It.Kma

PHP

Khoa CNTT - HvKTMM

PHP BASIC – Phần 1

1. Giới thiệu - cách thức cài đặt về PHP
2. Cách thức sử dụng PHP
3. Hiển thị, biến và kiểu dữ liệu trong PHP

1. Giới thiệu – cách thức cài đặt

- Là một ngôn ngữ kịch bản chạy trên server, được dùng để xây dựng các ứng dụng web động.
- Được ra đời năm 1994 do Rasmus Lerdorf
- PHP được (công bố ngày 02/02/2012)
- Mã PHP trong một kịch bản có thể là các câu truy vấn csdl, tạo ảnh, đọc/ghi file…
- Kết quả sau khi thực hiện mã PHP được kết hợp với HTML và gửi đến cho trình duyệt, do đó người dùng không thể xem được mã của PHP từ trình duyệt.
- PHP được biên dịch như một module của Apache (Apache là một web-server được sử dụng phổ biến trên thế giới)

1. Giới thiệu – cách thức cài đặt

- Mã nguồn mở (open source code)
- Miễn phí, dễ dàng download từ Internet.
- Ngôn ngữ rất dễ học, dễ viết.
- PHP có thể chạy trên các môi trường (platforms) khác nhau như (Windows, Linux, Unix, etc.)
- Có thể kết nối với nhiều DDMS một các đơn giản như MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL

1. Cách thức cài đặt

- Môi trường chạy PHP XAMPP, WAMP,...
- Xampp là chương trình tạo máy chủ Web (Web Server) trên máy tính cá nhân (Localhost)
- Được tích hợp sẵn Apache, PHP, MySQL, FTP Server
- Tải Xampp (https://www.apachefriends.org)
- Cài Đặt Xampp

1. Cách thức cài đặt

- Khi cài đặt cài XAMPP, các chương trình sau cài được tự động:
  - **Apache**: là một chương trình dành cho máy server đảm nhận việc giao tiếp bằng giao thức HTTP.
  - **PHP**: là một "plugin" của Apache, giúp Apache biết cách làm việc với các trang PHP.
  - **MySQL**: là hệ quản trị cơ sở dữ liệu
  - **PhpMyAdmin**: Là một tool giúp quản trị cơ sở dữ liệu trong MySQL

2. Sử dụng PHP

- PHP bao gồm:
  - Các file PHP có thể chứa text, HTML tags và các đoạn scripts
  - Các PHP sau khi Thông dịch trả về cho trình duyệt là các trang HTML
  - File PHP có phần mở rộng là ".php", ".php3", hoặc ".phtml"
- Nhúng mã PHP vào HTML
  - Mã PHP vào vị trí bất kì trong trang HTML
  - Một khối mã lệnh PHP được đặt giữa <?php và ?>
  - Một cấu trúc lệnh thông thường của PHP có thể được tách làm nhiều phần, mỗi phần đặt giữa <?php…?>
  - Kết quả do đoạn lệnh PHP thực thi được đưa vào vị trí mà đoạn lệnh PHP đang chiếm chỗ.

3. Hiển thị, biến và kiểu dữ liệu trong PHP

- Cú pháp:
  - Kết thúc lệnh bằng dấu ;
- Comment
  - Từng dòng: // hoặc #
  - Block: /* */
- Biến:
  - Trong PHP biến được bắt đầu bằng $
  - Biến cục bộ, biến toàn cục và biến tĩnh

3. Hiển thị, biến và kiểu dữ liệu trong PHP

- Sử dụng print hoặc echo để hiển thị
- Bên trong có chèn được các thẻ của html

```
<?php
echo "<h2>Bắt đầu vào học
PHP!</h2>";
echo "Hello world!<br>";
echo “Tôi muốn học giỏi lập trình
web với PHP!<br>";
echo "This ", "string ", "was
", "made ", "with multiple
parameters.";
?>
```

Hiển thị biến

```
<?php
$txt1 = “Học PHP";
$txt2 = “Khoa CNTT";
$x = 10;
$y = 2;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2
. "<br>";
echo $x + $y;
?>
```

3. Hiển thị, biến và kiểu dữ liệu trong PHP

- Kiểu dữ liệu
  - String
  - Integer
  - Float
  - Boolean
  - Array
  - Object
  - Null
- Kiểm tra kiểu và giá trị sử dụng hàm: var_dump()

Kiểu String

- Một số hàm hay sử dụng
  - Strlen(): Đếm số ký tự
  - Str_word_count(): Đếm số từ
  - Strrev(): Viết ngược chữ
  - Strpos(): Trả về vị trí của từ được tìm thấy đầu tiên trong đoạn, nếu không thấy trả về False.
  - Str_replace("Từ thay ra", "từ được thay vào", "choỗi ban đầu"): Thay thế trong chuỗi

3. Hiển thị, biến và kiểu dữ liệu trong PHP
Kiểu Number

- Kiểu interger: cả số nguyên âm, nguyên dương
- Kiểu float: có dấu chấm thập phân: 2.0, 10.35, 7.43E+5
- Kiểu Infinity
- Giá trị không phải là số trả về NaN
- Số + chuỗi số => vẫn cộng bình thường
- Số + chuỗi chữ => sai có thông báo
- Kiểm tra có phải là số hay không: is_numeric()

```
$x = acos(8);
var_dump($x);
```

```
$x = “2020";
var_dump(is_numeric($x));
```

- Ép kiểu số thực về số nguyên: (int) hoặc (integer)$x

Kiểu Number – Một số hàm toán học

- Pi()
- Min(), max()
- Abs(), sqrt(), round(), rand(), rand(10,100)

3. Hằng trong PHP

- Để khởi tạo một constant sử dụng hàm define()

```
define(name, value, case-insensitive)
```

  - Name: tên cả constant
  - Value: giá trị của constant
  - Case-insensitive: xem có phân biệt chữ hoa hay chữ thường hay không. Mặc định là False.

```
<?php
define("cntt", "Khoa công nghệ thông tin!");
echo cntt;
?>
```

  - Constant có thể thay thế string, number, array
  - Constant sử dụng cho cả biết toàn cục và hàm cục bộ

Các toán tử trong PHP

Số học	Gán	So sánh	Tăng giảm	Logic
+	X=y	==	++$x	And
=	X+=y	===	$x++	Or
*	X-=y	!= <>	--$x	Xor
/	X*=y	< <= > >=	$x--	!
%	x/=y	<=>		&&
** (mũ)	X%=y			||

