

Khoa Công Nghệ Thông Tin Trường Đại Học Đà Lạt

Ngôn ngữ Lập trình PHP

Trần Thị Phương Linh
linhttp@dlu.edu.vn

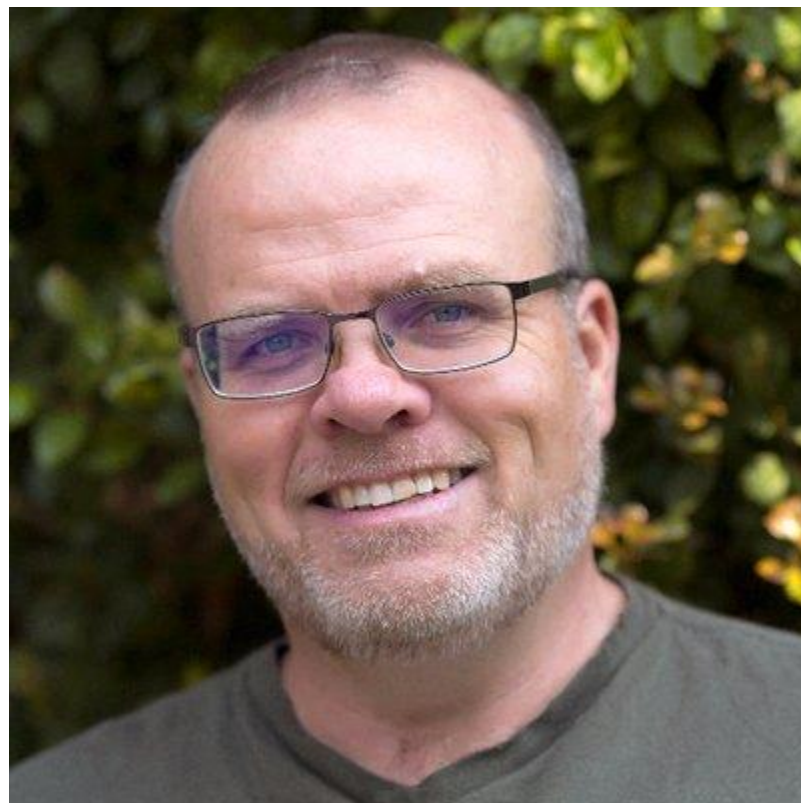
Nội dung

- Giới thiệu về PHP
- Biến, kiểu dữ liệu, phép toán
- Lệnh điều khiển
- Hàm
- PHP kết hợp với forms
- PHP-MySQL

Giới thiệu về PHP

■ PHP là gì ?

- Ra đời năm 1994, dựa trên PHP/FI (Personal Home Page / Forms Interpreter)
- 11/1997: PHP/FI v2.0
- 06/1998: PHP 3.0 - Andi Gutmans và Zeev Suraski
- 05/2000: PHP 4
- 14/07/2005: PHP 5.1 Beta 3
- 28/11/2019: PHP 7.4



Giới thiệu về PHP

■ PHP là gì ?

- PHP là Hypertext Preprocessor
- Ngôn ngữ script chạy trên server
- Mã nguồn mở, miễn phí
- PHP hỗ trợ để làm việc với nhiều hệ QTCSDL khác nhau: MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.
- Chạy trên nhiều platforms (Unix, Linux, Windows)



Giới thiệu về PHP

■ Tại sao chọn PHP?

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

■ Làm thế nào để sử dụng PHP

- Cài web server (Apache, IIS, Xampp, Wamp)
- Cài MySQL
- Cài PHP

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>
```



```
<html>
```

```
<body>
```

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

```
</body>
```

```
</html>
```

Cú pháp PHP

- 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
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 (\$sv \neq \$Sv)
- Kiểu được tính ở thời điểm gán giá trị
- Gán giá trị với =
- Sử dụng ký tự & để 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
```

```
?>
```

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;
$b = 2;
function Sum() {
    global $a, $b;
    $b = $a + $b;
}
Sum();
echo $b;
```

?>

<?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
```

```
?>
```

Biến

■ Phạm vi biến

- Biến hằng:
- Cú pháp: `define('ten_hang', 'gia_tri');`
- Ví dụ :

```
<?php
    define('SDT', '0909090909');
    echo SDT;
?>
```

