2, 3, 4);  
$names = array ("Pat", "Dave", "Surf", "K");
```

### ***Simple arrays usage***

```
$array[index]
```

- Index starts at 0 ! (zero)

```
echo "Second element of $names is: $names[1];
```

## Example 3-2: Simple variables and some HTML generation

url: See: </guides/php/examples/simple/simple-arrays.php>

```
<?php

// Some simple HTML
echo "<h1>Simple arrays</h1>";

$utilisateur = "cher étudiant";
$no_utilisateur = 3;

$numbers = array (1, 2, 3, 4);
$names = array ("Pat", "Dave", "Surf", "K");
$names[ ] = "Zorro";

// Note html <br> tag
echo "Salut $utilisateur. Vous êtes le numéro $no_utilisateur.<br>";

// echo with concatenation, use it to print complex things
echo "La quatrième personne s'appelle " . $names[3] . " ";

// simple echo
echo "et la cinquième personne s'appelle $names[4].<p>" ;
$n = sizeof($numbers);

// note that we have to use \ in order to print a $ !
echo "We have $n numbers in array \$numbers.";
?>
```

## C. Associative arrays and multi-dimensional tables

```
$fruits = array(  
    "fruits"  => array("a"=>"orange", "b"=>"banana", "c"=>"apple") ,  
    "numbers" => array(1, 2, 3, 4, 5, 6)  
    "holes"    => array("first", 5 => "second", "third")  
) ;
```

## D. Summary - variables

- You should, but don't need to initialize variables

```
$a = 1234; # decimal number  
$a = -123; # a negative number  
$a = 1.234; $a = 1.2e3; # floating point number  
$str = "This is a string"; # string  
$a[0] = "abc"; # element zero of un array  
$a[1] = "def"; # element 1 of an array  
$b["foo"] = 13; # element "foo" of an array
```

## 3.4 Constants.

- Constants are "variables" with information that cannot change.
- Do not use "\$".
- By convention, use capital letters.

### A. Definition

```
Syntax: define(<NAME>, <value>);  
define("PI", 3.14);  
define("REMERCIEMENTS", "Thanx for using our program<br>");  
define("SALUTATIONS", "Je vous prie d'agr er, Madame, Monsieur, l'expression  
de nos sentiments d vou s");  
$radius = 12;  
$perimeter = 2 * $radius * PI;  
echo REMERCIEMENTS;  
echo "le p rim tre du cercle is de " . $perimeter . "<br>";  
echo SALUTATIONS;
```

#### **result:**

Thanx for using our program  
le p rim tre du cercle is de 77.76  
Je vous prie d'agr er, Madame, Monsieur, l'expression de nos sentiments  
d vou s.

## 3.5 Simple expressions and operators

### A. Arithmetic operators

- Like normal math:

example	name	Returns the result:
\$a +\$b	Addition	Sum of \$a and \$b
\$a - \$b	Subtraction	\$b minus \$a
\$a * \$b	Multiplication	
\$a / \$b	Division	
\$a % \$b	Modulo	Rest of integer division \$a by \$b

### Example 3-3: Simple arithmetic

Application	url: <a href="#">/guides/php/examples/simple/simple-calcul.php</a>
Source	url: <a href="#">/guides/php/examples/simple/simple-calcul.php</a>
To copy	url: <a href="#">/guides/php/examples/simple/simple-calcul.text</a>

```
$leisure_satisfaction = 5;
$work_satisfaction = 7;
$family_satisfaction = 8;
$index = ($leisure_satisfaction + $work_satisfaction + $family_satisfaction)
        / 3 ;
echo "<p align=center> Satisfaction Index = $index <b>" ;
```

assignment + addition in one step:

```
// sets $a to 8, as if we had said: $a = $a + 5;
$a += 5;
```

## B. Operators for strings

### concatenation of strings

Use the `".` operator

example:

```
$a = "Hello ";
$b = $a . "World!"; // now $b = "Hello World!"
```

- Note: There are dozens of string manipulation functions in PHP !!

assignment + concatenation in one step

