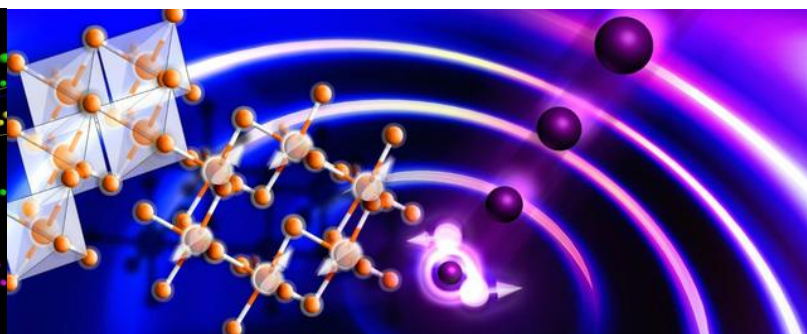
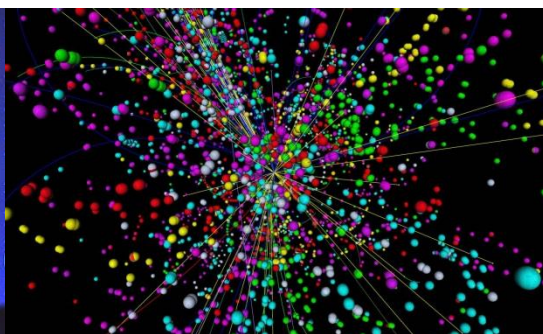


LẬP TRÌNH MÃ NGUỒN MỞ - PHP

BỘ MÔN KHOA HỌC DỮ LIỆU & TRÍ TUỆ NHÂN TẠO



NỘI DUNG

1. Tổng quan về lập trình PHP
2. Lập trình HĐT trong PHP – Kết hợp PHP và JavaScript
3. Cơ sở dữ liệu trong PHP
4. Một số mô hình và ứng dụng trong PHP

[Xem đề cương chi tiết](#)

TÀI LIỆU THAM KHẢO

1. Khát Thuỳ Phương, Lập trình Web động với PHP tập 1,2,3,4,5, NXB Hà Nội
2. Beginning PHP and MySQL
3. Andrew Curioso - Ronald Bradford - Patrick Galbraith, Expert PHP and MySQL, Wrox, 2010

KIỂM TRA – ĐÁNH GIÁ

- Dự lớp: 80% trở lên tính theo số tiết lên lớp.
- Các yêu cầu khác: 40% (Làm đề tài theo nhóm)
- Thi kết thúc học phần: 60%

Chương 1.

TỔNG QUAN VỀ LẬP TRÌNH PHP

1. Giới thiệu và cài đặt
2. Các kiểu dữ liệu cơ sở
3. Các cấu trúc điều khiển
4. Xây dựng phương thức (hàm)
5. Sử dụng hàm của PHP
6. Xử lý tập tin
7. Form và các điều khiển
8. Làm việc với XML
9. Session và Cookies

GIỚI THIỆU VÀ CÀI ĐẶT

1. Khái niệm về phần mềm nguồn mở:

Phần mềm nguồn mở (PMNM) là phần mềm với mã nguồn được công bố và sử dụng một giấy phép nguồn mở. Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi và cải tiến phần mềm, và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi .

GIỚI THIỆU VÀ CÀI ĐẶT

1. Khái niệm về phần mềm nguồn mở:

Nhà cung cấp PMNM có quyền yêu cầu người dùng trả một số chi phí về các dịch vụ bảo hành, huấn luyện, nâng cấp, tư vấn, vv... tức là những dịch vụ để phục vụ người dùng, nhưng không được bán các sản phẩm nguồn mở vì nó là tài sản của trí tuệ chung, không phải là tài sản riêng của một nhà cung cấp nào.

GIỚI THIỆU VÀ CÀI ĐẶT

- PHP (Hypertext Preprocessor) ? Là một ngôn ngữ lập trình mã nguồn mở được thực thi trên **máy chủ** (server).
- Một tập tin PHP có phần mở rộng là ***.php**, nó có thể chứa các văn bản, mã nguồn html, CSS, JavaScript,...
- Server là gì với trang PHP ? Server chỉ quan tâm đến mã nguồn PHP, nó sẽ chuyển mã nguồn PHP sang mã nguồn html rồi gửi lại cho người dùng.
- Trang web PHP phát sinh chỉ được gửi đến 1 client duy nhất.
- Server PHP phải mạnh hơn so với một server html

GIỚI THIỆU VÀ CÀI ĐẶT

2. Lợi ích của phần mềm nguồn mở:

- Các chuẩn mở, không lệ thuộc vào nhà cung cấp, miễn phí
- Phổ biến (70% số lượng website)
- Mạnh (facebook, yahoo, google dùng PHP)
- Chuyên dùng viết trong môi trường Web
- Có thể lập trình với dạng OPP
- Nhiều hỗ trợ và tài liệu hướng dẫn
- An toàn

GIỚI THIỆU VÀ CÀI ĐẶT

3. Công cụ cần có để lập trình PHP:

- Cài đặt Xampp server

<https://www.apachefriends.org/download.html>

- Cài đặt trình viết code

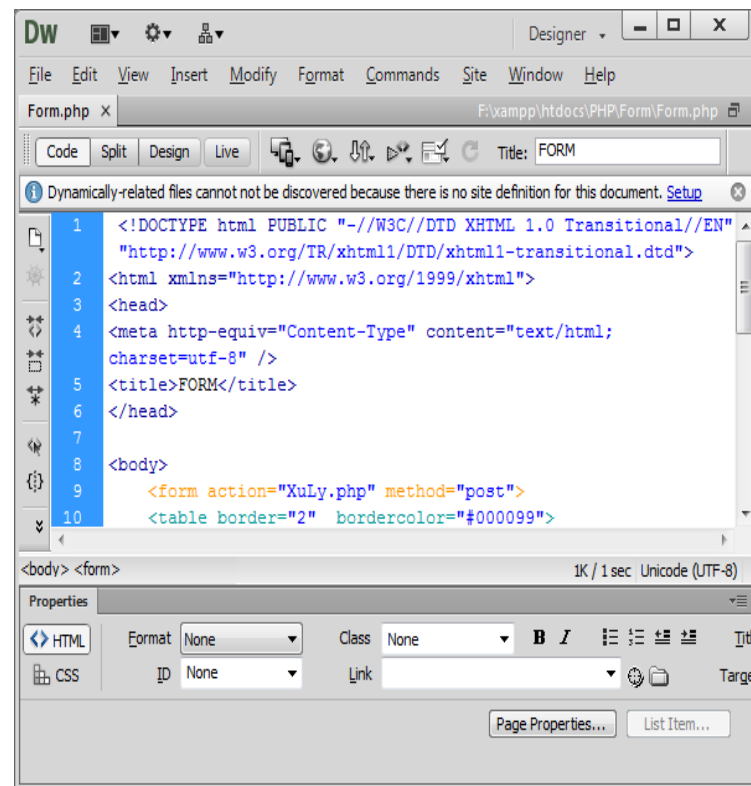
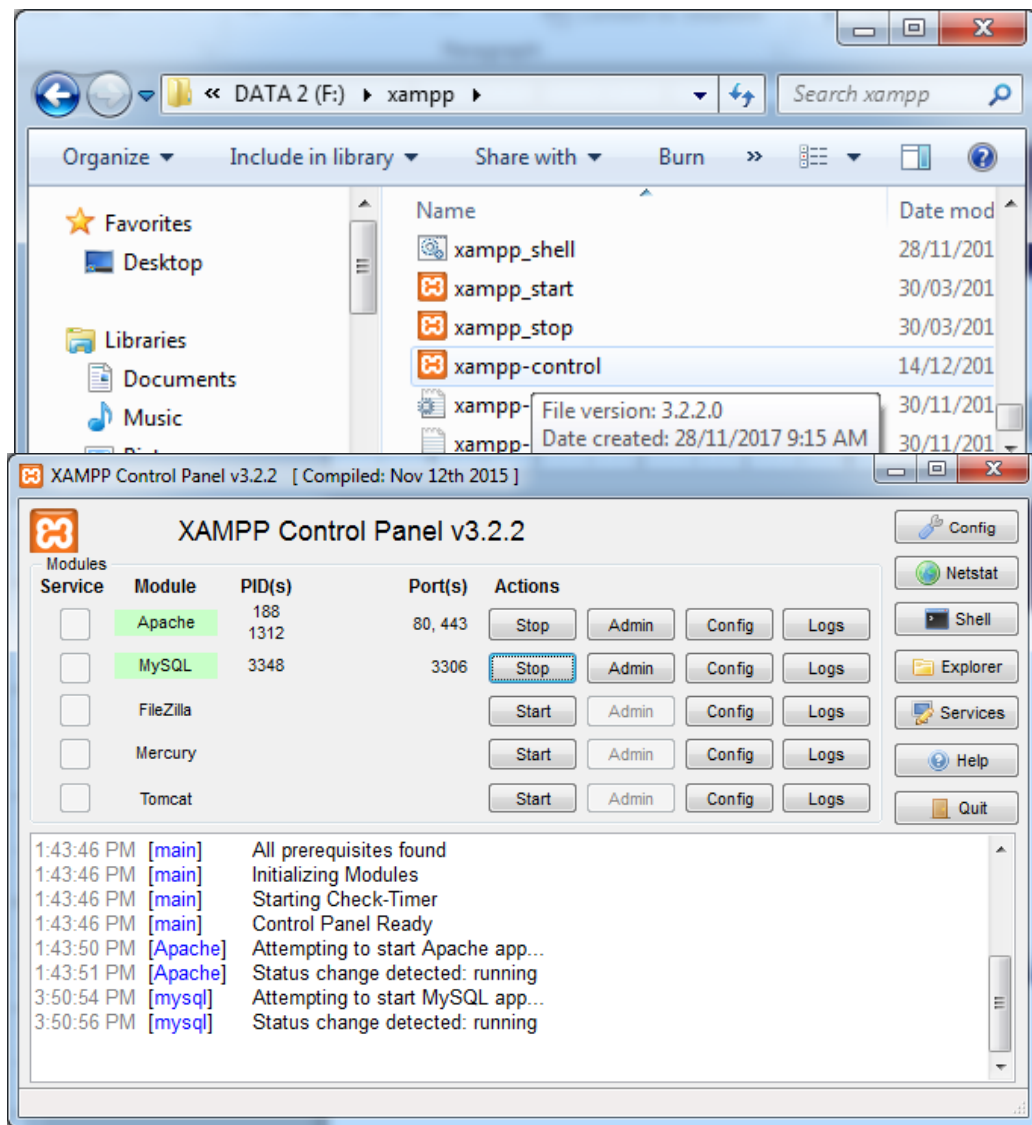
PHP design 8

Notepad ++

DreamWeaver

- Trình duyệt web

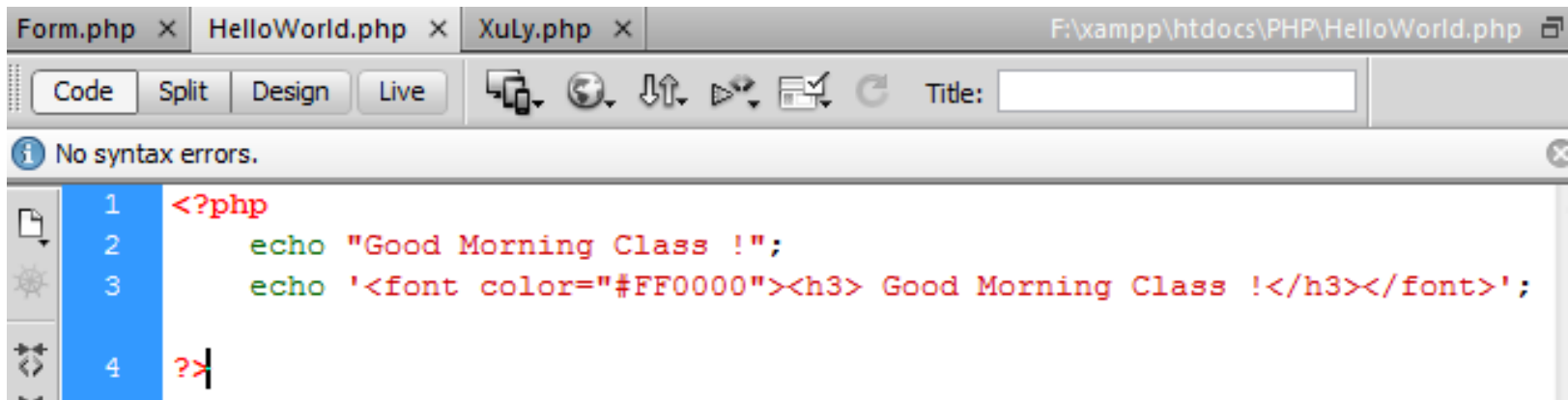
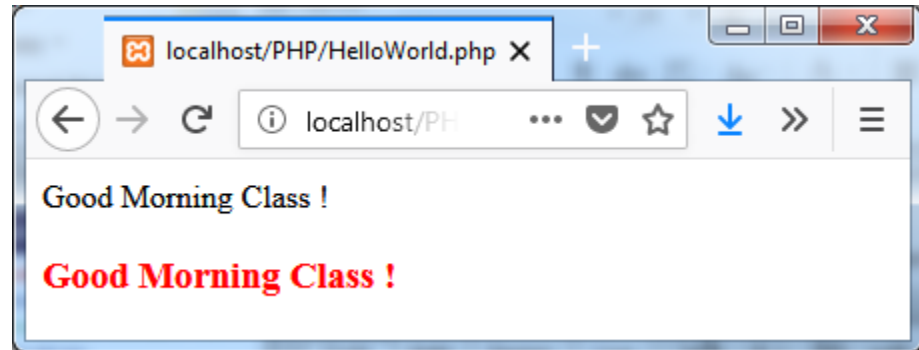
GIỚI THIỆU VÀ CÀI ĐẶT



GIỚI THIỆU VÀ CÀI ĐẶT

4. Ứng dụng ban đầu

- `<?php ... ?>`
- `<? ... ?>`
- `<% ... %>`
- `<script language = "php"> ... </script>`



GIỚI THIỆU VÀ CÀI ĐẶT

Chú thích trong php



```
8 <body>
9 <form action="XuLy.php" method="post">
10     Ngày Sinh
11     <select>
12     <?php
13         // Minh họa vòng for xuất danh sách ngày
14         for ($i = 1 ; $i <= 31; $i++)
15         {
16             echo "<option>$i</option>";
17         }
18     ?>
19 </select>
20     Tháng
21     <select>
22     <?php
23         /* Danh sách xuất ra số tháng trong năm
24         ( từ 1 - 12) */
25         for ($i = 1; $i <= 12; $i++)
26         {
27             echo "<option>$i</option>";
28         }
```

The screenshot shows a code editor with two tabs: 'XuLy.php' and 'Form.php*'. The code is in PHP and HTML. Two green boxes with yellow arrows point to specific comment lines in the code. The first box points to line 13, which contains a single-line comment starting with '//' (double slash). The second box points to line 23, which contains a multi-line comment starting with '/*' (asterisk) and ending with '*/' (asterisk slash).

Dùng // nếu nội dung cần ghi chú ngắn

Dùng /*...*/ nếu nội dung cần ghi chú nằm trên nhiều dòng liên tục

CÁC KIỂU DỮ LIỆU CƠ SỞ

1. Các kiểu dữ liệu

- Kiểu **Boolean**: chỉ có một trong hai giá trị là true hoặc False
- Kiểu **Integer**: số nguyên ex: 7
- Kiểu **Float/ Double**: số thực ex 1.2345
- Kiểu **String**: (chuỗi, ký tự)
 - Mỗi ký tự chiếm 1 Byte
 - Mỗi chuỗi có thể chứa một hay nhiều ký tự thuộc 256 ký tự khác nhau
 - Chuỗi không giới hạn về kích thước

CÁC KIỂU DỮ LIỆU CƠ SỞ

- Kiểu **array**: mảng các phần tử

Ex: array (1,2,3,4,5)

- Kiểu **Object**: kiểu đối tượng

Ex: \$sv = new Sinh_Vien();

CÁC KIỂU DỮ LIỆU CƠ SỞ

2. Kiểm tra kiểu dữ liệu

- `gettype($x)` : kiểm tra x có kiểu dữ liệu là gì
- `is_integer($x)`: kiểm tra x có phải kiểu integer không
- `is_double($x)`:
- `is_string($x)`:
- `is_array($x)`:
- `is_object($x)`:
- `isset($x)`: kiểm tra 1 biến đã được khởi tạo gt chưa
- `empty($x)`: kiểm tra biến có gt = null không

CÁC KIỂU DỮ LIỆU CƠ SỞ

3. Chuyển đổi kiểu

- Trong quá trình tính toán, kiểu dl cũ có thể không còn phù hợp (kết quả vượt khỏi phạm vi) → phải chuyển đổi kiểu dữ liệu.
- Có hai cách chuyển kiểu:

- Cách 1: (automatic)

Ví dụ .

<code>\$var = "100" + 15 ;</code>	115 : integer
<code>\$var = "100" + 15.0 ;</code>	115.0 : double
<code>\$var = 100 + 15.0 ;</code>	115.0: double
<code>\$var = 38 + "step"</code>	38 : integer
<code>\$var = "Hello" + "world"</code>	0 : integer

CÁC KIỂU DỮ LIỆU CƠ SỞ

- Cách 2: ghi tên kiểu dữ liệu mà biến muốn chuyển đổi vào phía trước biến **(data type) \$var**

Ví dụ 1.

```
$don_gia = 50000;
```

```
$so_luong = 100;
```

```
$thanh_tien = (double)($so_luong * $don_gia);
```

Ví dụ 2.

\$var	(int) \$var	(bool)\$var	(string)\$var
Null	0	False	""
true	1		"1"
False	0		""
'6 feet'	6	True	
'foo '	0	true	

CÁC KIỂU DỮ LIỆU CƠ SỞ

3. Biến

- **Khái niệm:** Biến là một vùng nhớ trong bộ nhớ chính giúp biểu diễn thông tin thực tế trong chương trình.
- **Khai báo:**
`$ten_bien;`
- **Quy tắc đặt tên:**
 - + Bắt đầu bằng ký tự \$, theo sau là một ký tự hoặc dấu _
 - + Không trùng với tên hàm
 - + Không bắt đầu bằng ký số
 - + Có phân biệt chữ HOA chữ thường
 - + Không khai báo kiểu dữ liệu (kiểu dữ liệu động được khởi tạo ở lần đầu tiên khi gán giá trị cho biến)

CÁC KIỂU DỮ LIỆU CƠ SỞ

- Xuất giá trị biến:**

```
$a = 500;
```

```
echo $a;
```

```
echo "giá trị biến a = $a";
```

```
echo "giá trị biến a = " . $a;
```

→ Toán tử "." Cho phép nối chuỗi

```
$s = "Nguyễn Hải Yên" ;  
$ns = " 2000";  
$kq1 = $s.$ns;  
  
echo "kq1 = $kq1". "<br>";
```

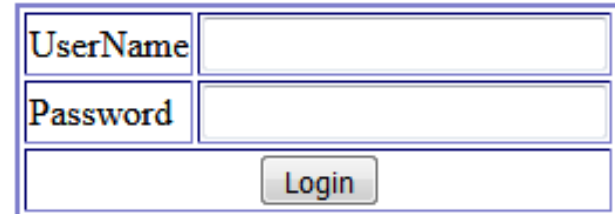
CÁC KIỂU DỮ LIỆU CƠ SỞ

- Biến nhận giá trị từ điều khiển :**

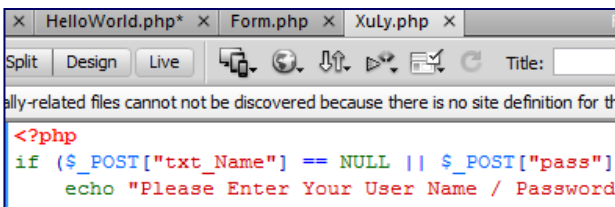
Cần nhận giá trị từ điều khiển và gán cho biến x

$X = \$_POST["\text{Tên điều khiển tương ứng}"]$

$X = \$_GET["\text{Tên điều khiển tương ứng}"]$



```
7 <body>
8 <form action="XuLy.php" method="post">
9 <table border="2" bordercolor="#000099">
10 <tr>
11 <td>UserName</td>
12 <td><input type="text" size="25" name = "txt_Name" /></td>
13 </tr>
14 <tr>
15 <td>Password </td>
16 <td><input type="password" size="25" name="pass" /></td>
17 </tr>
18 <tr>
19 <td colspan="2" align="center"><input type="submit" value=
20 "Login" /></td>
21 </tr>
22 </table>
23 </form>
24 </body>
25 </html>
```



```
<?php
if ($_POST["txt_Name"] == NULL || $_POST["pass"]==NULL)
    echo "Please Enter Your User Name / Password !";
else
{
    if($_POST["txt_Name"] == "yennh" && $_POST["pass"]=="123456" )
        echo "Login Successful !";
    else
        echo "Try again!";
}
?>
```

CÁC KIỂU DỮ LIỆU CƠ SỞ

- **Khai báo biến tham chiếu:**

C1:

```
$bien_1;
```

```
$bien_2 =& $bien_1;
```

C2:

```
$bien_1;
```

```
$bien_2 = &$bien_1;
```

→ Cùng tham chiếu đến một vùng nhớ, nên khi sửa giá trị một trong hai biến thì 2 biến đều thay đổi

CÁC KIỂU DỮ LIỆU CƠ SỞ

- **Phạm vi hoạt động của biến :**

LOẠI	VỊ TRÍ KHAI BÁO	PHẠM VI ẢNH HƯỞNG
Biến cục bộ	Trong hàm	Trong hàm
Tham số của hàm	Khai báo hàm	Trong hàm
Biến toàn cục	Sau từ khóa GLOBAL	Webpage
Biến tĩnh	Sau từ khóa STATIC	Webpage
Biến hệ thống	\$_SEVER, \$_POST,...	Website

CÁC KIỂU DỮ LIỆU CƠ SỞ

- **Biến của biến :**

`$bien_1;`

`$bien_1 = "abc";`

`$$bien_1 = value; ≡ $abc = value;`

- **Ví dụ :** cho biết cách khai báo biến nào sau đây đúng ?

`$size`

`$drink-size`

`$my_drink_size`

`$_drinks`

`$drink4you`

`$$2hot4u`

`$$hot4u2`

CÁC KIỂU DỮ LIỆU CƠ SỞ

4. Hằng:

- **K.n:** Hằng là một giá trị không thể chỉnh sửa được trong quá trình thực hiện chương trình.
- Ví dụ : PI, USD,...
- Quy tắc đặt tên giống quy tắc đặt tên biến, tên hằng thường IN HOA
- Khai báo: dùng hàm **define()** để khai báo hằng

define("TÊN_HÀNG", giá_trị);

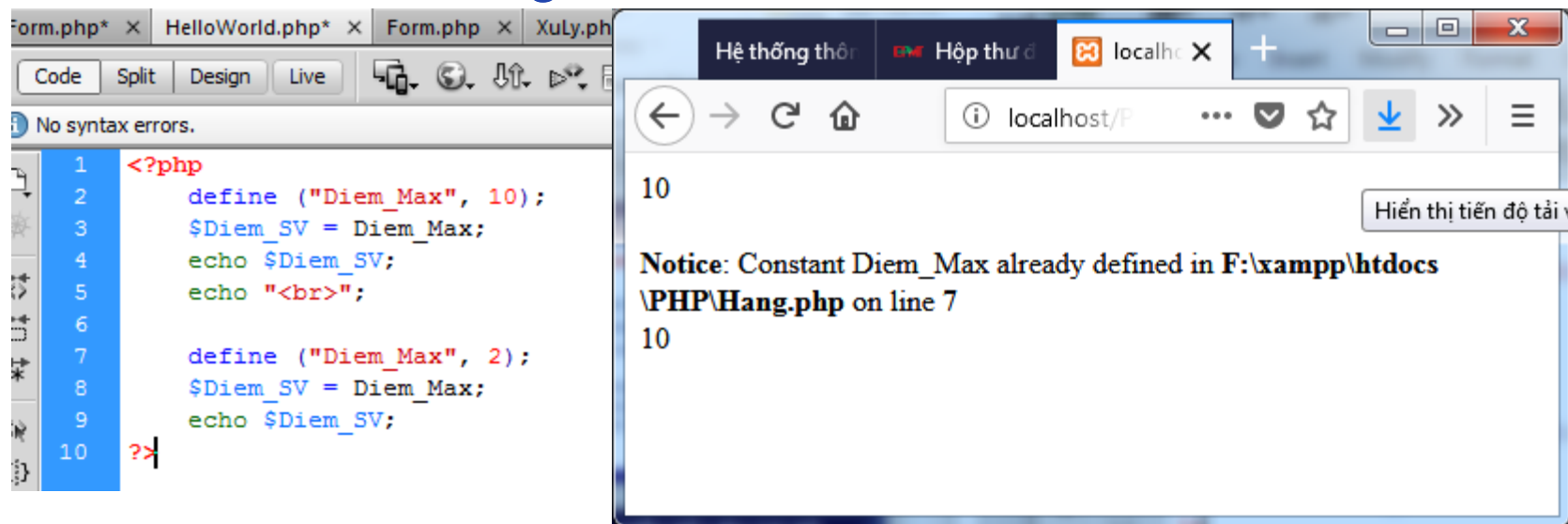
Vd: define("PI", 3.14);

define("Diem_Max", 10);

CÁC KIỂU DỮ LIỆU CƠ SỞ

4. Hằng (tt):

- Khi hằng được định nghĩa, nó không bị thay đổi
- Chỉ có kiểu dữ liệu boolean, integer, float, string mới có thể chứa các hằng



```
<?php
1  define ("Diem_Max", 10);
2  $Diem_SV = Diem_Max;
3  echo $Diem_SV;
4  echo "<br>";
5
6
7  define ("Diem_Max", 2);
8  $Diem_SV = Diem_Max;
9  echo $Diem_SV;
10 ?>
```

10

Notice: Constant Diem_Max already defined in F:\xampp\htdocs\PHP\Hang.php on line 7

10

CÁC KIỂU DỮ LIỆU CƠ SỞ

5. Xuất dữ liệu ra màn hình trình duyệt:

- **Echo/ echo** (danh sách tham số)
- **Print** (tham số)

vd: `print ("I love PHP");`

- boolean **Printf** (string format, mixed) // xuất theo định dạng

```
1 <?php
2     $num1 = 10;
3     $num2 = 20;
4     printf("%d + %d =%d", $num1, $num2, $num1+$num2);
5     ?>
```



localhost/PHP/printf.php

10 + 20 =30

CÁC KIỂU DỮ LIỆU CƠ SỞ

Ký hiệu	Kiểu tham số	Thể hiện ra
%b	Số nguyên	Số nhị phân
%c	Số nguyên	Ký tự ASCII
%d	Số nguyên	Số nguyên
%f	Số thực	Số thực
%O	Số nguyên	Số bát phân
%s	Chuỗi	Chuỗi
%u	Số nguyên	Số nguyên không dấu
%x	Số nguyên	Số thập lục phân thường
%X	Số nguyên	Số thập lục phân

Ví dụ:

Printf (“\$%.2f”, 43.2); → **\$43.20**

CÁC KIỂU DỮ LIỆU CƠ SỞ

- **Sprintf ()** → tạo ra một chuỗi

Ví dụ:

`$cost = Sprintf (“$%.2f”, 43.2)` → **\$43.20**

Ghi chú :

Xuất chuỗi tĩnh : **echo, print()**

Xuất chuỗi động có định dạng : **printf() , sprintf()**

CÁC CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc điều kiện **IF**
- Cấu trúc điều kiện **SWITCH CASE**
- Cấu trúc lặp **FOR / FOR EACH**
- Cấu trúc lặp **WHILE / DO WHILE**

CẤU TRÚC IF

Cú pháp 1:

if (biểu thức)

{

 // Các Câu Lệnh;

}

Cú pháp 2:

if (biểu thức)

{

 // Các Câu Lệnh1;

}

else

{

 // Các Câu Lệnh2;

}

CẤU TRÚC IF

Cú pháp 3:

```
if (điều_kiện_1)
    //Các câu lệnh khi điều
    kiện 1 là true;
elseif (điều_kiện_2)
    //Các câu lệnh khi điều
    kiện 2 là true;
else
    //Các câu lệnh khi các
    điều kiện là false;
```

Cú pháp 4:

```
if (bieu_thuc1)
{
    // Các câu lệnh;
    if (bieu_thuc2)
    {
        // Các câu lệnh;
    }
}
```

MỘT SỐ VÍ DỤ

- **VD1:** Kiểm tra tính chẵn lẻ của một số.
- **VD2:** Kiểm tra một năm nhuận hay không nhuận (*Năm nhuận là năm chia hết cho 4 hoặc 400 nhưng không chia hết cho 100*)
- **VD3:** Kiểm tra tính hợp lệ của form đăng nhập sau.

