



# Khoa Công Nghệ Thông Tin Trường Đại Học Cần Thơ

---

## Lập Trình PHP



**Đỗ Thanh Nghi**  
*[dtngchi@cit.ctu.edu.vn](mailto:dtngchi@cit.ctu.edu.vn)*

Cần Thơ  
24-11-2017

# Nội dung

---

- Giới thiệu về PHP
- Biến, kiểu dữ liệu, phép toán
- Lệnh điều khiển
- Hàm
- PHP kết hợp với forms
- Cookies, SSI (Server side includes), Date
- PHP-MySQL

- 
- **Giới thiệu về PHP**
  - Biến, kiểu dữ liệu, phép toán
  - Lệnh điều khiển
  - Hàm
  - PHP kết hợp với forms
  - Cookies, SSI (Server side includes), Date
  - PHP-MySQL

# Giới thiệu về PHP

---

## ■ PHP là gì ?

- PHP là Hypertext Preprocessor
- Ngôn ngữ script chạy trên server
- PHP scripts chứa text, thẻ HTML, script
- Sử dụng phần mở rộng tên file : .php, .phtml
- PHP scripts sẽ trả về kết quả cho trình duyệt một plain HTML
- PHP hỗ trợ để làm việc với nhiều hệ QTCSDL khác nhau
- MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.
- Phần mềm mã nguồn mở, miễn phí
- Chạy trên nhiều platforms (Unix, Linux, Windows)

# Giới thiệu về PHP

---

## ■ MySQL là gì ?

- Hệ quản trị cơ sở dữ liệu
- Dùng cho các ứng dụng vừa và nhỏ
- Hỗ trợ chuẩn SQL
- Phần mềm mã nguồn mở, miễn phí
- Chạy trên nhiều platforms (Unix, Linux, Windows)
- Phổ biến
- PHP + MySQL : Web động chạy trên nhiều platforms khác nhau

# Giới thiệu về PHP

---

## ■ Tại sao PHP ?

- Chạy trên nhiều platforms khác nhau (Unix, Linux, Windows)
- Phần mềm mã nguồn mở, miễn phí
- Tương thích với hầu hết các web server (Apache, IIS, etc)
- Dễ học và phát triển nhanh các ứng dụng trên Web

## ■ Làm thế nào để sử dụng PHP

- Cài web server (Apache, IIS, etc)
- Cài MySQL
- Cài PHP
- Địa chỉ : [www.apache.org](http://www.apache.org), [www.php.net](http://www.php.net), [www.mysql.com](http://www.mysql.com)

- 
- Giới thiệu về PHP
  - **Biến, kiểu dữ liệu, phép toán**
  - Lệnh điều khiển
  - Hàm
  - PHP kết hợp với forms
  - Cookies, SSI (Server side includes), Date
  - PHP-MySQL

# Cú pháp PHP

---

## ■ Cú pháp

- PHP scripts chứa text, thẻ HTML, script
- Ví dụ : in ra màn hình chuỗi “Hello World”

```
<html>
```

```
<body>
```

```
<?php echo "Hello World"; ?>
```

```
</body>
```

```
</html>
```



# Cú pháp PHP

---

## ■ Cú pháp

- Khởi lệnh PHP script bắt đầu với

**<?php**

và kết thúc bởi

**?>**

- Khởi lệnh có thể được đặt bất cứ nơi nào trong tài liệu
- Mỗi lệnh cách nhau bởi dấu ;
- Có 2 lệnh cơ bản để xuất dữ liệu ra màn hình: **echo** và **print**
- Chú thích trong chương trình
- // chú thích là 1 dòng đơn
- /\* chú thích là 1 đoạn  
văn bản \*/

# Cú pháp PHP

---

## ■ Cú pháp

- Ví dụ :

```
<?php
```

```
    echo "This is a test"; // This is a one-line c++ style comment
```

```
    /* This is a multi line comment
```

```
        yet another line of comment */
```

```
    echo("This is yet another test");
```

```
    print "Hello World";
```

```
    print("Hello World");
```

```
?>
```

# Cú pháp PHP

---

- Không phân biệt ký tự thường hoa
  - Từ khóa
  - Lớp
  - Hàm, hàm được tạo bởi người lập trình
- Chỉ phân biệt ký tự thường hoa
  - Các biến

# Biến

---

## ■ Biến trong PHP

- Chứa dữ liệu
- Biến được bắt đầu bởi dấu \$
- Tên biến bắt đầu bằng một ký tự chữ cái hoặc \_
- Phân biệt giữa ký tự thường và hoa
- Kiểu được tính ở thời điểm gán giá trị
- Gán giá trị với =
- Sử dụng & như tham chiếu

# Biến

---

## ■ Biến trong PHP

- Ví dụ :

```
<?php
```

```
$var = 'Bob';
```

```
$Var = 'Joe';
```

```
echo "$var, $Var";    // outputs "Bob, Joe"
```

```
$4site = 'not yet';   // invalid; starts with a number
```

```
$_4site = 'not yet';  // valid; starts with an underscore
```

```
$täyte = 'mansikka';  // valid; 'ä' is (Extended) ASCII 228.
```

```
?>
```

# Biến

---

## ■ Biến trong PHP

- Ví dụ :

```
<?php
```

```
$foo = 'Bob';           // Assign the value 'Bob' to $foo
```

```
$bar = &$foo;           // Reference $foo via $bar.
```

```
$bar = "My name is $bar"; // Alter $bar...
```

```
echo $bar;              // My name is Bob
```

```
echo $foo;              // My name is Bob
```

```
?>
```

# Biến

---

## ■ Biến trong PHP

- Ví dụ :

```
<?php
$foo = 'Bob';
echo $foo;           // Bob
$foo = 12
echo $foo;           // 12
$foo = array(1, 2, 3, 4, 5);
for($i = 0; $i < 5; $i++)
    echo $foo[$i] . "<br>";
?>
```

# Biến

---

## ■ Biến có sẵn trong PHP

- `$GLOBALS` : tất cả các biến trong phạm vi toàn cục của script
- `$_SERVER` : tập hợp biến môi trường của Web server
- `$_GET`, `$_POST` : biến được cung cấp các chuỗi query URL cho script
- `$_COOKIE` : biến cung cấp HTTP\_cookies cho script
- `$_FILES` : biến cung cấp HTTP POST file uploads cho script
- `$_ENV` : biến cung cấp môi trường cho script
- `$_REQUEST` : cung cấp các `$_GET`, `$_POST`, `$_COOKIE`