```
$b = "Hello ";
// sets $b to "Hello There!", just like $b = $b . "There!";
$b .= "There!" ;
```

## C. Logical operators

example	name	result
<code>\$a and \$b</code>	"and"	result true, if \$a et \$b are true
<code>\$a &amp;&amp; \$b</code>	"and"	"
<code>\$a or \$b</code>	"or"	result true, if \$a or \$b or both are true
<code>\$a    \$b</code>	"or"	"
<code>\$a xor \$b</code>	Or exclusive	result true, if \$a or \$b are true, but not both
<code>! \$a</code>	"not"	result true, if \$a is false(

## D. comparison

example	name	result
<code>\$a == \$b</code>	equal	True if \$a is equal to \$b.
<code>\$a===\$b</code>	identical	True if \$a==\$b and same type (php 4.x)
<code>\$a != \$b</code>	different	True if \$a is not equal to \$b.
<code>\$a!==\$b</code>	not identical	True if \$a!=\$b or not same type (php4.x)
<code>\$a &lt; \$b</code>	inferior	True if \$a is strictly less than \$b.
<code>\$a &gt; \$b</code>	superior	True if \$a is strictly greater than \$b.
<code>\$a &lt;= \$b</code>	inferior or equal	True if \$a is less than or equal to \$b.
<code>\$a &gt;= \$b</code>	superior or equal	True if \$a is greater than or equal to \$b

- You can use parenthesis if you like to group operators !

### Example 3-4: Simple comparisons

*url: /guides/php/examples/simple/simple-compare.php*

*url: /guides/php/examples/simple/simple-compare.php5*

- Note: in PHP each number equal or small than 0 is FALSE, each superior is TRUE

```
$a = "Migros";
$b = "Coop";
$result = $a==$b;
$result2 = $a > $b;
$result3 = $result==TRUE;
echo "Result One = $result. Result TWO = $result2. Result THREE = $result3.";
```

## 3.6 Selection (Conditions and tests)

Principle (several typical situations):

- If a condition is true then do ...
- If a condition is true then do ... , else do .....
- If a condition is true then do ... , else if an other condition is true do ... , else .....

### "IF" (several variants)

Syntax: `if (expr) statements`

Syntax: `if (expr) statements else statements`

Syntax: `if (expr) statements elseif (expr) statements else ...`

Syntax: `if (expr) statements elseif (expr) statements [ elseif (expr)  
... ]`

- *expr* = Expression must return TRUE or FALSE
- *statements* = simple instructions or a block or instructions
  - simple: `$a = 10;`
  - block: `{ $a =12; echo "salut"; ..... }`
- Execution model
  - If expression = TRUE then execute statement(s)
  - If expression = FALSE then go to the next clause

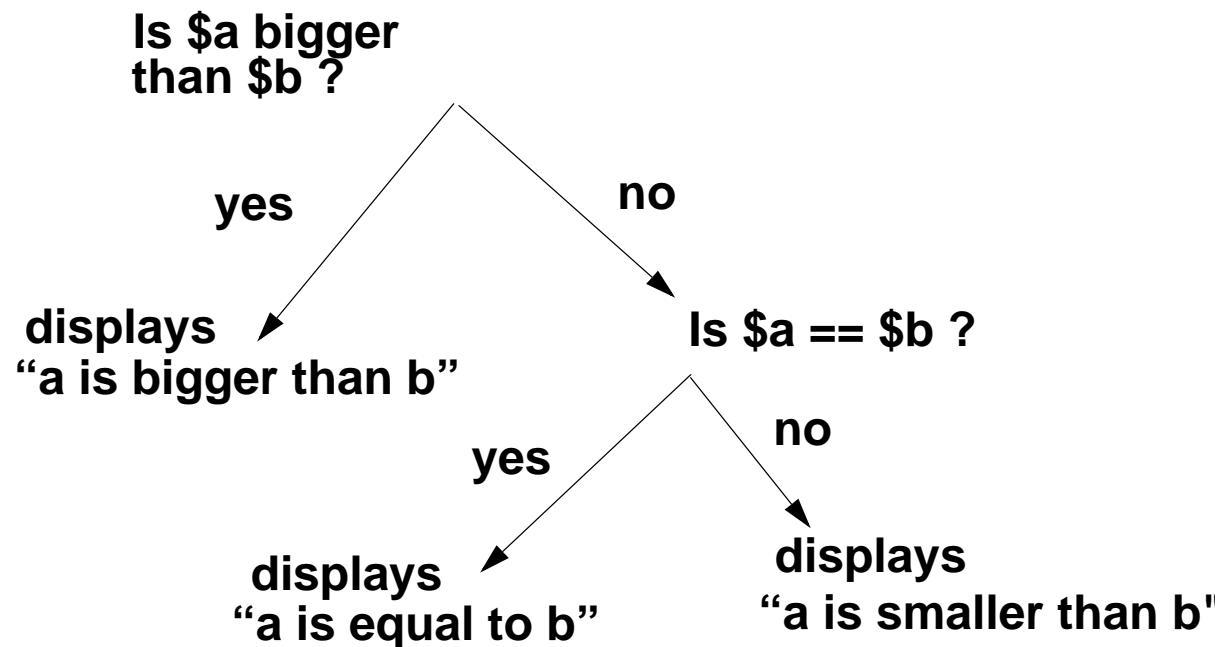
## Example 3-5: Simple "if" (comparison)

url: </quides/php/examples/simple/simple-if.php>

url: </quides/php/examples/simple/simple-if.php> (source)

- Compares two numbers: \$a and \$b, and displays a message.
- Here is a decision tree.

Figure 3-2: Simple decision tree