The image displays five screenshots of a web login form, illustrating different states of the application:

- Top Left:** The initial state of the form. It contains two input fields: "UserName" with the value "Nguyễn Hải Yến" and "Password" which is empty. Below the fields is a "Login" button.
- Top Middle:** The form after the user has entered their password. The "Password" field now contains six dots, indicating masked input. The "Login" button is still visible.
- Top Right:** The form with both fields empty, showing the state after a password reset or clearing the inputs.
- Bottom Left:** The form displaying an error message: "Please Enter Your User Name / Password !". The input fields are empty.
- Bottom Middle:** The form displaying an error message: "Try again!". The input fields are empty.
- Bottom Right:** The form displaying a success message: "Login Successful !". The input fields are empty.

CẤU TRÚC SWITCH CASE

Cú pháp:

```
switch (variable)
{
    case value_1:
        // Statement 1;
        break;
    case value_2:
        // Statement 2;
        break;

    ...
    default:
        // Statement 3
        break;
}
```

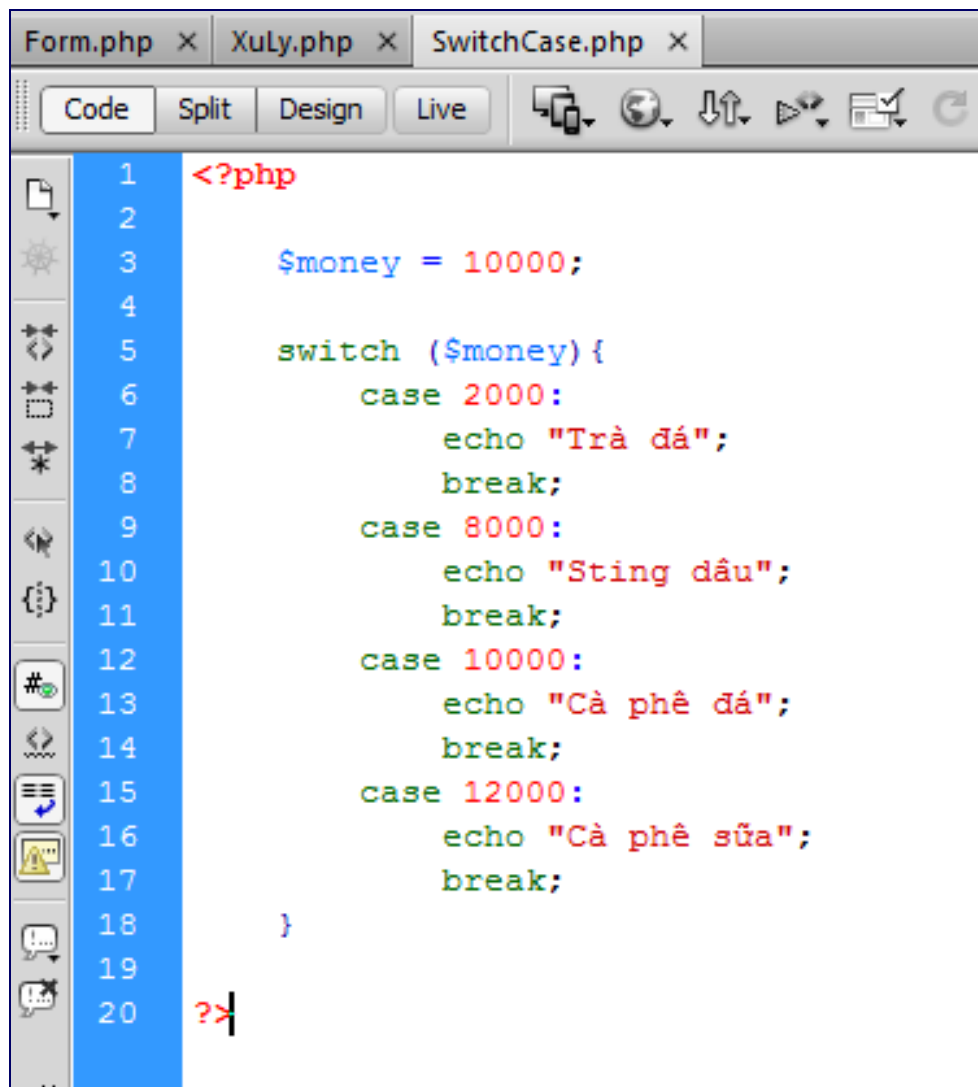
CẤU TRÚC SWITCH CASE

VÍ DỤ:

1. Viết chương trình kiểm tra một tháng $month$ có bao nhiêu ngày ?
2. Khi đến một quán nước, ở đó có một cái menu như sau và trong tay bạn chỉ có đúng mười nghìn. Nếu yêu cầu chọn một món nước uống có giá bằng đúng với số tiền mà bạn đang có, thì món nước uống đó là món gì !?

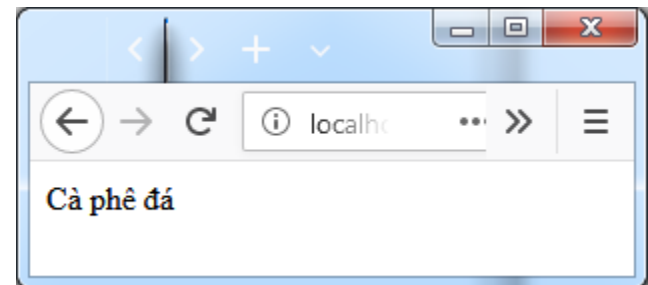
MENU		
<i>Cà phê sữa</i>		<i>12.000đ</i>
<i>Cà phê đá</i>		<i>10.000đ</i>
<i>Sting dâu</i>		<i>8.000đ</i>
<i>Trà đá</i>		<i>2.000đ</i>

CẤU TRÚC SWITCH CASE



The screenshot shows a code editor with three tabs: Form.php, XuLy.php, and SwitchCase.php. The SwitchCase.php tab is active, displaying a PHP script. The script starts with a PHP opening tag, followed by a variable assignment for \$money. A switch statement is then used to check the value of \$money against four cases: 2000, 8000, 10000, and 12000. Each case has an associated echo statement and a break statement. The script ends with a PHP closing tag. The code is as follows:

```
1 <?php
2
3     $money = 10000;
4
5     switch ($money){
6         case 2000:
7             echo "Trà đá";
8             break;
9         case 8000:
10            echo "Sting dâu";
11            break;
12        case 10000:
13            echo "Cà phê đá";
14            break;
15        case 12000:
16            echo "Cà phê sữa";
17            break;
18    }
19
20 ?>
```



CẤU TRÚC LẶP FOR

Cú pháp 1:

```
for(biểu thức 1; biểu thức 2; biểu thức 3)  
{  
    //Đoạn mã mà bạn muốn được thực thi ;  
}
```

Ví dụ :

```
1  <?php  
2  for($i = 1; $i <= 3; $i++)  
3  { echo "<p>Lập Trình Web</p>";  
4  }  
5  ?>  
  
7  <?php  
8  for($i = 0; $i < 10; $i++)  
9  { echo "Số: " . ($i+1) . "<br>";  
10 }  
11 ?>
```

CẤU TRÚC LẶP FOR

Cú pháp 2:

For each (*array as variable*)

```
{  
    //Đoạn mã mà bạn muốn được thực thi ;  
}
```

Ví dụ :

```
<?php  
$meal = array("breakfast"=> "Bread and milk", "lunch"=> "Rice", "dinner" => "Instance Noodle");  
print "<table border='1' ><br>";  
foreach ($meal as $key => $value)  
{  
    print "<tr><td bgcolor = 'Yellow' > $key </td><td> $value </td><tr>";  
    print "<br>";  
}  
print "</table>"  
?>
```

breakfast	Bread and milk
lunch	Rice
dinner	Instance Noodle

CẤU TRÚC LẶP WHILE

Cú pháp 1:

While (expression)

```
{  
    Statements;  
}
```

Cú pháp 2:

Do

```
{  
    Statements;  
} While ( expression);
```

Ví dụ :

Navigation: ⬅ ➡ ↺ 🏠

Ngày Sinh Tháng

Untitled Document x W Cách sử dụng vòng lặp while 8 x

Navigation: ⬅ ➡ ↺ 🏠 | localhost/PHP/While1.php

Nokia
SamSung
IPhone
Oppo
BPhone

```
XuLy.php x While.php x  
Split Design Live  
3x errors.  
<body>  
    <form action="XuLy.php" method="post">  
        Ngày Sinh  
        <select>  
            <?php  
                $i = 1;  
                while ($i <= 31)  
                {  
                    echo "<option>$i</option>";  
                    $i++;  
                }  
            ?>  
        </select>
```

SỬ DỤNG HÀM CỦA PHP

1. Giới thiệu hàm

Có những công việc được thực hiện đi, thực hiện lại nhiều lần, liên quan đến việc tính toán, xử lý chuỗi, thời gian,... → Cần xây dựng hàm

- Hàm có thể sử dụng lặp lại nhiều lần trong chương trình
- Hàm được thực thi khi chúng ta thực hiện lệnh gọi hàm
- Tham số hàm có thể là bất kỳ kiểu dữ liệu nào trong php: string, float, integer, array, object

Hàm có 2 loại (do người dùng tự viết, thư viện PHP (>700)

SỬ DỤNG HÀM CỦA PHP

2. Hàm do người dùng tự viết

Cú pháp khai báo:

```
Function Tên_hàm (Danh_sách_các_tham_số)
{
    Khối_lệnh;
    return giá_trị;
}
```

Danh sách tham số có thể gồm (tham trị, tham biến, **tham số tùy chọn**)

SỬ DỤNG HÀM CỦA PHP

Tham số tùy chọn :

- Là những tham số có thể được truyền giá trị hoặc không
- Cho phép tạo sẵn các giá trị mặc định cho tham số.
- Những tham số này chỉ xuất hiện ở cuối danh sách các tham số.

Cú pháp chung:

```
Function Tên_hàm ($Ds tham trị, &$ ds tham biến,  
                    $tham số tùy chọn = giá trị )  
{  
    Khối lệnh;  
    return giá_trị;  
}
```

SỬ DỤNG HÀM CỦA PHP

Ví dụ :

```
function so_thich($nuoc_uong = "cà phê")  
{  
    return "Tôi thích uống $nuoc_uong."  
}  
echo so_thich(); // "Tôi thích uống cà phê."  
echo so_thich("nước ép trái cây"); // "Tôi thích  
uống nước ép trái cây."
```

SỬ DỤNG HÀM CỦA PHP

3. Hàm của PHP – Các hàm toán học

- `abs ($x)` : Lấy giá trị tuyệt đối của x
- `Pow($x,$n)`: trả về x lũy thừa n
- `Round($x,2)`: làm tròn x lấy 2 số lẻ
- `Ceil(num)` : làm tròn lên (`ceil(-5.1) = -5`)
- `Floor(num)`: làm tròn xuống (`floor(9.99) = 9`)
- `Rand()` : sinh một số nguyên ngẫu nhiên
- `Rand(min, max)`: sinh một số ngẫu nhiên trong đoạn từ min → max

SỬ DỤNG HÀM CỦA PHP

3. Hàm của PHP – Các hàm chuỗi

- `strlen($s)` : trả về chiều dài của chuỗi `s`
- `Trim ($s)`: loại khoảng trắng thừa đầu và cuối chuỗi `s`
- `Substr(str, start,end)`: trả về chuỗi con của `str` từ `s` → `e`
- `Strtolower(str)`: chuỗi thường
- `strtoupper(str)`: chuỗi hoa
- `strcmp(s1, s2)` : so sánh chuỗi `s1` và `s2`

 `= 0` : `s1 = s2` ; `< 0` : `s1 < s2`; `> 0` : `s1 > s2`
- Nối chuỗi `'.'`

SỬ DỤNG HÀM CỦA PHP

3. Hàm của PHP – Các hàm thời gian

- **checkdate(\$tháng, \$ngày, \$năm)** : trả về true / False cho phép kiểm tra ngày hợp lệ

- **date ("d/m/y h:m:s")** : Định dạng ngày hiện tại

```
$ngay_hien_tai = date ("d/m/y h:m:s");
```

- **mktime(...)**: Đổi thời gian sang đơn vị giây

```
$Tong_so_giay = mktime(0, 0, 0,  
1, 1, 2014); // tính từ 01/01/1970
```

- **Time()** : lấy thời gian hiện tại

```
$Time = time();
```

XỬ LÝ TẬP TIN

- Ý nghĩa
- Các thao tác trên tập tin
- Upload tập tin

XỬ LÝ TẬP TIN

1. Ý nghĩa

- Khi dữ liệu lớn, cần được lưu trữ và xử lý thường xuyên thì một giải pháp là sử dụng tập tin (file)

2. Các thao tác trên tập tin

- Mở tập tin
- Đóng tập tin
- Đọc tập tin
- Ghi tập tin
- Các hàm xử lý

XỬ LÝ TẬP TIN

Mở tập tin:

- Cú pháp: `fopen ( filename, filemode)`
- Khi mở file bằng `fopen()` thì tất cả các nội dung trong file sẽ được gán vào biến `$f`.
- Ví dụ: Mở file `Data.txt` trong thư mục `PHP/File` với chế độ chỉ đọc

```
$f = fopen("PHP/File/Data.txt", "r");
```

XỬ LÝ TẬP TIN

Mở tập tin:

Một số File Mode

Filemode	Mô tả
r	Chỉ đọc bắt đầu từ đầu của tập tin
r ⁺	Đọc / ghi bắt đầu từ đầu tập tin
w	Ghi . Mở và xóa nội dung đang có của tập tin hoặc tạo mới nếu tập tin không tồn tại
w ⁺	Đọc /Ghi . Mở và xóa nội dung đang có của tập tin hoặc tạo mới nếu tập tin không tồn tại
a	Thêm. Mở và ghi nội dung mới vào cuối tập tin hoặc tạo mới nếu tập tin không tồn tại
a ⁺	Đọc và thêm . Ghi vào cuối tập tin
x	Chỉ ghi. Tạo tập tin mới. Nếu tập tin đã $\exists$ thì trả về False + thông báo lỗi
x ⁺	Đọc/ ghi. Tạo tập tin mới. Nếu tập tin đã $\exists$ thì trả về False + thông báo lỗi

XỬ LÝ TẬP TIN

Đóng tập tin:

- Cú pháp: `fclose ( $f)`
- Trả về TRUE nếu đóng thành công, ngược lại trả về FALSE nếu đóng không thành công
- Ví dụ: Đóng file `Data.txt` trong thư mục `PHP/File`

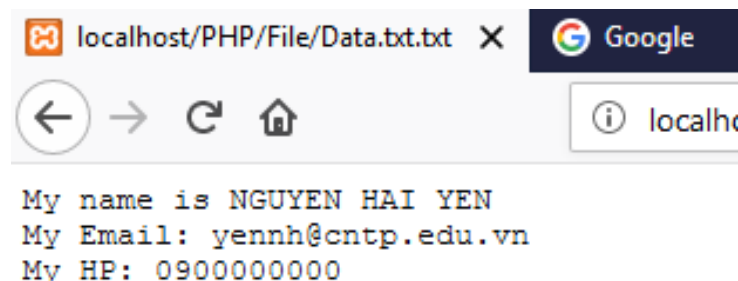
```
fclose($f);
```

XỬ LÝ TẬP TIN

Đọc tập tin:

- Duyệt tập tin và đọc từng dòng
- Ví dụ: Đọc file **Data.txt** trong thư mục **PHP/File**

```
1 <?php
2     $f = fopen("PHP/File/Data.txt", "r");
3     while (!feof($f))
4     {
5         $noi_dung = fgets($f);
6         echo $noi_dung. "<br>";
7     }
8     fclose($f);
9     ?>
```



XỬ LÝ TẬP TIN

Đọc tập tin:

- Đọc và duyệt toàn bộ nội dung tập tin bằng readfile ()
- Cú pháp:

 readfile (“đường dẫn và tên file”)
- Kết quả trả về là số byte của file nếu đọc thành công ,
ngược lại trả về FALSE

XỬ LÝ TẬP TIN

Ghi tập tin:

- Cú pháp:

`fwrite($f, nội dung, [độ dài theo số byte])`

```
1 <?php
2     $noi_dung = " NỘI DUNG CẦN GHI ...";
3     // Thực hiện mở và ghi file
4     $f = fopen("PHP/File/Data.txt", "a");
5     fwrite ($f,$noi_dung);
6     fclose($f);
7 ?>
8
```

XỬ LÝ TẬP TIN

Các hàm xử lý tập tin:

- Kiểm tra tập tin tồn tại **file_exists(\$f)**
- Lấy kích thước tập tin **filesize(\$f)**
- Xóa tập tin **unlink(\$f)**
- Kiểm tra tập tin có phải là file không **is_file(\$f)**

3. Upload tập tin

- **Bước 1:** Tạo form

- **Bước 2:**

2.1. Kiểm tra File upload có hợp lệ hay không (kiểu file, dung lượng,...)

2.2. Nếu hợp lệ dùng hàm `move_uploaded_file` để di chuyển file cần upload vào thư mục định sẵn là TM **upload**

```
Move_uploaded_file($_FILES["file_upload"]["tmp_name"],  
"upload/" . $_FILES["file_upload"]["tmp_name"]);
```

XỬ LÝ TẬP TIN

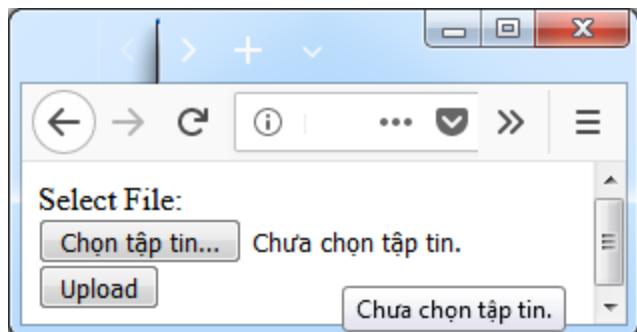
Thông tin của \$_FILES

- \$_FILE["filefield"]["name"] : tên File
- \$_FILE["filefield"]["type"] : kiểu File
- \$_FILE["filefield"]["size"] : kích thước File
- \$_FILE["filefield"]["tmp_name"] : tên tạm của File
- \$_FILE["filefield"]["error"] : lỗi của File

XỬ LÝ TẬP TIN

Ví dụ

```
<body>
  <form name = "f1" action="XuLy_Upload_File.php"
    method="post" enctype="multipart/form-data">
    Select File: <input type="file" name = "file_upload" size="20"><br />
    <input type="submit" value="Upload" name="oK" />
  </form>
</body>
</html>
```



```
<?php
if(isset($_POST['ok'])){ // Người dùng đã ấn submit
    if($_FILES['file']['name'] != NULL){ // Đã chọn file
        // Tiến hành code upload file
        if($_FILES['file']['type'] == "image/jpeg"
        || $_FILES['file']['type'] == "image/png"
        || $_FILES['file']['type'] == "image/gif"){
            // là file ảnh
            // Tiến hành code upload
            if($_FILES['file']['size'] > 1048576){
                echo "File không được lớn hơn 1mb";
            }else{
                // file hợp lệ, tiến hành upload
                $path = "data/"; // file sẽ lưu vào thư mục data
                $tmp_name = $_FILES['file']['tmp_name'];
                $name = $_FILES['file']['name'];
                $type = $_FILES['file']['type'];
                $size = $_FILES['file']['size'];
                // Upload file
                move_uploaded_file($tmp_name,$path.$name);
                echo "File uploaded! <br />";
                echo "Tên file : ".$name."<br />";
                echo "Kiểu file : ".$type."<br />";
                echo "File size : ".$size;
            }
        }else{
            // không phải file ảnh
            echo "Kiểu file không hợp lệ";
        }
    }else{
        echo "Vui lòng chọn file";
    }
}
?>
```

FORM VÀ CÁC ĐIỀU KHIỂN

1. Form:

- Form được dùng để gửi yêu cầu đến Server
- GV nhắc lại form **POST** và form **GET**
- Thuộc tính quan trọng action quy định trang xử lý yêu cầu khi người dùng submit form.

```
<form name = "f1"  
      action="XuLy_Upload_File.php"  
      method="post" enctype="multipart/form-data">
```

FORM VÀ CÁC ĐIỀU KHIỂN

2. Các điều khiển:

- TextField / TextArea
- Button (Submitutton/ Resetutton)
- Checkbox
- RadioButton / RadioGroup

→ Lưu ý xem lại tất cả các điều khiển trên form đã học trong học phần thiết kế Web

FORM VÀ CÁC ĐIỀU KHIỂN

3. Đọc các giá trị từ điều khiển:

\$_POST:

- Được dùng để lấy giá trị của các điều khiển trên Form thông qua phương thức POST.
- Thông tin được gửi từ Form thông qua phương thức POST không giới hạn lượng thông tin gửi đi và sẽ không hiển thị trên địa chỉ URL nên thông tin sẽ bảo mật hơn

FORM VÀ CÁC ĐIỀU KHIỂN

\$_POST:

- Cú pháp: lấy giá trị của một đối tượng trên form sau khi Form Submit .

\$_POST[" Tên điều khiển"]

- Ví dụ: Lấy giá trị TextField có name là txtTen

`$ten = $_POST[" txtTen "];`

FORM VÀ CÁC ĐIỀU KHIỂN

\$_GET:

- Được dùng để lấy giá trị của các điều khiển trên Form thông qua phương thức GET.
- Thông tin được gửi từ Form thông qua phương thức GET giới hạn lượng thông tin gửi đi (tối đa 100 ký tự)và sẽ hiển thị trên địa chỉ URL nên thông tin sẽ không được bảo mật.

FORM VÀ CÁC ĐIỀU KHIỂN

\$_GET:

- Cú pháp: lấy giá trị của một đối tượng trên form sau khi Form Submit .

\$_GET[" Tên điều khiển"]

- Ví dụ: Lấy giá trị TextField có name là txtTen

`$ten = $_GET[" txtTen "];`

FORM VÀ CÁC ĐIỀU KHIỂN

\$_REQUEST:

- Có thể được sử dụng để lấy thông tin trên form được gửi đi bằng cả phương thức POST và GET.
- Mức độ bảo mật thông tin khi sử dụng iến \$_REQUEST thấp → chỉ nên sử dụng biến \$_REQUEST khi trang web không có yêu cầu về bảo mật

FORM VÀ CÁC ĐIỀU KHIỂN

\$_REQUEST:

- Cú pháp: lấy giá trị của một đối tượng trên form sau khi Form Submit.

`$_REQUEST["Tên điều khiển"]`

- Ví dụ: Lấy giá trị TextField có name là txtTen

`$ten = $_REQUEST["txtTen"];`

`$_FILES:` Tìm hiểu thêm!

LÀM VIỆC VỚI XML

- Giới thiệu
- Cấu trúc cú pháp
- Phần tử
- Thuộc tính
- Không gian tên
- Mã hóa
- Truy xuất tài liệu XML với DOM

GIỚI THIỆU

XML (eXtensible Markup Language) Ngôn ngữ đánh dấu mở rộng

- Được thiết kế để chuyển và lưu trữ dữ liệu
- Do W3C (World Wide Web Consortium) đề nghị sử dụng
- Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với mạng Internet.

GIỚI THIỆU

Đặc điểm

- Dễ dàng viết được những chương trình xử lý tài liệu XML
- Dễ đọc và có tính hợp lý cao
- Giảm thiểu những thuộc tính tùy chọn
- Dễ dàng được sử dụng trên Internet
- Hỗ trợ nhiều ứng dụng.