# Biến

---

## ■ Phạm vi biến

- Toàn cục : sử dụng từ khóa global hoặc biến \$GLOBALS
- Ví dụ :

```
<?php
```

```
$a = 1;
```

```
include 'b.inc'; // biến $a sẽ dùng trong b.inc
```

```
?>
```

# Biến

---

## ■ Phạm vi biến

- Toàn cục : sử dụng từ khóa global hoặc biến \$GLOBALS
- Ví dụ :

```
<?php
```

```
$a = 1;
```

```
$b = 2;
```

```
function Sum() {
```

```
    global $a, $b;
```

```
    $b = $a + $b;
```

```
}
```

```
Sum();
```

```
echo $b;
```

```
?>
```

# Biến

---

## ■ Phạm vi biến

- Toàn cục : sử dụng từ khóa global hoặc biến \$GLOBALS
- Ví dụ :

```
<?php
```

```
$a = 1;
```

```
$b = 2;
```

```
function Sum() {
```

```
    $GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
```

```
}
```

```
Sum();
```

```
echo $b;
```

```
?>
```

# Biến

## ■ Phạm vi biến

- Cục bộ
- Ví dụ :

```
<?php
```

```
$a = 1; /* global scope */
```

```
function Test() {
```

```
    $a = 10;
```

```
    echo "in Test a = " . $a; /* reference to local scope variable */
```

```
}
```

```
Test();
```

```
echo "<br> out Test a = " . $a;
```

```
?>
```

# Biến

## ■ Phạm vi biến

- Biến tĩnh : sử dụng từ khóa static
- Ví dụ :

```
<?php
```

```
function Test() {
```

```
    static $a = 10;
```

```
    echo "in Test a = " . $a;
```

```
    $a++;
```

```
}
```

```
Test(); // 10
```

```
Test(); // 11
```

```
?>
```

# Kiểu

---

## ■ Kiểu dữ liệu cơ bản

- Số nguyên : 4 bytes, số có dấu
- Số thực
- Luận lý : TRUE/FALSE
- Chuỗi ký tự

## ■ Kiểu dữ liệu phức hợp

- mảng
- Đối tượng
- Kiểu giả
- Etc.

# Kiểu

---

## ■ Kiểu dữ liệu

- Ví dụ : số nguyên, số thực

<?php

\$a = 1234; // decimal number

\$a = -123; // a negative number

\$a = 0123; // octal number (equivalent to 83 decimal)

\$a = 0x1A; // hexadecimal number (equivalent to 26 decimal)

\$b = 1.234;

\$c = 1.2e3;

\$d = 7E-10;

?>

# Kiểu<sup>2</sup>

## ■ Kiểu dữ liệu

- Ví dụ : luận lý

```
<?php
```

```
$foo = True; // assign the value TRUE to $foo
```

```
if ($action == "show_version") {
```

```
    echo "The version is 1.23";
```

```
}
```

```
// this is not necessary...
```

```
if ($show_separators == TRUE) {
```

```
    echo "<hr>\n";
```

```
}
```

```
// ...because you can simply type
```

```
if ($show_separators) {
```

```
    echo "<hr>\n";
```

```
} ?>
```



# Kiểu

## ■ Kiểu dữ liệu

- Ví dụ : chuỗi

```
<?php
```

```
$beer = 'Heineken';
```

```
echo "$beer's taste is great"; // works, "" is an invalid character for varnames
```

```
echo "He drank some $beers"; // won't work, 's' is a valid character for varnames
```

```
echo "He drank some ${beer}s"; // works
```

```
echo "He drank some {$beer}s"; // works
```

```
$str = 'This is a test.';
```

```
$third = $str{2}; // Get the third character of a string
```

```
$str = "This is still a test.";
```

```
$last = $str{strlen($str)-1}; // Get the last character of a string.
```

```
$str = 'Look at the sea';
```

```
$str{strlen($str)-1} = 'e'; // Modify the last character of a string
```

```
?>
```

# Kiểu

## ■ Kiểu dữ liệu

- mảng

`array( [key =>] value`

`, ...`

`)`

`// key may be an integer or string`

`// value may be any value`

- Ví dụ :

`<?php`

`$arr = array("foo" => "bar", 12 => 1);`

`echo $arr["foo"]; // bar`

`echo $arr[12]; // 1`

`?>`

# Kiểu

## ■ Kiểu dữ liệu

- mảng, ví dụ :

```
<?php
```

```
$arr = array("somearray" => array(6 => 5, 13 => 9, "a" => 42));
```

```
echo $arr["somearray"][6]; // 5
```

```
echo $arr["somearray"][13]; // 9
```

```
echo $arr["somearray"]["a"]; // 42
```

```
// This array is the same as ...
```

```
$a = array(5 => 43, 32, 56, "b" => 12);
```

```
// ...this array
```

```
$a_n = array(5 => 43, 6 => 32, 7 => 56, "b" => 12);
```

```
?>
```

# Kiểu

---

## ■ Kiểu dữ liệu

- Truy xuất các phần tử mảng: `$array_name[key]`
- Ví dụ :

```
<?php
```

```
$arr = array(5 => 1, 12 => 2);
```

```
$arr[] = 56; // This is the same as $arr[13] = 56;
```

```
$arr["x"] = 42; // This adds a new element to the array with key "x"
```

```
unset($arr[5]); // This removes the element from the array
```

```
unset($arr); // This deletes the whole array
```

```
?>
```

# Kiểu

---

## ■ Kiểu dữ liệu

- Ví dụ : mảng

```
<?php
```

```
$array = array(1, 2, 3, 4, 5); // Create a simple array.
```

```
print_r($array);
```

```
foreach ($array as $i => $value) // Now delete every item, but leave the array itself intact:
```

```
    echo $array[$i] . "<br>";
```

```
?>
```