❖Echo ($x<=>$y) => nếu x<y trả về -1, = trả về =0, > trả về +1

Các toán tử trong PHP

❖Toán tử mảng

+	$x+$y	Gộp 2 mảng lại thành 1
==	$x==$y	So sánh 2 mảng có bằng nhau hay không. Trả về true nếu $x và $y có cùng key/value
===	$x===$y	Trả về true nếu $x và $y có cùng cặp key/value theoc ùng một thứ tự và cùng kiểu.
!=	$x!=$y	
<>	$x<>$y	
!==	$x!==$y	

Các toán tử trong PHP

❖Toán tử gán có điều kiện

?:	$x=expr1?expr2:expr3	Giá trị $x = expr2 nếu expr1 đúng, nếu sai thì = expr3
??	$x = expr1?? expr2	$x=expr1 nếu expr1 là tồn tại và khác NULL. Nếu không thì $x=expr2

```
<?php
  // if empty($user) = TRUE, set $status = "anonymous"
  echo $status = (empty($user)) ? "anonymous" : "logged in";
  echo("<br>");

  $user = “Le Thuan";
  // if empty($user) = FALSE, set $status = "logged in"
  echo $status = (empty($user)) ? "anonymous" : "logged in";
?>
```

```
anonymous
logged in
```

```
<?php
  // variable $user is the value of $_GET['user']
  // and 'anonymous' if it does not exist
  echo $user = $_GET["user"] ?? "anonymous";
  echo("<br>");
?>
```

PHP BASIC – Phần 2

1. Câu lệnh điều kiện - Vòng lặp
2. Sử dụng hàm và mảng
3. Biến globals - superglobals

Câu lênh IF

- Câu lệnh IF sử dụng giống các ngôn ngữ lập trình khác

```
if (điều kiện) {
  Đoạn mã nếu điều kiện đúng;
}
```

```
if (Điều kiện) {
  Đoạn mã nếu điều kiện đúng;
} else {
  Đoạn mã nếu điều kiện sai
}
```

```
if (Điều kiện1) {
   Đoạn mã nếu điều kiện1 đúng;
} elseif (Điều kiện2) {
  Nếu điều kiện 1 sai và điều kiện
2 là đúng;
} else {
  Nếu tất cả các điều kiện đều
sai;
}
```

Câu lênh Switch

❖Cú pháp của câu lệnh switch

```php
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

```
switch (n) {
  case label1:
    code to be executed if n=label1;
    break;
  case label2:
    code to be executed if n=label2;
    break;
  case label3:
    code to be executed if n=label3;
    break;
    ...
  default:
    code to be executed if n is different from all labels;
}
```

Vòng lặp while – do while

❖Thực hiện while khi điều kiện đúng

```
<?php
$x = 0;

while($x <= 100) {
  echo "The number is: $x <br>";
  $x+=10;
}
?>
```

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

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

```
<?php
$x = 1;

do {
  echo "The number is: $x <br>";
  $x++;
} while ($x <= 5);
?>
```

Vòng lặp for – Foreach

Cú pháp vòng lặp for

```
for (init counter; test counter; increment counter) {
  code to be executed for each iteration;
}
```

Cú pháp vòng lặp foreach: làm việc với mảng với mỗi key/value trong mảng.

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

```
<?php
$colors
= array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
} ?>
```

```
<?php
$age
= array("Peter"=>"35", "Ben"=>"
37", "Joe"=>"43");

foreach($age as $x => $val) {
  echo "$x = $val<br>";
}
?>
```

2. Hàm và mảng

❖Hàm không có đối số hoặc có đối số

```
function ten_ham() {
  mã thực thi trong hàm;
}
```

❖Hàm trả về bằng return

```php
<?php
function sum(int $x, int $y) {
  $z = $x + $y;
  return $z;
}

echo "5 + 10 = " . sum(5, 10)
. "<br>";
echo "7 + 13 = " . sum(7, 13)
. "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
```

```php
<?php
function familyName($fname, $year)
{
  echo “Tên là $fname. Năm sinh
$year <br>";
}

familyName(“Linh", "1999");
familyName(“Nam", “2000");
familyName(“Mai", “2001");
?>
```

2. Hàm và mảng

- Mảng lưu trữ nhiều giá trị trong một tên biến
  - Mảng có thể truy xuất theo chỉ số
  - Truy xuất theo key
  - Trong mảng có thể chứa một hoặc nhiều mảng
  - Có thể gán mảng trực tiếp hoặc thông qua index

```php
<?php
$cars
= array("Volvo", "BMW", "Toyota");
echo count($cars);
echo "I like " . $cars[0] . ", " . $cars[1] . " and
" . $cars[2] . ".";
?>

$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
```

```php
<?php
$cars
= array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++)
{
  echo $cars[$x];
  echo "<br>";
}
?>
```

2. Hàm và mảng

❖Sử dụng mảng liên kết

```
$age = array("Peter"=>"35",
"Ben"=>"37", "Joe"=>"43");
```

```
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
```

```
<?php
$age
= array("Peter"=>"35", "Ben"=>"37"
, "Joe"=>"43");
echo "Peter is " . $age['Peter']
. " years old.";
?>
```

```
<?php
$age
= array("Peter"=>"35", "Ben"=>"37"
, "Joe"=>"43");

foreach($age as $x => $x_value) {
  echo "Key=" . $x . ", Value=" .
$x_value;
  echo "<br>";
}
?>
```

2. Hàm và mảng

❖Mảng nhiều chiều

```
<?php
echo $cars[0][0].": In stock:
".$cars[0][1].", sold:
".$cars[0][2].".<br>";
echo $cars[1][0].": In stock:
".$cars[1][1].", sold:
".$cars[1][2].".<br>";
echo $cars[2][0].": In stock:
".$cars[2][1].", sold:
".$cars[2][2].".<br>";
echo $cars[3][0].": In stock:
".$cars[3][1].", sold:
".$cars[3][2].".<br>";
?>
```

```
$cars = array (
  array("Volvo",22,18),
  array("BMW",15,13),
  array("Saab",5,2),
  array("Land Rover",17,15)
);
<?php
for ($row = 0; $row < 4; $row++) {
  echo "<p><b>Row number
$row</b></p>";
  echo "<ul>";
  for ($col = 0; $col < 3; $col++)
{
    echo "<li>".$cars[$row][$col].
"</li>";
  }
  echo "</ul>";
}
?>
```

2. Hàm và mảng

Sắp xếp mảng

- Sort(): Sắp xếp theo thứ tự tăng dần
- Rsort(): Sắp xếp theo thứ tự giảm dần
- Asort(): Sắp xếp các mảng kết hợp theo thứ tự tăng dần, theo value
- Ksort(): Sắp xếp các mảng kết hợp theo thứ tự tăng dần, theo key
- Arsort(): Sắp xếp các mảng kết hợp theo thứ tự giảm dần, theo value
- Krsort(): Sắp xếp các mảng kết hợp theo thứ tự giảm dần, theo key