CẤU TRÚC – CÚ PHÁP

Cấu trúc

- Tài liệu XML có cấu trúc cây (tree) và bắt đầu bằng nút gốc (root) và các nhánh lá (leave)
- Tài liệu XML là tài liệu tự mô tả và có cấu trúc đơn giản

```
<?xml version="1.0" encoding="UTF-8"?>
```

Khai báo phiên bản và loại mã hóa

```
<note>
```

```
<to> Student </to>
```

```
<from> Teacher </from>
```

```
<heading> Reminder </heading>
```

```
<body> Please come back Class after Tembin Storm !!!
```

```
    on 27/12/2017
```

```
</body>
```

Nút gốc (root element)

Nút con (child element)

```
</note>
```

Kết thúc Nút gốc

CẤU TRÚC – CÚ PHÁP

Cú pháp:

- Có một root element duy nhất.
- Mỗi tag mở ra cần được đóng lại
- Tag có phân biệt chữ hoa và chữ thường
- Nút con (child element) phải nằm trong nút cha
- Ghi chú trong XML: `<!-- Ghi chú -->`
- Lưu trữ dấu xuống dòng: LF
- Chú ý trong file .xml cần phải có dòng khai báo

```
<?xml version="1.0" encoding="UTF-8"?>
```

CẤU TRÚC – CÚ PHÁP

Ví dụ:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
  <Don_hang>
    <Ngay_dat_hang>2010-01-27</Ngay_dat_hang>
    <Ten_khach_hang>Khuất Thuỳ Phương</Ten_khach_hang>
    <Noi_dung>
      <Ma_san_pham>1</Ma_san_pham>
      <So_luong>2</So_luong>
    </Noi_dung>
    <Noi_dung>
      <Ma_san_pham>4</Ma_san_pham>
      <So_luong>1</So_luong>
    </Noi_dung>
  </Don_hang>
```

PHẦN TỬ - ELEMENT

- Phần tử XML (XML element) là các nội dung được bắt đầu bằng tag mở và kết thúc bằng tag đóng.
- Các phần tử có thể chứa: các phần tử khác, văn bản, thuộc tính, dữ liệu tổng hợp,...
- Các phần tử XML có thể không chứa nội dung:
<book></book> hoặc **<book/>**
- Quy tắc đặt tên XML element: tên chứa ký tự, ký số, không bắt đầu bằng dấu chấm câu (.), không được bắt đầu là xml (XML, hoặc Xml,...), tên không chứa khoảng trắng.

PHẦN TỬ - ELEMENT

- Các phần tử XML có thể mở rộng thêm thông tin mà không làm cho ứng dụng bị ảnh hưởng.
- Ví dụ:

```
<note>
<to>Students</to>
<from>Teacher</from>
<body>Come to class on
time</body>
</note>
```

```
<note>
<date>2014-03-01</date>
<to>Students</to>
<from>Teacher</from>
<heading>Reminder</heading>
<body>Come to class on
time</body>
</note>
```

Mở rộng

THUỘC TÍNH - ATTRIBUTE

- Thuộc tính cung cấp thêm thông tin về phần tử
- Ví dụ:
`<file type = "gif "> computer.gif </file>`
`<person sex = " female "> Nguyễn Thị Mơ </person>`
- Không có quy định nào yêu cầu khi nào thì sử dụng thuộc tính , khi nào thì sử dụng phần tử

<pre><person sex="female"> <firstname>Anna</firstn ame> <lastname>Smith</lastna me> </person></pre>	<pre><person> <sex>female</sex> <firstname>Anna</firstn ame> <lastname>Smith</lastna me> </person></pre>
---	--

THUỘC TÍNH - ATTRIBUTE

- Tuy nhiên nên sử dụng phần tử thay cho thuộc tính vì:
- Thuộc tính không thể chứa nhiều giá trị
- Thuộc tính không thể chứa cấu trúc cây.
- Thuộc tính không dễ mở rộng (khi có yêu cầu thay đổi trong tương lai).

KHÔNG GIAN TÊN – NAME SPACE

- XML NameSpace cung cấp một phương pháp để **tránh xung đột tên các phần tử** → sử dụng tiền tố

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

Xung đột

```
<table>
  <name>African Coffee
  Table</name>
  <width>80</width>
  <length>120</length>
</table>
```



```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

```
<f:table>
  <f:name>African Coffee
  Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

- Cú pháp

Xmlns : prefix = "URI"

KHÔNG GIAN TÊN – NAME SPACE

- Ví dụ:

```
<root>
```

```
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
```

```
    <h:tr>
```

```
      <h:td>Apples</h:td>
```

```
      <h:td>Bananas</h:td>
```

```
    </h:tr>
```

```
  </h:table>
```

```
  <f:table xmlns:f="http://www.w3schools.com/furniture">
```

```
    <f:name>African Coffee Table</f:name>
```

```
    <f:width>80</f:width>
```

```
    <f:length>120</f:length>
```

```
  </f:table>
```

```
</root>
```

KHÔNG GIAN TÊN – NAME SPACE

- Không gian tên mặc định: việc sử dụng không gian tên mặc định cho một phần tử giúp chúng ta không phải sử dụng tiền tố trong các phần tử con.
- Cú pháp cho không gian tên mặc định

xmlns = " namespaceURI"

- Ví dụ:

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

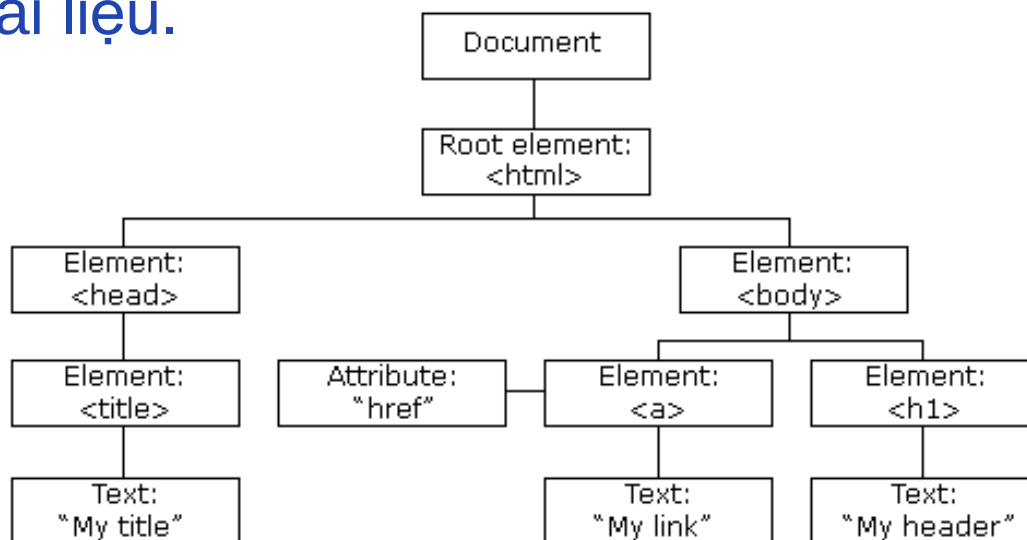
MÃ HÓA - ENCODING

- Tài liệu XML có thể chứa các ký tự như : êèé để tránh sai sót chúng ta cần phải xác định mã hóa sử dụng để lưu tập tin XML (UTF-8).
- UTF-8 là bảng mã ký tự chuẩn trên Web, là bảng mã mặc định cho HTML5, CSS, JavaScript, SQL, XML.
- Mã hóa trong XML

<?xml version =''1.0'' encoding = '' UTF-8'' ?>

TRUY XUẤT TÀI LIỆU XML VỚI DOM

- DOM (Document Object Model) là một giao diện lập trình ứng dụng (API). Thường DOM có dạng một cây cấu trúc dữ liệu, được dùng để truy xuất các tài liệu dạng HTML và XML. Mô hình DOM độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng đối tượng để mô tả tài liệu.



MỞ TÀI LIỆU XML

Các bước thực hiện

- Tạo đối tượng DOMDocument

```
$tai_lieu = new DOMDocument();
```

- Gọi phương thức load để mở tài liệu XML

```
$tai_lieu -> load(' file.xml ');
```

ĐỌC TÀI LIỆU XML

Các bước thực hiện

- Mở tài liệu XML
- Truy xuất node gốc

`$node_goc= $tai_lieu -> documentElement;`

- Truy xuất các node con
 - Dùng thuộc tính `childNodes` để truy xuất các node con
 - Mỗi Node là một đối tượng class `XMLElement` hoặc `text`
- Đọc nội dung node
 - Thuộc tính `nodeValue` trả về nội dung của node.
 - Dùng thuộc tính `nodeType` để kiểm tra 1 node là `text` hay `Element`

ĐỌC TÀI LIỆU XML

Các bước thực hiện

```
<?xml version="1.0"
encoding="utf-8"?>
<PHAN_SO>
    <TU_SO>2</TU_SO>
    <MAU_SO>3</MAU_SO>
</PHAN_SO>
```

```
//tạo đối tượng
$tai_lieu = new DOMDocument();
//mở tài liệu XML
$tai_lieu->load("PHAN_SO.xml");
//truy xuất node gốc
$node_goc = $tai_lieu->documentElement;
$mang = $node_goc->childNodes;

foreach($mang as $node)
{
    if($node->nodeType==XML_ELEMENT_NODE)
        echo $node->nodeValue;
}
```

ĐỌC TÀI LIỆU XML

- Đọc thuộc tính node sử dụng phương thức **getAttribute** của node.
- Đọc thuộc tính **tu_so** và **mau_so** của node **PHAN_SO**

```
<?xml version="1.0" encoding="utf-8"?>  
<PHAN_SO tu_so="2" mau_so="3" />
```

```
//tạo đối tượng  
$tai_lieu = new DOMDocument();  
//mở tài liệu XML  
$tai_lieu->load("PHAN_SO.xml");  
//truy xuất node gốc  
$node_goc = $tai_lieu->documentElement;  
$tu_so = $node_goc->getAttribute('tu_so');  
$mau_so = $node_goc->getAttribute('mau_so');
```

GHI TÀI LIỆU XML

- Tạo 1 node mới

```
$node = $tai_lieu->createElement($name)
```

Hoặc

```
$node = $tai_lieu->createTextNode($content)
```

- Tạo nội dung cho node

```
$node->nodeValue = $noi_dung;
```

- Thêm node mới vào node cha

```
$node_cha->appendChild($node_moi);
```

- Tạo thuộc tính cho node

```
$node->setAttribute('ten_thuoc_tinh', 'giá trị')
```

- Lưu tài liệu

```
$tai_lieu->save($duong_dan);
```

GHI TÀI LIỆU XML

- Ví dụ tạo tài liệu XML biểu diễn phân số

```
<?xml version="1.0"
encoding="utf-8"?>
```

```
<PHAN_SO>
```

```
    <TU_SO>2</TU_SO>
```

```
    <MAU_SO>3</MAU_SO>
```

```
</PHAN_SO>
```

```
//tạo đối tượng Dom
```

```
$tai_lieu = new DOMDocument('1.0', 'utf-8');
```

```
//tạo node PHAN_SO
```

```
$node_phan_so = $tai_lieu->createElement('PHAN_SO');
```

```
//gắn node PHAN_SO vào tài liệu xml
```

```
$tai_lieu->appendChild($node_phan_so);
```

```
//tạo node TU_SO và MAU_SO
```

```
$node_tu_so = $tai_lieu->createElement('TU_SO');
```

```
$node_mau_so = $tai_lieu->createElement('MAU_SO');
```

```
//Gán giá trị
```

```
$node_tu_so->nodeValue = 2;
```

```
$node_mau_so->nodeValue = 3;
```

```
//gắn node TU_SO và node MAU_SO vào node PHAN_SO
```

```
$node_phan_so->appendChild($node_tu_so);
```

```
$node_phan_so->appendChild($node_mau_so);
```

```
//lưu tài liệu
```

```
$tai_lieu->save('PHAN_SO.XML');
```

GHI TÀI LIỆU XML

- Ví dụ tạo tài liệu XML biểu diễn phân số

```
<?xml version="1.0"
encoding="utf-8"?>
<PHAN_SO tu_so="2"
        mau_so="3" />
```

```
//tạo đối tượng Dom
$toi_lieu = new DOMDocument('1.0', 'utf-
8');

//tạo node PHAN_SO
$node_ps =
    $toi_lieu->createElement('PHAN_SO');
//gắn node PHAN_SO vào tài liệu xml
$toi_lieu->appendChild($node_ps);
//tạo các thuộc tính tu_so và mau_so
$node_ps->setAttribute('tu_so', 2);
$node_ps->setAttribute('mau_so', 3);
//lưu tài liệu
$toi_lieu->save('PHAN_SO.XML');
```

SESSION

1. Khái niệm

- Session trong PHP cho phép lưu trữ thông tin người dùng trên trình duyệt (tên người dùng, danh mục hàng hóa,...) trong suốt phiên làm việc của họ.
- Session được dùng để lưu trữ thông tin trong biến và sử dụng nó từ trang này sang trang khác. Các biến thông thường trong PHP chỉ làm việc tại trang đó, sang trang khác sẽ mất, session thì không bị như thế.
- Biến session tồn tại cho đến khi người dùng đóng trình duyệt.
- Vì vậy biến session nắm giữ thông tin về một user nhất định và được sử dụng trên tất cả các trang trong ứng dụng đó

2. Cách hoạt động

- Session làm việc bằng cách tạo ra một địa chỉ duy nhất (UID) cho mỗi người sử dụng.
- UID có giá trị là một dãy số ngẫu nhiên
- UID có thể được lưu trong cookie hoặc được truyền lên URL
- Mỗi người truy cập đến web sẽ có một biến session khác nhau.

3. Khởi động Session

- Trước khi lưu trữ thông tin người dùng vào session, cần khởi động session.
- Hàm khởi động session phải đặt trên thẻ html
- Cú pháp:

```
session_start();
```

4. Đăng ký Session

- Sử dụng biến `$_SESSION` nhận và lưu trữ giá trị của biến session.
- Cú pháp:

`$_SESSION["tên biến session"] = "giá trị";`

- Ví dụ: Tạo biến session lưu tên đăng nhập của người dùng.

`$_SESSION["ho_ten"] = $_POST["ho_ten"];`

5. Hủy biến Session

- Khi không cần dùng biến session nữa có thể hủy biến session theo hai cách sau:
- Hủy toàn bộ các biến Session:

```
session_destroy();
```

- Hủy một biến Session:

```
Unset($_SESSION["Tên Biến Session"]);
```

COOKIES

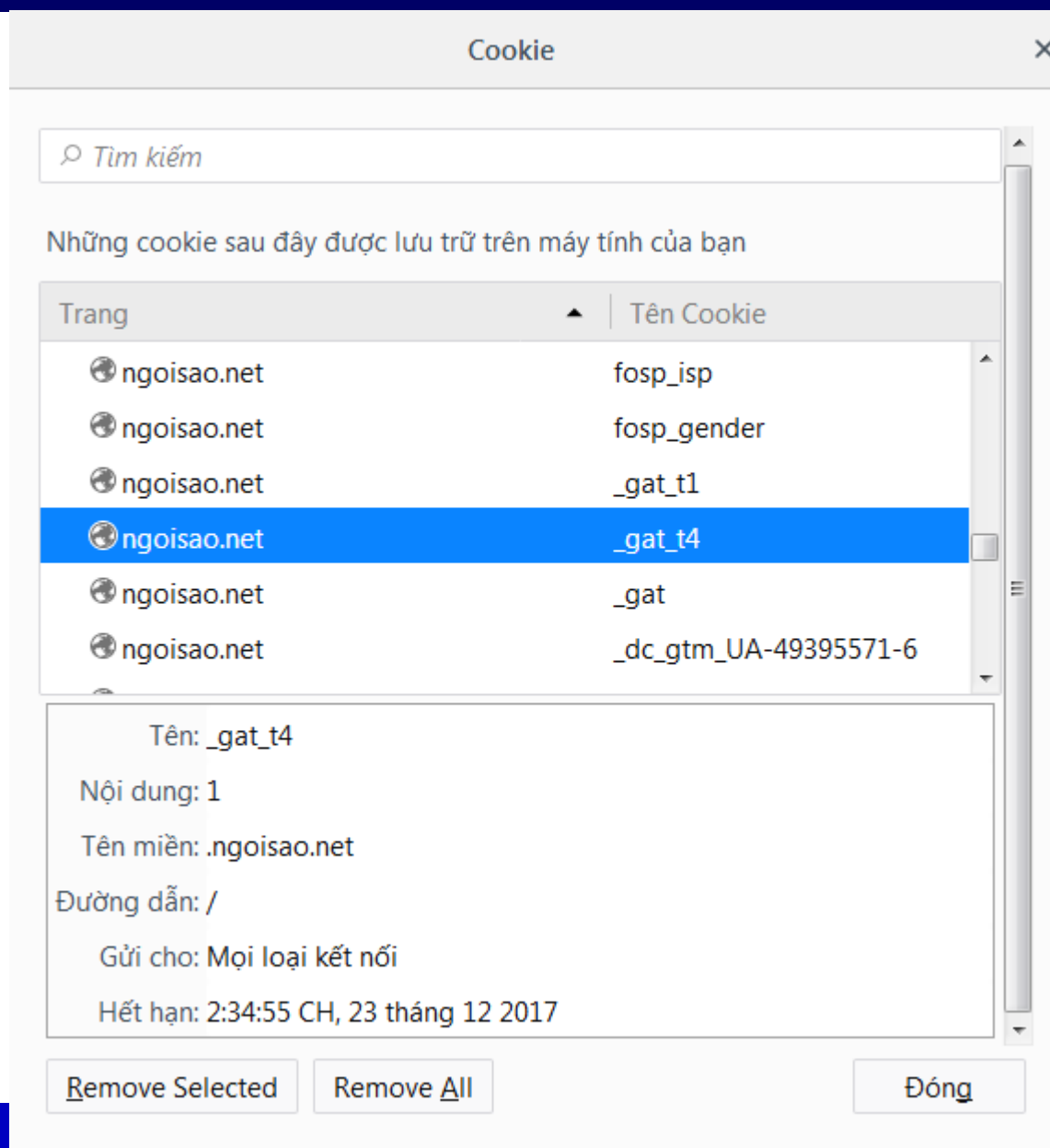
1. Khái niệm

- Cookies là một mẫu tin nhỏ, có cấu trúc, dạng text được lưu tự động vào đĩa cứng hoặc bộ nhớ của máy người dùng – client khi truy cập một website. Cookie thường được lưu

Mở web Browser → Alt → Tool (option) → Privacy & Security → Xóa các Cookie riêng lẻ

- Cookie được sử dụng để xác định thông tin của người dùng
- PHP có thể tạo ra và sử dụng giá trị của biến Cookie

COOKIES



COOKIES

2. Khai báo Cookie

- Khai báo Cookie được thực hiện phía trên thẻ html
- Cú pháp:

Setcookie (name, value, expire[, path, domain]);

- Name : tên biến cookie
- Value : giá trị
- Expire : thời gian giới hạn dành cho cookie - đvt là giây. Nếu thời gian này không được thiết lập trong setcookie thì biến cookie này sẽ có hiệu lực cho tới khi người dùng xóa tập tin cookie.
- Path : đường dẫn
- Domain là tên miền của web

COOKIES

3. Sử dụng Cookie

- Dùng biến `$_COOKIE` để đọc giá trị biến cookie
- Cú pháp:

```
$gia_tri = $_COOKIE[" tên biến cookie"] ;
```
- Ví dụ: đọc giá trị của biến cookie người dùng

COOKIES

4. Hủy Cookie

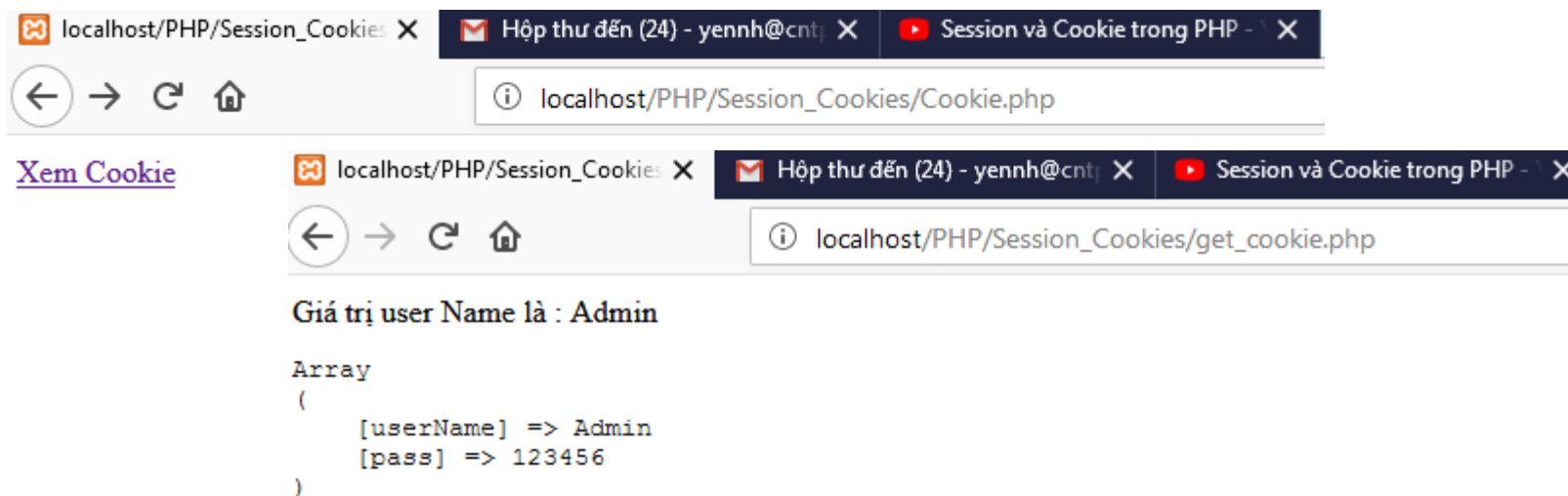
- Khi muốn hủy một biến cookie thì cần phải kiểm tra lại thời gian giới hạn dành cho biến cookie này.
- Sử dụng hàm `setcookie()` để hủy bằng cách đặt giá trị của biến cookie bằng `''` và thời gian bằng `-` thời gian giới hạn
- Cú pháp:
`Setcookie(name, '', time() - thời gian giới hạn) ;`
- Ví dụ: Hủy biến cookie người dùng với thời gian giới hạn của biến này là 3600s

VÍ DỤ VỀ SESSION - COOKIES

Ví dụ 1: Thiết lập một biến cookie có tên là 'userName' : 'Admin', tồn tại trong 10 phút. Thiết lập thêm 1 biến cookie có tên là 'pass': '123456' tồn tại trong 10 phút.

(1) Lấy giá trị biến cookie 'userName' xuất ra màn hình trình duyệt.

(2) Xuất tất cả các biến cookie ra màn hình trình duyệt



VÍ DỤ VỀ SESSION - COOKIES

Ví dụ 2: Thiết kế form đăng nhập .

THÔNG TIN ĐĂNG NHẬP

Họ và tên	Nguyễn Hải Yến
Email	yennh@cntp.edu.vn
Tên ĐN	yennh
Password	

ĐĂNG NHẬP

THÔNG TIN ĐĂNG NHẬP

Họ và tên	
Email	
Tên ĐN	
Password	

ĐĂNG NHẬP

YC:

- Tạo và gán giá trị cho các biến Session ?
- Đọc biến Session tại trang hiện hành ?
- Đọc biến Session tại một trang khác ?

Xin Chào Nguyễn Hải Yến
Email yennh@cntp.edu.vn
Tên Đăng Nhập **yennh**
Mật Khẩu 123456
[Sang trang kế tiếp](#)

VÍ DỤ VỀ SESSION - COOKIES



TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HỒ CHÍ MINH



Xin Chào Nguyễn Hải Yến
Email yennh@cntp.edu.vn
Tên Đăng Nhập yennh
Mật Khẩu 123456
[Trở về trang đăng Nhập](#)

THÔNG TIN ĐĂNG NHẬP

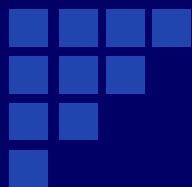
Họ và tên

Email

Tên ĐN

Password

ĐĂNG NHẬP



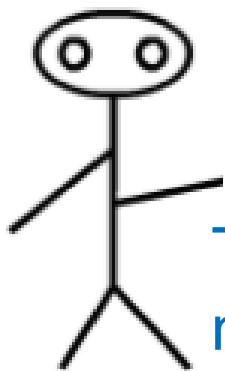
Q & A

Chương 3.

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG PHP

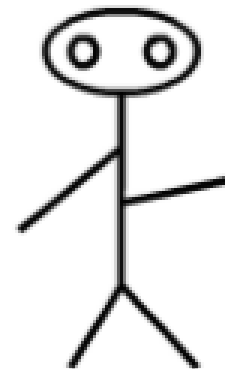
1. Lập trình hướng đối tượng trong PHP
2. Kết hợp PHP và Javascript

2.1.1. Tổng quan đối tượng



Thực hiện
nhiệm vụ f
trong thực tế
liên quan đến
đối tượng

Xây dựng
chương trình



Object
class

- Đối tượng (Object) là những **thực thể** tồn tại trong **thế giới thực**: Bàn, ghế, xe,...
- Là các **khái niệm** như: Khoa, phòng ban,...
- Có các thuộc tính và hành vi.

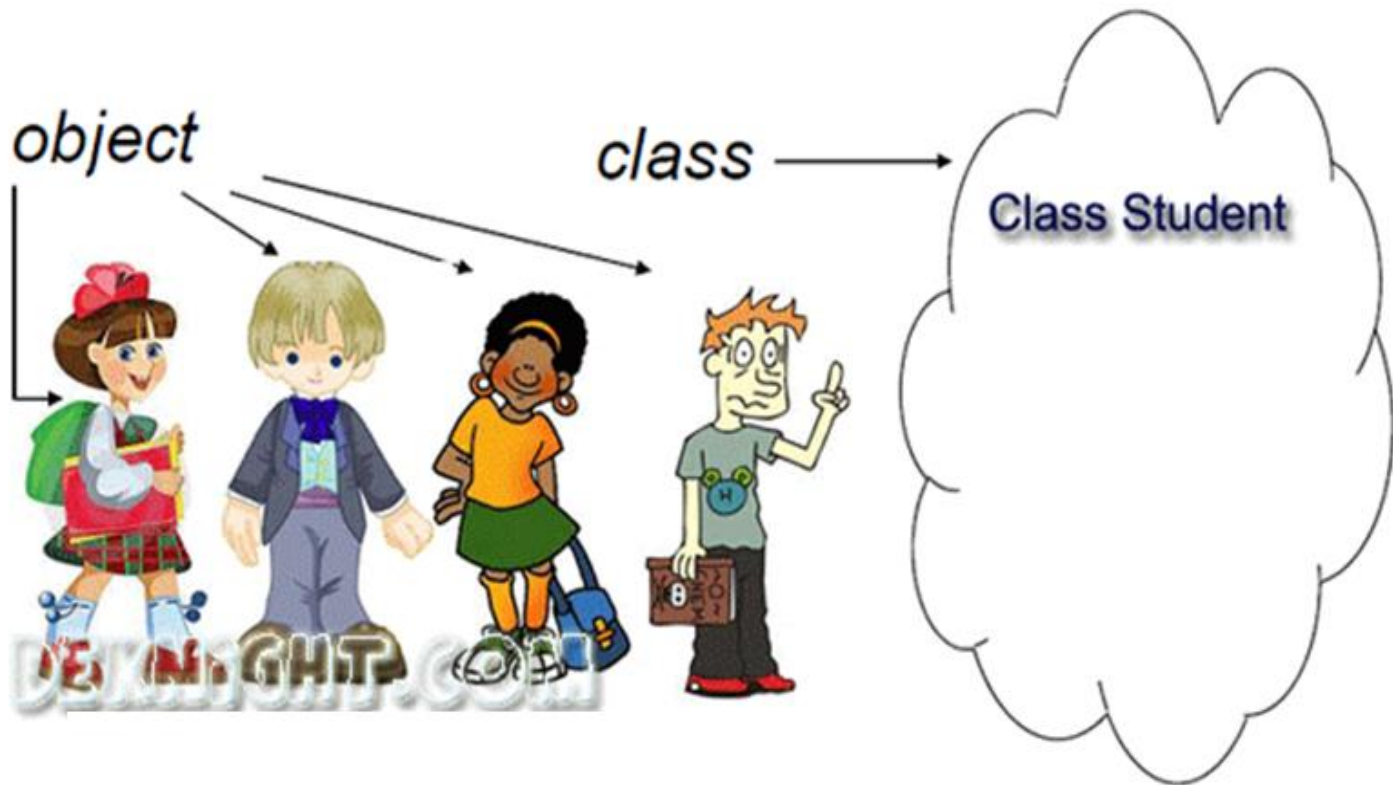
2.1.1. Tổng quan đối tượng (tt)

Lớp (Class):

- ❖ Các đối tượng có đặc tính tương tự nhau được gom chung lại thành lớp đối tượng. Ví dụ **NGƯỜI** là một lớp đối tượng.
- ❖ Một lớp đối tượng được đặc trưng bởi các **thuộc tính** và các **hoạt động** (hành vi, thao tác)

2.1.1. Tổng quan đối tượng (tt)

❖ Phân biệt giữa đối tượng và lớp



2.1.1. Tổng quan đối tượng (tt)

Các thành phần của lớp

- ❖ **Tên lớp:** duy nhất, dùng phân biệt lớp này với lớp khác trong cùng một phạm vi.
- ❖ **Thành phần dữ liệu, thuộc tính:** là các thành phần cấu tạo nên đối tượng, đặc trưng cho đối tượng.
- ❖ **Phương thức:** các hoạt động, của đối tượng thuộc lớp.

2.1.1. Tổng quan đối tượng (tt)

Ví dụ: Lớp Phan_So gồm các thành phần:

- Thuộc tính:

- Tử số
- Mẫu số

- Phương thức:

- Nhap(): nhập dữ liệu cho phân số
- Xuat (): xuất dữ liệu cho phân số
-

```
class PHAN_SO  
  
{  
  
    var $tu_so = 1;  
  
    var $mau_so = 1;  
  
    ...  
  