Kiểu dữ liệu

■ Kiểu dữ liệu cơ bản

- Số nguyên INT: 4 bytes, số có dấu
- Số thực (DOUBLE/FLOAT)
- Luận lý BOOLEAN: TRUE/FALSE
- Chuỗi ký tự STRING

■ Kiểu dữ liệu phức hợp

- Mảng: Array
- Đối tượng: Objects
- Kiểu NULL

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 8 decimal)

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

\$b = 1.234;

\$c = 1.2e3;

\$d = 7E-10;

?>

Kiểu dữ liệu

- Ví dụ: luận lý

```
<?php
```

```
$a = 123; // TRUE
```

```
$b = 0; // FALSE
```

```
$c = '0'; // FALSE
```

```
$d = 'a123b' // TRUE
```

```
$e = null; // FALSE
```

```
$f = "; // FALSE
```

```
?>
```

Kiểu dữ liệu

- Ví dụ: chuỗi

```
<?php
```

```
$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 dữ liệu

`print_r($array_name);`

- Ví dụ: Mảng

```
<?php
    $sinhvien = array('Nguyễn Văn A', 'Nguyễn Văn B');
    print_r($sinhvien);
?>
```

Hoặc:

```
<?php
    $sinhvien = array(0 => 'Nguyễn Văn A', 1 => 'Nguyễn Văn B');
    print_r($sinhvien);
?>
```

Hoặc:

```
<?php
    $sinhvien = array();
    $sinhvien[0] = 'Nguyễn Văn A';
    $sinhvien[1] = 'Nguyễn Văn B';
    print_r($sinhvien);
?>
```

Kiểu dữ liệu

- Truy xuất các phần tử mảng:

`$array_name[key];`

`$array_name[key/name];`

`$array_name[key/name][key/name];`

- Ví dụ:

```
<?php
```

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

```
$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 dữ liệu

- Ví dụ: Mảng kết hợp

```
<?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
```

```
?>
```

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> . |

Phép toán

15-7. Logical Operators

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

Phép toán

15-4. Comparison Operators

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

Phép toán

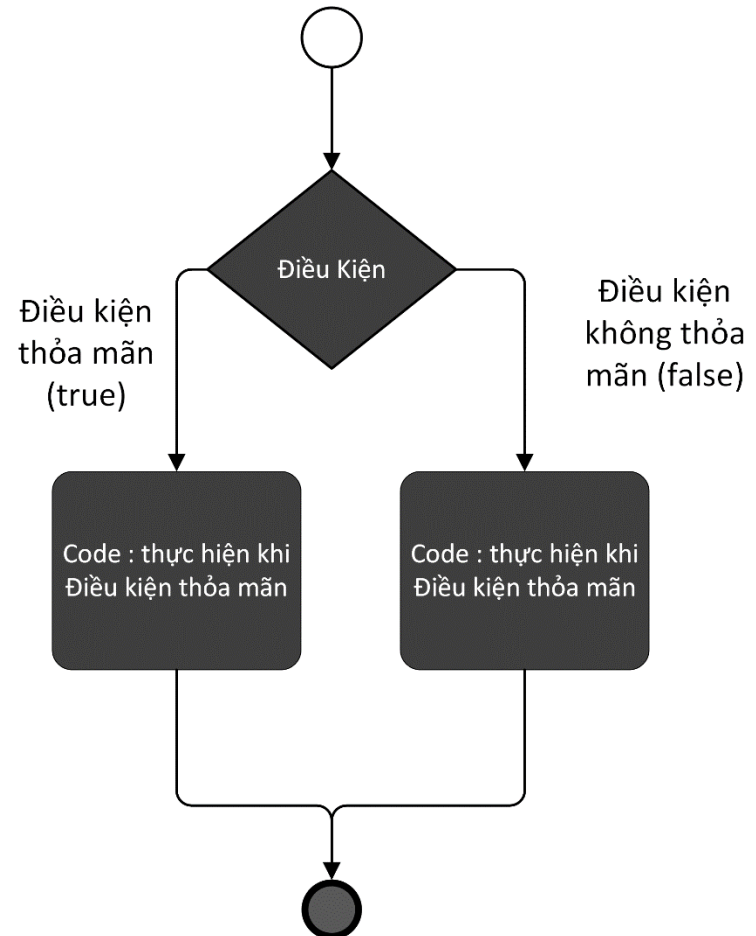
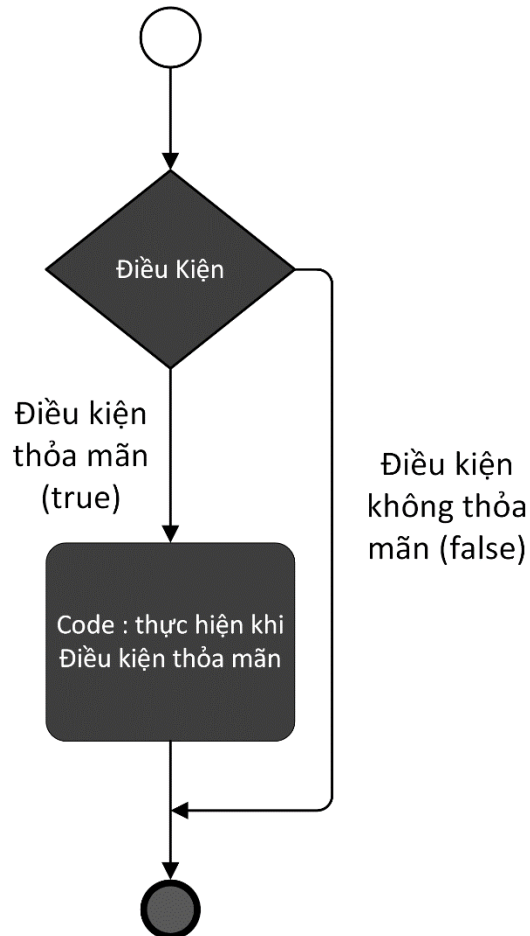
15-8. Array Operators