# Phép toán

## 15-1. Operator Precedence

| Associativity   | Operators                                       | Additional Information                                 |
|-----------------|---|--|
| non-associative | new   | <a href="#">new</a>                                    |
| left            | [   | <a href="#">array()</a>                                |
| non-associative | ++ --   | <a href="#">increment/decrement</a>                    |
| non-associative | ! ~ - (int) (float) (string) (array) (object) @ | <a href="#">types</a>                                  |
| left            | * / %   | <a href="#">arithmetic</a>                             |
| left            | + - .   | <a href="#">arithmetic</a> and <a href="#">string</a>  |
| left            | << >>   | <a href="#">bitwise</a>                                |
| non-associative | < <= > >=                                       | <a href="#">comparison</a>                             |
| non-associative | == != === !==                                   | <a href="#">comparison</a>                             |
| left            | &   | <a href="#">bitwise</a> and <a href="#">references</a> |
| left            | ^   | <a href="#">bitwise</a>                                |
| left            |   | <a href="#">bitwise</a>                                |
| left            | &&  | <a href="#">logical</a>                                |
| left            |   | <a href="#">logical</a>                                |
| left            | ? :   | <a href="#">ternary</a>                                |
| right           | = += -= *= /= .= %= &=  = ^= <<= >>=            | <a href="#">assignment</a>                             |
| left            | and   | <a href="#">logical</a>                                |
| left            | xor   | <a href="#">logical</a>                                |
| left            | or  | <a href="#">logical</a>                                |
| left            | ,   | many uses  |

# Phép toán

---

## 15-2. Arithmetic Operators

| Example                | Name           | Result  |
|------------------------|----------------|---|
| <code>-\$a</code>      | Negation       | Opposite of <code>\$a</code> .                              |
| <code>\$a + \$b</code> | Addition       | Sum of <code>\$a</code> and <code>\$b</code> .              |
| <code>\$a - \$b</code> | Subtraction    | Difference of <code>\$a</code> and <code>\$b</code> .       |
| <code>\$a * \$b</code> | Multiplication | Product of <code>\$a</code> and <code>\$b</code> .          |
| <code>\$a / \$b</code> | Division       | Quotient of <code>\$a</code> and <code>\$b</code> .         |
| <code>\$a % \$b</code> | Modulus        | Remainder of <code>\$a</code> divided by <code>\$b</code> . |

## 15-7. Logical Operators

| Example                         | Name | Result  |
|---------------------------------|------|---|
| <code>\$a and \$b</code>        | And  | <b>TRUE</b> if both <code>\$a</code> and <code>\$b</code> are <b>TRUE</b> .               |
| <code>\$a or \$b</code>         | Or   | <b>TRUE</b> if either <code>\$a</code> or <code>\$b</code> is <b>TRUE</b> .               |
| <code>\$a xor \$b</code>        | Xor  | <b>TRUE</b> if either <code>\$a</code> or <code>\$b</code> is <b>TRUE</b> , but not both. |
| <code>! \$a</code>              | Not  | <b>TRUE</b> if <code>\$a</code> is not <b>TRUE</b> .                                      |
| <code>\$a &amp;&amp; \$b</code> | And  | <b>TRUE</b> if both <code>\$a</code> and <code>\$b</code> are <b>TRUE</b> .               |
| <code>\$a    \$b</code>         | Or   | <b>TRUE</b> if either <code>\$a</code> or <code>\$b</code> is <b>TRUE</b> .               |

# Phép toán

## 15-4. Comparison Operators

| Example                       | Name                     | Result  |
|-------------------------------|--------------------------|---|
| <code>\$a == \$b</code>       | Equal                    | <b>TRUE</b> if \$a is equal to \$b.   |
| <code>\$a === \$b</code>      | Identical                | <b>TRUE</b> if \$a is equal to \$b, and they are of the same type. (introduced in PHP 4)        |
| <code>\$a != \$b</code>       | Not equal                | <b>TRUE</b> if \$a is not equal to \$b.   |
| <code>\$a &lt;&gt; \$b</code> | Not equal                | <b>TRUE</b> if \$a is not equal to \$b.   |
| <code>\$a !== \$b</code>      | Not identical            | <b>TRUE</b> if \$a is not equal to \$b, or they are not of the same type. (introduced in PHP 4) |
| <code>\$a &lt; \$b</code>     | Less than                | <b>TRUE</b> if \$a is strictly less than \$b.   |
| <code>\$a &gt; \$b</code>     | Greater than             | <b>TRUE</b> if \$a is strictly greater than \$b.  |
| <code>\$a &lt;= \$b</code>    | Less than or equal to    | <b>TRUE</b> if \$a is less than or equal to \$b.  |
| <code>\$a &gt;= \$b</code>    | Greater than or equal to | <b>TRUE</b> if \$a is greater than or equal to \$b.   |



# Phép toán

## 15-8. Array Operators

| Example                       | Name         | Result  |
|-------------------------------|--------------|---|
| <code>\$a + \$b</code>        | Union        | Union of <code>\$a</code> and <code>\$b</code> .  |
| <code>\$a == \$b</code>       | Equality     | <b>TRUE</b> if <code>\$a</code> and <code>\$b</code> have the same key/value pairs.   |
| <code>\$a === \$b</code>      | Identity     | <b>TRUE</b> if <code>\$a</code> and <code>\$b</code> have the same key/value pairs in the same order and of the same types. |
| <code>\$a != \$b</code>       | Inequality   | <b>TRUE</b> if <code>\$a</code> is not equal to <code>\$b</code> .  |
| <code>\$a &lt;&gt; \$b</code> | Inequality   | <b>TRUE</b> if <code>\$a</code> is not equal to <code>\$b</code> .  |
| <code>\$a !== \$b</code>      | Non-identity | <b>TRUE</b> if <code>\$a</code> is not identical to <code>\$b</code> .  |

The `+` operator appends the right handed array to the left handed, whereas duplicated keys are NOT overwritten.

```
<?php
$a = array("a" => "apple", "b" => "banana");
$b = array("a" => "pear", "b" => "strawberry", "c" => "cherry");

$c = $a + $b; // Union of $a and $b
echo "Union of \$a and \$b: \n";
var_dump($c);

$c = $b + $a; // Union of $b and $a
echo "Union of \$b and \$a: \n";
var_dump($c);
?>
```

- 
- Giới thiệu về PHP
  - Biến, kiểu dữ liệu, phép toán
  - **Lệnh điều khiển**
  - Hàm
  - PHP kết hợp với forms
  - Cookies, SSI (Server side includes), Date
  - PHP-MySQL

# Điều kiện

---

## ■ IF

- Cú pháp :

if (condition)

code to be executed if condition is true;

else

code to be executed if condition is false;

- Ví dụ :

```
<?php
```

```
$d=date("D");
```

```
if ($d=="Fri")
```

```
    echo "Have a nice weekend!";
```

```
else
```

```
    echo "Have a nice day!";
```

```
?>
```