}
```

2.1.2. Xây dựng lớp đối tượng

- ❖ Tạo lớp
- ❖ Khai báo thuộc tính
- ❖ Xây dựng phương thức
- ❖ Minh họa

Tạo lớp

Cú pháp

```
<?php  
class <ten_class>{  
    //Khai báo các thuộc tính của lớp  
    //Gán và lấy giá trị của thuộc tính  
    //Các phương thức của lớp  
}  
?>
```

Trong đó: **ten_class** là tên class cần đặt

Khai báo thuộc tính

- ❖ Thuộc tính: thành phần lưu trữ các tính chất, đặc điểm của đối tượng.
- ❖ Cú pháp:

```
var <tên thuộc tính 1>;  
...
```

Chú ý: có thể thiết lập những giá trị mặc định ban đầu cho tất cả các thuộc tính được tạo ra từ lớp đó

Khai báo thuộc tính

❖ Ví dụ: khai báo lớp phân số

```
class PHAN_SO  
{  
  
    var $tu_so = 1;  
  
    var $mau_so = 1;  
  
    ...  
  
}
```

Xây dựng phương thức

- ❖ **Phương thức:** là chức năng mà đối tượng có thể thực hiện; có thể có hoặc không có giá trị trả về
- ❖ **Cú pháp:**

```
function <tên phương thức>(Các tham  
số truyền vào nếu có)  
{  
    //Khởi lệnh  
}
```

Trong đó:

- **Phạm_vi_pt:** phạm vi sử dụng phương thức
- **Quy tắc đặt tên phương thức:** như quy tắc đặt tên biến

Xây dựng phương thức

- ❖ Trong các phương thức của lớp, có thể truy cập các thuộc tính hay phương thức thông qua con trỏ \$this
- ❖ Con trỏ \$this được dùng để chỉ lớp hiện tại đang làm việc
- ❖ Ví dụ: nếu một thuộc tính của lớp phân số có tên \$tu_so thì có thể tham chiếu đến nó: \$this->tu_so

Xây dựng lớp phân số

```
class phan_so{
    //Khai báo các thuộc tính
    var $tu_so;
    var $mau_so;
    //Gán và lấy giá trị của các thuộc tính
    //Xây dựng các phương thức cần thiết cho lớp phân số
    function khai_tạo_phan_so($ptu_so,$pmau_so)
    {
        $this->$tu_so;
        $this->$mau_so;
    }
}
```

Xây dựng lớp phân số

```
//Xây dựng phương thức: tìm ước số chung lớn nhất  
function USCLN($a,$b)  
{  
    $so_nho=($a<$b)?$a:$b;  
    for ($i=$so_nho;$i>0;$i--)  
        if (($a%$i)==0 && ($b%$i)==0)  
        {  
            return $i;  
            break;  
        }  
}
```

Xây dựng lớp phân số

```
//Xây dựng phương thức: Tối giản phân số  
function Toi_gian_phan_so()  
{  
    $uscln=USCLN($this->tu_so,$this->mau_so);  
    $this->tu_so=$this->tu_so/$uscln;  
    $this->mau_so=$this->mau_so/$uscln;  
    return PHAN_SO;  
}
```

Xây dựng lớp phân số

```
//Xây dựng phương thức: Tính tổng hai phân số
function tong($ptu_so,$pmau_so)
{
    $phan_so=new PHAN_SO();
    $phan_so->khoi_tao_phan_so($ptu_so,$pmau_so);
    $phan_so->$tu_so=($this->tu_so*$phan_so->mau_so)
    +($phan_so->tu_so*$this->mau_so);
    $phan_so->mau_so=$this->mau_so*$phan_so->mau_so;
    $phan_so->Toi_gian_phan_so();
    return $phan_so;
}
```

Sử dụng lớp đối tượng

- ❖ Khởi tạo đối tượng
- ❖ Gán giá trị cho các thuộc tính của lớp
- ❖ Gọi sử dụng các phương thức của lớp

Khởi tạo đối tượng

Cú pháp:

```
$<Tên biến đối tượng>=new <tên lớp>();
```

Ví dụ:

```
$phan_so=new PHAN_SO();
```

Gán giá trị cho các thuộc tính của lớp

❖ Cú pháp:

❖ $\$ \langle \text{Tên biến đối tượng} \rangle \rightarrow \langle \text{Tên thuộc tính} \rangle = \langle \text{giá trị} \rangle$

```
$phan_so->tu_so=2;  
$phan_so->mau_so=4;
```

Gọi sử dụng phương thức của lớp

❖ Phương thức **không có** giá trị trả về

❖ Cú pháp:

❖ Ví dụ:

`$<Tên biến đối tượng>-><tên phương thức>(Các giá trị truyền vào nếu có);`

```
$phan_so->khoidao_phan_so($ptu_so, $pmau_so);
```

Sử dụng phương thức của lớp có giá trị trả về:

❖ Cú pháp:

\$<Tên biến nhận giá trị> -><tên biến đối tượng>
-><Tên phương thức> (Các giá trị truyền vào nếu có);

```
//Phương thức có giá trị trả về  
$phan_so_2=new PHAN_SO;  
$phan_so_2=$phan_so->tong($a,$b);
```

Phạm vi sử dụng của biến, phương thức:

- ❖ **protected**: biến có phạm vi là protected được sử dụng trong class cha (class cho các class khác kế thừa/sử dụng lại) và class con (class được kế thừa từ class cha/được sử dụng lại lớp cha), ngoài 2 class này không thể sử dụng các biến/phương thức mang phạm vi protected.
- ❖ **private**: với kiểu phạm vi này, các biến/phương thức chỉ có thể sử dụng trong chính class chứa các phương thức/biến có phạm vi private.
- ❖ Nếu không khai báo phạm vi biến/phương thức. Mặc định sẽ được hiểu phạm vi là public.

Ví dụ

- ❖ Khai báo phạm vi sử dụng của phương thức:

```
public                                     function  
danh_sach_sv_hufi(){...}
```

- ❖ Khai báo phạm vi sử dụng của biến:

```
private $ten_sv;
```

Ví dụ 2: phương thức có phạm vi là public:

```
<?php
    class hufi{
        public function sinhvien(){
            return "Đây là sinh viên HUFI";
        }
    }
    //khởi tạo class hufi và gán vào biến $cntp:
    $cntp = new hufi;
    //sử dụng hàm sinhvien() trong class hufi:
    echo $cntp->sinhvien();
?>
```

Ví dụ 3: sử dụng biến có phạm vi là public ở ngoài class

```
<?php
class hufi{
    public $mon_hoc = "Môn học lập trình mã nguồn mở";
    public function sinhvien(){
        return "Đây là sinh viên HUFI";
    }
}
//khởi tạo class hufi và gán vào biến $cntp:
$cntp = new hufi;
//sử dụng biến $khoa_hoc trong class devpro:
echo $cntp->mon_hoc;
?>
```

Lưu ý: biến bắt đầu bằng ký tự '\$' nhưng khi gọi, không ghi ký tự đô la vào.

Ví dụ 3: biến có phạm vi là public ở ngay trong class nhưng ở một vị trí khác:

```
<?php
    class hufi{
        //đặt phạm vi của biến $mon_hoc là public
        public $mon_hoc = "Môn học lập trình mã nguồn mở";
        //sử dụng biến $mon_hoc vào function tenmonhoc() bằng từ khóa $this
        public function tenmonhoc(){
            return $this->mon_hoc;
        }
    }
    //sử dụng hàm tenmonhoc() trong class hufi
    $cntp = new hufi;
    echo $cntp->tenmonhoc();
?>
```

Trong đó: **\$this** đại diện cho class bao ngoài, **mon_hoc** là tên biến **\$mon_hoc**.

Ví dụ 4: phương thức có phạm vi static trong class:

```
<?php
    class hufi{
        //đặt phạm vi của hàm sinhvien() là static
        static function sinhvien(){
            return "Đây là sinh viên HUFİ";
        }
    }
    //sử dụng hàm SINHvien() trong class hufi
    echo hufi::sinhvien();
?>
```

Khi khai báo phạm vi là static chúng ta không cần khởi tạo class, mà có thể sử dụng luôn.

Cú pháp: `tên_class::phương_thức()`;

Tên class và tên phương thức sẽ được ngăn cách bởi 2 dấu 2 chấm.

Ví dụ 5: sử dụng biến có phạm vi static ngoài class

```
<?php
class hufi{
    //đặt phạm vi của biến $monhoc là static
    static $monhoc = "Môn học lập trình mã nguồn mở";
}
//sử dụng biến $monhoc trong class hufi
echo hufi::$monhoc;
?>
```

Lưu ý: biến **\$monhoc** được đặt static → khi sử dụng phải ghi theo cú pháp **tên_class::\$tên_biến**;
Khác với khi sử dụng biến có phạm vi là public khi gọi tên biến không có ký tự đô la '\$', **khi gọi biến static bắt buộc phải có dấu đô la.**

Ví dụ 6: sử dụng biến có phạm vi static ở trong class

```
<?php
class hufi{
    //đặt phạm vi của biến $monhoc là static
    static $monhoc = "Môn học lập trình mã nguồn mở";
    //gọi biến có phạm vi static vào hàm (cùng 1 class) bằng từ khóa self
    static function tenmonhoc(){
        return self::$monhoc;
    }
}
//sử dụng hàm tenmonhoc( trong class devpro
echo hufi::tenmonhoc();
?>
```

Khi sử dụng biến có phạm vi là static ở bất kỳ vị trí nào trong class, phải thêm từ khóa self trước tên biến.

Cú pháp: **self::\$tên_biến;**

Ví dụ 7: Sử dụng phương thức, biến có phạm vi là protected ở class con và class cha:

```
<?php
class hufi{
    //khai báo biến $hocphi là protected
    protected $hocphi = "4,000,000 vnd";
}
class hocphiLTMMN extends hufi{
    public function giahocphi(){
        return $this->hocphi;
    }
}
//khởi tạo class hocphiLTMMN và gán vào biến $dev
$dev = new hocphiLTMMN;
//in ra giá trị biến $hocphi ở class cha (class hufi)
echo $dev->giahocphi();
?>
```



Ví dụ 8:



```
<?php
    class hufi{
        //khai báo biến $hocphi là private
        private $hocphi = "4,000,000 vnd";
    }
    //khởi tạo class hufi và gán vào biến $dev
    $dev = new hufi;
    //in ra giá trị biến $hocphi
    echo $dev->hocphi;
?>
```

Ví dụ 9:

```
<?php
class hufi{
    //khai báo biến $hocphi là private
    private $hocphi = "4,000,000 vnd";
    //tạo một hàm có phạm vi là public để sử dụng biến $hocphi (private)
    public function layhocphi(){
        return $this->hocphi;
    }
}
//khởi tạo class hufi và gán vào biến $dev
$dev = new hufi;
//in ra giá trị của hàm layhocphi()
echo $dev->layhocphi();
?>
```

Kế thừa trong PHP

- ❖ Kế thừa trong php được hiểu là class này có thể sử dụng lại các thuộc tính, phương thức của class khác mà không cần viết lại từ đầu. Mỗi thuộc tính, phương thức chỉ cần viết một lần duy nhất và có thể chia sẻ thông tin với class khác nhờ sự kế thừa.
- ❖ Class cho các class khác kế thừa được gọi là class cha, class được kế thừa gọi là class con. Ngôn ngữ PHP cho phép đa kế thừa, tức là class A có thể kế thừa từ class B, class B có thể kế thừa từ class C....

Kế thừa trong PHP (tt)

- ❖ Khi class A kế thừa từ class B, class A sẽ có tất cả các thuộc tính, phương thức của class B, ngoài ra class A còn có thể có các phương thức, thuộc tính mới hoặc loại bỏ đi một số thuộc tính, phương thức của class B. Tôi gọi đây là kế thừa có chọn lọc. Class A chỉ lấy những cái mình cần từ class B, và tạo nên các thuộc tính mới, phương thức mới, thể hiện cá tính và chức năng của mình.
- ❖ Không phải class nào cũng cho class khác kế thừa, để quy định điều này chúng ta sử dụng các từ khóa mà PHP đã định sẵn.

Cú pháp kế thừa trong PHP:

Class A extends class B{...}

Trong đó “extends” là từ khóa khai báo có sự kế thừa.

Ví dụ:

- ❖ Có class xe, class xe là class quản lý chung tất cả các loại xe (xe đạp, xe máy, ô tô,...)
- ❖ Class xe chứa các đặc điểm chung mà tất cả các loại xe đều có: bánh xe, màu sắc, vận tốc.
- ❖ Có 1 class là xe máy kế thừa từ class xe → không cần viết lại bánh xe, màu sắc, và vận tốc nữa, và có thể dùng luôn những thứ đã khai báo ở class xe.

/í dụ:

```
//class xe là class cha:
class xe{
    public $mausac = "Màu vàng";
    public function banhxe(){
        return "Bánh xe làm bằng kim cương";
    }
    public function vantoc(){
        return "Vận tốc mà tôi đạt được là 2000km/h";
    }
}
```

```
//class xe máy là class con:
class xemay extends xe{
    public function mausackemay(){
        return $this->mausac;
    }
}
```

```
//khởi tạo class xe máy:
$xml = new xemay;
//in ra màu sắc của xe máy:
echo $xml->mausackemay();
//in ra vận tốc mà xe máy có thể đạt được:
echo "<br>".$xml->vantoc();
```

Class “xemay” không khai báo phương thức nào hết, nhưng lúc gọi hàm vẫn gọi được
Lý do: class “xemay” đã được kế thừa từ class “xe” và có tất cả những gì class “xe” có.

Nạp chồng (ghi đè):

- ❖ Nạp chồng sử dụng khi class con muốn thay đổi các thuộc tính, phương thức của lớp cha.
- ❖ Ví dụ ở trên, vận tốc của xe máy thì sẽ in ra vận tốc là 2000km/h, bây giờ sửa lại vận tốc này → phải nạp chồng (ghi đè) lại phương thức vận tốc bằng cách tạo 1 function mới trong class con có tên trùng với function muốn ghi đè ở lớp cha.

```
//class xe là class cha:
class xe{
    public $mausac = "Màu vàng";
    public function banhxe(){
        return "Bánh xe làm bằng kim cương";
    }
    public function vantoc(){
        return "Vận tốc mà tôi đạt được là 2000km/h";
    }
}
```

```
//class xe máy là class con:
class xemay extends xe{
    public function mausacxemay(){
        return $this->mausac;
    }
    public function vantoc(){
        return "Tôi là xe máy và vận tốc mà tôi đạt được là 5000km/h";
    }
}
```

Viết một hàm **vantoc**, ở class con và hàm này trùng tên với 1 hàm có trong class cha → vận tốc của xe máy bây giờ là 5000.

```
//khởi tạo class xe máy:
$xml = new xemay;
//in ra màu sắc của xe máy:
echo $xml->mausacxemay();
//in ra vận tốc mà xe máy có thể đạt được:
echo "<br>".$xml->vantoc();
```

Sử dụng từ khóa parent

- ❖ Trong trường hợp muốn khai báo thêm thuộc tính, phương thức ở class con nhưng vẫn sử dụng những cái ở class cha
 - ➔ Sử dụng từ khóa parent để thực thi thuộc tính/phương thức ở class cha, sau đó ghi thêm những thuộc tính, phương thức mà class con có.

```
//class xe là class cha:
class xe{
    public $mausac = "Màu vàng";
    public function banhxe(){
        return "Bánh xe làm bằng kim cương";
    }
    public function vantoc(){
        echo "Vận tốc của tất cả các loại xe ở Devpro là 2000km/h";
    }
}

//class xe máy là class con:
class xemay extends xe{
    public function vantoc(){
        parent::vantoc();
        return "Tôi là xe máy và vận tốc của tôi là 9000km/h";
    }
}

//khởi tạo class xe máy:
$xml = new xemay;
//in ra vận tốc mà xe máy có thể đạt được:
echo "<br>".$xml->vantoc();
```

Không cho class khác kế thừa:

- ❖ Để không cho class khác kế thừa: thêm từ khóa final ở ngay trước. Ví dụ:

```
final class xe(){...}
```

```
final function vantoc(){...}
```

- ❖ Tại sao phải chống kế thừa: khi kế thừa, class con có khả năng thay đổi, bổ sung lại nội dung của class cha. Những file như file kết nối database (có lưu mật khẩu, tài khoản,...). Chúng ta phải chống kế thừa, tránh việc tài khoản mật khẩu bị thay đổi.

Lớp Abstract

- ❖ Abstract là một lớp trừu tượng.
- ❖ Lớp trừu tượng là lớp chứa các thuộc tính/phương thức trừu tượng.
- ❖ Lớp trừu tượng, thuộc tính trừu tượng, phương thức trừu tượng bắt đầu bằng từ khóa abstract.

Lớp Abstract (tt)

Ví dụ 1

```
//class xe là class cha:
abstract class xe{
    abstract function banhxe();
}

//class xe máy là class con:
class xemay extends xe{
}

//khởi tạo class xe máy:
$xml = new xemay;
//in ra bánh xe của xe máy:
echo $xml->banhxe();
//kết quả sẽ xảy ra lỗi do chưa định nghĩa lại banhxe()
```

Lớp Abstract (tt)

Ví dụ 2

```
//class xe là class cha:
abstract class xe{
    abstract function banhxe();
}
//class xe máy là class con:
class xemay extends xe{
    public function banhxe(){
        return "Bánh xe của xe máy làm bằng bạch kim :v";
    }
}
//khởi tạo class xe máy:
$xml = new xemay;
//in ra bánh xe của xe máy:
echo $xml->banhxe();
//kết quả sẽ không có lỗi
```

Lớp interface

- ❖ Về bản chất lớp interface cũng là một lớp có khả năng cho lớp khác kế thừa. Khác với abstract, lớp interface không được phép có các thuộc tính/phương thức bình thường.
- ❖ Lớp con của lớp interface cũng bắt buộc phải định nghĩa lại các thuộc tính/phương thức của class cha. Nếu không định nghĩa lại sẽ gặp lỗi.
- ❖ Lớp con kế thừa từ lớp cha không sử dụng từ khóa extends mà dùng từ khóa implements.

Lớp interface

Ví dụ 1

```
//interface xe là class cha:
interface xe{
    public function banhxe();
}
//class xe máy là class con:
class xemay implements xe{
}
//khởi tạo class xe máy:
$xml = new xemay;
//in ra bánh xe của xe máy:
echo $xml->banhxe();
//kết quả sẽ xảy ra lỗi
```

Lớp interface

Ví dụ 2

```
//interface xe là class cha:
interface xe{
    public function banhxe();
}
//class xe máy là class con:
class xemay implements xe{
    //định nghĩa lại phương thức bánh xe của interface
    public function banhxe(){
        return "Bánh xe của xe máy làm bằng bạch kim :v";
    }
}
//khởi tạo class xe máy:
$xml = new xemay;

//in ra bánh xe của xe máy:
echo $xml->banhxe();
//kết quả sẽ không có lỗi
```

2.2. Kết hợp PHP và JavaScript trong PHP

- 2.2.1. Tổng quan về JavaScript
- 2.2.2. Sử dụng JavaScript trong trang HTML
- 2.2.3. Ngôn ngữ lập trình JavaScript
- 2.2.4. Kết hợp PHP và JavaScript

2.2.1. Tổng quan về JavaScript

- ❖ JavaScript là một ngôn ngữ kịch bản được hãng Sun Microsystems và Netscape phát triển
- ❖ JavaScript là ngôn ngữ kịch bản dùng để tạo các client-side scripts.
- ❖ JavaScript làm cho việc tạo các trang Web động và tương tác dễ dàng hơn
- ❖ JavaScript được phát triển từ Livescript của Netscape

2.2.1. Tổng quan về JavaScript (tt)

- ❖ JavaScript có thể tăng cường tính động và tính tương tác của các trang web.
 - Cung cấp sự tương tác người dùng
 - Thay đổi nội dung động
 - Xác nhận tính hợp lệ của dữ liệu

2.2.1. Tổng quan về JavaScript (tt)

❖ Trình duyệt hỗ trợ:

- Netscape Navigator: bắt đầu phiên bản 2.0
- Microsoft Internet Explorer: bắt đầu từ phiên bản 3.0
- Những trình duyệt khác có hỗ trợ JavaScript: Firefox, Chrome, Opera, . . .

2.2.1. Tổng quan về JavaScript

Các bước thực thi của JS:

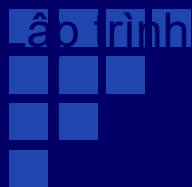
1. Trình duyệt tải trang web về
2. Trình duyệt kiểm tra xem có mã JS trong web hay không
3. Nếu có, trình duyệt sẽ chuyển mã JS cho bộ thông dịch
4. Bộ thông dịch xử lý và thực thi các mã lệnh JS
5. Các mã lệnh có thể tác động đến các thành phần của trang web.
6. Trình duyệt hiển thị toàn bộ nội dung web.

2.2.1. Tổng quan về JavaScript (tt)

- ❖ Các công cụ sinh mã JavaScript:
 - Thuận lợi khi soạn thảo
 - Mã lệnh sẵn có
- ❖ Môi trường thực thi:
 - Các Scripting ở Client
 - Java Script trên Web Server

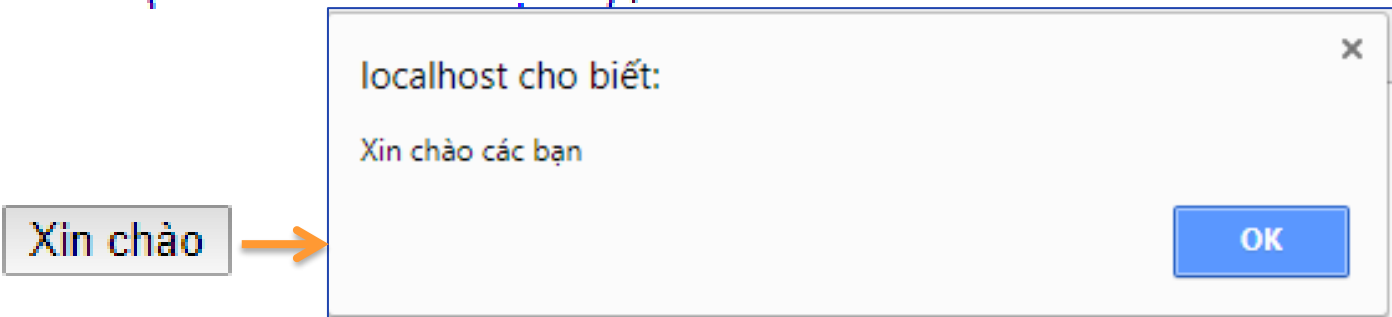
2.2.2. Sử dụng JavaScript trong trang HTML

- ❖ Những JavaScript trong thẻ `<head></head>`
- ❖ Những JavaScript trong thẻ `<body></body>`
- ❖ Những JavaScript trong thẻ `<body></body>` và những JavaScript trong thẻ `<head></head>`
- ❖ Những file JavaScript trong thẻ `<head></head>`.



Nhúng JavaScript trong thẻ <head></head>

```
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5   <title>Untitled Document</title>
6   <script type="text/javascript">
7       function xinChao() {
8           alert("Xin chào các bạn!");
9       }
10  </script>
11 </head>
12 <body>
13   <input type="button" onclick="xinChao();" value="Xin chào" />
14 </body>
```



Nhúng JavaScript trong thẻ

`<body></body>`

- ❖ Khi cần một script để tải nội dung trong trang → nhúng JavaScript trong thẻ `<body></body>`:
- ❖ Trong trường hợp này, không có bất kỳ phương thức JavaScript nào được định nghĩa, đoạn code nhúng JavaScript trong thẻ `<body></body>`:

```
<body>
  <script type="text/javascript">
    document.write("Xin chào các bạn!");
  </script>
  <p>Cách nhúng JavaScript vào HTML</p>
</body>
```

Nhúng JavaScript trong thẻ <body></body> và nhúng JavaScript trong thẻ <head></head>

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Nhúng JavaScript trong thẻ HTML</title>
  <script type="text/javascript">
    function xinChao() {
      alert("Xin chào các bạn");
    }
  </script>
</head>
<body>
  <script type="text/javascript">
    document.write("Xin chào các bạn!");
  </script>
  <p>Cách nhúng JavaScript vào HTML</p>
  <input type="button" onclick="xinChao()" value="Xin chào" />
</body>
```

Xin chào các bạn!

Cách nhúng JavaScript vào HTML

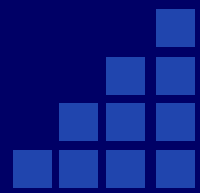
Xin chào



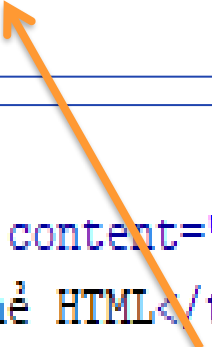
Nhúng file JavaScript trong thẻ <head></head>:

- ❖ Khi viết một ứng dụng web có những file JavaScript được dùng chung cho nhiều trang được viết trong một file và lưu với phần mở rộng .js
- ❖ Sử dụng thuộc tính src (Source) của tag <script> để khai báo tập tin *.js

```
<script type="text/javascript" src="tenFile.js" ></script>
```



```
function xinChao() {  
    alert("Chúc mừng bạn đã nhúng file thành công");  
}
```



```
<head>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <title>Nhúng JavaScript trong thẻ HTML</title>  
    <script type="text/javascript" src="tenFile.js" ></script>  
</head>  
<body>  
    <script type="text/javascript">  
        document.write("Xin chào các bạn!");  
    </script>  
    <p>Cách nhúng file JavaScript vào HTML</p>  
    <input type="button" onclick="xinChao()" value="Xin chào" />  
  
</body>
```

2 2 3 Ngôn ngữ lập trình JavaScript

- ❖ Chú thích
- ❖ Biến
- ❖ Hàm
- ❖ Một số hàm cơ bản trong JavaScript

Chú thích

Viết chú thích trên một dòng: đặt cặp dấu // phía trước nó

Viết chú thích trên nhiều dòng: nội dung chú thích trải dài trên nhiều dòng → dùng cặp dấu /* */ để bao quanh chúng.

Biến

- ❖ Khai báo: `Var <Tên biến>`
Hoặc `var <Tên biến>=<Giá trị>;`
Hoặc `<Tên biến>=<Giá trị>;`
- ❖ Ví dụ:

```
<script>  
  var a, b, c;  
  a = "Nguyễn Văn Nam";  
  b = 1999;  
  c= 2018-b;  
  d=a+ ", Số tuổi:"+ c;  
</script>
```



Giá trị của biến a là: Nguyễn Văn Nam
Giá trị của biến b là: 1999
Giá trị của biến c là: 19
Giá trị của biến d là: Nguyễn Văn Nam, Số tuổi:19

Quy tắc đặt tên biến

- ❖ Tên biến bao gồm: chữ hoa, chữ thường, chữ số, dấu gạch dưới, dấu đô la (khuyến khích **KHÔNG NÊN** sử dụng dấu \$)
- ❖ Tên biến tuyệt đối không được phép chứa các ký tự đặt biệt (*Ví dụ như: @, #, !, %, ^, &,*)
- ❖ **Tên biến không được bắt đầu bằng một chữ số.**
- ❖ Ví dụ:
- ❖ Tên biến đúng: webcoban, Webcoban, webCobAn, we9co3an
_webc_oban.
- ❖ Tên biến sai: 9webcoban, web%^coban

Giá trị của biến

- ❖ Giá trị của biến rất đa dạng: *Một chuỗi ký tự, một số, một biến khác, một biểu thức, undefined, null, . . .*
- ❖ Phân biệt giá trị null và undefined:
 - null: biến được khai báo và được gán =null
 - undefined: biến được khai báo nhưng chưa được gán giá trị

```
var test;
```

→ undefined

```
var test=null;
```

→ null

Hàm

❖ Khái báo: hàm có thể được khai báo trong tag <head> hoặc tag <body> hoặc trong tập tin *.js

❖ Cú pháp:

```
function <Tên hàm> ([tham số 1, tham số 2, . .  
.])  
{  
    /*Khối lệnh xử lý*/  
    [return <giá trị trả về>]
```

❖ Ví dụ:

```
function dt_hinhvuong(canh)  
{  
    return canh*canh;  
}
```

Phân loại hàm

- ❖ Có 2 loại: *hàm không có tham số*, *hàm có tham số*
- ❖ *Hàm không có tham số*: hàm mà kết quả thực thi của nó luôn luôn không thay đổi.
- ❖ *Hàm có tham số*: hàm mà khi gọi hàm ta phải truyền giá trị vào cho nó. Tùy vào giá trị được truyền mà hàm sẽ thực thi và cho ra kết quả khác nhau.

