

# PHP and MySQL

## Introduction

### General

HTML (Hypertext Markup Language), as described in the course HTML, makes it possible to create documents independent of any platform, and thus well adapted for exchanging informations in a heterogeneous environment like the Web. It is a well tested method of creating Web pages. Why is it necessary to program in PHP? Why would one need "dynamic" Web pages?

### Limits of HTML

Majority of the Web sites are carried out using HTML and offer static contents such as publications, articles etc ... The pages of these sites consist of simple text decorated of some images and hypertextes links leading towards others pages. Javascript allows to addition of more sophisticated effects if necessary.

However internet and the intranets are used more and more for applications, that include databases. The sites and their applications are dynamic, because the contents are modified according to the implied data and actions from the user. You can create powerful applications acting in an interactive way with a database, and generating dynamic contents by executing PHP on a server.

### What PHP can do and not HTML

PHP :

- facilitates the modification of the contents of a web page, intervening on the data of a database and not directly in the HTML code.
- creates personalized pages in order to only display what interests a particular user.
- displays and refreshes database included in the web page, can handle the data, for example by carrying out a sorting or by displaying part of the database.
- creates pages which carry out a cycle among various images
- obtains an answer from the user, then returns the information according to this answer.
- can still do well other things ...

This course is not exhaustif. It gives basics concepts to any programmer who wish to take his first steps in PHP and MySQL. That being, a clear understanding, at least intuitive, of the basic data-processing concepts (what a computer network, a file, a text editor, a programming language, a compilation etc.) is preferable. We suppose moreover that you have at least one of the following environments:

- a computer, connected to internet through an ISP proposing the MySQL/PHP environment.

·a computer under Linux or Windows, laying out in room an Apache/MySql/PHP complete environment.

## **A. PHP**

### **I. Bases of PHP**

#### **a. Definition**

PHP (Personal Tools Home Page - Preprocessor Hypertext) has been created by Rasmus Lerdorf in the year 1994, for his personal needs. The provision of the language on Internet will make pass its development from one person to a kernel of programmers composed by Andi Gutmans, Zeev Suraski, Stig Bakken, Shane Caraveo Jim Winstead, and of course Rasmus Lerdorf. It is a programming language very close to C, from which it takes again a great part of syntax, and intended to be integrated in HTML pages. After various evolutions, PHP is in its 5<sup>th</sup> version.

#### **b. PHP file and its structure**

PHP file is save in an ASCII format, so that you can write a PHP page with allmost all text editor: notepad, wordpad, VI, emacs ... PHP code is a script registered in a HTML page.

## II. Programming in PHP

### 1. syntax

#### a. general informations

It is very simple. Any PHP code have to be included in a tag `<?php ...?>`. "short" tags such as `<?>` are sometimes accepted, but are not recommended, because they risk to be in conflict with other languages such as XML.

As in C, the instruction separator is the semicolon "; " .

#### b. comments

If the comment can be written on one line, it can start with the signs `"/"` or `"#"`.

Else, if the comment have to be written on several lines, it should be placed between the signs `"/**"` and `"*/"`.

Of course, one can mix the two styles of comments in the same script.

#### c. variables and data

##### • Variables

A variable name always begins with one "\$", followed by at least one nonnumerical character (the "\_" is authorized), then by any combination of figures and characters. In PHP, it is not necessary to declare the variables nor to define the type of the data that they'll contain before being able to use them; PHP automatically creates a variable as soon as new symbol preceeded by "\$" appears in the script. The variable type can change if we modifies its contents.

PHP distinguishes the uppercases letters from the lowercase letters in the name of the variables; thus "\$mavariabLe" and "maVariable" indicate two different variables; on the other hand the names of functions are insensitive to the case.

##### • Data

The basic types are:

**Integer:** use 4 bytes of memory and is used to represent an ordinary figure deprived of decimals.

**Double:** also known under the name of real number or for floating point, is used to represent decimals alues with exponent.