# Điều kiện

---

## ■ Switch

- Cú pháp :

```
switch (expression) {
```

```
case label1:
```

```
    code to be executed if expression = label1;
```

```
    break;
```

```
case label2:
```

```
    code to be executed if expression = label2;
```

```
    break;
```

```
default:
```

```
    code to be executed
```

```
    if expression is different
```

```
    from both label1 and label2;
```

```
}
```

# Điều kiện

---

## ■ Switch

- Ví dụ :

```
<?php
switch ($x) {
case 1:
    echo "Number 1"; break;
case 2:
    echo "Number 2"; break;
case 3:
    echo "Number 3"; break;
default:
    echo "No number between 1 and 3";
}
?>
```

# Lặp

---

## ■ While

- Cú pháp :

while (condition)

code to be executed;

- Ví dụ :

```
<?php
```

```
$i=1;
```

```
while($i<=5) {
```

```
    echo "The number is " . $i . "<br />";
```

```
    $i++;
```

```
}
```

```
?>
```

# Lặp

---

## ■ Do ... while

- Cú pháp :

```
do {  
    code to be executed;  
} while (condition);
```

- Ví dụ :

```
<?php  
$i=0;  
do {  
    $i++;  
    echo "The number is " . $i . "<br />";  
} while ($i<5);  
?>
```

# Lặp

---

## ■ For

- Cú pháp :

```
for (initialization; condition; increment) {  
    code to be executed;  
}
```

- Ví dụ :

```
<?php  
for ($i=1; $i<=5; $i++)  
{  
    echo "Hello World!<br />";  
}  
?>
```



# Lặp

---

## ■ Foreach

- Cú pháp :

```
foreach (array as value) {  
    code to be executed;  
}
```

- Ví dụ :

```
<?php  
$arr=array("one", "two", "three");  
foreach ($arr as $value)  
{  
    echo "Value: " . $value . "<br />";  
}  
?>
```

- 
- Giới thiệu về PHP
  - Biến, kiểu dữ liệu, phép toán
  - Lệnh điều khiển
  - **Hàm**
  - PHP kết hợp với forms
  - Cookies, SSI (Server side includes), Date
  - PHP-MySQL

- Function Reference
- Apache-specific Function:
- Advanced PHP debugger
- Array Functions
- Aspell functions [deprecat
- BCMath Arbitrary Precisi
- PHP bytecode Compiler
- Bzip2 Compression Funct
- Calendar Functions
- CCVS API Functions [dep
- Classkit Functions
- Class/Object Functions
- COM and .Net (VWindows)
- ClibPDF Functions
- Crack Functions
- Character Type Functions
- CURL, Client URL Library
- Cybercash Payment Func
- Cyrus IMAP administratio
- Date and Time Functions
- Database (dbm-style) Abs
- dBase Functions
- DBM Functions [deprecat
- DB++ Functions
- dbx Functions
- Direct IO Functions
- Directory Functions
- DOM Functions
- DOM XML Functions
- .NET Functions
- Error Handling and Loggin
- Program Execution Funct
- Exif Functions

?????

## VI. Function Reference

????

- I. [.NET Functions](#)
- II. [Apache-specific Functions](#)
- III. [Alternative PHP Cache](#)
- IV. [Advanced PHP debugger](#)
- V. [Array Functions](#)
- VI. [Aspell functions \[deprecated\]](#)
- VII. [BCMath Arbitrary Precision Mathematics Functions](#)
- VIII. [PHP bytecode Compiler](#)
- IX. [Bzip2 Compression Functions](#)
- X. [Calendar Functions](#)
- XI. [CCVS API Functions \[deprecated\]](#)
- XII. [Class/Object Functions](#)
- XIII. [Classkit Functions](#)
- XIV. [ClibPDF Functions](#)
- XV. [COM and .Net \(Windows\)](#)
- XVI. [Crack Functions](#)
- XVII. [Character Type Functions](#)
- XVIII. [CURL, Client URL Library Functions](#)
- XIX. [Cybercash Payment Functions](#)
- XX. [Credit Mutuel CyberMUT functions](#)
- XXI. [Cyrus IMAP administration Functions](#)
- XXII. [Date and Time Functions](#)
- XXIII. [DB++ Functions](#)
- XXIV. [Database \(dbm-style\) Abstraction Layer Functions](#)
- XXV. [dBase Functions](#)
- XXVI. [DBM Functions \[deprecated\]](#)
- XXVII. [dbx Functions](#)
- XXVIII. [Direct IO Functions](#)
- XXIX. [Directory Functions](#)
- XXX. [DOM Functions](#)
- XXXI. [DOM XML Functions](#)
- XXXII. [Error Handling and Logging Functions](#)
- XXXIII. [Exif Functions](#)

Hàm định  
nghĩa sẵn  
trong PHP

# Hàm do người sử dụng định nghĩa

---

## ■ Hàm

- Cú pháp :

```
<?php
```

```
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
```

```
{
```

```
    echo "Example function.\n";
```

```
    return $retval;
```

```
}
```

```
?>
```

# Hàm do người sử dụng định nghĩa

## 17-2. Conditional functions

```
<?php

$makefoo = true;

/* We can't call foo() from here
   since it doesn't exist yet,
   but we can call bar() */

bar();

if ($makefoo) {
    function foo()
    {
        echo "I don't exist until program execution reaches me.\n";
    }
}

/* Now we can safely call foo()
   since $makefoo evaluated to true */

if ($makefoo) foo();

function bar()
{
    echo "I exist immediately upon program start.\n";
}

?>
```

# Hàm do người sử dụng định nghĩa

## 17-3. Functions within functions

```
<?php
function foo()
{
    function bar()
    {
        echo "I don't exist until foo() is called.\n";
    }
}

/* We can't call bar() yet
   since it doesn't exist. */

foo();

/* Now we can call bar(),
   foo()'s processing has
   made it accessible. */

bar();

?>
```

## 17-4. Recursive functions

```
<?php
function recursion($a)
{
    if ($a < 20) {
        echo "$a\n";
        recursion($a + 1);
    }
}

?>
```

# Hàm do người sử dụng định nghĩa

---

## ■ Tham số

- Truyền tham số : giá trị, tham chiếu
- Hàm : `func_num_args()`, `func_get_arg()`
- Ví dụ tham số là mảng:

```
<?php
function takes_array($input) {
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

# Hàm do người sử dụng định nghĩa

---

## ■ Tham số

- Ví dụ tham số có giá trị mặc định :