Phân loại hàm

//Hàm không có tham số:

```
function GioiThieuBanThan() {
```

```
    var name = "Nam";
```

```
    var year = 1999;
```

```
    document.write("Tôi là " + name + " sinh năm " + year);
```

```
}
```

//Hàm có tham số

```
function GioiThieuBanThan(x,y) {
```

```
    var name = x;
```

```
    var year = y;
```

```
    document.write("Tôi là " + name + " sinh năm " + year);
```

```
}
```

Gọi hàm

<Tên hàm> ([tham số 1, tham số 2, ...])

```
function GioiThieuBanThan(name, year){  
    document.write("Tôi là " + name + " sinh năm " + year);  
}  
GioiThieuBanThan();  
document.write("<br>");  
GioiThieuBanThan("Nguyễn Văn Nam");  
document.write("<br>");  
GioiThieuBanThan("Lê Thị Hồng", 1999);  
document.write("<br>");  
GioiThieuBanThan("Trần Ngọc Linh", 1998);
```



Tôi là undefined sinh năm undefined
Tôi là Nguyễn Văn Nam sinh năm undefined
Tôi là Lê Thị Hồng sinh năm 1999
Tôi là Trần Ngọc Linh sinh năm 1998

Gọi hàm

```
function GioiThieuBanThan(name="Nguyễn Văn Lê", year=1990) {  
    document.write("Tôi là " + name + " sinh năm " + year);  
}  
GioiThieuBanThan();  
document.write("<br>");  
GioiThieuBanThan("Nguyễn Văn Nam");  
document.write("<br>");  
GioiThieuBanThan("Lê Thị Hồng", 1999);  
document.write("<br>");  
GioiThieuBanThan("Trần Ngọc Linh", 1998);
```



Tôi là Nguyễn Văn Lê sinh năm 1990
Tôi là Nguyễn Văn Nam sinh năm 1990
Tôi là Lê Thị Hồng sinh năm 1999
Tôi là Trần Ngọc Linh sinh năm 1998

Gọi hàm thông qua một sự kiện

```
<button type="button" onclick="GioiThieu()">Bấm vào đây</button>  
<script>  
    function GioiThieu(){  
        alert("Tôi là Nguyễn Văn Nam sinh năm 1999");  
    }  
</script>
```

Bấm vào đây

localhost cho biết:

Tôi là Nguyễn Văn Nam sinh năm 1999

OK

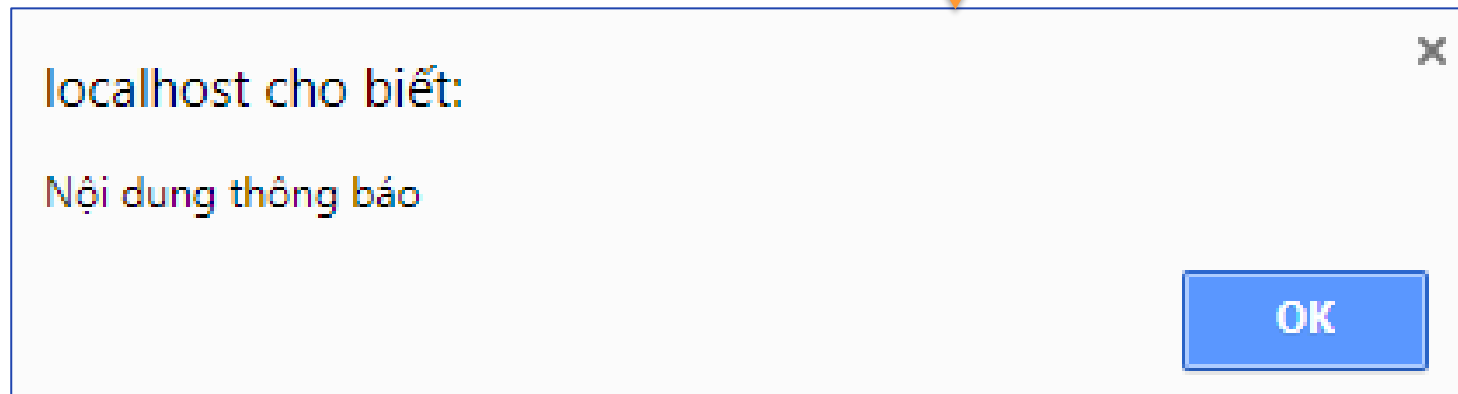
Một số hàm thông dụng

- ❖ **Hàm alert():** Hiện thị hộp thoại với một chuỗi thông tin và nút OK.
- ❖ **Hàm confirm():** Hiện thị hộp thoại với nút OK và Cancel. Thường dùng để xác minh lại một
`alert("Tôi là Nguyễn Văn Nam sinh năm 1999");`

Một số hàm thông dụng

- ❖ Hàm **alert()**: Hiện thị hộp thoại với một chuỗi thông tin và nút **OK**.
- ❖ Cú pháp:

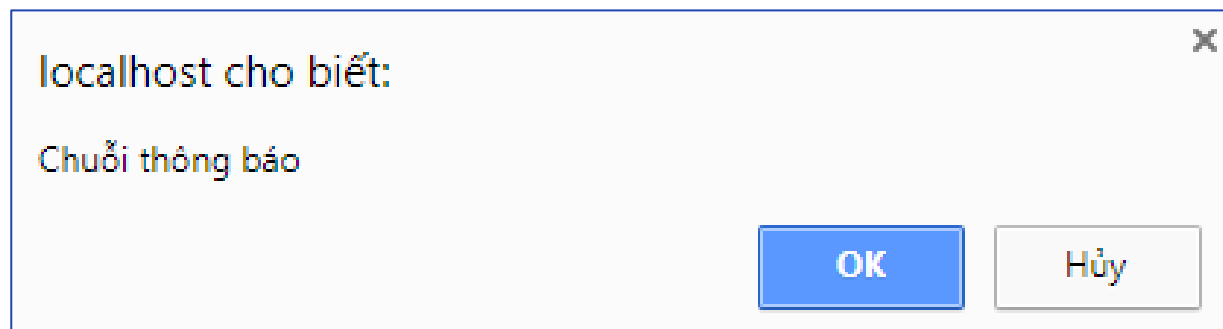
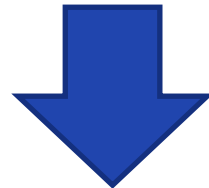
```
alert("Nội dung thông báo");
```



Một số hàm thông dụng (tt)

- ❖ Hàm **confirm()**: Hiển thị hộp thoại với nút **OK** và **Cancel (Huỷ)**. Thường dùng để xác minh lại một hành động do người dùng thực hiện. Cú pháp:

```
confirm("Chuỗi thông báo") ;
```



Một số hàm thông dụng (tt)

- ❖ Hàm **prompt()**: Hiển thị một hộp thoại cho phép người dùng nhập vào một giá trị. Cú pháp:

```
prompt("nội dung", giá trị khởi tạo)
```

```
document.write( prompt("Nhập", "Tên bạn là gì?"));
```



Một số hàm thông dụng (tt)

- ❖ Hàm **eval()**: định giá trị của một biểu thức và trả về kết quả.

```
a = eval(prompt("Nhập số a: "));  
b = eval(prompt("Nhập số b: "));  
c = a + b;  
alert("Kết quả: " + a + " + " + b + " = " + c);
```

localhost cho biết: x

Nhập số a:

OK Hủy

localhost cho biết: x

Nhập số b:

OK Hủy

localhost cho biết: x

Kết quả: 6 + 7 = 13

OK

Một số hàm thông dụng (tt)

- ❖ Hàm **parseInt()**: chuyển một giá chuỗi sang một giá trị số nguyên.
- ❖ Hàm **parseFloat ()**: chuyển một chuỗi số sang giá trị thực

```
var kq= "12345abcde";  
a= parseInt(kq);  
document.write(a);  
var kq1= "abcde12345";  
b= parseInt(kq1);  
document.write(b);  
var kqF= "12.345abcde";  
d= parseFloat(kqF);  
document.write(d);
```

→ 12345

→ NaN

→ 12.345

Một số hàm thông dụng (tt)

- ❖ Hàm **isNaN()**: kiểm tra xem giá trị là chuỗi hay chuỗi số, nếu là chuỗi thì kết quả trả về là **True**, ngược lại là chuỗi số sẽ là **False**

```
var kq = "12345abcd";  
a = isNaN(kq);  
document.write(a); → True  
  
var kq = "12345";  
b = isNaN(kq);  
document.write(b); → false
```

Các sự kiện trong JavaScript

Tên sự kiện	Mô tả
onclick	Sự kiện xảy ra khi người dùng click chuột vào phần tử
ondblclick	Sự kiện xảy ra khi người dùng click kép chuột vào phần tử
onmouseenter	Sự kiện xảy ra khi người dùng di chuyển con trỏ vào phần tử
onmouseleave	Sự kiện xảy ra khi người dùng di chuyển con trỏ ra khỏi phần tử.
onkeydown	Sự kiện xảy ra khi người dùng đang nhấn một phím

Các sự kiện trong JavaScript (tt)

Tên sự kiện	Mô tả
onkeyup	Sự kiện xảy ra khi người dùng nhấn phím ra
oncopy	Sự kiện xảy ra khi người dùng sao chép nội dung của phần tử
oncut	Sự kiện xảy ra khi người dùng cắt nội dung của phần tử
onpaste	Sự kiện xảy ra khi người dùng dán nội dung vào phần tử
onchange	Sự kiện xảy ra khi người dùng thay đổi giá trị của phần tử

Cách bắt sự kiện trong JavaScript

Trước khi bắt sự kiện thì ta cần phải xác định rõ ba thành phần:

- (1) Phần tử dùng để xảy ra sự kiện.
- (2) Sự kiện sẽ xảy ra.
- (3) Đoạn mã sẽ được thực thi khi sự kiện xảy ra.

Cú pháp bắt sự kiện:

<Tên phần tử **Tên sự kiện**="đoạn mã sẽ được thực thi khi sự kiện xảy ra">

```
<button onclick=  
"var a = 100;  
var b = 50;  
var result = (a + b)*2;  
alert('Kết quả của biểu thức là: ' + result)"  
>Xem kết quả</button>
```

Xem kết quả

localhost cho biết:

Kết quả của biểu thức là: 300

OK

2.2.5. Kết hợp PHP và JavaScript

- ❖ Kiểm tra dữ liệu nhập
- ❖ Di chuyển qua trang khác
- ❖ Xác nhận xoá
- ❖ Sử dụng thư viện JavaScript có sẵn

Chương 3.

CƠ SỞ DỮ LIỆU TRONG PHP

1. CSDL MySQL
2. Truy vấn trong MySQL
3. Kết hợp PHP và MySQL – Thư viện PDO
4. Hiển thị dữ liệu trên trang Web
5. Phân trang
6. PHP Email và Security Email

CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- MySQL là một hệ quản trị cơ sở dữ liệu đa luồng mã nguồn mở theo mô hình client/server, và ở mức độ chuyên dụng cho doanh nghiệp. MySQL được phát triển bởi một công ty tư vấn và phát triển ứng dụng của Thụy Điển có tên là TcX
- Hiện nay, MySQL đã được Oracle mua lại.
- MySQL là một hệ quản trị cơ sở dữ liệu có tốc độ truy xuất rất nhanh và uyển chuyển. MySQL được phát triển phổ biến cho hệ điều hành Linux, tuy nhiên, với các phiên bản mới hiện nay, nó đã có thể sử dụng tốt trên của hệ điều hành Windows.
- MySQL có nhiều phiên bản phát hành: các cá nhân có thể được dùng miễn phí tuy nhiên các doanh nghiệp thì cần phải trả phí mới được sử dụng.

CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- CSDL MySQL là tập hợp các đối tượng, bảng,...cho phép người dùng lưu trữ và truy xuất các thông tin đã được tổ chức và lưu trữ bên trong.
- Để thực hiện các thao tác trên CSDL, có thể sử dụng giao diện đồ họa hay dùng dòng lệnh (command line).
- Các tập tin vật lý lưu trữ CSDL
 - Mỗi bảng sẽ được lưu dưới 3 tập tin vật lý
 - .frm : lưu định dạng cấu trúc của mảng
 - .MYD : lưu nội dung của bảng
 - .MYI : lưu chỉ mục của bảng
 - Các tập tin này sẽ được lưu trữ tự động trong thư mục xampp\mysql\data\ten_CSDL

CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

Đối tượng chính của CSDL quan hệ

- Bảng (table)** → dòng , cột : dùng để lưu thông tin của CSDL

MKH	TEN_KH	PHAI	DI_A_CHI	DT	EMAIL
KH001	Trần Văn An	0	123 Nguyễn Du	8123456	tvan@yahoo.com
KH002	Nguyễn Thanh An	0	30 Lê Thánh Tôn	9852147	ntan@yahoo.com
KH003	Lê Thanh Thảo	1	22bis Pasteur	8976431	ltthao@gmail.com

- Quan hệ (relationship):** là thành phần được dùng để tạo mối liên kết giữa các bảng dữ liệu nhằm đảm bảo tính nhất quán, đúng đắn của dữ liệu trong CSDL. Có 3 loại quan hệ (1 – 1) ; (1 – n) ; (n – n)

CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- **Quan hệ 1 – 1** : Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu trong bảng này có quan hệ với duy nhất một dòng dữ liệu trong bảng kia và ngược lại.



CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- **Quan hệ 1 – n** : Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu trong bảng này có quan hệ với nhiều dòng dữ liệu trong bảng kia và một dòng dữ liệu trong bảng kia có quan hệ với duy nhất một dòng dữ liệu trong bảng này (thường gặp).



CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- Quan hệ n – n** : Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu trong bảng này có quan hệ với nhiều dòng dữ liệu trong bảng kia và ngược lại. Trong CSDL không lưu trữ quan hệ nhiều nhiều → chuyển thành các quan hệ 1 – n .



CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- Chiều dài của tên CSDL, bảng, chỉ mục , cột, định danh

Loại	Chiều dài tối đa (bytes)	Chiều dài tối đa (ký tự không dấu)
CSDL (Database)	64	64
Bảng (Table)	64	64
Chỉ mục (Index)	64	64
Cột (Column)	64	64
Định danh (Alias)	255	255

CƠ SỞ DỮ LIỆU MySQL

1. Tổng quan về CSDL MySQL:

- Quy tắc đặt tên
 - Tên không kết thúc bằng khoảng trắng
 - Tên CSDL không có các ký tự : /, \, ., :, *, ", <, >.
 - Tên bảng không có các ký tự: /, \, ., :, *, ", <, >, |.
 - Chiều dài của tên tối đa là 64 ký tự không dấu. Khi sử dụng các ký tự đa bytes thì chiều dài sẽ dựa trên tổng số bytes của tất cả các ký tự được dùng

CƠ SỞ DỮ LIỆU MySQL

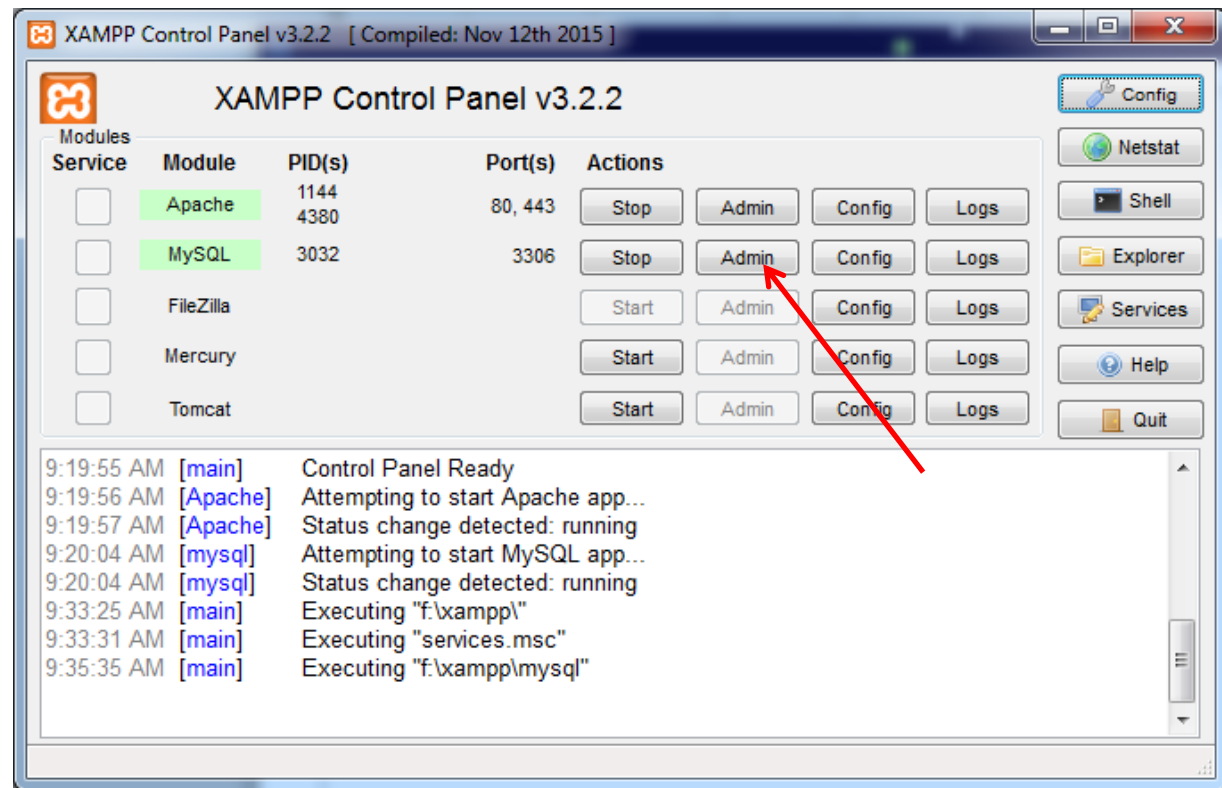
2. Tạo CSDL

- **Các thuộc tính của CSDL**
 - Tên CSDL phải duy nhất trên CSDL MySQL
 - Vị trí lưu trữ: khi tạo mới một CSDL hệ thống sẽ tự động tạo ra một thư mục có tên của CSDL và được lưu tại thư mục `xampp\mysql\data\`
- **Có hai cách để tạo một CSDL**
 - Dùng giao diện đồ họa
 - Dùng dòng lệnh

CƠ SỞ DỮ LIỆU MySQL

2. Tạo CSDL

- Dùng giao diện đồ họa

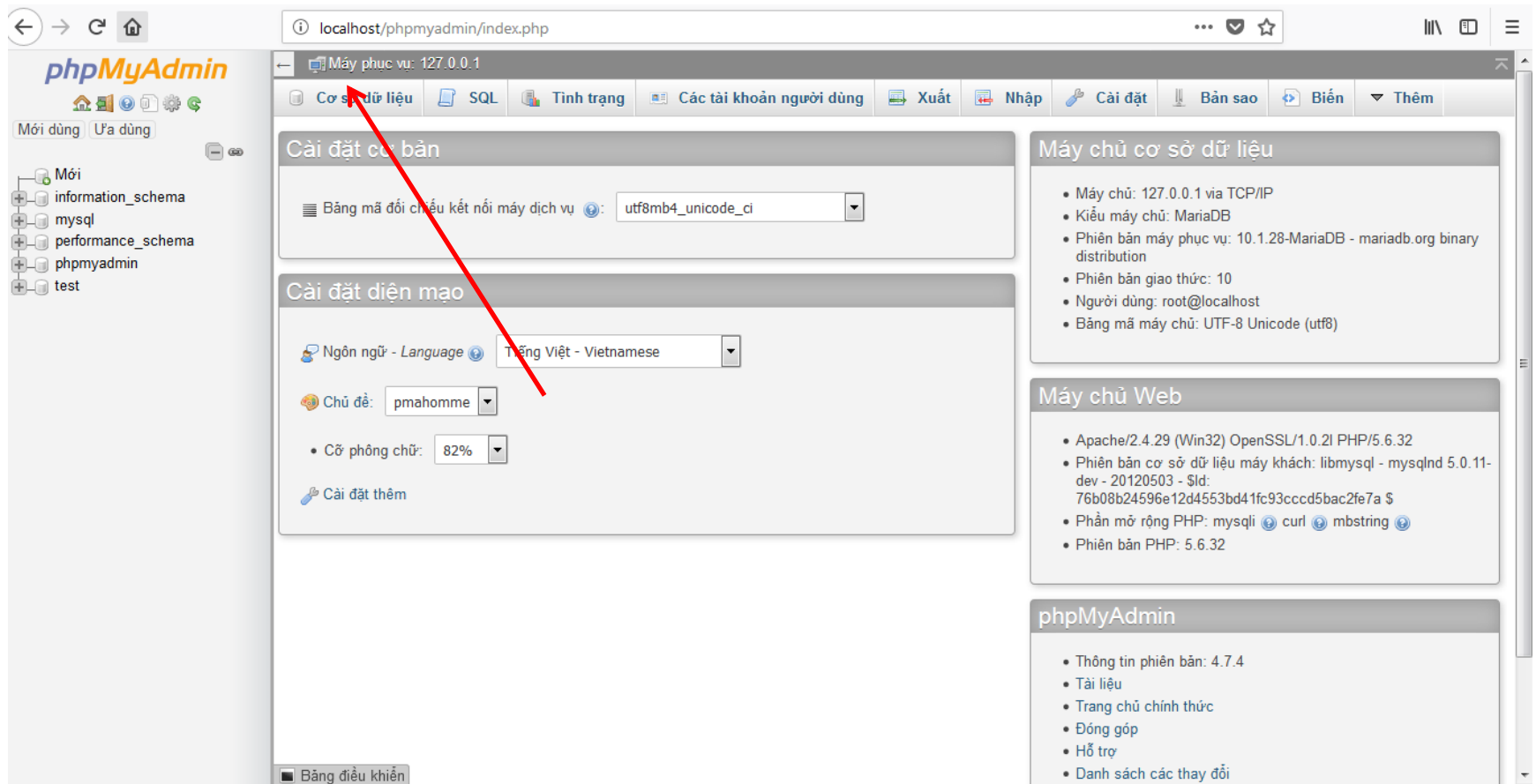


<http://localhost/phpmyadmin>

CƠ SỞ DỮ LIỆU MySQL

2. Tạo CSDL

- Dùng giao diện đồ họa



The screenshot shows the phpMyAdmin web interface. The top navigation bar includes tabs for 'Cơ sở dữ liệu' (Databases), 'SQL', 'Tình trạng' (Status), 'Các tài khoản người dùng' (Users), 'Xuất' (Export), 'Nhập' (Import), 'Cài đặt' (Settings), 'Bản sao' (Backup), 'Biến' (Variables), and 'Thêm' (Add). The 'Cơ sở dữ liệu' tab is selected and highlighted. A red arrow points to this tab. The main content area is divided into several sections: 'Cài đặt cơ bản' (Basic settings) with a dropdown for 'Bảng mã đối chiếu kết nối máy dịch vụ' (Service connection character set) set to 'utf8mb4_unicode_ci'; 'Cài đặt diện mạo' (Appearance settings) with a dropdown for 'Ngôn ngữ - Language' (Language) set to 'Tiếng Việt - Vietnamese' and a dropdown for 'Chủ đề' (Theme) set to 'pmahomme'; and a 'Máy chủ cơ sở dữ liệu' (Database server) section listing system information. The left sidebar shows a tree view of databases: 'Mới' (New), 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The bottom of the interface shows a 'Bảng điều khiển' (Control panel) section.

CƠ SỞ DỮ LIỆU MySQL

2. Tạo CSDL

- Dùng giao diện đồ họa

The screenshot displays the MySQL Workbench interface. At the top, there are tabs for 'Cơ sở dữ liệu' (Databases), 'SQL', 'Tình trạng' (Status), and 'Các tài khoản người dùng' (Users). The 'Cơ sở dữ liệu' tab is active, showing a tree view on the left with databases: 'Mới' (New), 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'ql_nha_hang' (highlighted with a red box), and 'test'. The main area is titled 'Cơ sở dữ liệu' and contains a 'Tạo cơ sở dữ liệu' (Create database) dialog. The dialog has a text field for the database name containing 'ql_nha_hang' and a dropdown menu for the character set set to 'utf8_unicode_ci'. A 'Tạo' (Create) button is highlighted with a red box. Below the dialog is a table listing existing databases and their character sets. A red arrow points from the 'Tạo' button to the 'ql_nha_hang' database in the sidebar.