```
<?php
$numbers = array(4, 6, 2, 22, 11);
sort($numbers);
$arrlength = count($numbers);
for($x = 0; $x < $arrlength; $x++) {
  echo $numbers[$x];
  echo "<br>";
}
?>
```

3. Biến global và superglobal

- Biến subperglobal là biến luôn được truy cập, bất kể phạm vi ở đâu từ bất kỳ function, class hoặc file nào mà không cần thao tác gì đặc biệt
  - $GLOBALS
  - $_SERVER
  - $_REQUEST
  - $_POST
  - $_GET
  - $_FILES
  - $_ENV
  - $_COOKIE
  - $_SESSION

3. Biến global và superglobal

- $GLOBALS: biến được truy cập ở bất cứ đâu trong đoạn scripts PHP.
  - Sử dụng: $GLOBALS[index] với index là tên biến

```php
<?php
$x = 75;
$y = 25;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x']
+ $GLOBALS['y'];
}

addition();
echo $z;
?>
```

3. Biến global và superglobal

- $_SERVER: chứa thông tin về các header, path và script locations.

```php
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

3. Biến global và superglobal

- $_SERVER: Danh sách một số phần tử quan trọng trong biến này:

$_SERVER['PHP_SELF']	**Trả về tên file của file đang chạy**
$_SERVER['GATEWAY_INTERFACE']	rả về phiên bản của Common Gateway Interface(CGI) của máy chủ đang dùng
$_SERVER['SERVER_ADDR']	
$_SERVER['SERVER_NAME']	Trả về tên của máy chủ
$_SERVER['SERVER_SOFTWARE']	
$_SERVER['SERVER_PROTOCOL']	
$_SERVER['REQUEST_METHOD']	
$_SERVER['REQUEST_TIME']	
$_SERVER['QUERY_STRING']	
$_SERVER['HTTP_ACCEPT']	Chấp nhận trả về tiêu đề từ các yêu cầu hiện tại
$_SERVER['HTTP_ACCEPT_CHARSET']	
$_SERVER['HTTP_HOST']	Trả về tiêu đề chủ từ yêu cầu hiện tại

3. Biến global và superglobal

❖$_SERVER: Danh sách một số phần tử quan trọng trong biến này:

$_SERVER['HTTP_REFERER']	Trả về URL hoàn chỉnh của trang hiện tại
$_SERVER['HTTPS']	Được kịch bản truy vấn thông qua một giao thức HTTP an toàn
$_SERVER['REMOTE_ADDR']	Trả về địa chỉ IP từ nơi người dùng đang xem một trang hiện hành
$_SERVER['REMOTE_HOST']	Trả về tên máy chủ từ nơi người dùng đang xem một trang hiện hành
$_SERVER['REMOTE_PORT']	Trả về các cổng được sử dụng trên máy tính của người dùng để giao tiếp với máy chủ web
$_SERVER['SCRIPT_FILENAME']	Trả lại tên đường dẫn tuyệt đối của kịch bản hiện đang thực hiện
$_SERVER['SERVER_ADMIN']	
$_SERVER['SERVER_PORT']	Trả về các cổng trên máy chủ được sử dụng bởi các máy chủ web để liên lạc(such as 80)
$_SERVER['SERVER_SIGNATURE']	Trả về phiên bản máy chủ và tên máy chủ ảo được thêm vào các trang máy chủ tạo ra
$_SERVER['PATH_TRANSLATED']	Trả về hệ thống tập tin dựa trên đường dẫn đến kịch bản hiện tại
$_SERVER['SCRIPT_NAME']	Trả về đường dẫn của kịch bản hiện tại
$_SERVER['SCRIPT_URI']	Trả về URI của trang hiện tại

3. Biến global và superglobal

❖$_REQUEST: lấy dữ liệu được gửi lên từ form HTML

```
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>
</body>
```

3. Biến global và superglobal

- $_POST: sử dụng để lấy dữ liệu gửi lên từ form HTML thông qua method = “post”. Lúc gửi không thể hiện gì thông qua đường dẫn URL
  - Vô hình đối với mọi người
- $_GET: sử dụng để lấy dữ liệu gửi lên từ form HTML thông qua method = “get”. Dữ liệu được gửi trong URL
  - Nên dùng với dữ liệu không nhạy cảm vì hiển thị với tất cả mọi người
- Cả POST và GET đều tạo một mảng gồm key/value để chứa giá trị. array(key1 => value1, key2 => value2, key3 => value3, ...))
  - Get: mảng các biến được truyền thông qua tham số URL
  - Post: truyền thông qua http POST không hiển thị trên URL

PHP FORM– Phần 3

1. Tạo form HTML
2. Kiểm tra tính đúng đắn của dữ liệu
3. Gửi dữ liệu xử lý bên Server

Quy trình

- Gửi các loại dữ liệu thường dùng từ form HTML đã được học:
  - Text (thông thường, mật khẩu, email, điện thoại….)
  - Radio button
  - Checkbox
  - Select
- Các trang đăng nhập, đăng ký, bổ xung thông tin, … (đã có form ở file (JS_ValidationForm.html + video hướng dẫn sau khi học xong JavaScript)
- Kiểm tra dữ liệu:
  - JavaScript kiểm tra trước khi gửi lên server (biểu thức chính quy)
  - PHP kiểm tra (biểu thức chính quy)
- Nhận và Xử lý dữ liệu trên Server (PHP)

Biểu thức chính quy PHP

- Giống nhau trên tất cả các ngôn ngữ lập trình
- Chỉ khác nhau ở hàm gọi
- Sử dụng hàm preg_match() để kiểm tra hay so sánh
  - **pattern**: được gọi là chuỗi Regular Expression (biểu thức chính quy)
  - **subject**: chuỗi so khớp với pattern
  - **matches**: kết qua so khớp

Modifier	Description
i	So sánh không phân biệt chữ hoa chữ thường (case-insensitive)
g	So sánh toàn bộ chuỗi dù trong chuỗi có xuống hàng (global)
m	So sánh nhiều dòng (multiline)

```
//Kiểm tra chuỗi chỉ là ký tự in HOA
$pattern = '/[A-Z]/';
$subject = 'ABCDGH';
if (preg_match($pattern, $subject)){
echo 'true'; }
```

```
//Kiểm tra chuỗi chỉ là ký tự in thường
$pattern = '/[a-z]/'; $subject = 'abcd'; if
(preg_match($pattern, $subject)){ echo 'true';
}
```

Các cờ này được đưa vào mẫu theo dạng /RegExp/flags

"/The/gi" => The fat cat sat on the mat.

"/.(at)/" => The fat cat sat on the mat.