```
<?php
```

```
function makecoffee($type = "cappuccino")
```

```
{
```

```
    return "Making a cup of $type.<br>";
```

```
}
```

```
echo makecoffee();
```

```
echo makecoffee("espresso");
```

```
?>
```



# Hàm do người sử dụng định nghĩa

---

## ■ Tham số

- Ví dụ truyền tham chiếu :

```
<?php
```

```
function add_some_extra(&$string)
```

```
{
```

```
    $string .= 'and something extra.';
```

```
}
```

```
$str = 'This is a string, ';
```

```
add_some_extra($str);
```

```
echo $str;    // outputs 'This is a string, and something extra.'
```

```
?>
```

# Hàm do người sử dụng định nghĩa

---

## ■ Giá trị trả về

- Ví dụ :

```
<?php
```

```
function square($num)
```

```
{
```

```
    return $num * $num;
```

```
}
```

```
echo square(4); // outputs '16'.
```

```
?>
```

# Hàm do người sử dụng định nghĩa

---

## ■ Giá trị trả về

- Ví dụ :

```
<?php
```

```
function small_numbers()
```

```
{
```

```
    return array (0, 1, 2);
```

```
}
```

```
list ($zero, $one, $two) = small_numbers();
```

```
?>
```

# Hàm do người sử dụng định nghĩa

---

## ■ Giá trị trả về

- Ví dụ :

```
<?php
function &returns_reference()
{
    return $someref;
}
$newref =& returns_reference();
?>
```

- 
- Giới thiệu về PHP
  - Biến, kiểu dữ liệu, phép toán
  - Lệnh điều khiển
  - Hàm
  - **PHP kết hợp với forms**
  - Cookies, SSI (Server side includes), Date
  - PHP-MySQL

# PHP + HTML Form

## ■ PHP kết hợp với HTML Form

- Hầu hết các thành phần của HTML Form đều có thể được truy xuất từ chương trình PHP script
- Sử dụng biến `$_GET` hay `$_POST` để truy xuất đến các thành phần của HTML Form
- Ví dụ : trang web là `welcome.html` nội dung như sau

```
<html>
```

```
<body>
```

```
<form action="welcome.php" method="POST">
```

```
Enter your name: <input type="text" name="name">
```

```
Enter your age: <input type="text" name="age">
```

```
<input type="submit" value="welcome">
```

```
</form>
```

```
</body>
```

```
</html>
```

# PHP + HTML Form

## ■ PHP kết hợp với HTML Form

- PHP script "**welcome.php**" sử dụng biến `$_POST` để truy xuất đến các thành phần của HTML Form do sử dụng **`method="POST"`**
- PHP script `welcome.php` nội dung như sau

```
<html>
```

```
<body>
```

```
Welcome <?php echo $_POST["name"]; ?>.<br>
```

```
You are <?php echo $_POST["age"]; ?> years old!
```

```
</body>
```

```
</html>
```

- 
- Giới thiệu về PHP
  - Biến, kiểu dữ liệu, phép toán
  - Lệnh điều khiển
  - Hàm
  - PHP kết hợp với forms
  - **Cookies, SSI (Server side includes), Date**
  - PHP-MySQL



# Cookies

---

## ■ Cookie

- Thường được sử dụng để xác định một user
- Server ghi 1 tập tin cookie lên web client
- PHP cho phép tạo và đọc lại những giá trị từ cookie
- Hàm tạo cookie : setcookie(name, value, expire, path, domain)
- Được đặt trước thẻ <html>
- Ví dụ :

```
<?php setcookie("uname", $name, time()+36000); ?>
```

```
<html>
```

```
<body>
```

```
<p> A cookie was set on this page! The cookie will be active when the client has sent the  
    cookie back to the server. </p>
```

```
</body>
```

```
</html>
```

# Cookies

---

## ■ Cookie

- Hàm isset() để đọc lại cookie đã được tạo

- Ví dụ :

```
<html>
```

```
<body>
```

```
<?php
```

```
if (isset($_COOKIE["uname"]))
```

```
    echo "Welcome " . $_COOKIE["uname"] . " !<br />";
```

```
else
```

```
    echo "You are not logged in!<br />";
```

```
?>
```

```
</body>
```

```
</html>
```

# Server side includes

---

## ■ SSI

- Chèn đoạn code chương trình của một file vào file khác trước khi thực thi
- Sử dụng hàm require()
- Ví dụ :

```
<html>
```

```
<body>
```

```
<?php require("header.htm"); ?>
```

```
<p> Some text </p> <p>Some text</p>
```

```
</body>
```

```
</html>
```

# Hàm thời gian

## ■ Date()

- Cú pháp : `string date (date_format[,int timestamp])`

---

### Date Formats

The table below shows the characters that may be used in the format string:

| Character | Description  |
|-----------|--|
| a         | "am" or "pm"   |
| A         | "AM" or "PM"   |
| B         | Swatch Internet time (000-999)                                 |
| d         | Day of the month with a leading zero (01-31)                   |
| D         | Three characters that represents the day of the week (Mon-Sun) |
| F         | The full name of the month (January-December)                  |
| g         | The hour in 12-hour format without a leading zero (1-12)       |
| G         | The hour in 24-hour format without a leading zero (0-23)       |
| h         | The hour in 12-hour format with a leading zero (01-12)         |
| H         | The hour in 24-hour format with a leading zero (00-23)         |
| i         | The minutes with a leading zero (00-59)                        |
| I         | "1" if the date is in daylight savings time, otherwise "0"     |
| j         | Day of the month without a leading zero (1-31)                 |
| l         | The full name of the day (Monday-Sunday)                       |

# Hàm thời gian

---

|   |  |
|---|--|
| L | "1" if the year is a leap year, otherwise "0"                            |
| m | The month as a number, with a leading zero (01-12)                       |
| M | Three letters that represents the name of the month (Jan-Dec)            |
| n | The month as a number without a leading zero (1-12)                      |
| O | The difference to Greenwich time (GMT) in hours                          |
| r | An RFC 822 formatted date (e.g. "Tue, 10 Apr 2005 18:34:07 +0300")       |
| s | The seconds with a leading zero (00-59)                                  |
| S | The English ordinal suffix for the day of the month (st, nd, rd or th)   |
| t | The number of days in the given month (28-31)                            |
| T | The local time zone (e.g. "GMT")   |
| U | The number of seconds since the Unix Epoch (January 1 1970 00:00:00 GMT) |
| w | The day of the week as a number (0-6, 0=Sunday)                          |
| W | ISO-8601 week number of year, weeks starting on Monday                   |
| Y | The year as a 4-digit number (e.g. 2003)                                 |
| y | The year as a 2-digit number (e.g. 03)                                   |
| z | The day of the year as a number (0-366)                                  |

# Hàm thời gian

---

## ■ Date()

- Ví dụ :