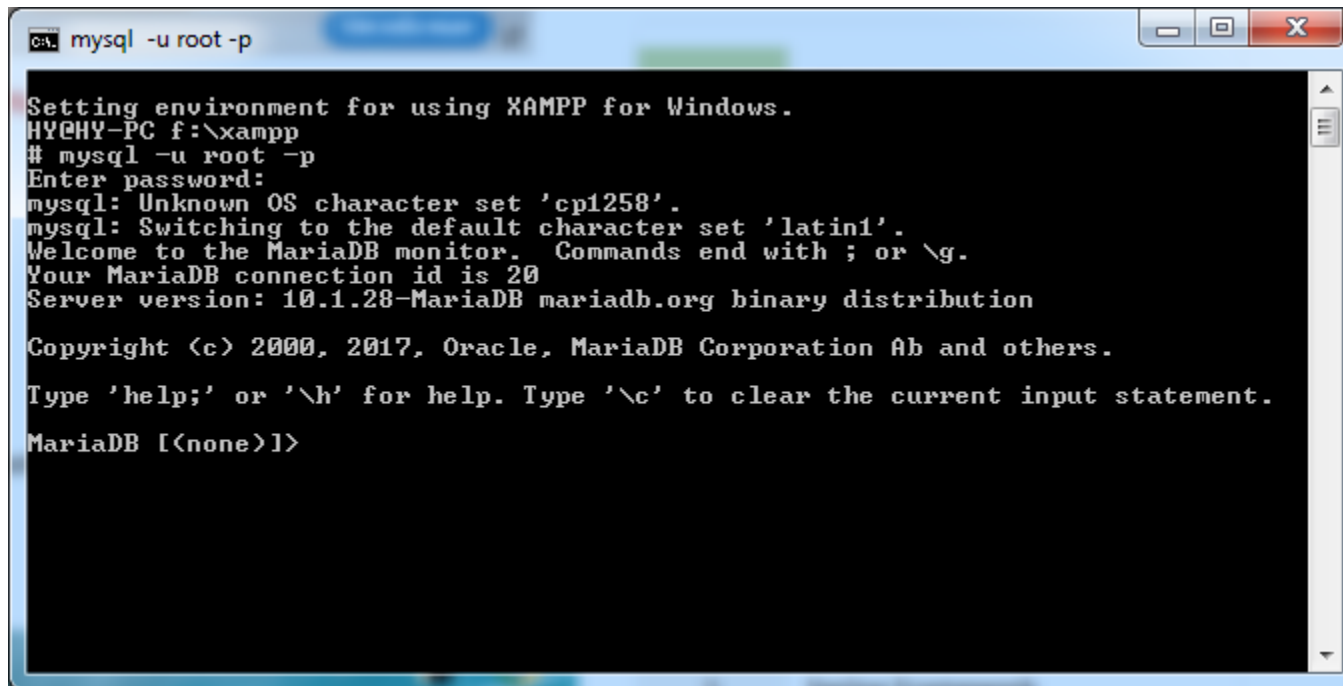
Cơ sở dữ liệu	Bảng mã đối chiếu	Hành động
☐ information_schema	utf8_general_ci	Kiểm tra đặc quyền
☐ mysql	latin1_swedish_ci	Kiểm tra đặc quyền
☐ performance_schema	utf8_general_ci	Kiểm tra đặc quyền
☒ phpmyadmin	utf8_bin	Kiểm tra đặc quyền
☒ test	latin1_swedish_ci	Kiểm tra đặc quyền
Tổng cộng: 5		

CƠ SỞ DỮ LIỆU MySQL

2. Tạo CSDL

- Dùng dòng lệnh

Tài khoản đăng nhập mySQL: root và không có mật khẩu



```
mysql -u root -p

Setting environment for using XAMPP for Windows.
HYCHY-PC f:\xampp
# mysql -u root -p
Enter password:
mysql: Unknown OS character set 'cp1258'.
mysql: Switching to the default character set 'latin1'.
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 20
Server version: 10.1.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

CƠ SỞ DỮ LIỆU MySQL

2. Tạo CSDL

- Dùng dòng lệnh

Cú pháp:

CREATE DATABASE Tên_CSDL

[[DEFAULT] CHARACTER SET<character set name>]

[[DEFAULT] COLLATE <collation name>] ;

Ví dụ:

```
MariaDB [(none)]> CREATE DATABASE TEMP
-> CHARACTER SET utf8
-> COLLATE utf8_unicode_ci;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| dich_vu_an_uong |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| ql_nha_hang |
| temp |
| test |
+-----+
```

```
CREATE DATABASE ql_ban_thuoc|
```

CƠ SỞ DỮ LIỆU MySQL

3. Xóa CSDL

Cú pháp:

DROP DATABASE Tên_CSDL;

Ví dụ:

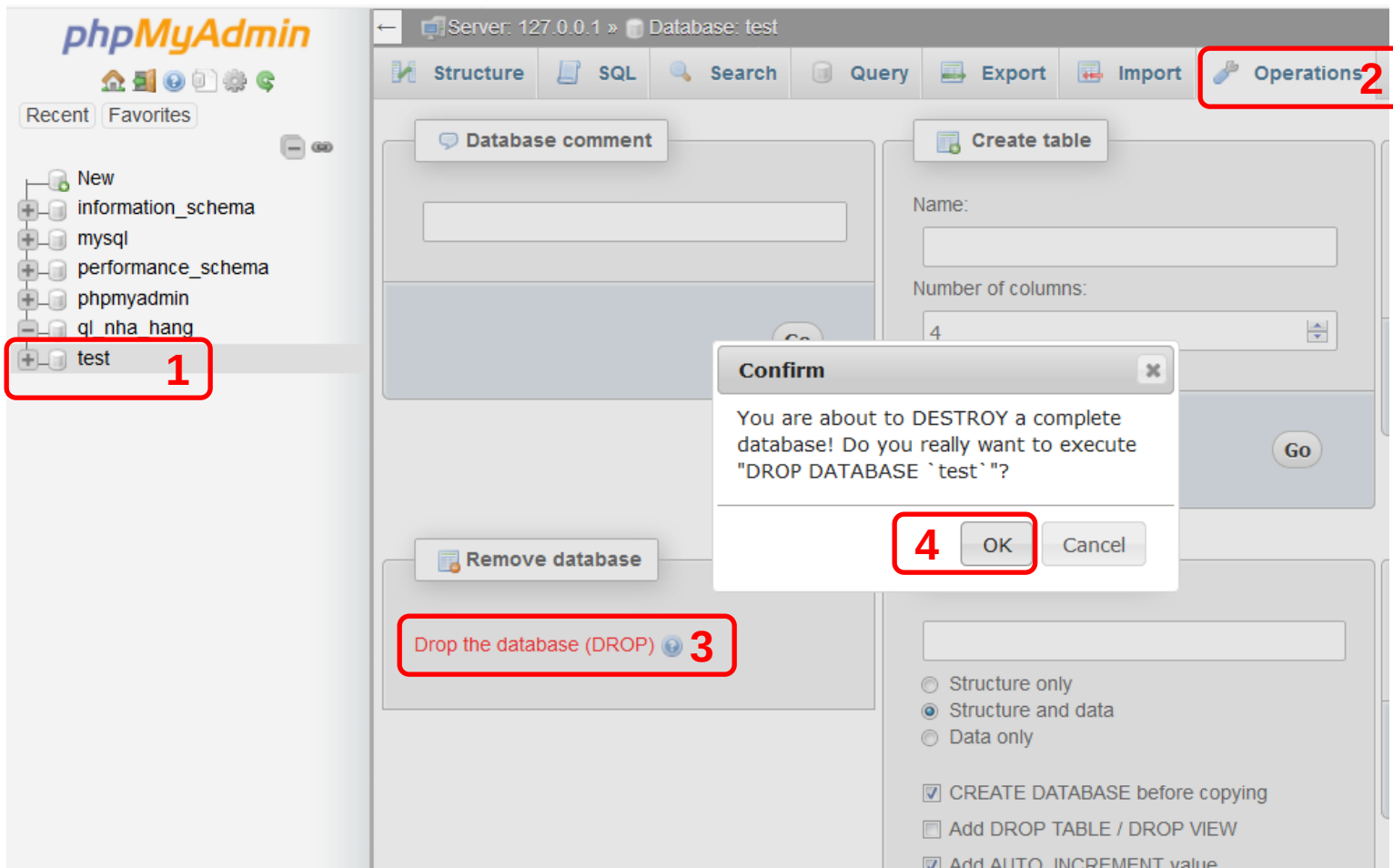
```
MariaDB [(none)]> DROP DATABASE temp;
Query OK, 0 rows affected (0.02 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| dich_vu_an_uong |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| ql_nha_hang |
| test |
+-----+
7 rows in set (0.00 sec)

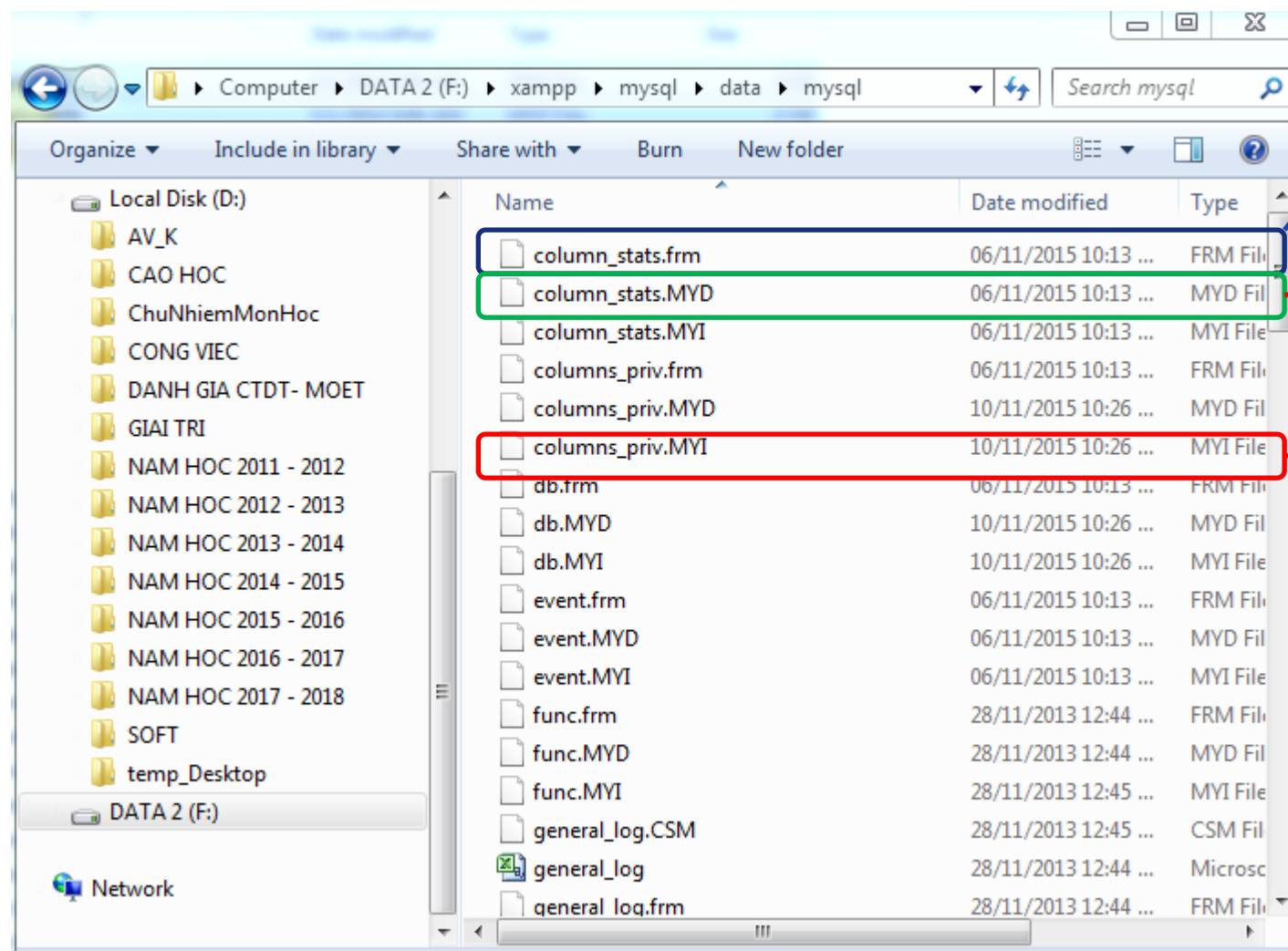
MariaDB [(none)]>
```

CƠ SỞ DỮ LIỆU MySQL

3. Xóa CSDL Dùng giao diện đồ họa



CƠ SỞ DỮ LIỆU MySQL



Lưu cấu trúc

Lưu nội dung

Lưu chỉ mục

- CSDL khi tạo xong thì lưu ở đâu trong ổ đĩa vật lý ???
- F:\xampp\mysql\data\ql_nha_hang

BẢNG

1. Quy tắc đặt tên

- **Tên bảng (table name)**

Do người dùng tạo ra

Duy nhất trong CSDL

- **Tên cột**

Do người dùng tạo ra

Duy nhất trong bảng

VD: Đặt tên cho cột họ tên trong bảng sinh viên là :

Ho_ten

BẢNG

2. Kiểu dữ liệu trong bảng

- Số nguyên

Kiểu dữ liệu	Kích thước	Miền Giá trị
Tinyint	1 byte	- 127 -> 128 hay 0 .. 255
Smallint	2 byte	-32767 -> 32767 hay 0 .. 65535
Mediumint	3 byte	-8388608 -> 8388608
Int	4 byte	-2^{31} -> $2^{31}-1$ hay $0..2^{32}-1$
Bigint	8 byte	-2^{63} -> $2^{63}-1$ hay $0..2^{64}-1$

- Boolean (True / False)

Kiểu dữ liệu	Kích thước	Miền Giá trị
Bool / Boolean	1 byte	Có hai giá trị là True và False

2. Kiểu dữ liệu trong bảng

- Số thập phân

Decimal(M[,N])

M : là tổng ký số

N : là số ký số thập phân

Kiểu dữ liệu	Kích thước	Miền Giá trị
Decimal / Numeric	4 bytes	Có hai giá trị là True và False

- Số thực

Kiểu dữ liệu	Kích thước	Miền Giá trị
Float	4 bytes	
Double	8 bytes	

BẢNG

2. Kiểu dữ liệu trong bảng

- Date / time

Kiểu dữ liệu	Miền giá trị	Diễn giải
Date	1000-01-01-> 9999-12-31	Ngày với định dạng yyyy-mm-dd
Datetime	1000-01-01 00:00:00 -> 9999-12-31 23:59:59	Ngày giờ với định dạng yyyy-mm-dd hh:mm:ss
Time	00:00:00 -> 23:59:59	Giờ với định dạng hh:mm:ss
Year[(2 4)]	4 ký số: 1901 -> 2155 2 ký số: 1970 -> 2069	Năm với định dạng 2 ký số hoặc 4 ký số
Timestamp [(Kích cỡ định dạng)]	1970-01-01 00:00:01	Timestamp trình bày dưới dạng yyyy-mm-dd hh:mm:ss

BẢNG

2. Kiểu dữ liệu trong bảng

- Chuỗi Char và Varchar

Giá trị	Char(4)	Số bytes	Varchar(4)	Số bytes
''	' '	4 bytes	''	1byte
'ab'	'ab '	4 bytes	'ab'	3bytes
'abcd'	'abcd'	4 bytes	'abcd'	5bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5bytes

3. Thuộc tính của cột trong bảng

- Độ dài dữ liệu (length / value)
- Kiểu hiển thị (collation)
- Thuộc tính (attribute)
- Cho phép để trống dữ liệu (NULL)
- Giá trị mặc định (default)
- Thuộc tính mở rộng (extra)
- Ghi chú (comment)
- Tăng giá trị tự động (auto_increment)

BẢNG

4. Thao tác với bảng – Tạo bảng

- Cú pháp

CREATE TABLE Tên_Bảng

(

Tên_Cột_1 Kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL],

Tên_Cột_2 Kiểu_dữ_liệu[(kích_cỡ)] **DEFAULT g_trị**,

Tên_Cột_3 Kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL],

.....

UNIQUE (các_cột_độc_nhất),

PRIMARY KEY (Các cột tham gia làm khóa chính)

)

BẢNG

- **Ví dụ:** tạo bảng `hang_sua` với hai cột dữ liệu không được phép bỏ trống là `Ma_hang_sua` và `Ten_hang_sua`, `Ten_hang_sua` là duy nhất, `Ma_hang_sua` là khóa chính

```
CREATE TABLE hang_sua
(
    Ma_hang_sua varchar(20) NOT NULL PRIMARY KEY,
    Ten_hang_sua varchar(100) NOT NULL UNIQUE,
    Dia_chi varchar(200), Dien_thoai varchar(20),
    Email varchar(100)
)
```

BẢNG

Tạo bảng Bảng giao diện đồ họa



⚠ No tables found in database.

Create table

Name:

loai_mon_an

Number of columns: 3

1 Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I
ma_loai	INT	11	None			☐	PRIMARY	☒
Pick from Central Columns								
ten_loai	VARCHAR	200	None	utf8_unicode_ci		☐	---	☐
Pick from Central Columns								
mo_ta	VARCHAR	500	None	utf8_unicode_ci		☐	---	☐
Pick from Central Columns								
hinh	VARCHAR	300	None			☐	---	☐
Pick from Central Columns								



Preview SQL

Save

BẢNG

Thay đổi cấu trúc bảng- đổi tên cột và thay đổi kiểu dữ liệu cho cột

- Cú pháp:

ALTER TABLE Tên_bảng

CHANGE Tên_Cột_cũ Tên_cột_mới Kiểu_dữ_liệu_mới[KT]

- Ví dụ: đổi tên cột **Hình** trong table sua thành cột **Ghi_chu**

```
ALTER TABLE sua  
CHANGE Hình Ghi_chu varchar(50)
```

BẢNG

Thay đổi cấu trúc bảng Bảng giao diện đồ họa

Server: 127.0.0.1 » Database: copy_ql_ban_sua » Table: sua "Sửa"

Browser Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Ma_sua	varchar(5)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
2	Ten_sua	varchar(100)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
3	Ma_hang	varchar(20)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
4	Ma_loai	varchar(3)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
5	TL	int(11)			No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
6	Don_gia	int(11)			No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
7	TP_DD	text	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
8	LI	text	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values More
9	Ghi_chu	varchar(50)	utf8_unicode_ci		Yes	NULL			Change Drop Primary Unique Index Spatial Fulltext Distinct values More

1

2

3

4 Save

Preview SQL

BẢNG

Thay đổi cấu trúc bảng- thêm cột mới vào bảng

- Cú pháp:

ALTER TABLE Tên_bảng

ADD Tên_cột *Kiểu_dữ_liệu* [Kích_Thước]

- Ví dụ: thêm cột **mo_ta** vào table sua

```
ALTER TABLE sua ADD mo_ta varchar(255)|
```

	Browse	Structure	SQL	Search	Insert	More
☐	7	TP_DD	text	utf8_unicode_ci	No	None
☐	8	LI	text	utf8_unicode_ci	No	None
☐	9	Ghi_chu	varchar(50)	utf8_unicode_ci	Yes	NULL
☐	10	mo_ta	varchar(255)	utf8_unicode_ci	Yes	NULL

BẢNG

Thay đổi cấu trúc bảng Bảng giao diện đồ họa

☐	8	LI	text	utf8_unicode_ci	No	None	Change	Drop	Primary
☒	9	Ghi_chu	varchar(50)	utf8_unicode_ci	Yes	NULL	Change	Drop	Primary

Check all With selected: Browse Change Drop Primary Unique Index
 Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Improve table structure

Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Nullable
Pick from Central Columns	INT		None			

BẢNG

Thay đổi cấu trúc bảng- xóa cột khỏi bảng

- Cú pháp:

ALTER TABLE Tên_bảng

DROP COLUMN Tên_cột_cần_xóa

- Ví dụ: xóa cột **mo_ta** từ table sua

```
ALTER TABLE sua DROP mo_ta|
```

BẢNG

Thay đổi cấu trúc bảng- Bảng giao diện đồ họa

☐	8	LI	text	utf8_unicode_ci	No	None	Change Drop Primary More
☐	9	Ghi_chu	varchar(50)	utf8_unicode_ci	Yes	NULL	Change Drop Primary More
☒	10	mo_ta	varchar(255)	utf8_unicode_ci	Yes	NULL	Change Drop Primary More

1

Confirm

Do you really want to execute "ALTER
TABLE `sua` DROP `mo\_ta`;"?

OK

Cancel

3

☐	7	TP_DD	text	utf8_unicode_ci	No	None	Change Drop Primary Unique
☐	8	LI	text	utf8_unicode_ci	No	None	Change Drop Primary Unique
☐	9	Ghi_chu	varchar(50)	utf8_unicode_ci	Yes	NULL	Change Drop Primary Unique

4. Thao tác với bảng – Xóa bảng

- Cú pháp

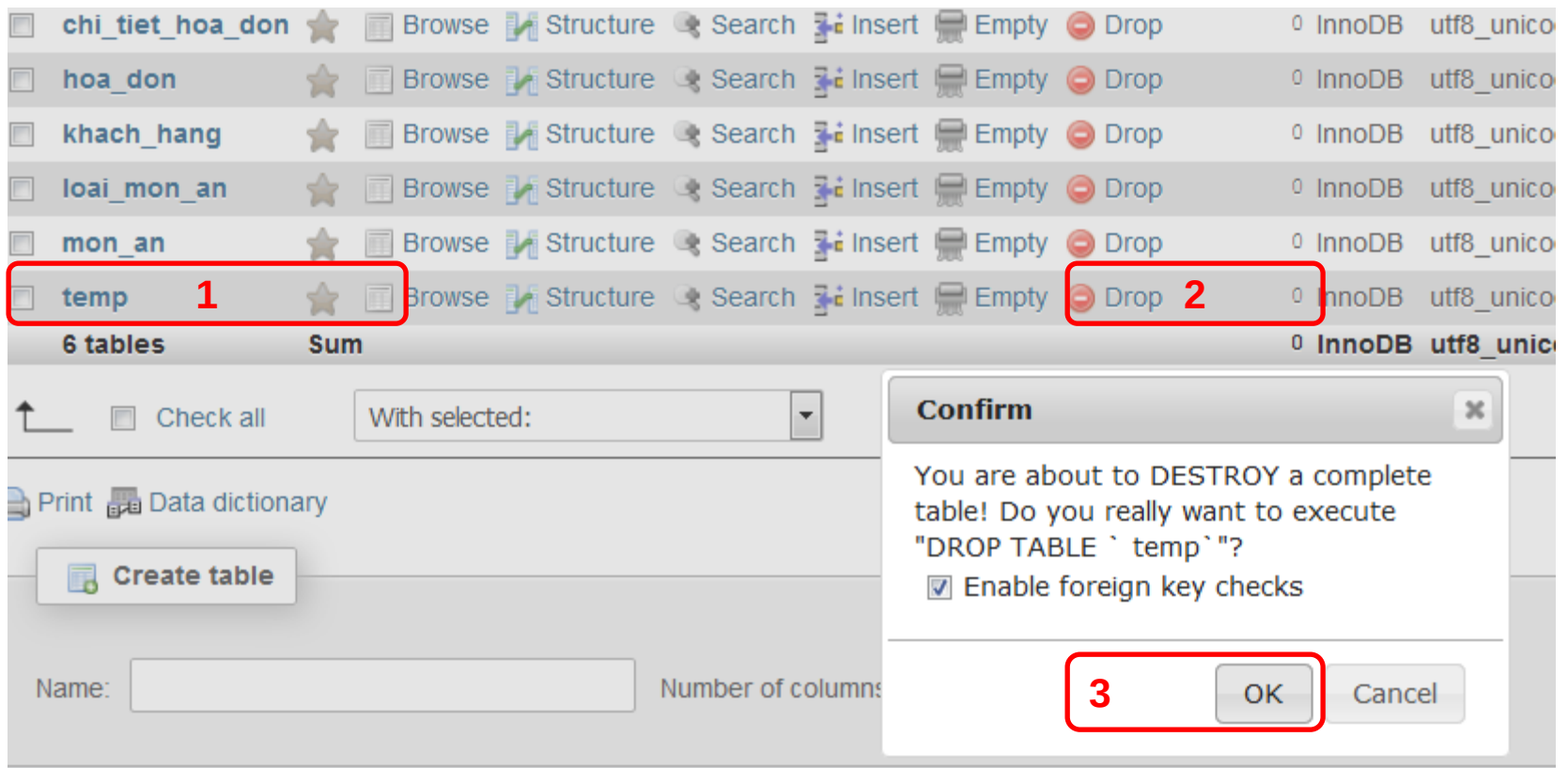
DROP TABLE Tên_Bảng

- Ví dụ: xóa table hang_sua

```
Drop TABLE hang_sua|
```

BẢNG

Xóa bảng Bảng giao diện đồ họa



BẢNG

5. Tạo mối quan hệ giữa các bảng

- Tạo mối quan hệ giữa các bảng với nhau (giữa **bảng cha** (khóa chính – primary key) và **bảng con** (khóa ngoại – foreign key))
- Giúp cho việc cập nhật , xóa dữ liệu được chính xác và nhanh chóng
- Tạo khóa ngoại

FOREIGN KEY (ten_cot_khoa_ngoai)

REFERENCES ten_bang_cha

(Ten_cot_khoa_chinh) **ON UPDATE CASCADE**,...

```
ALTER TABLE sua
```

```
ADD FOREIGN KEY (ma_loai_sua) REFERENCES  
loai_sua (ma_loai_sua) ON UPDATE CASCADE,
```