**String:** a string represents nonnumerical values, like letters, punctuation marks but also numerical characters.

Exercise 1 : my first program in PHP

```
<HTML>
< ?php
    echo « text generated by PHP » ;
?>
</HTML>
```

Exercise 2

```
<HTML>
<FORM>
    Please enter your name here: <BR>
    <INPUT TYPE = TEXT NAME = user> <BR><BR>
    <INPUT TYPE = SUBMIT VALUE = « submit »>
</FORM>
```

```
<BR><BR>
Your entered :
```

```
< ?php
    echo ($user) ;
?>
</HTML>
```

## 2. operators

### a. les arithmetics operators

- « + » : addition
- « - » : subtraction
- « \* » : multiplication
- « / » : division
- « % » : modulo

## b. comparison operators

- « == » : equal to
- « < » : lower than
- « > » : greater than
- « <= » : lower or equal to
- « >= » : greater of equal to
- « != » : different from
- « <> » : different from

## c. les logical operators

- « && » : and
- « || » : or
- « and » : and
- « or » : or
- « xor » : exclusive or
- « ! » : no

## 3. control structures

### a. conditionals instructions

- if instruction

```
if (condition)
    {instruction1;}
[else
    {instruction2;}]
```

- switch instruction

```
switch (variable) {  
    case value1 ;  
        instruction1 ;  
        break ;  
    case value2 ;  
        instruction2 ;  
        break ;  
  
    case value_n ;  
        instruction_n ;  
        break ;  
    default :  
        instruction ;  
}
```

## **b. loops**

- for loop

```
for (expr1 ; expr2 ; expr3) :  
    // instructions  
endfor ;
```

- while loop

```
while (condition) {  
    // instructions  
}
```

- do ... while loop

```
do {  
    instructions;  
} while (condition);
```

#### 4. Incorporate a file in a PHP page

insert the following code: `require ("NameOfTheFileToInsert.extensionOfTheFile")`.

#### 5. functions

The functions allows the written of modular and reusable code. A function can receive arguments (variables transmitted to the function to be used inside it) and return a value. A variable which is inside a function has a local range, which mean that it exists only inside the function and doesn't interfere with any variable located outside the function, even though it would bear the same name. A function can reach a global variable with the help of the instruction *global*. The local variables located inside a function are re-initialized at each call of the function, unless the *static* instruction is used: in this case, it will preserve the value it had before the last call.

The functions are declared with the *function* instruction.

##### a. declaration syntax

```
function function_name (parameters) {  
    function body  
}
```

E. g.

```
// function declaration et definition
```

```
function calculCube ($number) {  
    return $ number * $ number * $ number ; // return $number exponent 3  
}
```

```
// call of the function calculCube
```

```
echo (cube (9)) ;      // must display 729 on the screen
```

## B. MySql

### 1. databases units

PHP alone, i.e. taken in an isolated way, does not constitute a miracle solution to the lack of dynamism in HTML pages. The variables of this language can retain the data which are stored there, only during program execution. How to introduce more flexibility in displaying contents, regarding the user needs and tastes? How can we display only the informations that interested the user, he has chosen to receive, like news or announcements? The storage of structured data becomes necessary when these needs must be satisfied.

To take advantages of a dynamic and powerful site, the data must be stored, and grouped if necessary in tables, which will constitute a database. There are several stable and tested systems comprising powerful techniques allowing programmers to create applications meeting their needs: Databases Management Systems (DBMS), with the number of which Access, Oracle, SQL Server, Sybase, InterBase, MySQL. We will be interested in DBMS MySQL this paragraph, because the solution provided by the couple (MySQL, PHP) is of an increasing popularity for the management of dynamic Web sites; then because these two tools constitute a complete solution to the problems mentioned above, while remaining very simple to use, and finally because the tandem formed by MySQL and PHP can easily associate to the Apache server, and this, as well as under Linux and Windows.

### 2. Instructions of data definition

Also called requests, these instructions of the MySQL language make it possible to modify the diagram of the database by creating or modifying the objects in this one.