```
<?php
```

```
//Prints something like: Monday
```

```
echo date("l");
```

```
//Prints something like: Monday 15th of January 2003 05:51:38 AM
```

```
echo date("l dS of F Y h:i:s A");
```

```
//Prints something like: Monday the 15th
```

```
echo date("l \\t\\h\\e jS");
```

```
?>
```

- 
- Giới thiệu về PHP
  - Biến, kiểu dữ liệu, phép toán
  - Lệnh điều khiển
  - Hàm
  - PHP kết hợp với forms
  - Cookies, SSI (Server side includes), Date
  - **PHP-MySQL**

# MySQL

---

## ■ MySQL

- Download : [www.mysql.com](http://www.mysql.com), cài đặt
- Có thể cài thêm giao diện quản trị
- Hoặc sử dụng trình mysql (client)

```
mysql -u root -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 4 to server version: 5.0.15-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```



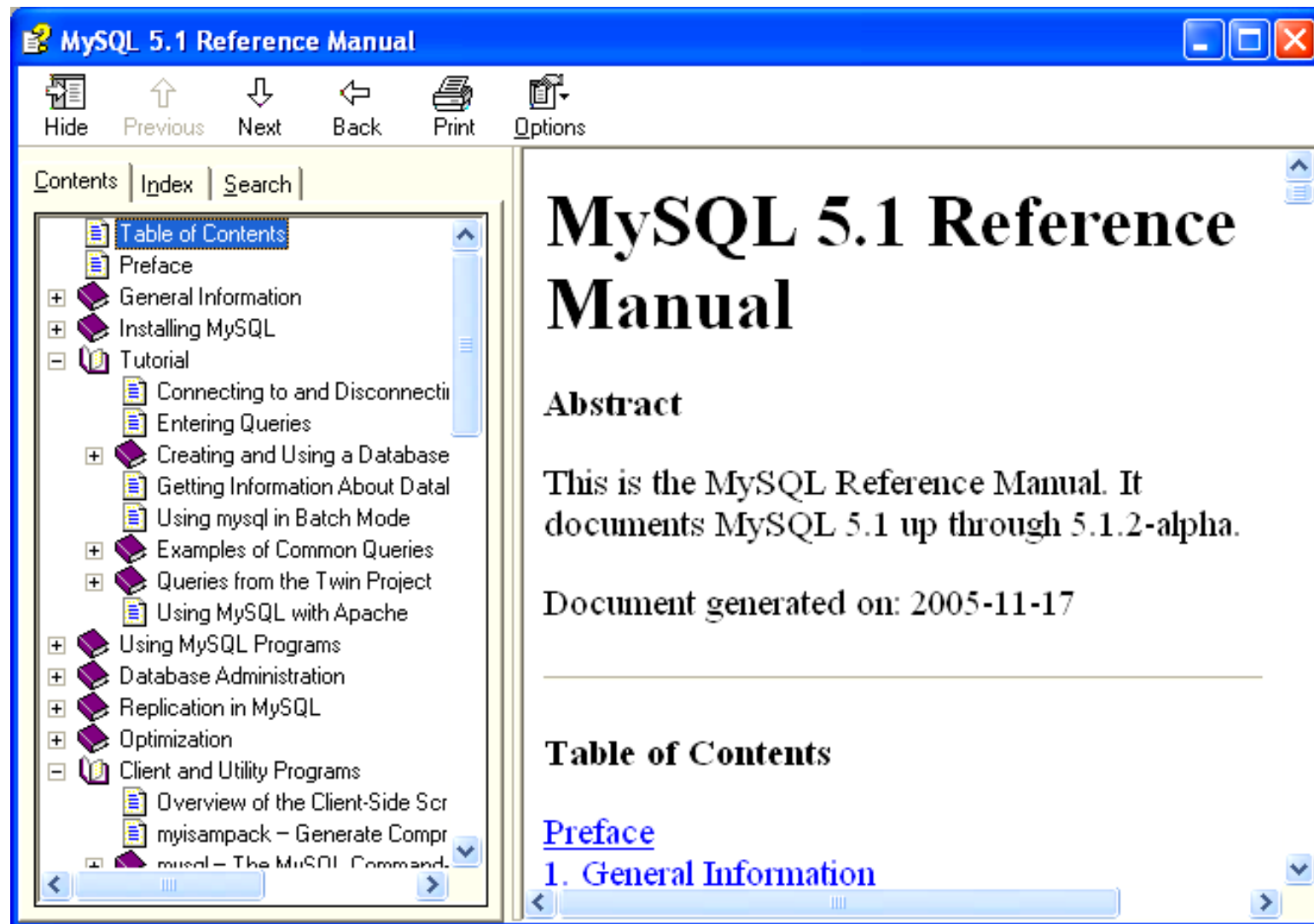
# MySQL

---

## ■ Lệnh cơ bản MySQL

- Tạo xóa cơ sở dữ liệu : `create (drop) database dbname`
- Tạo xóa người dùng : `create (drop) user uname`
- Tạo xóa quyền truy cập : `grant (revoke) ...`
- Tạo xóa bảng : `create (drop) table tname`
- Chèn mẫu tin : `insert into tname values (...)`
- Xóa mẫu tin : `delete ... from tname where ...`
- Cập nhật : `update tname set colname = value ...`

# MySQL



# MySQL

The screenshot shows the MySQL Administrator application window. The title bar reads 'MySQL Administrator - root@127.0.0.1:3306'. The interface includes a menu bar (File, Edit, View, Tools, Window, Help) and a left-hand sidebar with various system management options like Server Information, Service Control, and User Administration. Below these is a 'Schemata' section listing databases: information\_schema, mydb, mysql (selected), and test. The main pane is titled 'mysql All mysql indices' and displays a table of indices. The table has five columns: Index Name, Table Name, Type, Unique, and Not Null. It lists 25 indices, including 'user\_info\_Full\_name', 'User', and several 'PRIMARY' indices for tables like 'user\_info', 'user', and 'time\_zone'. At the bottom of the main pane, summary statistics are shown: 'Num. of Indices: 25', 'Num. of Columns: 45', 'Num. of Unique: 21', and 'Num. of NotNull: 24'. A 'Refresh' button is located at the bottom right of the main pane.