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

Các lệnh điều khiển trong PHP

- Câu lệnh **if**.
- Câu lệnh **switch**



Lệnh IF

■ 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
```

```
    $nam = 2014;
```

```
    $so_du = $nam % 2;
```

```
    if ($so_du == 0){ echo 'Năm ' . $nam . ' Là Năm Chẵn';}
```

```
    else{ echo 'Năm ' . $nam . ' Là Năm Lẻ';}
```

```
?>
```

Lệnh SWITCH

■ 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;
```

```
}
```


Lệnh 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";
    }
?>
```

Vòng 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 />";  
    }  
?>
```

Vòng 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++;
```

```
    }
```

```
?>
```

Vòng 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);  
?>
```

Vòng lặp FOREACH

- Cú pháp :

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

- Ví dụ :

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

Vòng lặp FOREACH

- Ví dụ :

```
<?php
    $nam = array(1990,1991,1992,1993,1994,1995);
    //Dùng foreach xuất ra các năm trong $nam
    foreach ($nam as $value)
    {
        echo $value;
    }
?>
```

Hàm định nghĩa sẵn trong PHP

| Array Functions | Mô tả |
|----------------------------|--|
| <code>array_diff()</code> | Hàm <code>array_diff()</code> sẽ so sánh sự khác nhau giữa hai hay nhiều mảng |
| <code>array_keys()</code> | Hàm <code>array_keys()</code> sẽ trả về một hoặc một tập hợp các khóa của mảng. |
| <code>array_map()</code> | Hàm <code>array_map()</code> sẽ lặp tất cả các phần tử của mảng và truyền vào hàm callback định nghĩa trước đó của người dùng. |
| <code>array_merge()</code> | Hàm <code>array_merge()</code> sử dụng để nối hai hay nhiều mảng lại thành một mảng. |
| <code>array_pad()</code> | Hàm <code>array_pad()</code> sẽ thêm một hoặc nhiều phần tử có cùng giá trị vào mảng cho đến khi chiều dài của mảng đạt đến chiều dài đã định sẵn. |

Hàm định nghĩa sẵn trong PHP

| String Functions | Mô tả |
|-------------------------------|---|
| <code>wordwrap()</code> | Hàm <code>wordwrap()</code> sẽ thêm một kí tự hoặc một thẻ nào đó vào chuỗi khi đã đếm đủ số kí tự nhất định. |
| <code>ucwords()</code> | Hàm <code>ucwords()</code> sẽ chuyển tất cả các chữ đầu tiên của mỗi từ trong chuỗi thành chữ in hoa. |
| <code>ucfirst()</code> | Hàm <code>ucfirst()</code> sẽ chuyển đổi kí tự đầu tiên trong chuỗi thành in hoa nếu kí tự đó là một chữ cái. |
| <code>trim()</code> | Hàm <code>trim()</code> sẽ loại bỏ khoảng trắng (hoặc bất kì kí tự nào) dư thừa ở đầu và cuối chuỗi. |
| <code>substr()</code> | Hàm <code>substr()</code> sẽ trích xuất một phần của chuỗi |
| <code>substr_replace()</code> | Hàm <code>substr_replace()</code> sẽ thay thế một đoạn của chuỗi bằng một chuỗi nào đó. |