```
<?php  
  
$a = 10; $b = 11;  
print "a was $a, b was $b. "  
if ($a > $b) {  
    print "a is bigger than b";  
} elseif ($a == $b) {  
    print "a is equal to b";  
} else {  
    print "==> a is smaller than b.";  
}  
  
?>
```

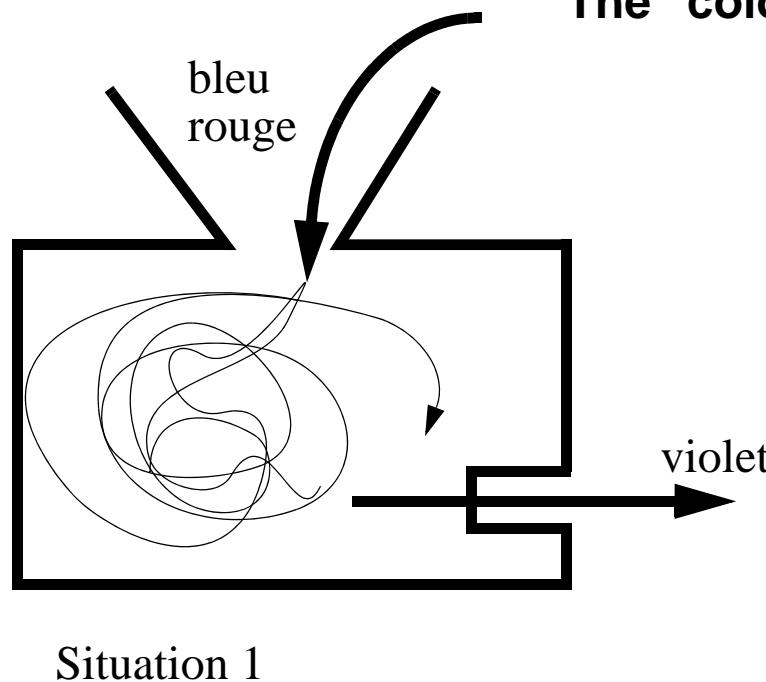
See also the following:

- [switch](#)
- [foreach](#)
- [do ... while](#)
- [break and continue](#)

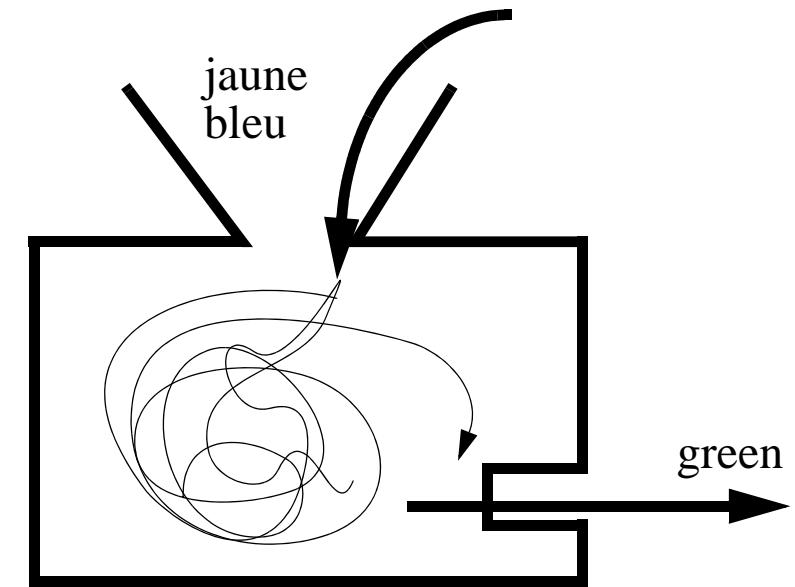
## 3.7 PHP functions

- Like all programming languages PHP allows to define procedures/functions.
- A function is a mini program that has a name and that you can "call" (invoke).  
"Hey, take these informations, do something and (maybe) return the result"
- Usually, you find function definition in the beginning of program (or within include files)

The "color-mix"



Situation 1



Situation 2

## Example 3-6: Color mixing for paint

url: <http://tecfa.unige.ch/guides/php/examples/simple/> (files color-mix.\*)