"/.(at)/g" => The fat cat sat on the mat.

```
"/.at(.)?$/gm" => The fat
                  cat sat
                  on the mat.
```

Tạo Form

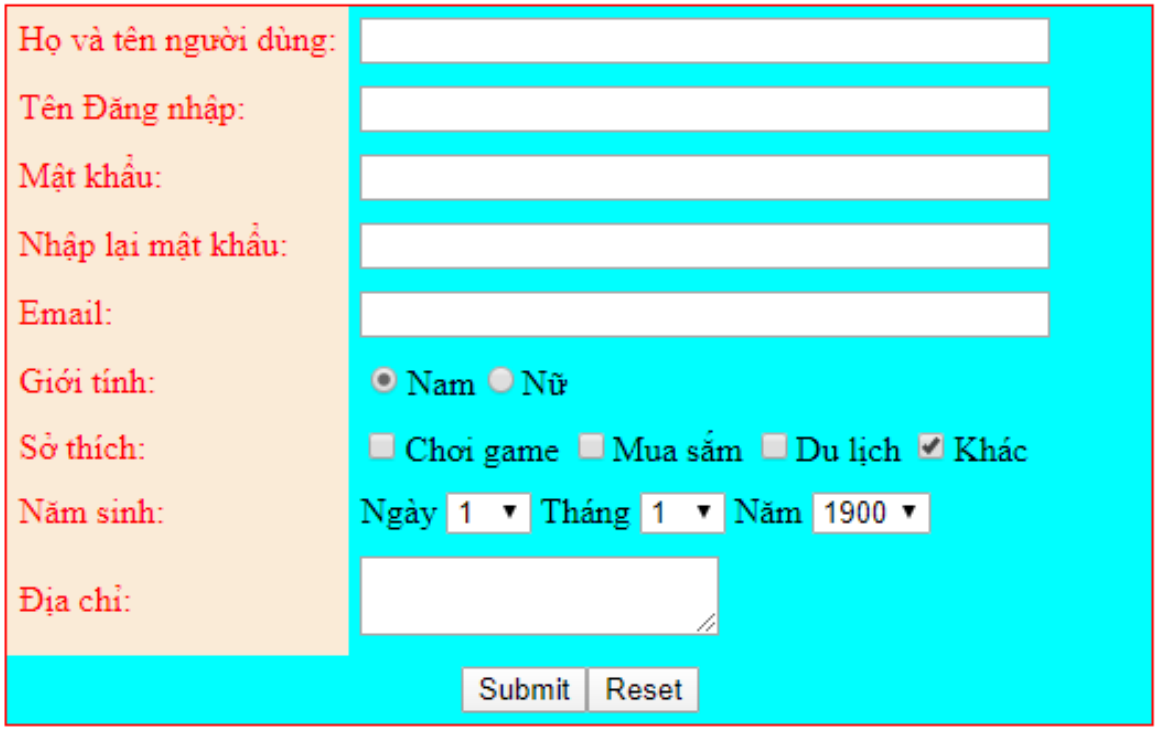
TT	Biểu thức chính quy	Mô tả
1	.	Khớp (match) với bất kỳ ký tự nào
2	^regex	Biểu thức chính quy phải khớp tại điểm bắt đầu
3	regex$	Biểu thức chính quy phải khớp ở cuối dòng.
4	[abc]	Thiết lập định nghĩa, có thể khớp với a hoặc b hoặc c.
5	[abc][vz]	Thiết lập định nghĩa, có thể khớp với a hoặc b hoặc c theo sau là v hay z.
6	[^abc]	Khi dấu ^ xuất hiện như là nhân vật đầu tiên trong dấu ngoặc vuông, nó phủ nhận mô hình. Điều này có thể khớp với bất kỳ ký tự nào ngoại trừ a hoặc b hoặc c.
7	[a-d1-7]	Phạm vi: phù hợp với một chuỗi giữa a và điểm d và con số từ 1 đến 7.
8	X|Z	Tìm X hoặc Z.
9	XZ	Tìm X và theo sau là Z.
10	$	Kiểm tra kết thúc dòng.

Kiểm tra có dấu chấm trong đoạn cần kiểm tra hay không: "/\./"

11	\d	Số bất kỳ, viết ngắn gọn cho [0-9]
12	\D	Ký tự không phải là số, viết ngắn gon cho [^0-9]
13	\s	Ký tự khoảng trắng, viết ngắn gọn cho [\t\n\x0b\r\f]
14	\S	Ký tự không phải khoản trắng, viết ngắn gọn cho [^\s]
15	\w	Ký tự chữ, viết ngắn gọn cho [a-zA-Z_0-9]
16	\W	Ký tự không phải chữ, viết ngắn gọn cho [^\w]
17	\S+	Một số ký tự không phải khoảng trắng (Một hoặc nhiều)
18	\b	Ký tự thuộc a-z hoặc A-Z hoặc 0-9 hoặc _, viết ngắn gọn cho [a-zA-Z0-9_].
19	*	Xuất hiện 0 hoặc nhiều lần, viết ngắn gọn cho {0,}
20	+	Xuất hiện 1 hoặc nhiều lần, viết ngắn gọn cho {1,}
21	?	Xuất hiện 0 hoặc 1 lần, ? viết ngắn gọn cho {0,1}.
22	{X}	Xuất hiện X lần, {}
23	{X,Y}	Xuất hiện trong khoảng X tới Y lần.
24	*?	* có nghĩa là xuất hiện 0 hoặc nhiều lần, thêm ? phía sau nghĩa là tìm kiếm khớp nhỏ nhất.

Tạo Form

- Form HTML đã được tạo trong JS_ValidationFORM.html khi học hết JavaScript.
- Chuyển thành dangky.php
- Viết kiểm tra bằng php thay thế cho JS đã viết



Xử lý Form

❖Lấy thông tin xử lý bằng PHP

```
function check_data($data){
        $data =  trim($data);  //Cắt khoảng trắng 2 đầu
        $data = stripslashes($data); //Cắt bỏ ký tự \
        $data = htmlspecialchars($data);  //Bỏ tác dụng
của thẻ HTML, tương tự hàm htmlentities()
        return $data;
    }
```

PHP Advanced– Phần 4

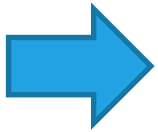
1. Date and Time trong PHP
2. Include, require file và header điều hướng trang
3. Thao tác với File
4. Cookies và Sessions
5. Filters

1. Sử dụng date và time

- Sử dụng hàm date(): định dạng cho dễ đọc hơn
  - Format: cần thiết, chỉ định định dạng của timestamp (mốc thời gian)
  - Timestamp: không bắt buộc, định timestamp cụ thể. Mặc định là ngày giờ hiện tại.
  - Hàm date() ta có thể định dạng hoặc lấy bất kỳ giá trị thời gian nào.

```
2010-<?php echo date("Y");?>
```

```
<?php
echo "Today is " . date("Y/m/d")
. "<br>";
echo "Today is " . date("Y.m.d")
. "<br>";
echo "Today is " . date("Y-m-d")
. "<br>";
echo "Today is " . date("l");
?>
```