```
ADD FOREIGN KEY (ma_hang_sua) REFERENCES  
hang_sua (ma_hang_sua) ON UPDATE CASCADE;
```

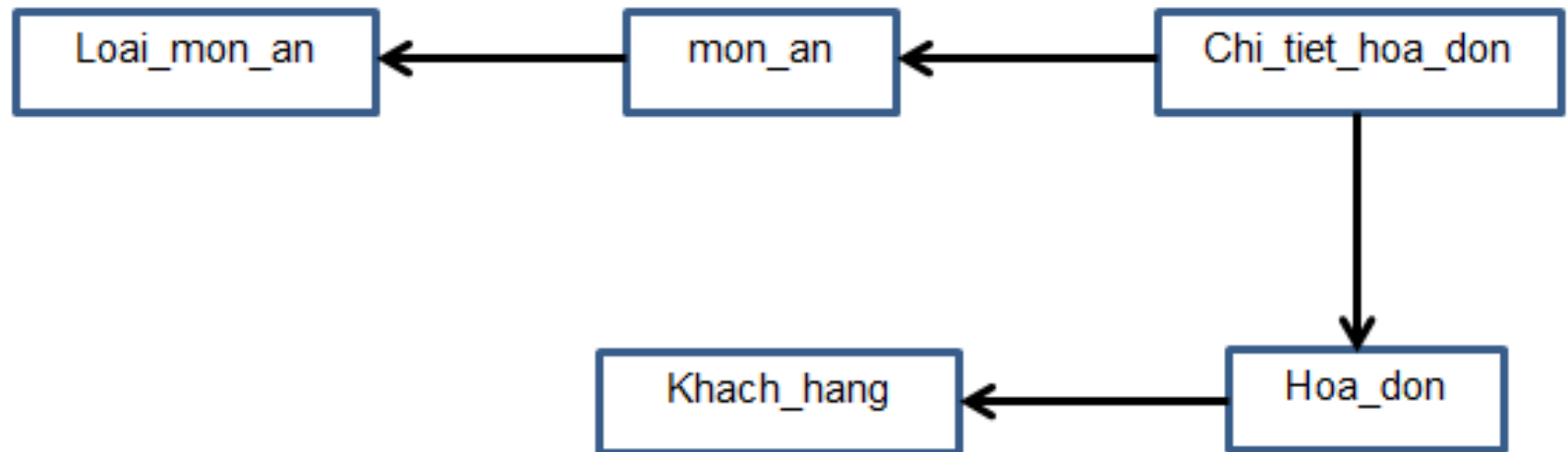
5. Tạo mối quan hệ giữa các bảng

- Bảng giao diện đồ họa (DEMO)
- Quan hệ (1 – 1) (1 – n)
- Chú ý : TH tạo qan hệ 1 – n giữa hai bảng thì :
- Phải mở bảng đầu nhiều trước , thiết lập quan hệ qua khóa ngoại.
- Khi nhập liệu phải nhập bảng đầu 1 trước , đầu nhiều sau

BẢNG

5. Tạo mối quan hệ giữa các bảng

Tạo mối quan hệ



BẢNG

5. Tạo mối quan hệ giữa các bảng

Server: 127.0.0.1 » Database: ql_nha_hang » Table: mon_an "Món ăn"

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [More](#)

[Table structure](#) [Relation view](#)

Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
FK_Monan_LoaiMonAn					
☐ Drop	ON DELETE NO ACTION	ma_loai	ql_nha_hang	loai_mon_an	ma_loai
	ON UPDATE CASCADE	+ Add column			
Constraint name					
ON DELETE RESTRICT			ql_nha_hang		
ON UPDATE RESTRICT		+ Add column			
+ Add constraint					

BẢNG

5. Tạo mối quan hệ giữa các bảng

Server: 127.0.0.1 » Database: ql_nha_hang » Table: hoa_don

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [More](#)

[Table structure](#) [Relation view](#)

Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
FK_HoaDon_KhachHang					
Drop	ON DELETE NO ACTION	ma_khach_hang	ql_nha_hang	khach_hang	ma_khach_hang
	ON UPDATE CASCADE	+ Add column			
Constraint name					
ON DELETE RESTRICT			ql_nha_hang		
ON UPDATE RESTRICT		+ Add column			
+ Add constraint					

BẢNG

5. Tạo mối quan hệ giữa các bảng

Server: 127.0.0.1 » Database: ql_nha_hang » Table: chi_tiet_hoa_don "Chi Tiết hóa đơn"

Browse Structure SQL Search Insert Export Import Privileges Operations More

Table structure Relation view

Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
FK_CTHD_Hoadon					
Drop	ON DELETE RESTRICT	ma_hoa_don	ql_nha_hang	hoa_don	ma_hoa_don
	ON UPDATE RESTRICT	+ Add column			
FK_CTHD_MonAn					
Drop	ON DELETE NO ACTION	ma_mon	ql_nha_hang	mon_an	ma_mon
	ON UPDATE CASCADE	+ Add column			

EXPORT DỮ LIỆU

The screenshot shows the phpMyAdmin interface with the 'Export' tab selected. The interface is for a database named 'ql_nha_hang'. The left sidebar shows a list of tables: 'ql_nha_hang', 'chi_tiet_hoa_don', 'hoa_don', 'mon_an', 'khach_hang', 'loai_mon_an', 'temp', 'sinhvien', and 'SinhVien'. The 'mon_an' table is selected. The main area displays the 'Exporting tables from "ql_nha_hang" database' screen. The 'Export' button in the top navigation bar is highlighted with a red box and the number 2. The 'Export method' section has two radio buttons: 'Quick - display only the minimal options' (selected) and 'Custom - display all possible options'. The 'Format' dropdown menu is set to 'SQL' and is highlighted with a red box and the number 4. The 'Go' button is highlighted with a red box and the number 5. A red arrow points from the text '1. Chọn CSDL hoặc bảng cần export' to the 'mon_an' table in the sidebar. A box on the right shows the filename 'ql_nha_hang.sql'.

phpMyAdmin

Server: 127.0.0.1 » Database: ql_nha_hang

Structure SQL Search Query **Export** Import Operations Privileges Routines More

Exporting tables from "ql_nha_hang" database

Export templates:

New template: Existing templates:

Template name Create Template: -- Select a template -- Update Delete

Export method:

3 ☒ Quick - display only the minimal options
☐ Custom - display all possible options

Format:

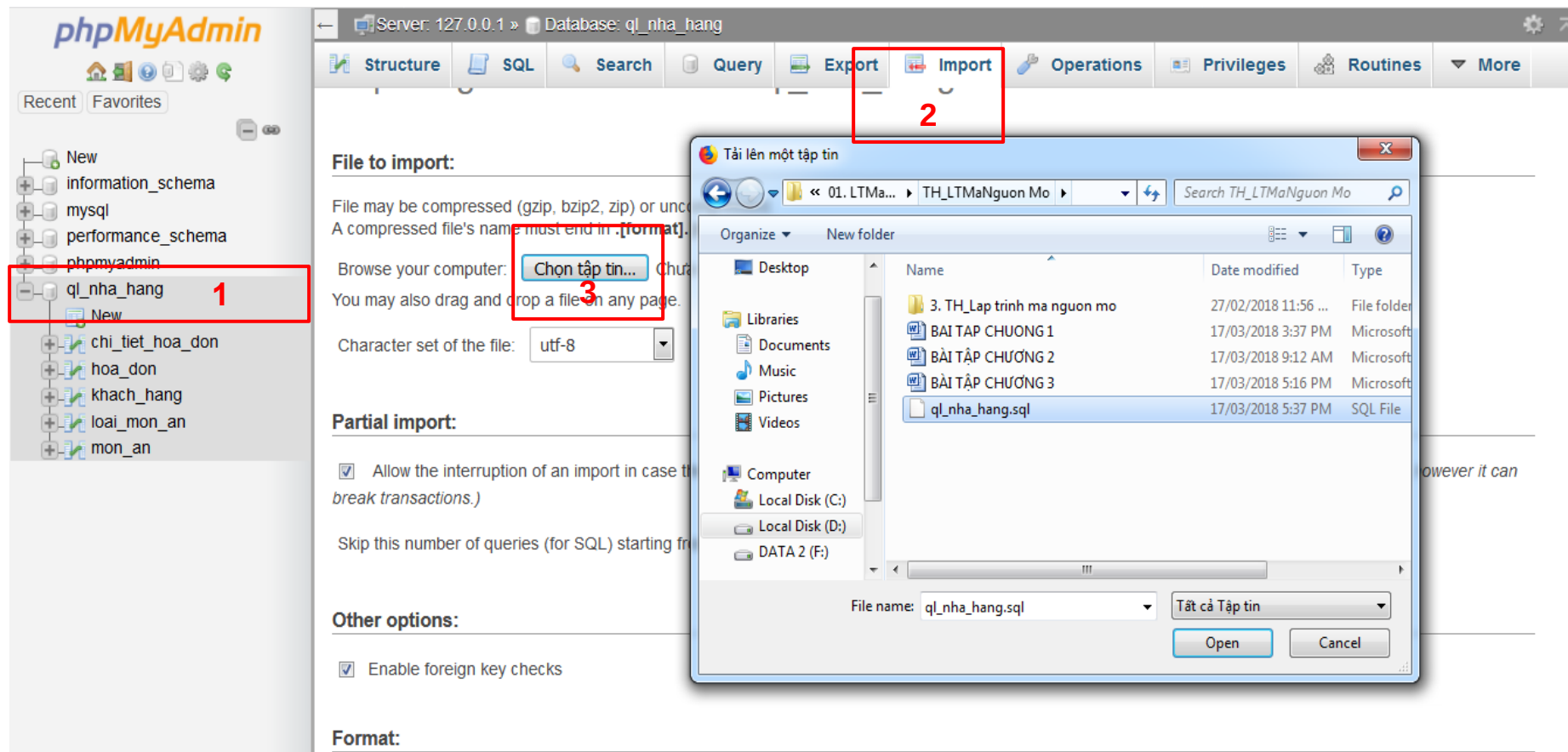
SQL 4

Go 5

1. Chọn CSDL hoặc bảng cần export

ql_nha_hang.sql

IMPORT DỮ LIỆU



The screenshot shows the phpMyAdmin interface for a server at 127.0.0.1 with the database 'ql_nha_hang'. The 'Import' tab is selected in the top navigation bar. In the left sidebar, the 'ql_nha_hang' database is selected. The main area shows the 'File to import:' section with a 'Chọn tập tin...' button. A file explorer window is open, showing the file 'ql_nha_hang.sql' selected. The 'Format:' section is visible at the bottom.

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed. A compressed file's name must end in **.[format]**.

Browse your computer: **Chọn tập tin...**

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☒ Allow the interruption of an import in case the server runs out of memory (this may break transactions.)

Skip this number of queries (for SQL) starting from:

Other options:

☒ Enable foreign key checks

Format:

Format-specific options:

SQL compatibility mode: NONE

☒ Do not use AUTO_INCREMENT for zero values

Go

4

TRUY VẤN TRONG MY SQL

1. Truy vấn đơn giản - SELECT : cho phép chọn ra dữ liệu của các cột có trong một bảng

- Cú pháp:

SELECT Danh_sach_cac_cot

FROM Ten_bang

- Với tên các cột trong Danh_sach_cac_cot phải chính xác
- Dùng * khi muốn lấy dữ liệu từ tất cả các cột trong bảng

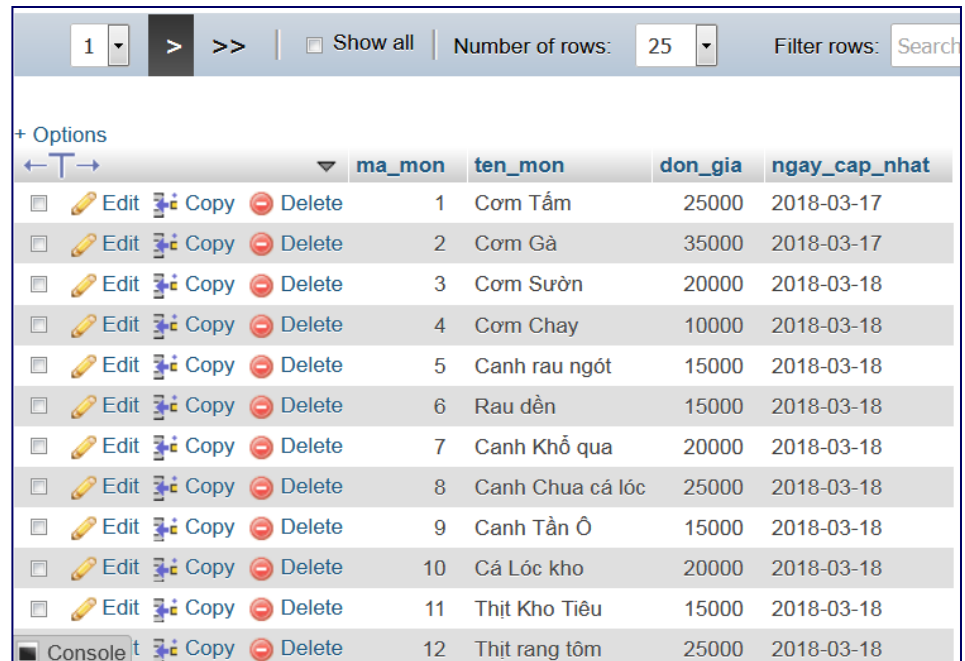
TRUY VẤN TRONG MY SQL

- Ví dụ: hiển thị thông tin của các món ăn có trong bảng món ăn gồm các cột như sau: mã món, tên món, đơn giá, ngày cập nhật.

```
SELECT  
ma_mon, ten_mon, don_gia, ngay_cap_nhat  
FROM `mon_an`
```

- hiển thị tất cả thông tin có trong bảng món ăn

```
SELECT * FROM `mon_an`
```



The screenshot shows a MySQL database interface. At the top, there are navigation buttons (1, >, >>), a 'Show all' checkbox, 'Number of rows: 25', and a 'Filter rows:' search bar. Below this is a '+ Options' section with a dropdown menu. The main table has four columns: 'ma_mon', 'ten_mon', 'don_gia', and 'ngay_cap_nhat'. Each row represents a food item with its ID, name, price, and update date. Each row also has action buttons: a checkbox, an 'Edit' icon, a 'Copy' icon, and a 'Delete' icon. The table contains 12 rows of data.

	ma_mon	ten_mon	don_gia	ngay_cap_nhat
☐ Edit Copy Delete	1	Cơm Tấm	25000	2018-03-17
☐ Edit Copy Delete	2	Cơm Gà	35000	2018-03-17
☐ Edit Copy Delete	3	Cơm Sườn	20000	2018-03-18
☐ Edit Copy Delete	4	Cơm Chay	10000	2018-03-18
☐ Edit Copy Delete	5	Canh rau ngót	15000	2018-03-18
☐ Edit Copy Delete	6	Rau dền	15000	2018-03-18
☐ Edit Copy Delete	7	Canh Khổ qua	20000	2018-03-18
☐ Edit Copy Delete	8	Canh Chua cá lóc	25000	2018-03-18
☐ Edit Copy Delete	9	Canh Tầm Ô	15000	2018-03-18
☐ Edit Copy Delete	10	Cá Lóc kho	20000	2018-03-18
☐ Edit Copy Delete	11	Thịt Kho Tiêu	15000	2018-03-18
☐ Edit Copy Delete	12	Thịt rang tôm	25000	2018-03-18

TRUY VẤN TRONG MY SQL

2. Truy vấn sắp xếp – ORDER BY : cho phép lấy dữ liệu của các cột có trong bảng đồng thời sắp xếp lại dữ liệu theo thứ tự tăng dần hoặc giảm dần

- Cú pháp:

```
SELECT Danh_sach_cac_cot
```

```
FROM Ten_bang
```

```
ORDER BY Ten_cot_sap_xep[DESC,...]
```

- Mặc định, cột sẽ được sắp tăng dần, nếu muốn sắp theo thứ tự giảm dần thì cần thêm từ khóa DESC

TRUY VẤN TRONG MY SQL

- Ví dụ:

```
SELECT *  
FROM `khach_hang`  
ORDER BY email
```

ma_khach_hang	ten_khach_hang	email	dia_chi	dien_thoai	ghi_chu
8	Nguyễn Linh Giang	giangnl@gmail.com	1254 Gò Dầu	093568745	
9	Trương Thị Thu Hiền	hienttt@gmail.com	54 Lương Thế Vinh	01234587	
3	Nguyễn Văn Hòa	hoanv@gmail.com	123 Cầu Xéo	08987654	
5	Đinh Duy Minh	minhdd@gmail.com	698 Đỗ Thừa Luông	01235478	
4	Ngô Thị Nguyệt	nguyetnt@gmail.com	456 Tân kỳ tân quý	086532148	
6	Nguyễn Chinh Tây	taync@gmail.com	546 Dương Văn Dương	12345678	
7	Nguyễn Việt Thu	thunv@gmail.com	457 Đỗ Thừa Luông	090123456	
2	Trương Thị Khánh Uyên	uyenttk@gmail.com	123 Lê Trọng Tấn	090756489	
10	Trương Thị Thanh Xuân	xuanttt@gmail.com	99 Trường Chinh	0124897333	
1	Nguyễn Hải Yến	yennh@cntp.edu.vn	140 Lê Trọng Tấn	08123456	

```
SELECT *  
FROM `khach_hang`  
ORDER BY email DESC
```

ma_khach_hang	ten_khach_hang	email	dia_chi	dien_thoai	ghi_chu
1	Nguyễn Hải Yến	yennh@cntp.edu.vn	140 Lê Trọng Tấn	08123456	
10	Trương Thị Thanh Xuân	xuanttt@gmail.com	99 Trường Chinh	0124897333	
2	Trương Thị Khánh Uyên	uyenttk@gmail.com	123 Lê Trọng Tấn	090756489	
7	Nguyễn Việt Thu	thunv@gmail.com	457 Đỗ Thừa Luông	090123456	
6	Nguyễn Chinh Tây	taync@gmail.com	546 Dương Văn Dương	12345678	
4	Ngô Thị Nguyệt	nguyetnt@gmail.com	456 Tân kỳ tân quý	086532148	
5	Đinh Duy Minh	minhdd@gmail.com	698 Đỗ Thừa Luông	01235478	
3	Nguyễn Văn Hòa	hoanv@gmail.com	123 Cầu Xéo	08987654	
9	Trương Thị Thu Hiền	hienttt@gmail.com	54 Lương Thế Vinh	01234587	
8	Nguyễn Linh Giang	giangnl@gmail.com	1254 Gò Dầu	093568745	

TRUY VẤN TRONG MY SQL

3. Truy vấn có sử dụng – AS: sử dụng AS để đặt lại tên (alias) cho các cột

- Cú pháp:

```
SELECT ten_cot AS Định_danh_cot,...  
FROM Ten_bang
```

- Ví dụ: Đặt định danh lại cho cột don_gia là DG

```
SELECT ma_mon, ten_mon, don_gia AS DG  
FROM `mon_an` |
```

TRUY VẤN TRONG MY SQL

3. Truy vấn có sử dụng – Limit: sử dụng Limit n,m để lấy ra m mẫu tin trong bảng tính từ vị trí n theo một tiêu chuẩn sắp xếp nào đó.

- Cú pháp:

```
SELECT Danh_sach_cac_cot  
FROM Ten_bang  
[ORDER BY Ten_cot_sap_xep[DESC,...]]  
LIMIT n,m
```

- Ví dụ: Đặt định danh lại cho cột don_gia là DG

TRUY VẤN TRONG MY SQL

- Ví dụ: Cho biết 5 món ăn trong bảng mon_an có đơn giá lớn nhất gồm các thông tin : ma_mon, ten_mon, don_gia

```
SELECT ma_mon,ten_mon,don_gia  
FROM `mon_an`  
ORDER BY don_gia DESC  
LIMIT 0,5|
```

+ Options				ma_mon	ten_mon	don_gia
☐	 Edit	 Copy	 Delete	26	Gỏi Tiến vua	110000
☐	 Edit	 Copy	 Delete	27	Gỏi Sứa	100000
☐	 Edit	 Copy	 Delete	25	Gỏi Bao tử	90000
☐	 Edit	 Copy	 Delete	28	Gỏi ngó sen	70000
☐	 Edit	 Copy	 Delete	17	Phở Gà	40000

TRUY VẤN TRONG MY SQL

3. Truy vấn có sử dụng – DISTINCT:

- Trong một bảng, một cột có thể chứa nhiều giá trị trùng lặp → sử dụng từ khóa DISTINCT để lấy ra danh sách những giá trị duy nhất không trùng lặp.
- Cú pháp:

```
SELECT DISTINCT Danh_sach_cac_cot  
FROM Ten_bang  
[ORDER BY Ten_cot_sap_xep[DESC,...]]
```

- Ví dụ: Cho biết các mã loại món ăn

```
SELECT DISTINCT ma_loai  
FROM `mon_an`  
|
```

TRUY VẤN TRONG MY SQL

4. Truy vấn lọc theo mẫu tin– WHERE:

```
SELECT DISTINCT ma_mon,ma_loai,ten_mon,don_gia,dvt  
FROM `mon_an`  
WHERE ma_loai="1"|
```

ma_mon	ma_loai	ten_mon	don_gia	dvt
5	1	Canh rau ngọt	15000	chén
6	1	Rau dền	15000	Tô
7	1	Canh Khổ qua	20000	Tô
8	1	Canh Chua cá lóc	25000	Tô
9	1	Canh Tần Ô	15000	tô

```
SELECT ma_mon,ma_loai,ten_mon,don_gia,dvt  
FROM mon_an  
WHERE (ma_loai="1") AND (don_gia IN(20000,25000))|
```

ma_mon	ma_loai	ten_mon	don_gia	dvt
7	1	Canh Khổ qua	20000	Tô
8	1	Canh Chua cá lóc	25000	Tô

TRUY VẤN TRONG MY SQL

4. Truy vấn trên nhiều bảng – dùng where

- Ví dụ: Hiển thị thông tin các món ăn trong bảng mon_an gồm : ma_mon, ma_loai, ten_mon,ten_loai, don_gia và đơn giá ≥ 50000

```
1 |SELECT ma_mon,mon_an.ma_loai,ten_mon,ten_loai,don_gia
   |FROM mon_an, loai_mon_an
2 |WHERE mon_an.ma_loai = loai_mon_an.ma_loai AND
   |don_gia>=50000
```

+ Options				
ma_mon	ma_loai	ten_mon	ten_loai	don_gia
25	6	Gỏi Bao tử	Món gỏi	90000
26	6	Gỏi Tiến vua	Món gỏi	110000
27	6	Gỏi Sứa	Món gỏi	100000
28	6	Gỏi ngó sen	Món gỏi	70000

TRUY VẤN TRONG MY SQL

4. Truy vấn trên nhiều bảng – dùng JOIN

- Khi sử dụng JOIN để nối các bảng cần phải lưu ý những bảng này phải có các cột liên hệ với nhau và thứ tự quan hệ được chỉ định giữa các bảng sẽ làm ảnh hưởng tới kết quả truy vấn.

INNER JOIN : chỉ định việc so sánh giá trị trong các cột của các bảng là tương đương → dữ liệu đều phải có ở cả hai bảng

- Cú pháp:

SELECT Danh_sách_các_cột

FROM Tên_bảng

TRUY VẤN TRONG MY SQL

INNER JOIN : chỉ định việc so sánh giá trị trong các cột của các bảng là tương đương → dữ liệu đều phải có ở cả hai bảng

- Cú pháp:

SELECT Danh_sách_các_cột

FROM Tên_bảng

INNER JOIN Tên_bảng_liên_kết **ON** Điều_kiện_liên_kết

[**WHERE** Điều_kiện_lọc]

[**ORDER BY** Danh_sách_các_cột_sắp_xếp [**DESC**]]

TRUY VẤN TRONG MY SQL

- Ví dụ : trong in ra thông tin chi tiết của các hóa đơn của mã hóa đơn ' 1' gồm : ma_hoa_don; ma_mon; so_luong

```
SELECT chi_tiet_hoa_don.ma_hoa_don,  
ma_mon,so_luong  
FROM chi_tiet_hoa_don  
INNER JOIN hoa_don ON chi_tiet_hoa_don.ma_hoa_don  
= hoa_don.ma_hoa_don  
WHERE chi_tiet_hoa_don.ma_hoa_don ="1"
```

ma_hoa_don	ma_mon	so_luong
1	1	1
1	2	2
1	30	2

TRUY VẤN TRONG MY SQL

LEFT / RIGHT JOIN : chỉ định việc so sánh giá trị trong các cột của các bảng được ưu tiên cho mối quan hệ bên nhánh trái / nhánh phải. Việc thay đổi thứ tự ưu tiên này sẽ làm ảnh hưởng đến kết quả truy vấn

- Cú pháp:

SELECT Danh_sách_các_cột

FROM Tên_bảng

LEFT | RIGHT JOIN Tên_bảng_liên_kết **ON** Điều_kiện_liên_kết

[WHERE Điều_kiện_lọc]

[ORDER BY Danh_sách_các_cột_sắp_xếp **[DESC]]**

TRUY VẤN TRONG MY SQL

- Ví dụ : hiển thị danh sách tất cả các khách hàng có trong bảng khách_hang và bảng hoa_don gồm :
ma_khach_hang; ten_khach_hang; ngay_dat.

```
SELECT khách_hang.ma_khach_hang,  
       khách_hang.ten_khach_hang,  
       hoa_don.ma_hoa_don,  
       hoa_don.ngay_dat  
FROM khách_hang LEFT JOIN hoa_don  
      ON khách_hang.ma_khach_hang = hoa_don.ma_hoa_don  
WHERE khách_hang.ma_khach_hang IN(1,2,3,4,5)
```

ma_khach_hang	ten_khach_hang	ma_hoa_don	ngay_dat
1	Nguyễn Hải Yến	1	2018-03-17
2	Trương Thị Khánh Uyên	2	2018-03-18
3	Nguyễn Văn Hòa	3	2018-03-07
4	Ngô Thị Nguyệt	4	2018-03-05
5	Đinh Duy Minh	5	2018-03-01

TRUY VẤN TRONG MY SQL

5. Truy vấn nhóm- GROUP BY

- Giúp nhóm dữ liệu của các dòng dữ liệu bên trong bảng và sử dụng thêm các hàm thống kê đi kèm để tính toán dữ liệu
- Cú pháp:

```
SELECT Danh_sách_các_cột,Hàm_thống_kê [AS Tên]  
FROM Tên_bảng  
[WHERE Điều_kiện_lọc]  
GROUP BY Danh_sách_các_cột_nhóm_dl  
[ORDER BY Danh_sách_các_cột_sắp_xếp [DESC] ]
```

TRUY VẤN TRONG MY SQL

Các hàm thống kê

- AVG : hàm trả về giá trị trung bình theo nhóm
- COUNT: hàm trả về số lượng mẫu tin theo nhóm
- MIN: hàm trả về giá trị nhỏ nhất theo nhóm
- MAX: hàm trả về giá trị lớn nhất theo nhóm
- SUM: hàm trả về tổng các giá trị theo nhóm

TRUY VẤN TRONG MY SQL

- Ví dụ: Thống kê số lượng món ăn của mỗi loại món ăn, sắp xếp tăng dần theo số lượng

```
SELECT loai_mon_an.ma_loai, loai_mon_an.ten_loai,  
       COUNT(mon_an.ma_mon) AS so_luong_mon  
FROM loai_mon_an INNER JOIN mon_an  
     ON loai_mon_an.ma_loai = mon_an.ma_loai  
GROUP BY loai_mon_an.ma_loai, loai_mon_an.ten_loai  
ORDER BY so_luong_mon
```

ma_loai	ten_loai	so_luong_mon	▲ 1
5	Món Súp	2	
8	Tráng miệng	2	
7	Món Sào	3	
2	Món Cơm	4	
6	Món gỏi	4	
1	Món canh	5	
3	Món Mặn	6	
4	Ăn sáng	7	

TRUY VẤN TRONG MY SQL

5. Truy vấn nhóm- GROUP BY - HAVING

- HAVING giúp lọc lại dữ liệu sau khi đã gom nhóm bằng GROUP BY.
- Cú pháp:

```
SELECT Danh_sách_các_cột,Hàm_thống_kê [AS Tên]  
FROM Tên_bảng  
[WHERE Điều_kiện_lọc]  
GROUP BY Danh_sách_các_cột_nhóm_dl  
HAVING Điều_kiện_lọc_sau_gom_nhóm  
[ORDER BY Danh_sách_các_cột_sắp_xếp [DESC] ]
```

TRUY VẤN TRONG MY SQL

- Ví dụ: Thống kê số lượng món ăn của mỗi loại món ăn, sắp xếp tăng dần theo số lượng và chỉ lọc ra những loại có số lượng món >5

```
SELECT loai_mon_an.ma_loai,  
       loai_mon_an.ten_loai,  
       COUNT(mon_an.ma_mon) AS so_luong_mon  
FROM loai_mon_an INNER JOIN mon_an  
     ON loai_mon_an.ma_loai = mon_an.ma_loai  
GROUP BY loai_mon_an.ma_loai,  
         loai_mon_an.ten_loai  
HAVING so_luong_mon >5  
ORDER BY so_luong_mon
```

ma_loai	ten_loai	so_luong_mon
3	Món Mặn	6
4	Ăn sáng	7

ma_loai	ten_loai	so_luong_mon
5	Món Súp	2
8	Tráng miệng	2
7	Món Sào	3
2	Món Cơm	4
6	Món gỏi	4
1	Món canh	5
3	Món Mặn	6
4	Ăn sáng	7

TRUY VẤN TRONG MY SQL

6. Truy vấn con

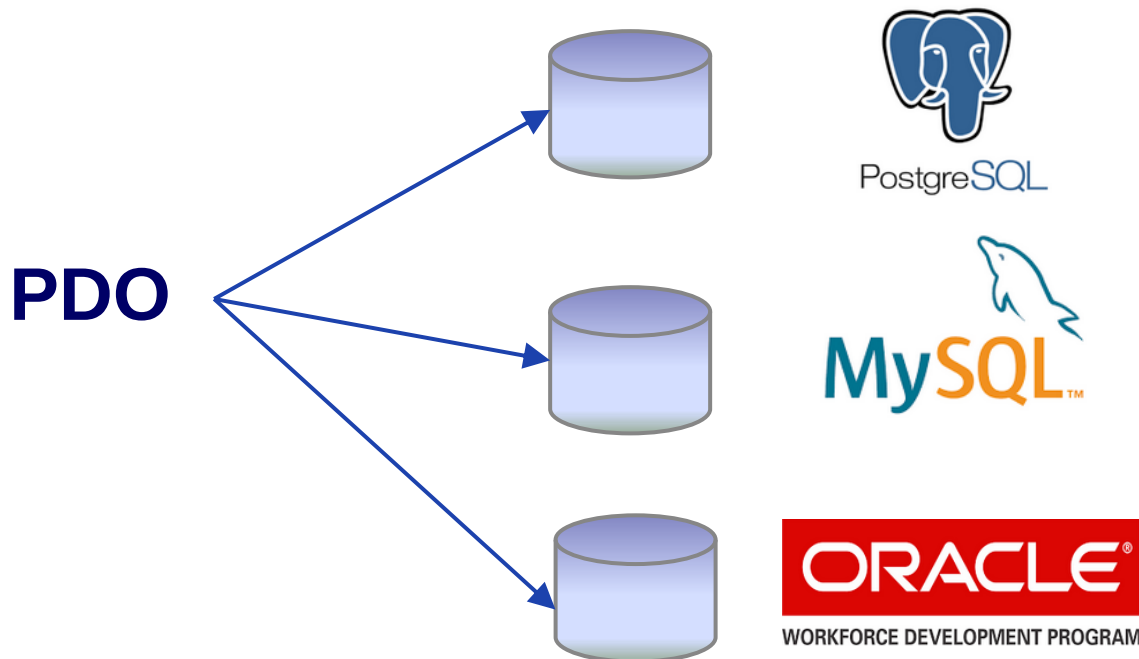
7. Truy vấn cập nhật (Insert, Update, Delete)

Về nhà xem lại học phần CSDL

KẾT HỢP PHP VÀ MYSQL – PDO (Library)

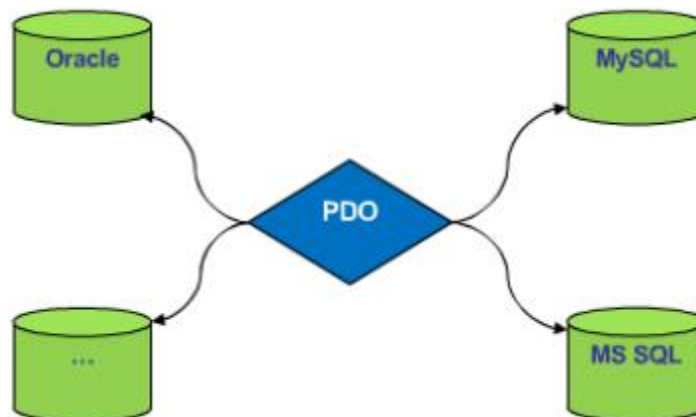
1. Giới thiệu

- **PDO** - PHP Data Object – là một lớp truy cập dữ liệu (data base access layer) cung cấp phương thức chuẩn để truy cập vào nhiều loại CSDL khác nhau



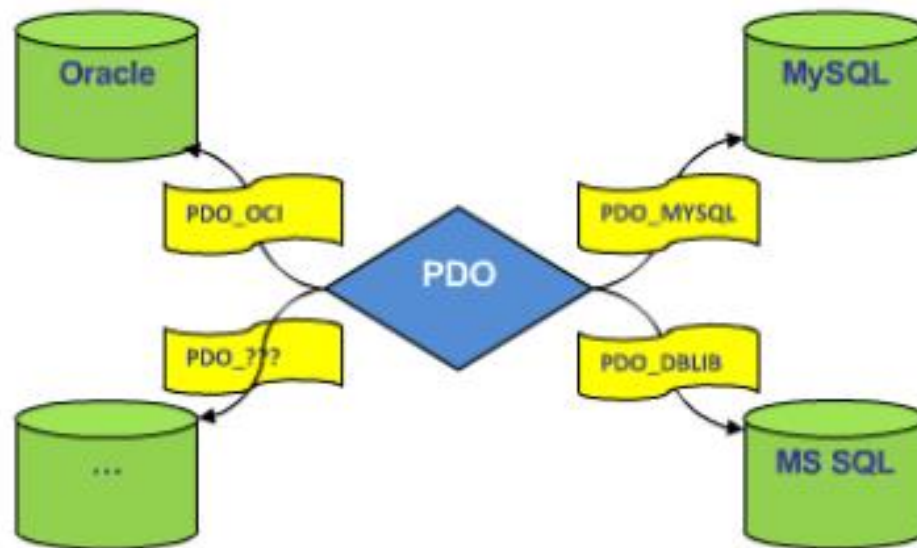
KẾT HỢP PHP VÀ MYSQL – PDO (Library)

- **PDO** cung cấp một lớp truy cập dữ liệu trừu tượng, nghĩa là dù ta đang sử dụng CSDL nào, thì ta đều có thể sử dụng các phương thức như nhau để truy vấn và lấy dữ liệu.
- **PDO** theo hướng đối tượng → phù hợp để tích hợp vào mô hình MVC (rất nhiều frame work lớn hiện nay đều sd PDO)



KẾT HỢP PHP VÀ MYSQL – PDO (Library)

- Việc thao tác với từng loại CSDL sẽ do từng driver tương ứng đảm nhiệm



2. Tạo kết nối

- **Cú pháp:**