Hàm định nghĩa sẵn trong PHP

| Variable handling Functions | Mô tả |
|-----------------------------|--|
| <code>var_export()</code> | Hàm <code>var_export()</code> dùng để in thông tin của biến truyền vào |
| <code>var_dump()</code> | hàm <code>var_dump()</code> sẽ in ra thông tin của biến gồm kiểu dữ liệu của biến và giá trị. |
| <code>unset()</code> | Hàm <code>unset()</code> sẽ loại bỏ một hoặc nhiều biến được truyền vào. |
| <code>strval()</code> | Hàm <code>strval()</code> sẽ chuyển đổi giá trị của biến truyền vào thành một chuỗi. |
| <code>settype()</code> | Hàm <code>settype()</code> sẽ định kiểu dữ liệu cho biến được truyền vào |
| <code>unserialize()</code> | Hàm <code>unserialize()</code> sẽ chuyển đổi chuỗi đã được hàm <code>serialize()</code> chuyển đổi trước đó về chuỗi ban đầu |
| <code>serialize()</code> | Hàm <code>serialize()</code> sẽ mã hóa giá trị biến truyền vào thành một chuỗi đặc biệt |

Hàm định nghĩa sẵn trong PHP

| Date / Time function | Mô tả |
|--------------------------------------|---|
| <code>timezone_open()</code> | Hàm <code>timezone_open()</code> sẽ khởi tạo một đối tượng <code>timezone</code> . |
| <code>timezone_offset_get()</code> | Hàm <code>timezone_offset_get()</code> sẽ trả về khoảng thời gian chênh lệch giữa múi giờ hiện truyền vào so với múi giờ số 0 tính bằng giây. |
| <code>timezone_name_get()</code> | Hàm <code>timezone_name_get()</code> sẽ lấy tên của múi giờ từ đối tượng <code>DateTimeZone</code> truyền vào. |
| <code>timezone_location_get()</code> | Hàm <code>timezone_location_get()</code> sẽ lấy thông tin về múi giờ từ đối tượng <code>DateTimeZone</code> được truyền vào. |

Hàm định nghĩa sẵn trong PHP

| File system Functions | Mô tả |
|------------------------|---|
| <code>touch()</code> | Hàm <code>touch()</code> sẽ thiết lập thời gian truy cập và sửa đổi nội dung của file truyền vào. |
| <code>tmpfile()</code> | Hàm <code>tmpfile()</code> sẽ tạo file tạm với tên file là duy nhất và quyền đọc và viết(w+). |
| <code>tempnam()</code> | Hàm <code>tempnam()</code> sẽ tạo file với tên file là duy nhất trong năm thư mục truyền vào. |
| <code>symlink()</code> | Hàm <code>symlink()</code> sẽ tạo liên kết tượng trưng tới đối tượng hiện tại với đối tượng chỉ định được truyền vào. |
| <code>rmdir()</code> | Hàm <code>rmdir()</code> sẽ xóa thư mục dựa theo đường dẫn truyền vào. |
| <code>rewind()</code> | Hàm <code>rewind()</code> sẽ đặt lại vị trí con trỏ nội bộ của file về đầu file. |

Hàm định nghĩa sẵn trong PHP

| MySQL Function | Mô tả |
|------------------------------------|--|
| <code>mysqli_stat()</code> | Hàm <code>mysqli_stat()</code> sẽ trả về trạng thái hiện tại của hệ thống. |
| <code>mysqli_set_charset()</code> | Hàm <code>mysqli_set_charset()</code> sẽ thiết lập bảng mã kí tự mặc định khi bạn gửi dữ liệu từ form đến database server. |
| <code>mysqli_select_db()</code> | Hàm <code>mysqli_select_db()</code> sẽ thay đổi database mặc định được thiết lập lúc khởi tạo kết nối MySQL. |
| <code>mysqli_refresh()</code> | Hàm <code>mysqli_refresh()</code> sẽ làm mới các bảng, caches, hoặc các thông tin máy chủ đã sao chép. |
| <code>mysqli_real_connect()</code> | Hàm <code>mysqli_real_connect()</code> sẽ khởi tạo kết nối đến máy chủ MySQL. |