Today is 2020/05/15
Today is 2020.05.15
Today is 2020-05-15
Today is Friday

1. Sử dụng date và time

Lấy date

- d - Represents the day of the month (01 to 31)
- m - Represents a month (01 to 12)
- Y - Represents a year (in four digits)
- l (lowercase 'L') - Represents the day of the week

Lấy time

- H - 24-hour format of an hour (00 to 23)
- h - 12-hour format of an hour with leading zeros (01 to 12)
- i - Minutes with leading zeros (00 to 59)
- s - Seconds with leading zeros (00 to 59)
- a - Lowercase Ante meridiem and Post meridiem (am or pm)

```php
<?php
echo "The time is " .
date("h:i:sa");
?>
```

Lấy Time Zone

```php
<?php
date_default_timezone_set("America/New_
York");
echo "The time is " . date("h:i:sa");
?>
```

1. Sử dụng date và time

- Khởi tạo datetime với mktime(): trả về thời gian kiểu Unix từ 1970 tới thời điểm cụ thể.

```
mktime(hour, minute, second, month, day, year)
```

Created date is 2014-08-12 11:14:54am

```php
<?php
$d=mktime(11, 14, 54, 8, 12, 2014);
echo "Created date is " . date("Y-m-d h:i:sa", $d);
?>
```

2. Include, require file và header

- Xây dựng web PHP cần sử dụng rất nhiều code. Nếu để trên 1 file .php rất khó quản lý, nâng cấp, bảo trì

=> sử dụng require, require_once, include, include_one

- Mục đích: import file PHP A vào file PHP B có thể sử dụng được các thư viện trong PHP A.
- Cách dùng
  - Require: Nếu import 2 lần cùng 1 file sẽ chạy cả 2file
  - Require_one: Import 2 lần cùng 1 file chỉ hiển thị 1 lần
  - Include: giống với require (nếu import bằng require thì khi chương trình bị lỗi lập tức tình biên dịch sẽ dừng và xuất ra lỗi còn include thì chỉ đưa ra cảnh báo và chương trình vẫn chạy cho đến cuối)
  - Include_one: giống với require_one.
- Sử dụng:

```
include "/import.php";
```

2. Include File và require file

❖Ví dụ: Nhúng trang menu.php vào trang web

Có trang menu.php

```
<?php
echo '<a
href="/default.php">Home</a> -
<a href="/html/default.php">Học
HTML</a> -
<a href="/css/default.php">Học
CSS</a> -
<a href="/js/default.php">Học
JavaScript</a> -
<a href="default.php">Học PHP
</a>';
?>
```

```
<html>
<body>

<div class="menu">
<?php include 'menu.php';?>
</div>

<h1>Chào mừng bạn đến với trang
chủ!</h1>
<p>Nội dung đoạn<p>
<p>Đoạn khác</p>

</body>
</html>
```

2. Header: điều hướng trang

- Để điều hướng trang web với hàm header trong PHP các bạn sử dụng cú pháp:

```
header('location:' . $url);
```

```
header('location: http://cntt.actvn.edu.vn');
```

3. Thao tác File

- PHP cung cấp các hàm creating, reading, uploading và editing Files.

```
<?php
$myfile = fopen("truyencuoi_kma.txt", "r") or die("Không mở được file!");
echo fread($myfile,filesize(" truyencuoi_kma.txt"));
fclose($myfile);
?>
```

- Fopen mở file gồm 2 tham số: tên file và chế độ mở file
  - r: chỉ đọc. Con trỏ ở đầu file
  - w: mở file cho phép chỉ ghi. Xoá nội dung cũ của file hoặc tạo file mới nếu file đó chưa tồn tại. Con trỏ ở đầu file
  - a: mở file và cho phép chỉ ghi. Ghi tiếp file đã tồn tại, con trỏ ở cuối file. Nếu file chưa tồn tại thì tạo file mới.
  - x: Khởi tạo file mới và chỉ cho phép ghi file. Trả về FALSE nếu file đã có.
  - r+: mở file cho đọc và ghi. Con trỏ đầu file
  - w+: Mở file cho đọc và ghi. Xoá nội dung trong file đã tồn tại. Tạo file nếu chưa có.
  - a+: Mở file cho đọc và ghi. Ghi tiếp file đã có. Tạo mới nếu chưa có
  - x+: Tạo mới file cho phép đọc/ghi. False nếu file đã tồn tại

3.Thao tác File

- fgets(): Đọc 1 dòng của file
- feof(): kiểm tra đã đến cuối file hay chưa (end-of-file)
- fgetc(): sử dụng để độc 1 ký tự từ file

```php
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable
to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
  echo fgets($myfile) . "<br>";    // echo fgetc($myfile);
}
fclose($myfile);
?>
```

Vẫn đọc hết file vì đọc theo dòng hay ký tự thì nó vẫn nằm trong vòng lặp while để đọc tế cuối file.

3. Thao tác File

❖Tạo file và ghi nội dụng vào file: fwrite()

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open
file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

3. Thao tác File
Upload File

❖Đẩy file lên server rất dễ dàng

- Chắc chắn rằng đã cho phép upload trong file php.ini: file_uploads = On
- Sử dụng thuộc tính enctype="multipart/form-data": chỉ định nội dung sẽ sử dụng khi gửi form lên server
- sử dụng thuộc tính method="post"
- trong form sử dụng type="file" để chọn file gửi lên server

```
<form action="upload.php" method="post" enctype="multipart/form-
data">
  Chọn ảnh để upload:
  <input type="file" name="fileToUpload" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>
```

3. Thao tác File
Upload File

❖Bên phía server trang PHP:

$target_dir: thư mục file sẽ được đặt
$target_file: Đường dẫn file được tải lên
$imageFileType: phần mở rộng của file

```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
  $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
  if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
  } else {
    echo "File is not an image.";
    $uploadOk = 0;
  }
}
?>
```

3. Thao tác File
Upload File

- Kiểm tra file đã tồn tại hay chưa?
- Kiểm tra kích thước file có vượt quá kích thước cho phép.
- Kiểm tra kiểu định dạng file cho phép.

```
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType !=
"png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
  echo "Sorry, only JPG, JPEG, PNG & GIF files
are allowed.";
  $uploadOk = 0;
}
```

```
// Check if file already exists
if (file_exists($target_file)) {
  echo "Sorry, file already
exists.";
  $uploadOk = 0;
}
```

```
// Check file size
if
($_FILES["fileToUpload"]["size"] >
500000) {
  echo "Sorry, your file is too
large.";
  $uploadOk = 0;
}
```

4. Cookies và Sessions

- Cookie cung cấp cho ứng dụng web phương thức lưu trữ thông tin trên trình duyệt của người dùng và truy xuất khi người dùng gửi yêu cầu xem trang
- Cookie được lưu theo cặp thông tin key/value
- Cookie được lưu ở trình duyệt và có thể truy xuất từ server khi người dùng truy cập trang
- Thời gian sống của server được quy định tối đa 3 năm
- Cookie có tầm ảnh hưởng đến các vùng trên server do lập trình quy định.