| Index Name          | Table Name                | Type  | Unique | Not Null |
|---------------------|---------------------------|-------|--------|----------|
| user_info_Full_name | user_info                 | BTREE |        |          |
| User                | db                        | BTREE |        | NOT NULL |
| PRIMARY             | user_info                 | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | user                      | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | time_zone_transition_type | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | time_zone_transition      | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | time_zone_name            | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | time_zone_leap_second     | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | time_zone                 | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | tables_priv               | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | procs_priv                | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | proc                      | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | host                      | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | help_topic                | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | help_relation             | BTREE | UNIQUE | NOT NULL |
| PRIMARY             | help_keyword              | BTREE | UNIQUE | NOT NULL |

Num. of Indices: 25 | Num. of Columns: 45 | Num. of Unique: 21 | Num. of NotNull: 24

Refresh

# MySQL

---

## ■ Ví dụ :

- Tạo cơ sở dữ liệu mydb : `create database mydb;`
- Tạo bảng Person

```
mysql> use mydb;
```

```
Database changed
```

```
mysql> CREATE TABLE Person
```

```
-> (
```

```
-> lastname varchar(30),
```

```
-> firstname varchar(10),
```

```
-> address varchar(30),
```

```
-> age int
```

```
-> );
```

```
mysql>
```

# MySQL

---

- Ví dụ :
  - Chèn các mẫu tin vào bảng Person

```
mysql> insert into Person values ('Thanh-Nghi', 'Do', '84/40, CMT8',31);
```

```
mysql> insert into Person values ('Nguyen-Khang', 'Pham', '43/20, Mau Than',27);
```

```
mysql> insert into Person values ('Nguyen-Binh', 'Le', '12, Nguyen Thong',18);
```

```
mysql> insert into Person values ('Trung-Tin', 'Nguyen', '31, Ngo Quyen',12);
```

```
mysql> insert into Person values ('Binh-Minh', 'Bui', 'C8, Truong Dinh',22);
```

```
mysql>
```

# MySQL

---

## ■ Ví dụ :

- Thực hiện câu truy vấn trên bảng Person

```
mysql> select * from Person;
```

| lastname     | firstname | address          | age |
|--------------|-----------|------------------|-----|
| Thanh-Nghi   | Do        | 84/40, CMT8      | 31  |
| Nguyen-Khang | Pham      | 43/20, Mau Than  | 27  |
| Nguyen-Binh  | Le        | 12, Nguyen Thong | 18  |
| Trung-Tin    | Nguyen    | 31, Ngo Quyen    | 12  |
| Binh-Minh    | Bui       | C8, Truong Dinh  | 22  |

```
5 rows in set (0.00 sec)
```

```
mysql>
```

# PHP nối kết đến MySQL

---

## ■ PHP nối kết đến MySQL

- Tạo kết nối :

```
$conn = mysql_connect("ip_db_serv", "username", "passwd");
```

- Chọn cơ sở dữ liệu để kết nối

```
$db = mysql_select_db("dbname", $conn);
```

- Thực hiện câu SQL

```
$result = mysql_query("SQL command", $conn);
```

- Lấy 1 dòng kết quả

```
$row = mysql_fetch_array($result);
```

- Đọc giá trị một trường của mẫu tin

```
$val = $row["col-name"];
```

# PHP nối kết đến MySQL

---

## ■ PHP nối kết đến MySQL

- Giải phóng tài nguyên của kết quả

`mysql_free_result($result);`

- Đóng kết nối

`mysql_close($conn);`



# Ví dụ : PHP nối kết đến MySQL để hiển thị bảng Person

---

```
<html>
```

```
<body>
```

```
<?php
```

```
$conn = mysql_connect("127.0.0.1", "nghi", "pass")  
    or die("Could not connect: " . mysql_error());
```

```
$db = mysql_select_db("mydb", $conn)  
    or die("Could not select database");
```

```
$result = mysql_query("SELECT * FROM Person", $conn);
```

```
echo "<TABLE BORDER=1>";
```

```
echo "<TR><TH> LASTNAME </TH> <TH> FIRSTNAME </TH>  
    <TH> ADDRESS </TH> <TH> AGE </TH> </TR>";
```

# Ví dụ : PHP nối kết đến MySQL để hiển thị bảng Person

---

```
while ($row = mysql_fetch_array($result)) {  
    echo "<TR>";  
    echo "<TD> " . $row["lastname"]. " </TD>";  
    echo "<TD> " . $row["firstname"]. " </TD>";  
    echo "<TD> " . $row["address"] . " </TD>";  
    echo "<TD> " . $row["age"] . " </TD>";  
    echo "</TR>";  
}  
echo "</TABLE>";  
  
mysql_free_result($result);  
mysql_close($conn);  
  
?>  
</body>  
</html>
```

## Ví dụ : PHP nối kết đến MySQL để hiển thị bảng Person

---

| LASTNAME     | FIRSTNAME | ADDRESS          | AGE |
|--------------|-----------|------------------|-----|
| Thanh-Nghi   | Do        | 84/40, CMT8      | 31  |
| Nguyen-Khang | Pham      | 43/20, Mau Than  | 27  |
| Nguyen-Binh  | Le        | 12, Nguyen Thong | 18  |
| Trung-Tin    | Nguyen    | 31, Ngo Quyen    | 12  |
| Binh-Minh    | Bui       | C8, Truong Dinh  | 22  |

# Ví dụ : Trang web **insert.html** để thêm một mẫu tin vào bảng Person

---

```
<html>
<head>
<title>Vi du form insert </title>
</head>
<body>
<form method="post" action="insert.php">
Nhap vao ten: <input type="text" name="ln"> <br>
Nhap vao ho: <input type="text" name="fn"> <br>
Nhap vao tuoi: <input type="text" name="age"> <br>
Nhap vao dia chi: <input type="text" name="add"> <br>
<input type="submit" value="Insert">
</form>
</body>
</html>
```

# Ví dụ : Chương trình **insert.php** để thêm một mẫu tin vào bảng Person

---

```
<?php
```

```
$conn = mysql_connect("127.0.0.1", "ngchi", "pass")  
    or die("Could not connect: " . mysql_error());
```

```
$db = mysql_select_db("mydb",$conn)  
    or die("Could not select database");
```

```
$sql = "insert into Person values (" .  
$_POST["ln"] . "," .  
$_POST["fn"] . "," .  
$_POST["add"] . "," .  
$_POST["age"] . ")";
```