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

■ 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

```
// Số cần kiểm tra
$number = 12;

// gọi đến hàm kiem_tra_so_chan và truyền biến cần kiểm tra vào
// vì hàm kiem_tra_so_chan trả về true/false nên ta có thể đặt nó trong câu điều
// kiện if như thế này
if (kiem_tra_so_chan($number)){
    echo 'Số chẵn';
}
else{
    echo 'Số lẻ';
}

// Hàm kiểm tra số chẵn sẽ trả về true nếu $number là số chẵn và ngược lại.
// biến $number gọi là biến truyền vào hàm, đó chính là biến cần kiểm tra
function kiem_tra_so_chan($number)
{
    if ($number % 2 == 0)
        return true;
    else return false;
}
```

Hàm do người sử dụng định nghĩa

```
// Số cần kiểm tra
$number = 12;

// gọi đến hàm kiểm_tra_so_chan và truyền biến cần kiểm tra vào
kiem_tra_so_chan($number);

// Hàm này có nhiệm vụ xuất ra màn hình số chẵn nếu biến truyền vào ($number)
// là Số chẵn, và ngược lại sẽ xuất ra màn hình là Số lẻ
function kiem_tra_so_chan($number)
{
    if ($number % 2 == 0)
    {
        echo 'Số chẵn';
    }
    else {
        echo 'Số lẻ';
    }
}
```


Hàm do người sử dụng định nghĩa

■ Cách gọi hàm:

- Truyền bằng giá trị/tham trị

```
// Biến
```

```
$a = 1;
```

```
// Hàm tăng giá trị tham số truyền vào lên 1
```

```
function tang_len_1($a)
```

```
{
```

```
    return $a + 1;
```

```
}
```

```
// Xuất giá trị trả về của hàm
```

```
echo tang_len_1($a);
```

```
// Xuất giá trị của biến
```

```
echo $a;
```

Hàm do người sử dụng định nghĩa

■ Cách gọi hàm:

- Truyền bằng tham chiếu/tham biến

```
// Biến  
$a = 1;
```

```
// Hàm tăng giá trị tham số truyền vào lên 1  
function tang_len_1(&$a)  
{  
    $a = $a + 1;  
    return $a;  
}
```

```
// Xuất giá trị trả về của hàm  
echo tang_len_1($a);
```

```
// Xuất giá trị của biến  
echo $a;
```