Xây dựng chức năng ghi nhớ mật khẩu trong ứng dụng đăng nhập

4. Cookies và Sessions

❖Thiết lập COOKIE

```
setcookie($name, $value, $expire, $path, $domain)
```

- $name: tên cookie
- $value: Giá trị cookie (mặc định là chuỗi rỗng)
- $expire: Thời gian sống cookie (theo giây), nếu thiết lập là 0 thì sau khi tắt trình duyệt cookie tự mất. Tối đa là 3 năm
- $path: Đường dẫn trên server mà cookie có hiệu lực
- $domain: Tên miền cookie có hiệu lực

```
<?php
$name = 'is_login';
$value = true;
$expire = time()+3600; $path =
'/'; setcookie($name,
$value,$expire ,$path); ?>
```

```
<?php
setcookie('is_login', true, time()+
3600, '/');
?>
```

4. Cookies và Sessions

Lấy giá trị của COOKIE

- Biến toàn cục $_COOKIE là nơi lưu trữ thông tin của COOKIE

```
<?php
$is_login = $_COOKIE['is_login'];
echo $is_login;
?>
```

Xoá cookie

- Để xoá cookie ta cập nhật thời gian sống bằng một thời gian trong quá khứ
- Khi cookie được xoá thông tin của cookie được loại bỏ ra khỏi trình duyệt và biến $_COOKIE

```
<?php
setcookie('is_login', true, time() - 3600, '/');
?>
```

4. Cookies và Sessions

```php
<?php
$cookie_name = “thuanld";
$cookie_value = “Le Thuan";
setcookie($cookie_name, $cookie_value, time() +
(86400 * 30), "/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

4. Cookies và Sessions

- Session dùng để lưu trữ thông tin của người dùng hoặc tuỳ chọn cấu hình hệ thống cho người dùng.
- Mỗi client có ID session khác nhau (32 số hexa) nếu các client không bị ảnh hưởng lẫn nhau.
- Thông thường sử dụng session để lưu thông tin đăng nhập, giỏ hàng hoặc những giữ liệu mang tính tạm thời và mỗi client có dữ liệu khác nhau.
- Vị trí tạm thời của session được cài đặt trong file php.ini gọi là session.save_path
- Session kết thúc khi người dùng tắt trình duyệt hoặc sau khi rời khỏi site, Server sẽ chấm dứt session sau một thời gian định trước thường là 30 phút.

4. Cookies và Sessions

- Đăng ký session bằng session_start(): bắt buộc phải có, khi đó Server tạo ra ID riêng không trùng lặp để nhận diện client hiện tại.
- Lưu trữ:
  - Sử dụng biến toàn cục $_SESSION
  - Trước khi dùng biến nên kiểm tra session đó đã tồn tại hay chưa.

```
$_SESSION['session_name'] = $session_value
```

- Lấy giá trị trong Sesion
  - $tenbien = $_SESSION['session_name']
- Xoá session
  - unset($_SESSION['session_name'])
  - Nếu muốn xoá toàn bộ session: session_destroy()

4. Cookies và Sessions

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

5. Filter

- Mọi dữ liệu nhận được từ bên ngoài mã nguồn gọi là extrarmal input. Chúng đều chứa nguy cơ gây ra lỗi hệ thống. Sử dụng bộ lọc dữ liệu giúp dữ liệu đầu vào chính xác – đúng quy chuẩn.
- Dữ liệu đầu vào gồm 2 phần:
  - Xác thực tính đúng đắn (tính hợp lệ của dữ liệu)
  - Loại bỏ các ký tự trái phép ra khỏi dữ liệu.
- Các nguồn dữ liệu từ bên ngoài gửi đến trang xử lý có thể từ:
  - Form nhập liệu của người dùng
  - Dữ liệu cookie gửi lên từ trình duyệt
  - Giá trị trả về từ CSDL
  - Từ dịch vụ web
  - Giá trị nhận từ các biến của server (các biến toàn cục)

5. Filter

Lọc dữ liệu với hàm filter_var()

- Xác thực dữ liệu
- Loại bỏ ký tự trái phép
  - **FILTER_SANITIZE_STRING**
  - **FILTER_SANITIZE_NUMBER_INT**
  - **FILTER_SANITIZE_NUMBER_FLOAT**
  - **FILTER_SANITIZE_EMAIL**
  - **FILTER_SANITIZE_URL**
  - **FILTER_VALIDATE_IP (ver4)**

```php
<?php
$str = "A nice day!";
$new_str =
filter_var($str,FILTER_SANITIZE
_STRING);
echo $new_str;
?>
```

```php
<?php
$in = 10;
if(!filter_var($in,FILTER_VALIDATE_INT) === false)
{
echo 'Là số nguyên';
}
else
{
echo 'không phải là số nguyên - hãy kiểm tra lại';
}
?>
```

5. Filter

❖Xác thực số nguyên với miền giá trị

- Thêm tham số min_range và max_range vào hàmf filter_var()

```
<?php
    $in = 99;
    $min_range = 1;
    $max_range = 1000;
    if(!filter_var($in, array("option" => array("min_range" => min_range, "max_range" =>
    maxrange))) === false)
    {
        echo 'Là số nguyên - nằm trong miền giá trị cho phép';
    }
    else
    {
        echo 'Không phải là số nguyên - không nằm trong miền giá trị cho phép';
    }
?>
```

5. Filter

❖Filter_var() và vấn đề với số 0

```php
<?php
$int = 0;

if (filter_var($int, FILTER_VALIDATE_INT) === 0 ||
!filter_var($int, FILTER_VALIDATE_INT) === false) {
  echo("Integer is valid");
} else {
  echo("Integer is not valid");
}
?>
```

5. Filter

- Xác thực IP ver4

```php
<?php
$ip = "127.0.0.1";

if (!filter_var($ip, FILTER_VALIDATE_IP) === false) {
  echo("$ip is a valid IP address");
} else {
  echo("$ip is not a valid IP address");
}
?>
```

- Xác thực với IP ver6