```
function color_mix($color1,$color2) {  
    $result= "unknown";  
  
    if ($color1 == "bleu" and $color2 == "rouge") {  
        $result = "violet";    }  
    elseif ($color1 == "jaune" and $color2 == "bleu") {  
        $result = "green";    }  
    elseif ($color1 == "noire" and $color2 == "blanc") {  
        $result = "gris";    }  
    else {  
        $result = "orange";    }  
    return $result;  
}  
  
// Two calls to this function, results saved in variables  
  
$situation1 = color_mix ("bleu", "rouge") ;  
  
$situation2 = color_mix ("jaune", "bleu") ;  
  
// Print  
  
echo "Bleu et rouge donne $situation1 <br>" ;  
echo "Jaune et bleu donne $situation2" ;
```

## Example 3-7: HTML generation with functions

url: </guides/php/examples/simple/function-demo.php>

url: </guides/php/examples/simple/function-demo.php>

```
<?php

// html formats a data element
function pretty_print ($output) {
    separator ();
    echo "<p align='center'> <strong>ELEMENT:</strong> $output </p>" ;
}
// outputs a separator
function separator () {
    echo "<hr size=4 width=70%>" ;
}
// data we have
$el1 = "Un arbre jaune";
$el2 = "Ein gelber Hund";
$el3 = "A yellow sky";
// dump the data
pretty_print($el1);
pretty_print($el2);
pretty_print($el3);

separator ();
echo "<hr>";
?>
```

## 3.8 Loops (iterations)

### The "for loop" syntax

FOR (expr1; expr2; expr3) statement

- expr1 is evaluated at start
- expr2 is evaluated at start of each loop,  
if result = TRUE the loop will continue, else it will stop
- expr3 is evaluated at the end of each loop,
- statement is executed for each loop.

### Example 3-8: Love generation

url: see: [/guides/php/examples/html-generate/love.php](#)

url: see: [/guides/php/examples/html-generate/love.php](#)

```
for ($i=1; $i<=10; $i++) {  
    print "I love you so ! ";
```

result:

love you so ! I love you so ! .....

```
echo "Je t'aime plus que toi.<br>  
for ($i=2; $i<=10; $i++) {  
    echo "Non, je t'aime $i fois plus que toi ! ";
```

result:

Je t'aime plus que moi.

Non, je t'aime 2 fois plus que moi ! Non, je t'aime 3 fois plus que moi ! Non,  
je t'aime 4 fois plus que moi ! Non, je t'aime 5 fois plus que moi ! Non, je  
t'aime 6 .....

## Other PHP elements:

- `$i` is used as so-called iteration variable. At start `$i = 1` or `2`.
- `echo`
- `print` works like `print`.

### Example 3-9: Generation of html tables

url: see: [/guides/php/examples/html-generate/love.php](#)

url: see: [/guides/php/examples/html-generate/love.php5](#)

url: see: [/guides/php/examples/html-generate/love.text](#)

```
$love_list = array ("a lot", "a bit", "somewhat", "à mourir", "forever", "until
notice", "more than I love my dog");
<table border align="center">
<?
// define a function to generate a table
function build_table($list) {
    for ($i=0; $i < sizeof($list); $i++) {
        $love_text = $list[$i];
        echo "<tr> <td> ... I love you</td> <td>$love_text</td>";
    }
}
// call the function, generate the table
build_table($love_list);
?>
</table>
```

#### Note:

- PHP is used with the HTML `<table>` element
- The `build_table` function is called with an array
- There exist more looping constructs in PHP (like while or for-each) !

## 4. Practical advice

### 4.1 Debugging

- Look at the generated HTML code "View Source")
- Insert this in you PHP file (will give you lots of information !)

```
phpinfo();
```

- Insert print statements!