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.'
```

```
?>
```

PHP + 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>
```

PHP & MySQL

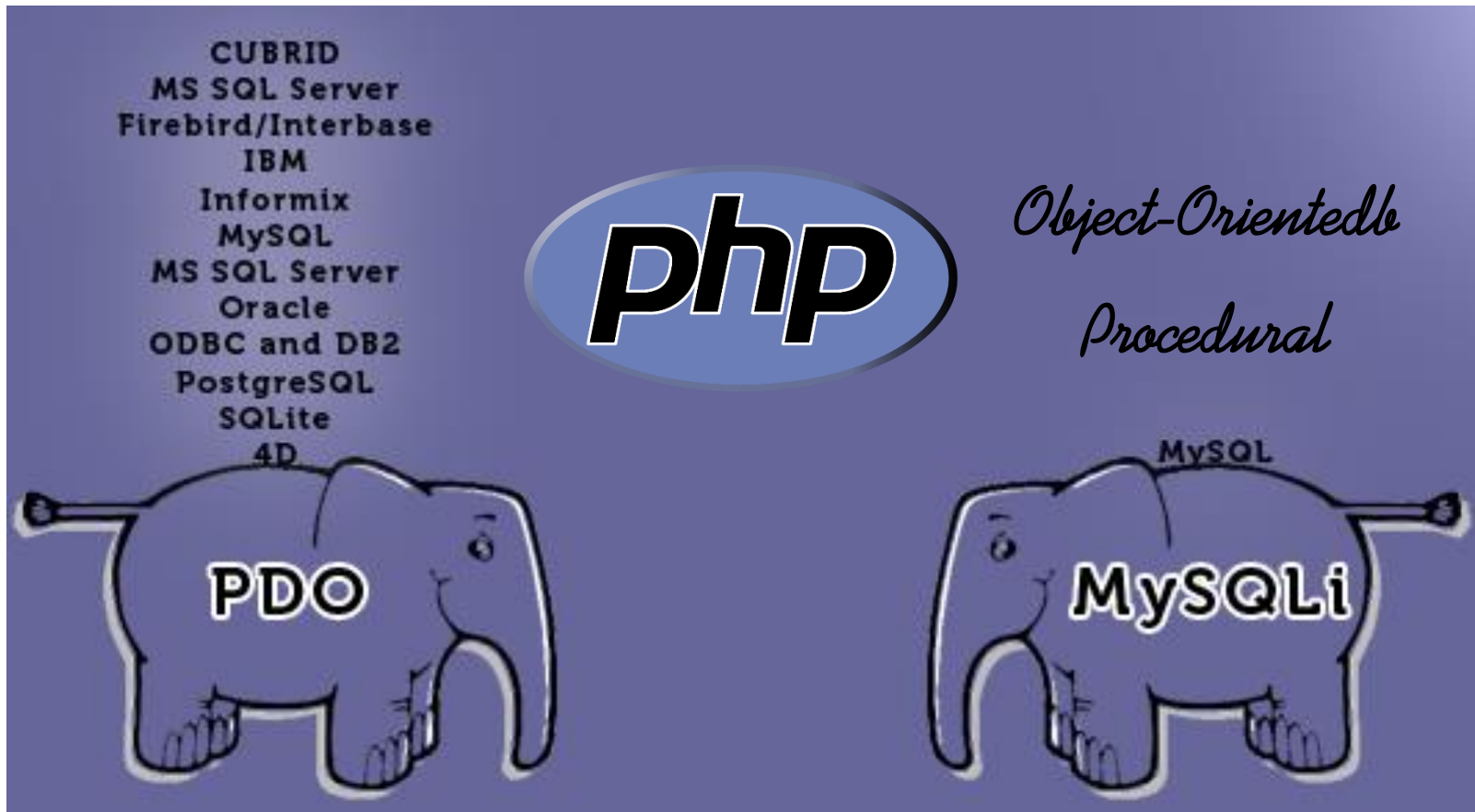
■ MySQL là gì ?

- Hệ quản trị cơ sở dữ liệu
- Hỗ trợ chuẩn SQL
- Phần mềm mã nguồn mở, miễn phí
- Chạy trên nhiều platforms (Windows, Linux, MacOS)
- PHP + MySQL : Web động chạy trên nhiều platforms khác nhau



PHP kết nối đến MySQL

- PHP kết nối đến MySQL



PHP kết nối đến MySQL

Một số hàm MySQLi cơ bản

| Kiểu thủ tục (Procedural) | Kiểu hướng đối tượng (Object-Oriented) | Mô tả |
|---|---|---|
| <code>\$ketnoi = mysqli_connect()</code> | <code>\$ketnoi = new mysqli()</code> | Tạo kết nối database |
| <code>mysqli_connect_error()</code> | <code>\$ketnoi->connect_error</code> | Lỗi kết nối |
| <code>mysqli_close(\$ketnoi)</code> | <code>\$ketnoi->close()</code> | Ngắt kết nối MySQLi |
| <code>\$ketqua = mysqli_query(\$ketnoi, \$sql)</code> | <code>\$ketqua = \$ketnoi->query(\$sql)</code> | Truy vấn table từ <i>\$ketnoi</i> Với <i>\$sql</i> là câu truy vấn |
| <code>mysqli_num_rows(\$ketqua)</code> | <code>\$ketqua->num_rows</code> | Số lượng số hàng có trong table. |
| <code>mysqli_fetch_assoc(\$ketqua)</code> | <code>\$ketqua->fetch_assoc()</code> | Số lượng số hàng có trong table. |

Sử dụng 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 kết nối đến MySQL

- PHP kết nối đến MySQL
 - Giải phóng tài nguyên của kết quả
`mysql_free_result($result);`
 - Đóng kết nối
`mysql_close($conn);`

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

```
<html>
<body>
<?php
$conn = mysql_connect("localhost", "abc", "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 kết nối 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 kết nối 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("localhost", "abc", "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)

- Tạo kết nối :