```php
<?php
$ip = "2001:0db8:85a3:08d3:1319:8a2e:0370:7334";

if (!filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV6)
=== false) {
  echo("$ip is a valid IPv6 address");
} else {
  echo("$ip is not a valid IPv6 address");
}
?>
```

5. Filter

❖Xác thực Email

```php
<?php
$email = "john.doe@example.com";

// Remove all illegal characters from email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);

// Validate e-mail
if (!filter_var($email, FILTER_VALIDATE_EMAIL) ===
false) {
  echo("$email is a valid email address");
} else {
  echo("$email is not a valid email address");
}
?>
```

5. Filter

❖Xác thực URL

```php
<?php
$url = "https://www.w3schools.com";

// Remove all illegal characters from a url
$url = filter_var($url, FILTER_SANITIZE_URL);

// Validate url
if (!filter_var($url, FILTER_VALIDATE_URL) === false) {
  echo("$url is a valid URL");
} else {
  echo("$url is not a valid URL");
}
?>
```

Với query string

```php
<?php
$url = "https://www.w3schools.com";

if (!filter_var($url, FILTER_VALIDATE_URL,
FILTER_FLAG_QUERY_REQUIRED) === false) {
  echo("$url is a valid URL with a query string");
} else {
  echo("$url is not a valid URL with a query string");
}
?>
```

PHP OOP– Phần 5 (đọc thêm)

1. Tạo đối tượng trong PHP
2.
3.

MySQL– Phần 6 (Quan trọng)

1. Cách quản trị MySQL trong XAMPP
2. Hướng dẫn quản trị MySQL trong Php my Admin
3. Thao tác với cơ sở dữ liệu

PHP - MySQL– Phần 7

1. Kết nối PHP với MySQL
2. Tạo, thêm, sửa, xoá dữ liệu bằng PHP
3. Câu lệnh truy vấn dữ liệu từ PHP

Kết nối PHP và MySQL

- Có thể kết nối bằng MySQLi (i- improved => cải tiến) và PDO (PHP Data Objects)
- PDO làm việc với 12 hệ CSDL khác nhau (việc chuyển đổi dễ dàng)
- MySQLi chỉ làm việc với MySQL
- Cả 2 đều làm việc theo hướng đối tượng, MySQLi có thể làm việc với thủ tục API
- Có 3 cách thức để kết nối
  - MySQLi (object-oriented)
  - MySQLi (procedural)
  - PDO
- *Note: Nắm chắc 1 cách kết nối để thường xuyên làm việc. Nên biết cả 3 cách.*

Kết nối PHP và MySQL

❖Mở kết nối: MySQLi Object-Oriented

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username,
$password);

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

Kết nối PHP và MySQL

❖Mở kết nối: MySQLi Procedural

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username,
$password);

// Check connection
if (!$conn) {
  die("Connection failed: " .
mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Kết nối PHP và MySQL

❖Mở kết nối: PDO

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
  $conn
= new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
  echo "Connected successfully";
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
?>
```

Kết nối PHP và MySQL

- Đóng kết nối: Object-Oriented:
  - $conn->close();
- Đóng kết nối Procedural:
  - mysqli_close($conn);
- Đóng kết nối: PDO
  - $conn = null;

Tạo cơ sở dữ liệu

- Sử dụng cú pháp CREATE DATABASE ten_db
- MySQLi Object-oriented

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername,
$username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " .
$conn->connect_error);
}
```

```php
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
  echo "Database created
successfully";
} else {
  echo "Error creating database:
" . $conn->error;
}

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

Tạo cơ sở dữ liệu

- Sử dụng cú pháp CREATE DATABASE ten_db
- MySQLi Procedural

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn =
mysqli_connect($servername,
$username, $password);
// Check connection
if (!$conn) {
  die("Connection failed: " .
mysqli_connect_error());
}
```

```php
// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
  echo "Database created
successfully";
} else {
  echo "Error creating database:
" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Tạo cơ sở dữ liệu

- Sử dụng cú pháp CREATE DATABASE ten_db
- MySQLi PDO

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
  $conn = new PDO("mysql:host=$servername", $username,
$password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
  $sql = "CREATE DATABASE myDBPDO";
  // use exec() because no results are returned
  $conn->exec($sql);
  echo "Database created successfully<br>";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

Tạo bảng

- Trong bảng: tên trường, kiểu dữ liệu và thuộc tính tuỳ chọn:
  - NOT NULL: Không cho phép trường đó để null
  - DEFAULT: Đặt giá trị mặc định
  - UNSIGNED: Giá trị số không âm
  - AUTO INCREMENT: Tự động tăng
  - PRIMARY KEY: Khoá chính

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP
)
```

Tạo bảng

MySQLi Object-oriented

```php
<?php
$servername
= "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn
= new mysqli($servername,
$username, $password,
$dbname);
// Check connection
if ($conn-
>connect_error) {
  die("Connection failed:
" . $conn-
>connect_error);
}
```

```php
// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
  echo "Table MyGuests created successfully";
} else {
  echo "Error creating table: " . $conn->error;
}
$conn->close();
?>
```

Tạo bảng

❖Trong MySQLi Procedural

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn =
mysqli_connect($servername,
$username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " .
mysqli_connect_error());
}
```

```php
// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT
PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT
CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)";
if (mysqli_query($conn, $sql)) {
  echo "Table MyGuests created
successfully";
} else {
  echo "Error creating table: " .
mysqli_error($conn);
}
mysqli_close($conn);
?>
```

Tạo bảng

❖PDO

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
  $conn
= new PDO("mysql:host=$servername;
dbname=$dbname", $username,
$password);
  // set the PDO error mode to
exception
  $conn-
>setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
// sql to create table
  $sql = "CREATE TABLE MyGuests (
  id INT(6) UNSIGNED AUTO_INCREMENT
PRIMARY KEY,
  firstname VARCHAR(30) NOT NULL,
  lastname VARCHAR(30) NOT NULL,
  email VARCHAR(50),
  reg_date TIMESTAMP DEFAULT
CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
  )";
  // use exec() because no results are
returned
  $conn->exec($sql);
  echo "Table MyGuests created
successfully";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e-
>getMessage();
}
$conn = null;
?>
```

Chèn dữ liệu

Trường tự tăng thì không chèn dữ liệu vào

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

MySQLi Object-oriented

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername,
$username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " .
$conn->connect_error);
}
```

```php
$sql = "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('John', 'Doe',
'john@example.com')";

if ($conn->query($sql) === TRUE) {
  echo "New record created
successfully";
} else {
  echo "Error: " . $sql . "<br>" .
$conn->error;
}

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

Chèn dữ liệu

MySQLi Procedural

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn =
mysqli_connect($servername,
$username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " .
mysqli_connect_error());
}
```

```php
$sql = "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('John', 'Doe',
'john@example.com')";

if (mysqli_query($conn, $sql)) {
  echo "New record created
successfully";
} else {
  echo "Error: " . $sql . "<br>" .
mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Chèn dữ liệu

PDO

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
  $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  $sql = "INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('John', 'Doe', 'john@example.com')";
  // use exec() because no results are returned
  $conn->exec($sql);
  echo "New record created successfully";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

Chèn dữ liệu

- Chèn nhiều dòng một lúc