```
echo "DEBUG: \$var = $var";  
echo "TEST: var = $var";
```

- Raise "error reporting" to its maximum !!!

Insert this on top:

```
error_reporting(E_ALL);
```

### 4.2 Portals

- Warning: NEVER insert blank lines at start or end of a file !!
- Most files should stop like this (no line feed !!) ?>
- ... because PHP starts producing HMTL headers as soon as it sees a little blank space before or after php code <?php .... ?>

# 5. HTML forms processing with PHP

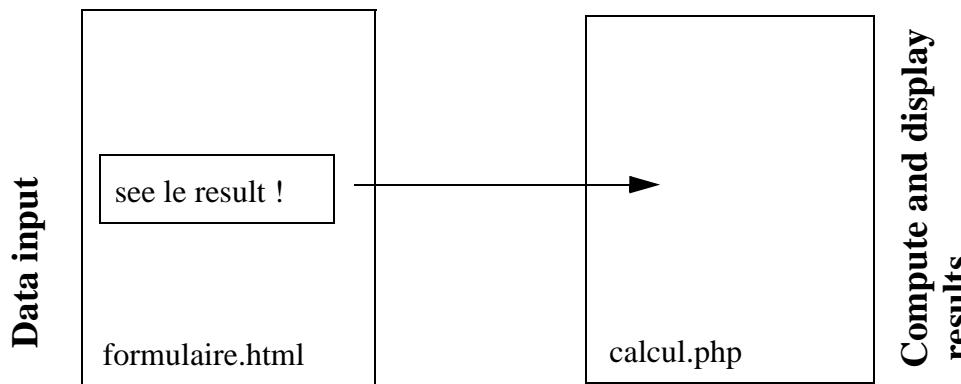
## 5.1 Forms processing with PHP I (Calcul)

### Example 5-1: Simple quiz and POST to a php file

url: See: </guides/php/examples/simple-calculate/formulaire.html>

url: Source: </guides/php/examples/simple-calculate/formulaire.text>

- This example shows:
  - how to treat an HTML form
  - how to compute and display a result.



## Part of the HTML form:

```
<form action="calcul.php" method="post">

Quelles sont vos connaissances de HTML ?
<input type="radio" name="choice" value="1" checked>faibles
<input type="radio" name="choice" value="2">moyennes
<input type="radio" name="choice" value="3">bonnes
<br>
Indiquez votre expertise en programmation:
<input type="radio" name="choice2" value="1" checked>absente
<input type="radio" name="choice2" value="2">moyenne
<input type="radio" name="choice2" value="3">bonne
<P>
<input type="submit" value="Voir le résultat!">
</form>
```

The diagram illustrates the mapping of HTML form fields to PHP variables. It shows a screenshot of a web page with two sets of radio buttons and a submit button. Arrows point from the labels to the corresponding HTML code in the previous block.

Quelles sont vos connaissances de HTML ?  faibles  moyennes  bonnes

Indiquez votre expertise en programmation:  absente  moyenne  bonne

Voir le résultat!

NAME="choice"

NAME="choice2"

Send to the PHP page.

## A. Retrieve values of an HTML form

- Data from a form is stored by the server in so-called super global variables
- Use `$_POST` to deal with POST variables
  - POST: values are handed over attached (and not visible) to the server
- Use `$_GET` for GET variables
  - GET: values are handed over in the URL string (user can see these)
- You can use the "name" attribute of the form to retrieve values

In our example, we use `$_POST`:

```
$choice = $_POST[ 'choice' ];  
$choice2 = $_POST[ 'choice2' ];  
• In our example, we will use two PHP variables:  
$choice and $choice2
```

## B. Computing and display of results

*url: see: [/guides/php/examples/simple-calculate/calcul.php](#)*

- we add result of the two values and compute a result with an if clause.

```
<?php

// Get values from the form
$choice = $_POST['choice'];
$choice2 = $_POST['choice2'];

// Compute score
$score = $choice + $choice2;

// Compute message as function of result
echo "<h3>Votre score est de " . $score . "</h3>";
if ($score < 3) {
    echo "<p>Vous &ecirc;tes un d&acute;butant</p>";
} elseif ($score < 5) {
    echo "<p>Vous avez un niveau moyen</p>";
} else {
    echo "<p>Vous &ecirc;tes un expert !</p>";
}
?>
```

## C. Inhibit direct access to PHP (without data)

- (1) if (isset(\$\_POST['choice'])) then { ..... } else { echo "sorry ....."; }
- (2) Alternative: if (!isset(\$\_POST['choice'])) {echo "sorry"; exit; }

## 5.2 Forms processing with PHP II

### Example 5-2: Checkboxes with PHP - arrays

url: [/guides/php/examples/simple-calculate/formulaire4.text](#)

url: [/guides/php/examples/simple-calculate/formulaire4.html](#)

#### Part of the HTML code:

```
<form action="calcul4.php" method=post>
Quels sont vos couleurs préférées?
<br>
<input type="checkbox" name="choice[]" value="Red">Red
<table bgcolor="red" width="50"><tr><td>&ampnbsp</td></tr></table>

