

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
BỘ MÔN MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

# LẬP TRÌNH WEB



## Chương 5: PHP

ThS. NGUYỄN CAO HỒNG NGỌC

# Nội dung



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

# Nội dung



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

# Giới thiệu về PHP



## ■ PHP (PHP: Hypertext Preprocessor)

- Ngôn ngữ script chạy trên server
- Tập tin PHP có thể chứa text, thẻ HTML, script
- Phần mở rộng: \*.php, \*.php3, \*.phtml
- Tập tin PHP sẽ trả kết quả cho trình duyệt thuần HTML
- PHP hỗ trợ để làm việc với nhiều HQT CSDL khác nhau: MySQL, Oracle, PostgreSQL,...
- PHP là phần mềm mã nguồn mở, miễn phí download và sử dụng

# Giới thiệu về PHP (tt)



## ■ Tại sao PHP?

- Chạy trên nhiều platform khác nhau (Windows, Linux, Unix,...)
- Tương thích với hầu hết các web server hiện nay (Apache, IIS,...)
- Download miễn phí tại [www.php.net](http://www.php.net)
- Dễ học và chạy hiệu quả trên phía server

# Giới thiệu về PHP (tt)



- Làm thế nào để sử dụng PHP?
  - Cài web server (Apache hoặc IIS)
    - [www.apache.org](http://www.apache.org)
  - Cài PHP
    - [www.php.net](http://www.php.net)
  - Cài MySQL (nếu sử dụng HQT CSDL MySQL)
    - [www.mysql.com](http://www.mysql.com)

Cài XAMPP chứa cả Apache, PHP và MySQL.

[www.apachefriends.org](http://www.apachefriends.org)

# Giới thiệu về PHP (tt)



## ■ Cú pháp

- Tập tin PHP chứa text, thẻ HTML, script
- Khối lệnh PHP bắt đầu với `<?php` và kết thúc với `?>`
- Câu lệnh PHP kết thúc bởi dấu `;`
- Ví dụ: Trang web hiển thị dòng chữ “Hello World”

```
<!DOCTYPE html>
<html>
<body>
<?php
echo "<p>Hello World</p>";
?>
</body>
</html>
```

# Giới thiệu về PHP (tt)



- Cú pháp (tt)
  - Khỏi lệnh có thể đặt bất cứ nơi nào trong tài liệu
  - Chú thích trong PHP
    - // chú thích 1 dòng đơn
    - # chú thích 1 dòng đơn
    - /\* chú thích đoạn văn bản \*/



# Nội dung



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

# Biến, kiểu dữ liệu



## ■ Biến

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

# Biến, kiểu dữ liệu (tt)



```
<?php
$var="Viet";
$Var="Nam";
echo "$var, $Var"; // output "Viet, Nam"
$4u="invalid"; // invalid
$_4u="valid";
?>
```

# Biến, kiểu dữ liệu (tt)



```
<?php
$var="Bob";
echo $var;
$var=123;
echo $var;
$var=array(1, 2, 3, 4, 5);
for ($i=0; $i<5; $i++)
    echo $var[$i] . "<br/>";
?>
```

# Biến, kiểu dữ liệu (tt)



## ■ Biến có sẵn trong PHP

- `$GLOBALS`: chứa tất cả các biến toàn cục
- `$_SERVER`: tập hợp biến môi trường của web server
- `$_GET`, `$_POST`
- `$_COOKIE`: biến cung cấp HTTP cookies cho script
- `$_FILES`: biến cung cấp HTTP POST file upload cho script
- `$_ENV`: biến cung cấp môi trường cho script
- `$_REQUEST`: cung cấp các `$_GET`, `$_POST`, `$_COOKIE`

# Biến, kiểu dữ liệu (tt)



## ■ Phạm vi biến

- Biến toàn cục (global)
  - Biến được định nghĩa bên ngoài hàm
  - Có thể truy cập tại bất cứ nơi nào của chương trình bằng cách dùng từ khóa **global** trước tên biến hoặc thông qua mảng **\$GLOBALS[]**

# Biến, kiểu dữ liệu (tt)



```
<?php
$a=1; // global scope
$b=2;
function sum() {
    global $a, $b;
    $b = $a + $b;
}
sum();
echo $b;
?>
```