#### ➤ **CREATE**

This instruction allows to create a database or a table inside an existing database.. The syntax of database creation is simple, while of the creation of a table is more complex because it is necessary to include the description of the fields.



- Creation of a database:

*CREATE DATABASE dataBase\_Name*

- Creation of a table

*CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table\_Name (CreationDefinition, ...) [optionsTable]  
[optionSelect]*

*CreationDefinition:*

*fieldName type [NOT NULL | NULL] [DEFAULT literal] [AUTO\_INCREMENT]  
[PRIMARY KEY] [definitionReference]*  
or *PRIMARY KEY (fieldName, ...)*  
or *KEY [nomIndex] (fieldName, ...)*  
or *INDEX [nomIndex] (fieldName, ...)*  
or *UNIQUE [INDEX] [nomIndex] (fieldName, ...)*  
or *[CONSTRAINT constraint] FOREIGN KEY nomIndex (fieldName, ...) [reference]*  
or *CHECK (expression)*

*optionsTable :*

*TYPE = {ISAM | MYISAM | HEAP}*  
or *AUTO\_INCREMENT = #*  
or *AVG\_ROW\_LENGTH = #*  
or *CHECKSUM = {0 | 1}*  
or *COMMENT = « comments »*  
or *MAX\_ROWS = #*  
or *MIN\_ROWS = #*  
or *PACK\_KEYS = {0 | 1}*  
or *PASSWORD = « passWord »*  
or *DELAY\_KEY\_WRITE = {0 | 1}*

*optionSelect:*

*[IGNORE | REPLACE] SELECT ... (request SQL)*

*reference:*

*REFERENCES TableName [(fieldName, ...)]*

*[MATCH FULL | MATCH PARTIAL]*  
*[ON DELETE optionRef]*  
*[ON UPDATE optionRef]*

*optionRef*

*RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT*

The instruction *CREATE table* creates a table in the current database. The option *TEMPORARY* indicates that the table is created for the current connection only. The option *optionTable* makes it possible to choose the table type. The index name is optional.

example of *CREATE* instruction in the creation of database and table:

*CREATE DATABASE university*

Let us create inside *university* database, the *student* table:

```
CREATE TABLE student (IDnumb VARCHAR (40) NOT NULL,  
                        lastname VARCHAR (30) NOT NULL,  
                        firstname VARCHAR (30) NOT NULL,  
                        sex CHAR (1),  
                        birthDate INTEGER,  
                        school VARCHAR (30),  
                        option VARCHAR (30),  
                        remark TEXT,  
                        PRIMARY KEY (IDnumb));
```

## ➤ **DELETE**

syntax :

```
DELETE [LOW_PRIORITY] FROM TableName  
      [WHERE clauseWhere] [LIMIT nbLines]
```

This instruction deletes all lines verifying the *WHERE* clause. The option *LOW\_PRIORITY* shows to MySQL that the deletions are lower priorities than the other current request accessing to the table.

E. g. of use *DELETE* instruction:

```
DELETE FROM TABLE student ;
```

This request delete all the *student table* records

```
DELETE FROM TABLE student WHERE school = 'health sciences ' AND  
option = 'medicine' ;
```

This request deletes from the *student* table all students registered in the option « medicine » of the « health science » school.

### ➤ **SELECT**

```
SELECT [STRAIGHT_JOIN] [SQL_SMALL_RESULT] [SQL_BIG_RESULT]  
[HIGH_PRIORITY] [DISTINCT | DISTINCTROW | ALL]  
fieldsList  
[INTO {OUTFILE | DUMPFILE} 'fileName' optionExport]  
[FROM clauseFROM  
[WHERE clauseWHERE]  
[GROUP BY nomAttribut, ...]  
[HAVING clauseWHERE]  
[ORDER BY {entire | fieldName | formule} [ASC | DESC], ...]  
[LIMIT [debut,] nbLines]  
[PROCEDURES]]
```