<input type="checkbox" name="choice[]" value="Blue">Blue
<table bgcolor="blue" width="50"><tr><td>&ampnbsp</td></tr></table>

<input type="checkbox" name="choice[]" value="Green">Green
<table bgcolor="green" width="50"><tr><td>&ampnbsp</td></tr></table>
.....
<input type="checkbox" name="choice[]" value="Black">Black
<table bgcolor="black" width="50"><tr><td>&ampnbsp</td></tr></table>

<input type="submit" value="Voir le result!">
</form>
```

- Remember the syntax to put all values into an array: "choice[ ]"

## PHP code:

```
<?php
$choice = $_POST['choice'];

echo( "<h3>Vos couleurs préférées sont </h3>" );

for ($i=0;$i<sizeof($choice);$i++) {
    if (isset($choice[$i])) {
        echo( "$choice[$i] - " );
    }
}
?>
```

## 5.3 All in one solution ?

- You can put both the form and the processing code in a single page
- In this case, test if the file is called with data from a form, see \$process below

```
<?
if (!isset($_POST['process'])) {
?>
//... lets display the form)

<FORM METHOD="POST" ACTION=" echo $PHP_SELF ?&gt;"&gt;
.....
&lt;/FORM&gt;

&lt;?
} else {
//... we got data, so let's process
}
?&gt;</pre
```

## 5.4 Polishing: Test if we have all the POST/GET variables

### 2 methods to test what we have in \$\_POST or \$\_GET

#### 1. "array\_key\_exists()"

```
if (array_key_exists('first', $_POST)) { .... do something ... };
```

#### 2. "isset()" to see if a variable exists:

```
if (isset($_POST['first'])) { .... do .... }
```

- The difference is that

- array\_key\_exists returns TRUE if value is NULL
- isset returns FALSE if value is NULL.

**ATTENTION, to test <input type="text"> you also may want to test if there is an empty string.**

#### 3. “empty()”

- to decide if user filled in a text field

```
if (empty ($input) ) { ... complain ... } else { ... do ... }
```

- empty() returns TRUE if a value is: "", 0, "0", NULL, FALSE, array(), ....

## 5.5 Session management

- PHP has session support (can keep variables over a whole user session).
- Each visitor gets an identifier (a "sessions id"). It is stored in a cookie (in the www client) or within the URL.
- This information is available in super global: `$_SESSION`

### Example 5-3: Restrict repetitive access to a page

url: <http://tecfa.unige.ch/guides/php/examples/sessions/>

```
session_start();
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 0;
} else {
    $_SESSION['count']++;
}
if ($_SESSION['count'] > 2) {
echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';
    echo "<html> <body>";
    echo "Sorry it's over you can't do it twice";
    echo "</body> </html>";
    exit;
}
// .... continue code with access time = 1 and 2
```

# 6. On-line surveys and file-based storage

## Example 6-1: Survey

url: see: <http://tecfa.unige.ch/guides/php/examples/form-file-demo/>

- new-entry.php contains the form and code
- dump\_results.php shows file contents

### A. The HTML form

- This time we use PHP to generate the HTML code

```
$scales = array( "food" , "work" , "love" , "leisure" , "sports" );
function scale ( $thing ) {
    echo "<TR> <TD align=right>Importance of <STRONG>$thing</STRONG>:</TD>" ;
    echo "<TD><select name=$thing>" ;
    echo "<option value=1>1 - totally unimportant" ;
    echo "<option value=2>2 - not important" ;
    echo "<option value=3 selected>3 - rather not important" ;
    echo "<option value=4>4 - slightly important" ;
    echo "<option value=5>5 - rather important" ;
    echo "<option value=6>6 - very important" ;
    echo "</select>" ;
    echo "</TD></TR>" ;
}
```

```
function dump_scales ()  {
    global $scales;
    reset($scales);
    do {
        $scale = scale(current($scales));
        echo "$scale\n";
    }
    while (next($scales));
} ?>
<form> <table>
.....
dump_scales();
.....
</table> </form>
Ecrire dans un fichier
// check existance of file (or try to create it)
    // a better alternative to touch() would be is_file, is_writable and so on.
    $try = touch($file_name);
if (!$try) {
    echo "<p>Sorry I can't open a file, something is wrong";
    exit;
}

// this is the stuff we get from the form, we insert it into an array
$input = array ($login, $password, $fullname, $url, $food, $work, $love,
$leisure, $sports);