# Biến, kiểu dữ liệu (tt)



```
<?php
$a=1; // global scope
$b=2;

function sum() {
    $GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
}

sum();

echo $b;

?>
```



# Biến, kiểu dữ liệu (tt)



- Phạm vi biến (tt)
  - Biến cục bộ (local)
    - Biến được định nghĩa bên trong hàm
    - Chỉ có thể truy cập bên trong hàm mà biến được định nghĩa
  - Biến tham số (parameter)
    - Là biến cục bộ mà giá trị của nó được truyền khi gọi hàm
  - Biến tĩnh (static)
    - Khi hàm kết thúc, tất cả các biến thông thường sẽ bị xóa, tuy nhiên, biến tĩnh sẽ không bị xóa và giá trị của nó được lưu lại cho lần gọi sau.
    - Sử dụng từ khóa **static** trước tên biến khi khai báo

# Biến, kiểu dữ liệu (tt)



```
<?php
function test() {
    static $a = 10;
    echo "a = " . $a;
    $a++;
}
test(); // output: a = 10
test(); // output: a = 11
?>
```

# Biến, kiểu dữ liệu (tt)



- Kiểu dữ liệu cơ bản
  - Số nguyên
  - Số thực
  - Luận lý: true, false
  - Chuỗi ký tự
- Kiểu dữ liệu phức hợp
  - Mảng
  - Đối tượng
  - Kiểu giả

# Biến, kiểu dữ liệu (tt)



## ■ Chuỗi

- Phép kết nối chuỗi: .
- Trả về chiều dài chuỗi: **strlen()**
- Tìm vị trí bắt đầu của chuỗi trong chuỗi: **strpos()**
- Mã hóa chuỗi: **md5()**
- Tham khảo các hàm trên chuỗi tại [http://www.w3schools.com/php/php\\_ref\\_string.asp](http://www.w3schools.com/php/php_ref_string.asp)

# Mảng



```
<?php // mảng chỉ mục  
$car=array("Toyota", "BMW", "Volvo");  
$car[3]="Saab";  
echo $car[0]." and ".$car[1]." are famous cars."  
?>
```

```
<?php // mảng dạng tự điển  
$ages=array('Quy'=>19, 'Lam'=>18);  
$ages['Viet']=20;  
$ages['Nam']=19;  
?>
```

# Nội dung



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

# Lệnh điều khiển



## ■ if

```
<?php
$d=date("D");
if ($d=="Fri")
    echo "Have a nice weekend!";
else
    echo "Have a nice day!";
?>
```

# Lệnh điều khiển (tt)



## ■ switch

```
<?php
switch ($num) {
    case 1:
        echo "Number 1"; break;
    case 2:
        echo "Number 2"; break;
    default:
        echo "It is not number 1 or number 2";
}
?>
```



# Lệnh điều khiển (tt)



## ■ while

```
<?php
$i=1;
while ($i<=5) {
    echo "The number is " . $i . "<br/>";
    $i++;
}
?>
```

# Lệnh điều khiển (tt)



- do ... while

```
<?php
$i=1;
do {
    echo "The number is " . $i . "<br/>";
    $i++;
} while ($i<5);
?>
```

# Lệnh điều khiển (tt)



## ■ for

```
<?php
for ($i=1; $i<=5; $i++) {
    echo "The number is " . $i . "<br/>";
}
?>
```

# Lệnh điều khiển (tt)



## ■ foreach

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

# Nội dung



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

# Hàm



## ■ Hàm

```
<html><body>

<?php

function writeName($fname) {
    echo $fname . "<br/>";
}

echo "I am "; writeName("Ruby");
echo "My friend is "; writeName("Bob");
?>

</body></html>
```

# Hàm (tt)



```
<html><body>

<?php

function add($x, $y) {
    $total = $x + $y;
    return $total;
}

echo "2 + 7 = " . add(2, 7) . " <br/>";

?>

</body></html>
```

# Nội dung



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



```
<html><body>                                     input_form.html
<form action= "input_xuly.php" method="POST">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body></html>
```

```
<html><body>                                     input_xuly.php
Welcome <?php echo $_POST['fname']; ?>!<br />
You are <?php echo $_POST['age']; ?> years
old.
</body></html>
```

# PHP form (tt)



- Nên kiểm tra tính hợp lệ của dữ liệu do người dùng nhập vào ở trình duyệt nếu có thể (ví dụ sử dụng javascript) => nhanh và giảm tải cho server.