```
$conn = mysqli_connect("host", "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>" . $row["address"] . "</td>";
echo "</tr>";
}
echo "</table>";
mysqli_free_result($result); //giai phong tai nguyen
mysqli_close($conn);
?> </body></html>
```

Sử dụng MySQLi (hướng đối tượng)

- Tạo kết nối :

```
$conn = new mysqli (“host”, “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("localhost", "abc", "pass", "mydb");
```

```
// Check connection
```

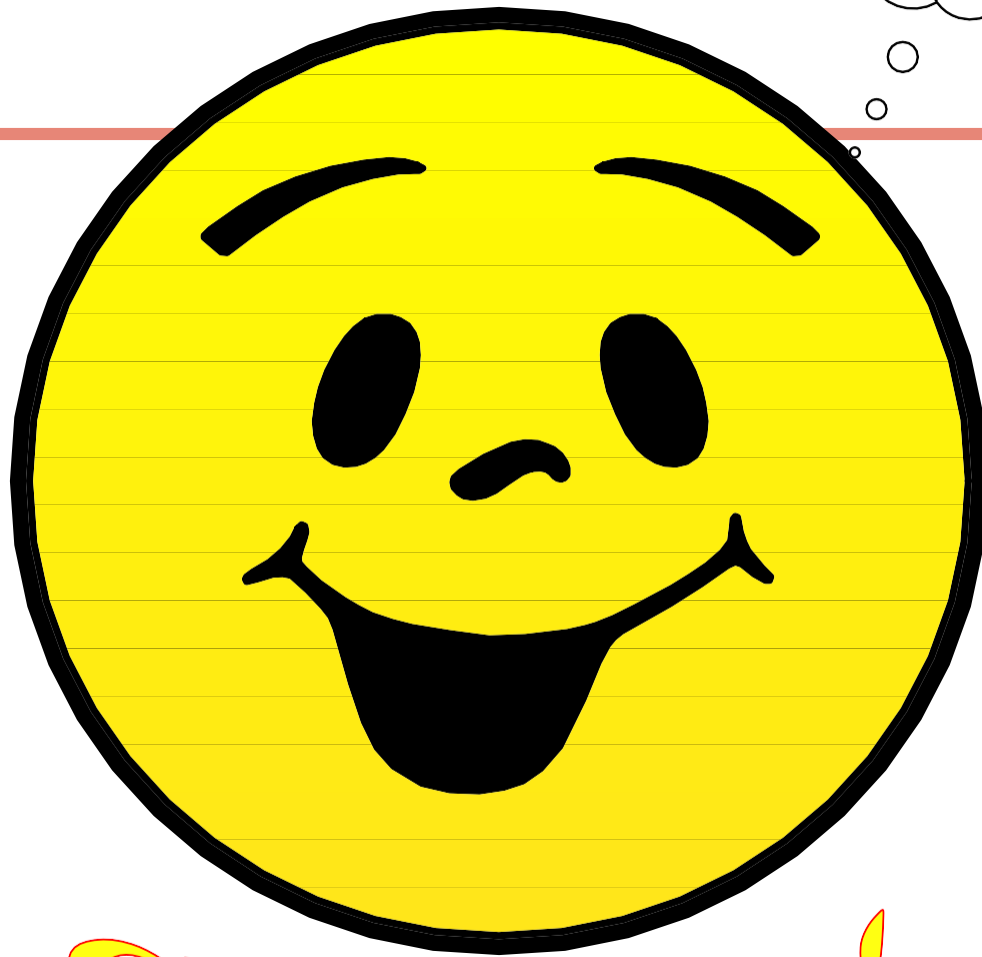
```
if ($conn->connect_error) {
```

```
    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 !

Câu hỏi ôn tập

1. Khái quát các ưu & nhược điểm của PHP?
2. PHP kết nối đến CSDL MySQL bằng cách nào?
So sánh MySQLi và PDO?
3. Trình bày các bước truy vấn CSDL MySQL sử dụng thư viện PDO?
4. Nêu ý nghĩa các biến có sẵn trong PHP?
5. Viết code PHP thực hiện những bài toán sau:
 - Kiểm tra một số là chẵn hay lẻ?
 - Giải phương trình bậc nhất 1 ẩn?
 - Giải phương trình bậc 2?
 - In ra màn hình bảng cửu chương?
 - Kiểm tra một số có phải là nguyên tố hay không?



THANK

YOU