// so we can make a big string with tabs between the elements
// note that we add a \n (line break) to the end of the string.
$output_line = implode ("")."\n";
```

```
// Now open the file (get a file pointer)
// We will append to it and therefore use the "a" option
$output_stream = fopen($file_name, "a");
// and dump the string into the file
$result = fputs ($output_stream, $output_line);

// give feedback
if ($result) {
    echo "<p>Your data have successfully been registered." ;
}
else {
    echo "<p>Too bad, the db did not want your data." ;
}
// close the file pointer
fclose($output_stream);
?>

<?
// EXIT here ... we don't want to see the form again. If you do, kill the exit
exit;
}
?>
```

## Remember

fopen (<file name>, "a")

- to open a file and then append.

fputs(<handle>, "string")

- to write to a file

- WARNING: This will attract spammers !!

## B. Dump contents of a file

.... we just insert it a <pre> with an “include”

```
<BODY>
    <H1>PHP/MySQL Demo – Dump Database Contents</H1>

<?
/* Daniel.Schneider@tecfa.unige.ch
Will dump the contents of the results file
*/
?>

    <strong>Results registered so far:</strong>
<pre>
<? readfile( "results/result.text" ); ?>
</pre>
.....
</BODY>
```

### Important:

- Use "readfile", and not "include" or "require", **else you will get hacked !!**

## 7. Other formats than HTML

Principle: in the FIRST line of your program you have to define the content-type:

- example: Header( "Content-type: image/gif" );

### Example XML

```
Header( "Content-type: text/xml");
```

### Example SVG

```
Header( "Content-type: image/svg+xml");
```

### Example RDF

```
Header( "Content-type: application/rdf+xml");
```

### Example 7-1: Generate some simple XML

url: <http://tecfa.unige.ch/guides/php/examples/simple/simple-calcul-xml.php>

url: <http://tecfa.unige.ch/guides/php/examples/simple/simple-calcul-xml.phps>

url: <http://tecfa.unige.ch/guides/php/examples/simple/simple-calcul-xml.css>

```
<?php  
header("Content-type: text/xml");  
print('<?xml version="1.0" encoding="iso-8859-1"?>' . "\n");  
print('<?xml-stylesheet href="simple-calcul-xml.css" type="text/css" ?>');  
  
$leisure_satisfaction = 5; $work_satisfaction = 7; $family_satisfaction = 8;  
$index = ($leisure_satisfaction + $work_satisfaction + $family_satisfaction) / 3 ;  
echo "<resultat> Satisfaction Index = $index </resultat>";  
?>
```

# 8. PHP and MySQL

- PHP can interact with most RDMS.
- SQL queries are built and sent via PHP.
- Results from MySQL are translated into HTML.

## 8.1 A complete documented example:

url: <http://tecfa.unige.ch/guides/php/examples/mysql-demo/main.html>

url: Details: <http://tecfa.unige.ch/guides/php/examples/mysql-demo/>

- Show all records ([dump\\_results.php](#))
- Add a record ([new-entry.php](#)s and [insert-entry.php](#)s)
- Edit a record ([edit-entry.php](#)s and [replace-entry.php](#)s)
- Kill a record ([delete-entry.php](#)s)

### To add records we need to scripts:

- new-entry.php produces the HTML form
- insert-entry.php inserts the values from the forms and gives some feedback.

### Same principle for editing:

- edit-entry.php produces the form
- replace-entry.php replaces old values by new ones.

## 8.2 Functions PHP - MySQL

### A. Connect to a database

- First, you have to connect to a server, you need:
  - the host name of the database (usually "localhost")
  - the database user name and password

Syntax: `mysql_pconnect( host, username, password );`

```
$link = mysql_pconnect( "localhost", "clavel", "secret" );
```

- `$link` will contain the link to the open connection (or "FALSE" if something went wrong).

### B. Select a database

- A MySQL server hosts several database. You need to tell which one you will use.

Syntax: `mysql_select_db( dbname, [linkID] );`

```
mysql_select_db( "demo" );
```

### Execute a SQL statement

- `mysql_query` will send a SQL statement
- This call will return an identifier for the result

```
$result = mysql_query( "SELECT * FROM demo1" );
```

`$result` is "boolean" and contains:

- 1 (TRUE) if ok
- 0 (FALSE) if there was a problem.

## C. Processing of results

- There exist several ways of dealing with the result. Here we show two

### ***Deal with the result row by row***

Syntax: `mysql_fetch_row`

`mysql_fetch_row(identificateur)`

example:

```
$result = mysql_query("SELECT * FROM demo1")  
$row = mysql_fetch_row($result);
```

- returns a line of the result as an array (a list of values).
- If you call it again, it will return the next line

### ***Number of fields returned***

Syntax: `mysql_num_fields`

`mysql_num_fields(identificateur)`

```
$nb_champs = mysql_num_fields($result);
```

## D. Error handling

- For most mysql operations, there can be errors and PHP does remember these:
- PHP functions:

Syntax: `mysql_errno()`

`mysql_errno(identificateur)`

- returns a number code for the last mySQL operation or zero (0) if no error was found.

Syntax: `mysql_error()`

`mysql_error(identificateur)`

- returns a string with some information.

### Example 8-1: simple error handling:

```
<?php
    mysql_pconnect( "nohost" , "basuser" , "wrongpass" );
    echo mysql_errno()." : ".mysql_error()." <BR>

    mysql_select_db( "nonexistentdb" );
    echo mysql_errno()." : ".mysql_error()." <BR>

    $conn = mysql_query( "SELECT * FROM nonexistenttable" );
    echo mysql_errno()." : ".mysql_error()." <BR>

?>
```

## Example 8-2: Generation of an HTML table with the results

url: [http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump\\_results\\_demo.php](http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo.php)  
url: [/guides/php/examples/mysql-demo/dump\\_results\\_demo.php](/guides/php/examples/mysql-demo/dump_results_demo.php)  
url: [/guides/php/examples/mysql-demo/dump\\_results\\_demo.source](/guides/php/examples/mysql-demo/dump_results_demo.source)

```
<?
mysql_pconnect("localhost", "nobody", "") ;
mysql_select_db("demo") ;
$result = mysql_query("SELECT * FROM demo1") ;
?>


||
||
||


```

## E. Processing a result and using field names

### *mysql\_num\_rows*

Syntax: `mysql_num_rows(identificateur);`

example:

```
$nb_enregistrements = mysql_num_rows($result);
```

- Returns number of returned records in \$result

### *mysql\_result*

Syntax: `mysql_result(identificateur, index, champ);`

example:

```
$nom = mysql_result($result, 0, 'fullname');
```

- *index* holds the record number. Indexation starts at 0 ! (zero).
- *champ* is the name of the field.
- see next slide ...

### Example 8-3: HTML generation using field names

url: [http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump\\_results\\_demo2.php](http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo2.php)

url: [/guides/php/examples/mysql-demo/dump\\_results\\_demo2.php](/guides/php/examples/mysql-demo/dump_results_demo2.php)

url: [/guides/php/examples/mysql-demo/dump\\_results\\_demo2.source](/guides/php/examples/mysql-demo/dump_results_demo2.source)

```
<?php  
mysql_pconnect( "localhost", "nobody", "" ) or die( "Unable to connect to  
SQL server");
```

```
mysql_select_db( "demo" ) or die ( "Unable to select database" );
$result = mysql_query( "select * from demol" );
?>
<table border="1">
<?php
$i = 0;
while ($i < mysql_num_rows($result)) {
    echo "<tr>";
    echo "<td>";
    echo mysql_result($result,$i,'id');
    echo "</td>";
    echo "<td>";
    echo mysql_result($result,$i,'fullname');
    echo "</td>";
    echo "<td>";
    echo mysql_result($result,$i,'love');
    echo "</td>";
    echo "<td>";
    echo mysql_result($result,$i,'sports');
    echo "</td>";
    echo "</tr>";
    $i++;
}
echo "</table>";
?>
```

# 9. Workthrough example: Guestbook.

## 9.1 Example files

- Create a `comments` directory in your web server.
- Copy files: `comments-list.source`, `comments-insert.source`, `comments.html`

*url: <http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/>*

**Rename** `.source` to `.php` !!!

## 9.2 Structure of the comments table

*url: <http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/comments-table.txt>*

```
create table comments (
    id int(10) default '0' not null auto_increment,
    nom char(20) default '' not null,
    prenom char(20) default '',
    email char(50) default '',
    computer char(10),
    browser char(10),
    version char(10),
    comments char(200),
    primary key (id),
    key nom (nom)
);
```

## 9.3 Task

- **comments.html** displays the form and sends values to **comments-insert.php**.
- **comments-insert.php** writes data to mySQL

Server : localhost

database name : .....

table : comments

database user name: .....

database user password : .....

- **comments-list.php** will display all comments in a table.

- the HTML is functional, but you will have to fix the 2 php files (comments are in french sorry)

Start with comments-insert.php then file comments-list.php

**Solution:**

url: <http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/comments.html>