# Phương thức GET



- Nếu dùng phương thức GET thì tất cả các biến và giá trị sẽ hiển thị trong URL
  - Ví dụ:  
`http://www.w3schools.com/welcome.php?fname=Peter&age=37`
- Không nên sử dụng với password và thông tin nhạy cảm
- Không thích hợp với dữ liệu lớn (tối đa khoảng 2000 ký tự)
- Bookmark được

# Phương thức POST



- Nếu sử dụng phương thức POST thì thông tin truyền đi sẽ ẩn và không giới hạn về kích thước (mặc định là 8MB nhưng có thể thay đổi tăng thêm bằng cách thay đổi `post_max_size` trong `php.ini`)
- Không bookmark được
- Biến `$_REQUEST` có thể được sử dụng để nhận dữ liệu gửi bằng phương thức GET và POST
- `$_REQUEST` chứa cả `$_GET`, `$_POST`, `$_COOKIE`

# Nội dung



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

# PHP và MySQL (Tạo và đóng nối kết)



```
<?php // hướng thủ tục
$con = mysqli_connect("ten_server", "taikhoan",
"matkhau", "ten_database");
mysqli_set_charset($con, "utf8");
// some code
mysqli_close($con);
?>
```

```
<?php // hướng đối tượng
$con = new mysqli("ten_server", "taikhoan",
"matkhau", "ten_database");
$con->set_charset("utf8");
// some code
$con->close();
?>
```

```
<?php // hướng thủ tục
$con = mysqli_connect("ten_server", "taikhoan",
"matkhau", "ten_database");
mysqli_set_charset($con, "utf8");
mysqli_query($con, "INSERT INTO Persons (FirstName,
LastName, Age) VALUES ('Ngoc', 'Nguyen', 27)");
mysqli_close($con);
?>
```

```
<?php // hướng đối tượng
$con = new mysqli("ten_server", "taikhoan",
"matkhau", "ten_database");
$con->set_charset("utf8");
$con->query("INSERT INTO Persons (FirstName,
LastName, Age) VALUES ('Ngoc', 'Nguyen', 27)");
$con->close();
?>
```

# PHP và MySQL

## (Làm việc với form)



### ■ input.html

```
<html><body>
<form action="input_xuly.php" method="POST">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body></html>
```



```
<?php // hướng thủ tục - INSERT  
// lấy dữ liệu về  
$ten=$_POST['firstname'];  
$ho=$_POST['lastname'];  
$tuoi=$_POST['age'];  
// làm việc với CSDL  
$con = mysqli_connect("ten_server",  
"taikhoan", "matkhau", "ten_database");  
mysqli_set_charset($con,"utf8");  
$sql="INSERT INTO Persons (FirstName,  
LastName, Age) VALUES ('$ten', '$ho', $tuoi)";  
// echo $sql;  
mysqli_query($con, $sql);  
mysqli_close($con);  
?>
```

```
<?php // hướng đối tượng - INSERT
// lấy dữ liệu về
$ten=$_POST['firstname'];
$ho=$_POST['lastname'];
$tuoi=$_POST['age'];
// làm việc với CSDL
$con = new mysqli("ten_server", "taikhoan",
"matkhau", "ten_database");
$con->set_charset("utf8");
$sql="INSERT INTO Persons (FirstName,
LastName, Age) VALUES ('$ten', '$ho', $tuoi)";
// echo $sql;
$con->query($sql);
$con->close();
?>
```

```
<?php // hướng thủ tục - SELECT
```

```
$con = mysqli_connect("ten_server", "taikhoan",  
"matkhau", "ten_database");
```

```
$sql="SELECT FirstName, LastName, Age FROM  
Persons WHERE FirstName='Ngoc' ";
```

```
$result=mysqli_query($con, $sql);
```

```
if (mysqli_num_rows($result) > 0) {  
    while($row = mysqli_fetch_assoc($result)) {  
        echo $row['FirstName']. " " .  
$row['LastName']. " " . $row['Age']. "<br>";  
    }  
}
```

```
} else {  
    echo "0 results";  
}
```

```
mysqli_close($con);
```

```
?>
```

```
<?php // hướng đối tượng - SELECT

$con = new mysqli("ten_server", "taikhoan",
"matkhau", "ten_database");

$con->set_charset("utf8");

$sql="SELECT FirstName, LastName, Age FROM
Persons WHERE FirstName='Ngoc' ";

$result = $con->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo $row['FirstName']. " " .
$row['LastName']. " " . $row['Age']. "<br/>";
    }
} else { echo "0 results"; }

$con->close();

?>
```