*clauseFROM:*

```
TableName, TableName  
or TableName [CROSS] JOIN TableName  
or TableName INNER JOIN TableName  
or TableName STRAIGHT_JOIN TableName  
or TableName LEFT [OUTER] JOIN TableName ON expression  
or TableName LEFT [OUTER] JOIN TableName USING (fieldsList)  
or TableName NATURAL LEFT [OUTER] JOIN TableName  
or TableName LEFT OUTER JOIN TableName ON expression
```

This instruction extracts from one or several tables the lines which satisfy the WHERE clause. FieldList is a list of fields coming from the FROM tables, or from expressions implying functions. We can refer to an attribute by its name, the name of its table and its name, or by the name of its database, the name of its table and his name: university.student.lastname indicates the field *lastname* from the table student of the database university.

The option:

- STRAIGHT\_JOIN indicates that the joint must reach the tables in the specified order.
- SQL\_SMALL\_RESULT warns MySQL that the result will contain just a few lines, which allows to speed up the request.
- SQL\_BIG\_RESULT indicates the inverse.
- HIGH\_PRIORITY requires the priority of execution for the request compared to those which carry out modifications.

**Caution:** these options are reserved to the experimented users and must be used with full knowledge of the facts.

## ➤ **INSERT**

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] TableName [(fieldName, ...)] VALUES (expression, ...), (...), ...
or    INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] TableName [(fieldName, ...)] SELECT...
or    INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] TableName SET fieldName = expression, ...
```

E. g. use the instruction *INSERT*

```
INSERT INTO student (IDnumb, lastname, firstname, sex, birthYear, school, option, remark) VALUES
('20194', 'GUINKO', 'Tonguim Ferdinand', 'M', '1979', 'computer sciences ', 'software engineering',
'none');
```

The instruction *INSERT* insert one or several lines into a table.

## ➤ **USE**

syntax :

*USE baseName*

The command *USE* allow to have access to the database.

E. g. use of *USE* instruction

*USE university*

### **3. functions**

The following functions can be used in the requests:

- ABS (number): returns the absolute value of a number.
- ASCII (char): returns the ASCII code of the character.
- CONCAT (chaîne1, [ chaîne2? ]) returns the concatenation of all the arguments.
- CONV (a number, base1, base2): return the conversion of a number, base1 in base2; the base is a number between 2 and 36.
- BIN (decimal): returns the binary value of a decimal number.
- CURDATE (): returns the current date into format AAAAMMJJ or Aaaa-mm-jj according to whether the context is numerical or alphanumeric.
- CURTIME (): returns current time into format HHMMSS or HH:MM:SS according to whether the context is numerical or alphanumeric.
- DATABASE (): returns the name of the current data base.
- DAYNAME (date): returns the name of the day in English.
- DAYOFMONTH (date): returns the number of the month.
- DAYOFWEEK (date): returns the number of the day in the week.
- DAYOFYEAR (date): returns the number of the day in a year.
- LTRIM (chain): withdraws all the blanks spaces at the beginning of string.
- MONTHNAME (date): returns the name of the month of date in English.
- NOW (): returns the current date and hour.
- PASSWORD (string): encoding of string with the function used for the MySQL.
- UCASE (string): return string in uppercase letter.

## C. Access to MySQL with PHP

PHP communicates with MySQL through some functions which make it possible to recover, modify, or create all kind of information relating to a database. Among this information, it is necessary to count the contents of the tables, but also their description, i. e. the diagram of the database. These functions are:

*Mysql\_connect*: is used to establish a connection with MySQL, for a user account, and a data server; its returns a value which can be used to dialogue with the server.

*Mysql\_pconnect*: Idem, but with a persistent connection. This second version is more powerful when the PHP interpreter is included in Apache.

*Mysql\_select\_db*: allows to go into a database. It is the equivalent of the *USES* database instruction under mysql.

*Mysql\_query*: is used to carry out a SQL request; it returns a variable representing the result of the request.

*Mysql\_fetch\_object*: allows to recover one of the lines of the result, and positions the cursor on the following line; the line is represented such as an object (a group of values).

*Mysql\_fetch\_row*: allows to recover one of the lines of the result, and positions the cursor on the following line; the line is represented such as a table (a list of values).

*Mysql\_error*: return the message of the last error met.

Exemple : acces to MySQL with PHP

To reach a MySQL data base, a script PHP must:

- be connected to the MySQL database server's.
- send a SQL request to the MySQL database server's, then recover the result.
- extract the data of the result with the help of functions.
- generate HTML page to display the datas.