## Ví dụ : Chương trình **insert.php** để thêm một mẫu tin vào bảng Person

---

```
//echo "exec: " . $sql . "<br>";
```

```
$res = mysql_query($sql, $conn) or die("error: " . mysql_error());
```

```
//echo "success!!";
```

```
mysql_close($conn);
```

```
?>
```

# Sử dụng MySQLi (thủ tục)

---

## ■ PHP nối kết đến MySQL

- Tạo kết nối :

```
$conn = mysqli_connect ("ip_dbs", "uname", "passwd", "dbname");
```

- Thực hiện câu SQL

```
$result = mysqli_query($conn, "SQL command");
```

- Lấy 1 dòng kết quả

```
$row = mysqli_fetch_array($result);
```

- Đọc giá trị một trường của mẫu tin

```
$val = $row["col-name"];
```

- Đóng kết nối

```
mysqli_close($conn);
```

# Sử dụng MySQLi (thủ tục)

---

```
<html>
<head>
<title>PHP-MySQL for select * from person </title>
</head>
<body>
<?php
// tao ket noi den server mysql
$conn = mysqli_connect("localhost", "user1", "puser1", "mydb");

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
//thuc hien cau truy van
$result = mysqli_query($conn, "select * from Person");
```



# Sử dụng MySQLi (thủ tục)

---

```
echo "<table border=1>";
echo "<tr><th>Ten</th><th>Ho</th><th>Tuoi</th><th>Dia chi</th></tr>";
while ($row = mysqli_fetch_array($result)) {
echo "<tr>";
echo "<td>" . $row["lastname"] . "</td>";
echo "<td>" . $row["firstname"] . "</td>";
echo "<td>" . $row["age"] . "</td>";
echo "<td>" . $row["address"] . "</td>";
echo "</tr>";
}
echo "</table>";
//giai phong tai nguyen
mysqli_free_result($result);
mysqli_close($conn);
?> </body></html>
```

# Sử dụng MySQLi (hướng đối tượng)

---

## ■ PHP nối kết đến MySQL

- Tạo kết nối :

```
$conn = new mysqli ("ip_db_serv", "uname", "passwd", "dbname");
```

- Thực hiện câu SQL

```
$result = $conn->query("SQL command");
```

- Lấy 1 dòng kết quả

```
$row = $result->fetch_assoc();
```

- Đọc giá trị một trường của mẫu tin

```
$val = $row["col-name"];
```

- Đóng kết nối

```
$conn->close();
```

# Sử dụng MySQLi (hướng đối tượng)

---

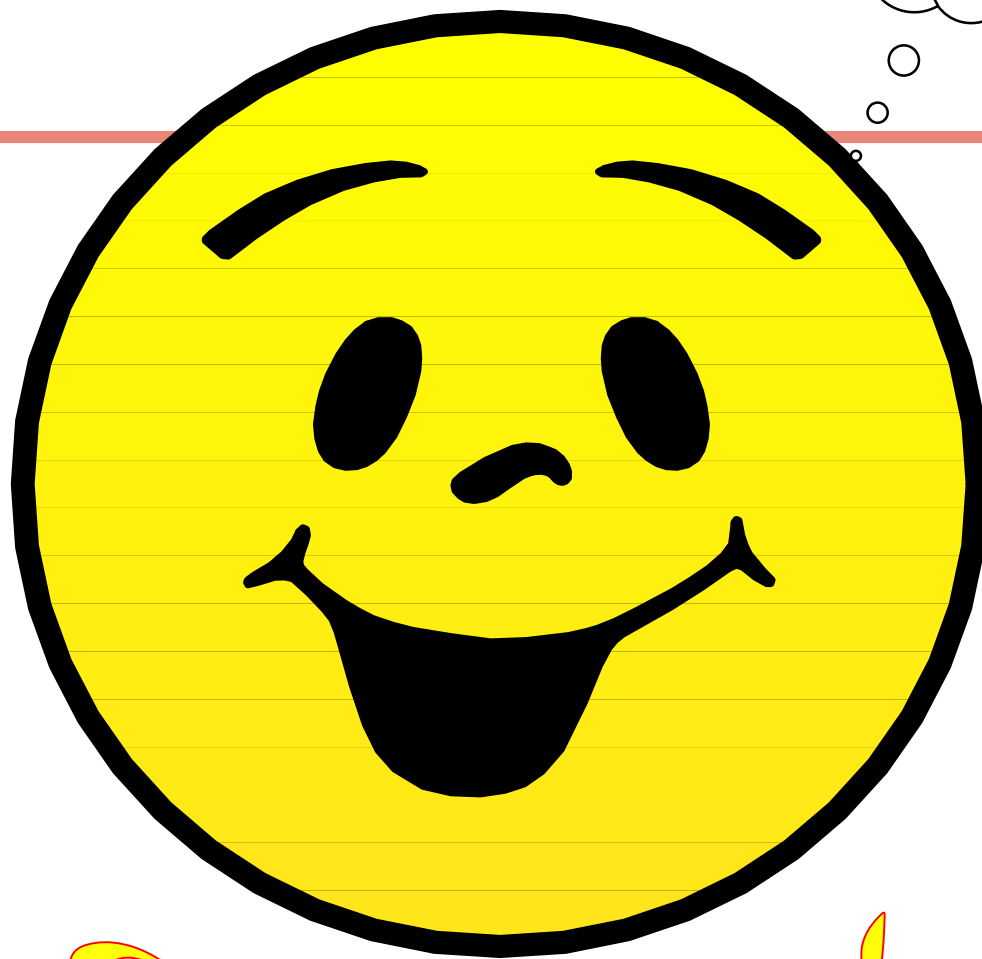
```
<html>
<head>
<title>PHP-MySQL for select * from person </title>
</head>
<body>
<?php
// tao ket noi den server mysql
$conn = new mysqli("127.0.0.1", "nghe", "pass", "mydb");

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

# Sử dụng MySQLi (hướng đối tượng)

---

```
$result = $conn->query("SELECT * FROM Person");
echo "<table border=1>";
echo "<tr><th>Ten</th><th>Ho</th><th>Tuoi</th><th>Dia chi</th></tr>";
while ($row = $result->fetch_assoc()) {
echo "<tr>";
echo "<td>" . $row["lastname"] . "</td>";
echo "<td>" . $row["firstname"] . "</td>";
echo "<td>" . $row["age"] . "</td>";
echo "<td>" . $row["address"] . "</td>";
echo "</tr>";
}
echo "</table>";
$result->free_result();
$conn->close();
?> </body></html>
```



Cám ơn !