# PHP và MySQL

## (Upload tập tin)



### ■ upload.html

```
<html><body>

<form action="upload_xuly.php" method="POST"
enctype="multipart/form-data">

Full name: <input type="text" name="fullname" />

Avatar: <input type="file" name="avatar" />

<input type="submit" />

</form>

</body></html>
```

# PHP và MySQL

## (Upload tập tin)



```
<?php // hướng đối tượng - UPLOAD FILE

$con = new mysqli("ten_server", "taikhoan", "matkhau",
"ten_database");

$con->set_charset("utf8");

$hoten=$_POST['fullname'];

$duongdan="./thumuc/" . $_FILES['avatar']['name'];

$sql="INSERT INTO Members (FullName, AvatarLink) VALUES
('$hoten', '$duongdan') ";

$con->query($sql);

move_uploaded_file($_FILES['avatar']['tmp_name'],
$duongdan);

$con->close();

?>
```

# Nội dung



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

# Cookie



- Thường được dùng 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)*
- Phải đặt trước thẻ `<html>`
- Truy xuất thông qua biến `$_COOKIE`



# Cookie (tt)



```
<?php
setcookie("user", "Ruby", time()+3600);
?>

<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>
```

# Cookie (tt)



- Hàm *isset()* kiểm tra xem biến đã được tạo chưa

```
<html><body>

<?php

if (isset($_COOKIE['user']))

    echo "Welcome " . $_COOKIE['user'] . "!<br />";

else

    echo "Welcome guest!<br />";

?>

</body></html>
```

# Cookie (tt)



- Xóa cookie

```
<?php  
// set the expiration date to one hour ago  
setcookie("user", "", time()-3600);  
?>
```

# Session



- Thường được dùng để lưu trữ thông tin một phiên làm việc của người dùng
- Lưu trữ thông tin người dùng ở server tạm thời và sẽ bị xóa khi người dùng đóng website
- Khởi tạo bằng hàm *session\_start()* và phải đặt trước thẻ `<html>`
- Truy xuất thông qua biến `$_SESSION`



```
<?php
session_start();
// store session data
if (isset($_SESSION['views']))
    $_SESSION['views']=$_SESSION['views']+1;
else
    $_SESSION['views']=1;
?>

<html><body>

<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];
?>

</body></html>
```

# Session (tt)



## ■ Xóa session

```
<?php
// hủy dữ liệu của biến views
unset($_SESSION['views']);
?>
```

```
<?php
// hủy toàn bộ dữ liệu của session
session_destroy();
?>
```

# Cookie hay Session



- Cookie giữ thông tin người dùng trên trình duyệt người dùng đến khi bị xóa.
  - Dùng cookie để lưu thông tin người dùng đã đăng nhập thành công => không cần đăng nhập mỗi khi trở lại website cũ
  - Cần nhắc khi dùng với giỏ hàng vì người dùng có thể mặc định không cho trình duyệt dùng cookie => không thể mua hàng
- Session lưu thông tin phía server nhưng bị xóa khi người dùng đóng website.
  - Người dùng sẽ phải đăng nhập mỗi lần sử dụng website.
- Tùy tình hình cụ thể mà có thể phối hợp sử dụng cả cookie và session.

# Server Side Includes



## ■ SSI

- Chèn đoạn code chương trình của một file vào file khác trước khi thực thi
- Sử dụng hàm *require* hoặc *include*
- Ví dụ

```
<html>
<body>
<?php require 'menu.php'; ?>
<p> Some text </p> <p> Another text </p>
<?php include 'footer.php'; ?>
</body>
</html>
```



# HẾT