```

<HTML>
  <HEAD>
    <TITLE> connection to MySQL </TITLE>
  </HEAD>
  <BODY>
    <H1>interrogation of the student table</H1>
    < ?php

        // connection to the MySQL database server's

        require ("connect.php") ;

        $connexion = mysql_pconnect (server, loginName, passWord) ;

        if ( !$connexion)
        {
            echo "Sorry, connection to " . server . " failed\n";
            exit ;
        }

        if ( !mysql_select_db (tableName, $connexion))
        {
            echo "Sorry, acces to table " . tableName . " failed\n";
            exit ;
        }

        /* send a SQL request to the MySQL database server's, then recover the
result */

        $result = myquery ( "SELECT * FROM student ", $connection) ;

        /* extract the data of the result with the help of functions */

        if ($result)
        {
            while ($Student = mysql_fetch_object ($result))
            {
                echo "$Student ->IDnumb, of the student "
                . " $Student->lastname $Student->firstname" <BR>\n" ;
            }
        }
    < /?php
  </BODY>
</HTML>

```

```

        }
    }

    else

    {
        echo "<B>Errors detected in the request</B><BR>";
        echo "<B>Message from MySQL : </B>" . mysql_error
            ($connection) ;
    }

?>
</BODY>
</HTML>

```

The instructions *require* allows to include the contents of a file in a script. Some informations are common to many scripts, and to repeat them systematically is at the same time a waste of time and a large source of matters the day when it is necessary to carry out a modification in *n* versions duplicated. Here we placed in the file *connect.php* some basic information on the site: the name of the server, the name of the database and the access account to the database.

```

< ?php

    // connect.php

    $server = "www.hosting_server.com";           // host of MySQL database execution
    $loginName = "php"                           // database user name
    $passWord = "php"                            // user password
    $databaseName = "university"                 // database name

?>

```



## Conclusion

This work is on one hand the fruit of long research on the Net and in several works, and on the other hand, of our experience in this area. Certainly It is not exhaustive, but allows the programmer who wishes to take his first steps in PHP and/or MySQL to understand the basic elements of these languages, in simplified terms.

PHP represents in fact one technology among several others which can be used to create more dynamic and more interactive Web pages. Among other technologies allowing to carry out dynamic Web sites we can quote:

- *Active Server Pages (ASP)*: it is similar to PHP; the fundamental inconvenient of this technique is that it cannot be used with an other server than a Microsoft Web server (IIS, PWS), or with an Microsoft operating system.

- *Javascript side waiter (SSJS)*: it combines with HTML like the two precedents; the inconvenient of this technique is that it requires the compilation of SSJS applications before being able to carry out them; moreover, SSJS is only taken charges by Netscape Enterprise Server.

Concerning the DBMS, they are numerous: Access, Oracle, SQL Server, Sybase, InterBase, MySQL... The MySQL/PHP solution presents the advantage to be powerful, and portable on the linux and Windows operating systems.

### To go further

#### **Bibliography**

J. Castagnetto, H. Rawat, S. Schumann, C. Scollo, D. Veliath, "*PHP professionnel*", Eyrolles, 2001

Philippe Rigaux, « *MySQL et PHP* », O'Reilly, édition originale, 2001

#### **Webography**

<http://www.wrox.fr>

<http://www.php.net>

<http://www.mysql.com>