```
$sql = "INSERT INTO MyGuests (firstname, lastname,
email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname,
email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname,
email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";
```

- MySQLi Object-oriented: if ($conn->multi_query($sql) === TRUE)
- MySQLi Procedural: if (mysqli_multi_query($conn, $sql))
- PDO:

Chèn dữ liệu

- Chèn nhiều dòng một lúc
- PDO:

```
// begin the transaction
  $conn->beginTransaction();
  // our SQL statements
  $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('John', 'Doe', 'john@example.com')");
  $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('Mary', 'Moe', 'mary@example.com')");
  $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('Julie', 'Dooley', 'julie@example.com')");

  // commit the transaction
  $conn->commit();
  echo "New records created successfully";
```

Chèn dữ liệu

- Chèn dữ liệu bằng thi thi tham số (MySQLi with Prepared Statements)

```
// prepare and bind
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname,
email) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();

echo "New records created
successfully";

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

- Đối số truyền vào bind_param:
  - i: integer d: double s: string

Chèn dữ liệu

❖Chèn dữ liệu bằng thi thi tham số (PDO with Prepared Statements

```
try {
  $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  // prepare sql and bind parameters
  $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
  VALUES (:firstname, :lastname, :email)");
  $stmt->bindParam(':firstname', $firstname);
  $stmt->bindParam(':lastname', $lastname);
  $stmt->bindParam(':email', $email);
 // insert a row
   $firstname = "John";
   $lastname = "Doe";
   $email = "john@example.com";
   $stmt->execute();
   // insert another row
   $firstname = "Mary";
   $lastname = "Moe";
   $email = "mary@example.com";
   $stmt->execute();
  // insert another row
    $firstname = "Julie";
    $lastname = "Dooley";
    $email = "julie@example.com";
    $stmt->execute();
    echo "New records created
successfully";
} catch(PDOException $e) {
  echo "Error: " . $e->getMessage();
}
$conn = null;
?>
```

Lấy ID bản ghi cuối cùng

- Khi sử dụng ID là trường tự tăng, ta có theerlaays ID cuối cùng được chèn vào

MySQLi Object-oriented

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
  $last_id = $conn->insert_id;
  echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
  echo "Error: " . $sql . "<br>" . $conn->error;
}

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

Lấy ID bản ghi cuối cùng

❖MySQLi Procedural

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn =
mysqli_connect($servername,
$username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " .
mysqli_connect_error());
}
```

```
$sql = "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('John', 'Doe',
'john@example.com')";

if (mysqli_query($conn, $sql)) {
  $last_id =
mysqli_insert_id($conn);
  echo "New record created
successfully. Last inserted ID is:
" . $last_id;
} else {
  echo "Error: " . $sql . "<br>" .
mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Lấy ID bản ghi cuối cùng

PDO

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
  $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  $sql = "INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('John', 'Doe', 'john@example.com')";
  // use exec() because no results are returned
  $conn->exec($sql);
  $last_id = $conn->lastInsertId();
  echo "New record created successfully. Last inserted ID is: " .
$last_id;
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

Select

- SELECT column_name(s) FROM table_name
- SELECT * FROM table_name
- MySQLi Object-oriented + table HTML

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. "
" . $row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

Select

MySQLi Procedural

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. "
" . $row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

Select

PDO

```
try {
  $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  $stmt = $conn->prepare("SELECT id, firstname, lastname FROM
MyGuests");
  $stmt->execute();

  // set the resulting array to associative
  $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
  foreach(new TableRows(new RecursiveArrayIterator($stmt-
>fetchAll())) as $k=>$v) {
    echo $v;
  }
} catch(PDOException $e) {
  echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

Select + Where

- SELECT column_name(s) FROM table_name WHERE column_name operator value

```php
$sql = "SELECT id, firstname, lastname FROM MyGuests WHERE lastname='Doe'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
  }
```

MySQLi Object-oriented

```php
$sql = "SELECT id, firstname, lastname FROM MyGuests WHERE lastname='Doe'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
  }
```

MySQLi Procedural

Select + Where

PDO

```
try {
  $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  $stmt = $conn->prepare("SELECT id, firstname, lastname FROM
MyGuests WHERE lastname='Doe'");
  $stmt->execute();
  // set the resulting array to associative
  $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
  foreach(new TableRows(new RecursiveArrayIterator($stmt-
>fetchAll())) as $k=>$v) {
    echo $v;
  }
}
catch(PDOException $e) {
  echo "Error: " . $e->getMessage();
}
$conn = null;
```

Select + Order By

- Tương tự như where, ta chỉ thêm phần Order by trong chuỗi SQL mà thôi

```
SELECT column_name(s) FROM table_name
ORDER BY column_name(s) ASC|DESC
```

Xoá và cập nhật dữ liệu

❖Để xoá dữ liệu trong bảng

```
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE
id=3";
if ($conn->query($sql) === TRUE) {
  echo "Record deleted
successfully";
} else {
  echo "Error deleting record: " .
$conn->error;
}
```

```
// sql to delete a record
$sql = "DELETE FROM MyGuests
WHERE id=3";
if (mysqli_query($conn, $sql)) {
  echo "Record deleted
successfully";
} else {
  echo "Error deleting record:
" . mysqli_error($conn);
}
```

PDO

```
// sql to delete a record
  $sql = "DELETE FROM MyGuests WHERE id=3";
  // use exec() because no results are returned
  $conn->exec($sql);
  echo "Record deleted successfully";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}
```

MySQLi Procedural

Xoá và cập nhật dữ liệu

❖Cập nhật dữ liệu trong chuỗi SQL:

```
UPDATE table_name
SET column1=value,
column2=value2,...
WHERE some_column=some_value
```

Limit – Chọn số lượng bản ghi trả về

- Lấy tất cả bản ghi từ 1 tới 30

```
$sql = "SELECT * FROM Orders LIMIT 30";
```

- Lấy 10 bản ghi từ bản ghi 16 (OFFSET = 15)
  - Cách 1

```
$sql = "SELECT * FROM Orders LIMIT 10 OFFSET 15";
```

  - Cách 2

```
$sql = "SELECT * FROM Orders LIMIT 15, 10";
```

Bài tập PHP