```
$pdo = new PDO (<driver>, <username>,  
                <password>)
```

- **Tham số:**

<driver> : tên driver dùng để làm việc với csdl

<username>: tên đăng nhập vào CSDL

<password>: password dùng để truy cập CSDL

KẾT HỢP PHP VÀ MYSQL – PDO (Library)

- **Server hỗ trợ khá nhiều driver:**
 - PDO_DBLIB (Microsoft SQL Server)
 - PDO_IBM (IBM DB2)
 - PDO_MYSQL (MySQL 3.x / 4.x / 5.x)
 - PDO_OCI (Oracle Call Interface)
 - PDO_PGSQL (PosgreSQL)
 -
- **Để xem Server hỗ trợ được những driver nào**

```
<?php
```

```
print_r(PDO::getAvailableDrivers());
```

```
?>
```

```
Array ( [0] => mysql [1] => sqlite )
```

KẾT HỢP PHP VÀ MYSQL – PDO (Library)

- Ví dụ : Kết nối đến MySQL

```
$pdo = new PDO
```

```
('mysql:host=localhost;dbname=ql_nha_hang', 'root', '');
```

- Ví dụ : Kết nối đến MsSQL

```
$pdo = new PDO
```

```
('dblib:host=localhost;dbname=ql_nha_hang', 'root', '');
```

3. Hủy kết nối

- Cú pháp:

```
$pdo = NULL;
```

4. Xử lý lỗi kết nối

```
Try {.....  
}
```

```
Catch (PDOException $ex)
```

```
{ echo “lỗi kết nối ! ”. $ex -> getMessage();  
  die(); // kết thúc  
}
```

KẾT HỢP PHP VÀ MYSQL – PDO (Library)

```
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Untitled Document</title>
6 <?php
7 try{
8     $pdo = new PDO("mysql:host=localhost;dbname=ql_nha_hang","root","");
9     $pdo->query("set names utf8");
10 }
11 catch (PDOException $ex){
12     echo "Lỗi kết nối" . $ex->getMessage();
13     die();
14 }
15 }
16 ?>
17 </head>
```



Kết nối thất bại

3. Thực thi câu lệnh SQL (Insert, Update, Delete)

- Cú pháp:

`$pdo->exec(<query>);`

- Tham số:

Query là tham số bắt buộc, là câu lệnh truy vấn được gửi đi

`$pdo->exec()`: kết quả là tổng số dòng bị ảnh hưởng

Nếu không có dòng nào bị tác động thì kết quả là 0.

Nếu không thực thi được câu lệnh thì kết quả là False

KẾT HỢP PHP VÀ MYSQL – PDO (Library)

- Ví dụ : Thực hiện xóa tất cả các dòng dữ liệu trong bảng thanh_vien (CSDL ql_ban_sua)

```
$sql ='delete from thanhvien ';
```

```
$pdo->exec($sql);
```

				IDTV	username	password	Level			
		Edit		Copy		Delete	0	mai	12345	1
		Edit		Copy		Delete	1	Linh	123456	0



IDTV	username	password	Level
------	----------	----------	-------

Xóa thành công !

3. Thực thi câu lệnh SQL (Select)

- Cú pháp:

`$pdo->query(<query>);`

- Tham số:

Query là tham số bắt buộc, là câu lệnh truy vấn được gửi đi

- Ví dụ : Thực hiện lấy thông tin khách hàng (ma_Khach_hang, ten_khach_hang, email)

```
$sql ='select ma_Khach_hang, ten_khach_hang, email from khách_hang ';
```

```
$pdo->query($sql);
```

PDO - Statement

SQL - injection

- **SQL injection** là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update, v.v. trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy. SQL injection thường được biết đến như là một vật trung gian tấn công trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase

PDO - Statement

1. Prepared statement

- Sử dụng Prepared statement sẽ giúp chúng ta tránh khỏi SQL injection

- **Cú pháp:**

```
$sta = $pdo ->prepare(<query>);
```

- **Tham số:**

Query là tham số bắt buộc, là câu lệnh truy vấn được gửi đi

- **Ví dụ:**

```
$sta = $pdo->prepare(' SELECT * FROM mon_an');
```

2. Thực thi Prepared statement

- Cú pháp:

```
$sta->execute();
```

PDO - Statement

3. Duyệt dữ liệu

- Duyệt lần lượt từng mẫu tin:

- Cú pháp:

`PDOStatement->fetch([int $fetch_style])`

- Dữ liệu được lấy thông qua phương thức `fetch()`.

Trước khi gọi `fetch()`, cần phải cho PDO biết các dữ liệu muốn lấy thông qua tham số tùy chọn `$fetch_style`



PDO - Statement

- **Có các \$fetch_style sau:**
 - PDO::FETCH_NUM : kết quả trả về một mảng (có chỉ số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng \$row[0], \$row[1],...
 - PDO::FETCH_ASSOC : kết quả trả về là một mảng (có chỉ số chuỗi) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng \$row["tên cột 1"], \$row["tên cột 2"],...

PDO - Statement

- **Có các \$fetch_style sau:**
 - PDO::FETCH_BOTH : kết quả trả về là một mảng (có chỉ số chuỗi và số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng \$row[0], \$row[1],...hoặc \$row["tên cột 1"], \$row["tên cột 2"],...
 - PDO::FETCH_OBJ : kết quả trả về là một mẫu tin trong bộ các mẫu tin như là một đối tượng → truy cập bằng cách gọi từng thuộc tính của đối tượng: \$Ten_doi_tuong → ten_cot_1 , \$Ten_doi_tuong → ten_cot_2,...

PDO - Statement

3. Duyệt dữ liệu

- Duyệt danh sách các mẫu tin:

- Cú pháp:

**PDOStatement->fetchAll([int \$fetch_style],
[**\$column_index**])**

- Tham số:

\$fetch_style : như trên

PDO::FETCH_COLUMN: kết quả trả về là mảng một chiều chứa tất cả các giá trị của một cột được chỉ định.

\$column_index: kết quả trả về là cột được chỉ định khi \$fetch_style có giá trị là PDO::FETCH_COLUMN

PDO - Statement

- Ví dụ: Hiển thị danh sách món ăn trong bảng mon_an như sau y/c dùng PDO- Statement

```
<?php
include ("Connection.php");
$sql = "select ma_mon, ten_mon, noi_dung_tom_tat, don_gia,
$sta = $pdo->prepare($sql);
$sta->execute();

if($sta->rowCount()>0)
{
    $mon_an = $sta->fetchAll(PDO::FETCH_OBJ);
}
```

```
<body>
<div id="main">
<h1>DANH SÁCH MÓN ĂN</h1>
<?php
foreach ($mon_an as $mon)
{
    <div class="khung">
        
        <h3><?php echo $mon->ten_mon; ?></h3>
        Com tấm từ lâu đã là một nét đặc trưng của ẩm thực
        nước mắm đậm đà và com dẻo thơm.
        <p>Đơn giá : 50.000 VNĐ</p>
    </div>
}
```

P/CSDL/DanhSachMonAn.php

DANH SÁCH MÓN ĂN



Com Tấm

Com tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa com tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm đà và com dẻo thơm.

Đơn giá : 50.000 VNĐ



Com Gà

Com tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa com tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm đà và com dẻo thơm.

Đơn giá : 50.000 VNĐ



Com Sườn

Com tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa com tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm đà và com dẻo thơm.

Đơn giá : 50.000 VNĐ

4. Đếm số lượng mẫu tin

- Cú pháp:

`$pdo->rowCount();` → số lượng mẫu tin bị tác động bởi câu lệnh Select;

`$sta->rowCount();` → số lượng mẫu tin bị tác động bởi câu lệnh Insert, Update, Delete

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Định dạng hiển thị dữ liệu

- Cách đơn giản nhất là hiển thị dữ liệu dạng bảng hoặc dạng text.

- Ví dụ :

THÔNG TIN KHÁCH HÀNG

Mã KH	Tên KH	Email	Địa Chỉ	Điện thoại	Hình
1	Nguyễn Hải Yến	yennh@cntp.edu.vn	140 Lê Trọng Tấn	08123456	
2	Trương Thị Khánh Uyên	uyenttk@gmail.com	123 Lê Trọng Tấn	090756489	
3	Nguyễn Văn Hòa	hoanv@gmail.com	123 Cầu Xéo	08987654	
4	Ngô Thị Nguyệt	nguyetnt@gmail.com	456 Tân kỳ tân quý	086532148	
5	Đinh Duy Minh	minhdd@gmail.com	698 Đỗ thừa Luông	01235478	
6	Nguyễn Chinh Tây	taync@gmail.com	546 Dương Văn Dương	12345678	
7	Nguyễn Việt Thu	thunv@gmail.com	457 Đỗ Thừa Luông	090123456	
8	Nguyễn Linh Giang	giangnl@gmail.com	1254 Gò Dầu	093568745	
9	Trương Thị Thu Hiền	hienttt@gmail.com	54 Lương Thế Vinh	01234587	

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Định dạng hiển thị dữ liệu

- Yêu cầu đơn giản nhất của một trang web là dữ liệu hiển thị phải được định dạng giúp cho người dùng dễ dàng đọc thông tin.
- Các định dạng thông thường là định dạng cột, dòng, kích thước, màu sắc, kiểu chữ,... Cho nội dung hiển thị.
- Để thực hiện được các yêu cầu về định dạng → Cần phải xem xét và tính toán cho các nội dung cần hiển thị.

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Ví dụ 1:

- Yêu cầu hiển thị thông tin khách hàng gồm MaKH, Tên KH, Email, địa chỉ, điện thoại từ table khách_hang như sau

THÔNG TIN KHÁCH HÀNG

Mã KH	Tên KH	Email	Địa Chỉ	Điện thoại	Hình
1	Nguyễn Hải Yến	yennh@cntp.edu.vn	140 Lê Trọng Tấn	08123456	
2	Trương Thị Khánh Uyên	uyenttk@gmail.com	123 Lê Trọng Tấn	090756489	
3	Nguyễn Văn Hòa	hoanv@gmail.com	123 Cầu Xéo	08987654	
4	Ngô Thị Nguyệt	nguyetnt@gmail.com	456 Tân kỳ tân quý	086532148	
5	Đinh Duy Minh	minhdd@gmail.com	698 Đỗ thừa Luông	01235478	
6	Nguyễn Chinh Tây	taync@gmail.com	546 Dương Văn Dương	12345678	
7	Nguyễn Việt Thu	thunv@gmail.com	457 Đỗ Thừa Luông	090123456	
8	Nguyễn Linh Giang	giangnl@gmail.com	1254 Gò Dầu	093568745	
9	Trương Thị Thu Hiền	hienttt@gmail.com	54 Lương Thế Vinh	01234587	

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Cách thực hiện:

- Kết nối CSDL
- Xuất dữ liệu lên bảng

HIỂN THỊ DỮ LIỆU TRÊN TRANG

- Kết nối CSDL

```
7  <?php
8      $pdo = new PDO ("mysql:host=localhost;dbname=ql_nha_hang","root","");
9      $pdo->query("set names utf8");
10     $sql = "select * from khách_hang";
11     $khach_hang = $pdo->query($sql);
12     echo "Số khách hàng : ".$khach_hang->rowCount()."<br>";
13
14  ?>
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

- Xuất dữ liệu lên bảng

```
<table width="800" border="2" cellspacing="0" cellpadding="5" align="center">
  <caption><H1>THÔNG TIN KHÁCH HÀNG</H1></caption>
  <tr align="center" style="font-weight:bold" bgcolor="#CCFFFF">
    <td>Mã KH</td>
    <td>Tên KH</td>
    <td>Email</td>
    <td>Địa Chỉ</td>
    <td>Điện thoại</td>
    <td>Hình</td>
  </tr>
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

- Xuất dữ liệu lên bảng

```
<?php
foreach ($khach_hang as $kh)
{
?>
<tr>
    <td align="center"><?php echo $kh[0]?></td>
    <td><?php echo $kh["ten_khach_hang"]?></td>
    <td><?php echo $kh[2]?></td>
    <td><?php echo $kh[3]?></td>
    <td><?php echo $kh[4]?></td>
    <td align="center"><img width="50px" src = "image/<?php echo $kh[5]?>" />
</td>
</tr>
<?php
}
?>
</table>

<?php
}
$pdo = NULL;
?>
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Ví dụ 2:

- Yêu cầu hiển thị thông tin danh sách món ăn như sau:

DANH SÁCH MÓN ĂN	
 960 x 960 - vi-vn.facebook.com	Cơm Tấm Cơm tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa cơm tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm đà và cơm dẻo thơm. Đơn giá : 50.000 VNĐ
	Cơm Gà Cơm tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa cơm tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm đà và cơm dẻo thơm. Đơn giá : 50.000 VNĐ
	Cơm Sườn Cơm tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn.

Cách thực hiện:

- Tạo file Default.css
- Kết nối CSDL bằng PDOStatement
- Xuất dữ liệu lên trang

HIỂN THỊ DỮ LIỆU TRÊN TRANG

- Tạo file Default.css

```
1  /* CSS Document */
2  body {
3      background-color:#CFF;
4  }
5  h1
6      {
7      text-align:center;
8      color:#F00;
9      }
10 #main
11     {
12     width:700px;
13     margin: 0 auto;
14     border:#906 solid 2px;
15     background-color:#FFF;
16     }
```

```
17 .khung
18     {
19     width:680px;
20     height:250px;
21     border-bottom:#000 solid 1px;
22     padding:10px;
23     }
24 .khung img
25     {
26     width:250px;
27     float:left;
28     margin-right:15px;
29     }
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

- Kết nối CSDL bằng PDOStatement

```
5 <title>Danh sách món ăn</title>
6 <!-- nhúng default.css-->
7 <link href="css/Default.css" rel="stylesheet" type="text/css" />
8
9 </head>
10 <?php
11     include ("Connection.php");
12     $sql = "select ma_mon, ten_mon, noi_dung_tom_tat, don_gia, hinh from
mon_an";
13     $sta = $pdo->prepare($sql);
14     $sta->execute();
15
16     if($sta->rowCount()>0)
17     {
18         $mon_an = $sta->fetchAll(PDO::FETCH_OBJ);
19     }
20
21 ?>
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

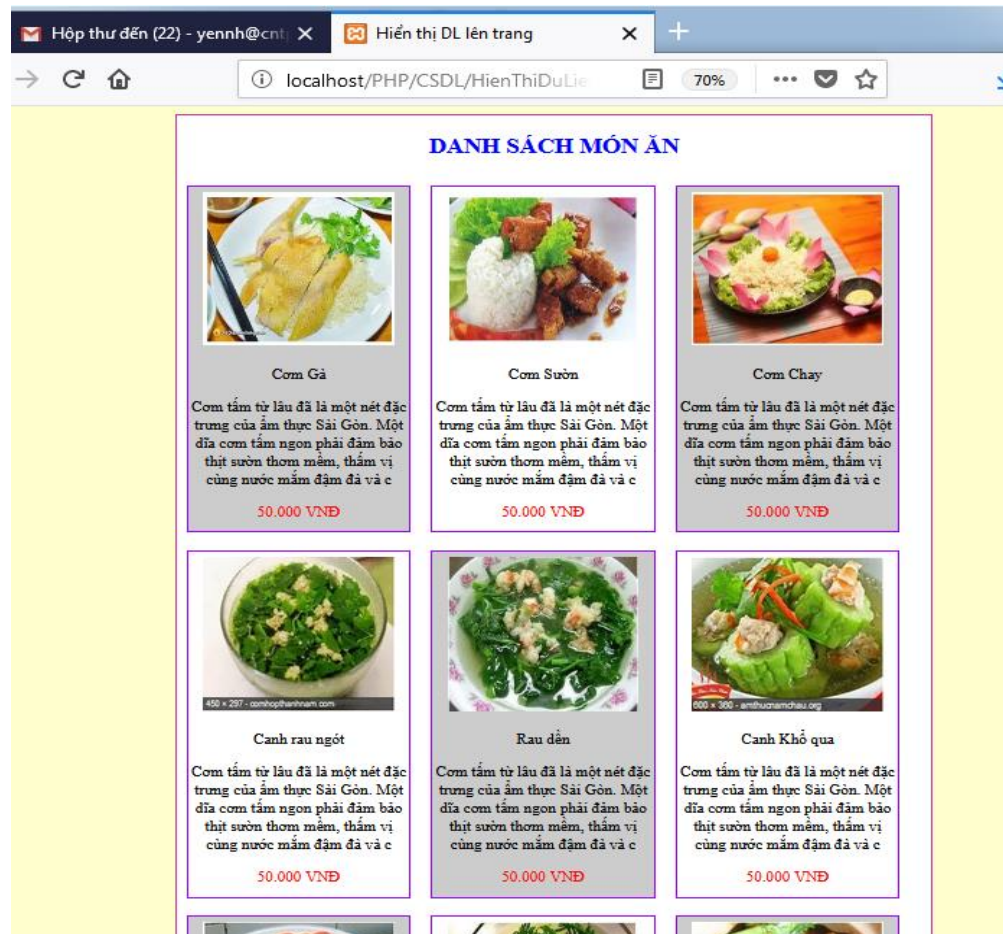
- Xuất dữ liệu lên trang

```
22 <body>
23     <div id="main">
24         <h1>DANH SÁCH MÓN ĂN</h1>
25         <?php
26         foreach ($mon_an as $mon)
27         {
28             <?>
29                 <div class="khung">
30                     
31                     <h3><?php echo $mon->ten_mon;?></h3>
32                     Com tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một
dĩa com tấm ngon phải đảm bảo thịt sườn thơm mềm, thấm vị cùng nước mắm đậm
dà và com dẻo thơm.
33                     <p>Đơn giá : 50.000 VNĐ</p>
34                 </div>
35                 <?php
36                 }
37             <?>
38         </div>
39 </body>
```

HIỂN THỊ DỮ LIỆU TRÊN TRANG

Ví dụ 3:

- Yêu cầu hiển thị thông tin danh sách món ăn như sau:



HIỂN THỊ DỮ LIỆU TRÊN TRANG

Cách thực hiện:

- Tạo file style.css
- Kết nối CSDL bằng PDOStatement
- Xuất dữ liệu lên trang (kiểm tra vị trí lẻ đổi màu , chặn bt).

1. Giới thiệu

- Đối với những bảng có số lượng mẫu tin lớn, khi hiển thị dữ liệu làm cho người dùng khó xem → thực hiện phân trang.
- Phân trang dữ liệu giúp cho việc xem và tìm kiếm thông tin dễ dàng, nhanh chóng, thuận tiện
- Phân trang là chia tách các kết quả từ cơ sở dữ liệu thành các trang riêng biệt và hiển thị các liên kết đến tất cả các trang

PHÂN TRANG

2. Làm sao để phân trang

Giả sử có danh sách gồm 19 mon_an, để dl hiển thị trên trang được rộng rãi , dễ theo dõi → chỉ muốn xem 2 món ăn trên một trang → chia nhỏ 19 món ăn này vào các trang:

❖ Trang 1 có 2 món từ món 0 => 1

❖ Trang 2 có 2 món từ món 2 => 3

.....

❖ Trang 9 có 2 món từ món 17 => 18

❖ Trang 10 có 1 món từ món 19 => 19

→ dữ liệu sẽ được hiển thị trong 10 trang.

2. Làm sao để phân trang

- Xác định được tổng số trang là bao nhiêu ?

Tiến hành một phép chia dựa trên tổng số records và số records/1 trang:

$\$totalPage = \$totalRecords / \$recordPerpage;$

19 records → $\$totalPage = (19/2);$

→ 9 trang

→ dư 1 record?

Phương án giải quyết?

→ Dùng hàm **ceil()**: Hàm làm tròn phép chia ở cận trên

$\$totalPage = \text{ceil}(\$totalRecords / \$recordPerPage);$

→ Trả về 10 trang

2. Làm sao để phân trang

- Để tính được 2 món tương ứng với từng trang hay bắt đầu của một trang là món nào và kết thúc là món nào ta dùng câu lệnh mySQL ?

`SELECT * FROM mon_an LIMIT vi_tri, 2`

Trong đó **vi_tri** tính bởi công thức

$$\text{vi_tri} = (\text{trang hiện tại} - 1) * 2$$

Ví dụ: Ở trang thứ 6: **vi_tri** = $(6 - 1) * 2 = 10$

➔ Ở trang thứ 6 thì vị trí trả về = 10

3. Các bước phân trang

- **Bước 1:** Kết nối CSDL
- **Bước 2 – 6 :** tạo trang Pagination.php
- **Bước 7:** tạo trang DSMonAn_PhanTrang1.php
- **Bước 8:** hiển thị link phân trang

3. Các bước phân trang

- **Bước 1:** Kết nối CSDL

```
5 <title>Untitled Document</title>
6 <?php
7 try{
8     $pdo = new PDO("mysql:host=localhost;dbname=ql_nha_hang","root","");
9     $pdo->query("set names utf8");
10 }
11 catch (PDOException $ex){
12     echo "Lỗi kết nối" . $ex->getMessage();
13     die();
14 }
15 }
16 ?>
17 </head>
18
19 <body>
20 </body>
```

3. Các bước phân trang

- **Bước 2,3:** tạo trang Pagination.php

//Bước 2: Cài đặt cấu hình dùng để phân trang

```
$page_url = 'http://localhost/PHP/CSDL/DanhSachMonAn_PhanTrang1.php';  
$display = 2;  
$num links = 5;
```

//Bước 3: Tính tổng số hàng trong bảng dữ liệu

```
$sql = "select ma_mon, ten_mon, noi_dung_tom_tat, don_gia, hinh from mon_an";  
$sta = $pdo->prepare($sql);  
$sta->execute();  
  
if($sta->rowCount() > 0)  
{  
    $mon_an = $sta->fetchAll(PDO::FETCH_OBJ);  
}  
$total_rows = $sta->rowCount();
```

3. Các bước phân trang

- **Bước 4:** tạo trang Pagination.php

Lấy trang hiện tại từ URL qua phương thức
`$_GET['page']`

```
//Bước 4: Lấy trang hiện tại từ URL qua phương thức $_GET['page']  
if(isset($_GET['page']) && is_numeric($_GET['page']))  
{  
    $curr_page = $_GET['page'];  
}  
else  
{  
    //nếu không tồn tại trang thì mặc nhiên sẽ là trang 1  
    $curr_page = 1;  
}
```

Nếu không tồn tại trang thì mặc định sẽ là trang 1

3. Các bước phân trang

- **Bước 5:** tạo trang Pagination.php

Tìm vị trí món bắt đầu của một trang

Tổng số trang

```
$position = (($curr_page - 1) * $display);  
$total_pages = ceil($total_rows / $display);
```

3. Các bước phân trang

- **Bước 6:** tạo trang Pagination.php

Tính các trang bắt đầu và kết thúc của link phân trang

```
//Bước 6: Tính các trang bắt đầu và kết thúc của link phân trang ($start, $end)
//Trang bắt đầu
if($curr_page > $num_links)
    $start = $curr_page - ($num_links - 1);
else
    $start = 1;
//trang kết thúc
if(($curr_page + $num_links ) < $total_pages)
    $end = $curr_page + $num_links;
else
    $end = $total_pages;
```

3. Các bước phân trang

- Bước 7: Tạo trang DanhSachMonAn_PhanTrang1.php**

```
10 <?php
11     include ("Pagination.php");
12     $sql = "select ma_mon, ten_mon, noi_dung_tom_tat, don_gia, hinh from mon_an limit $position, $display";
13     $sta = $pdo->prepare($sql);
14     $sta->execute();
15     $mon_an = $sta->fetchAll(PDO::FETCH_OBJ);
16
17 ?>
18 <body>
19     <div id="main">
20     <h1>DANH SÁCH MÓN ĂN</h1>
21     <?php
22     foreach ($mon_an as $mon)
23     {
24     ?>
25         <div class="khung">
26             
27             <h3><?php echo $mon->ten_mon; ?></h3>
28             Com tấm từ lâu đã là một nét đặc trưng của ẩm thực Sài Gòn. Một đĩa com tấm ngon phải đảm bảo th
29             nước mắm đậm đà và com dẻo thơm.
30             <p>Đơn giá : 50.000 VNĐ</p>
31         </div>
32     <?php
33     }
```

PHÂN TRANG

3. Các bước phân trang **Hiển thị link phân trang**

```
38 <?php
39     if(isset($total_pages))
40     {
41         if($total_pages > 1) // nếu tổng số trang > 1 in dòng Page..of..
42         {
43             echo 'Page '.$curr_page. ' of '.$total_pages.'-----';
44             // nếu trang hiện tại lớn hơn số link muốn hiển thị
45             if($curr_page > $num_links)
46             {
47                 // thì hiển thị nút 'First'
48                 echo '<a href="'. $page_url. '?page=1">First</a>';
49             }
50             // nếu trang hiện tại > 1
51             if($curr_page > 1)
52             {
53                 // hiển thị nút 'Previous'
54                 echo '<a href="'. $page_url. '?page=' . ($curr_page-1) . ">Previous</a> '";
55             }
56         }
57     }
```

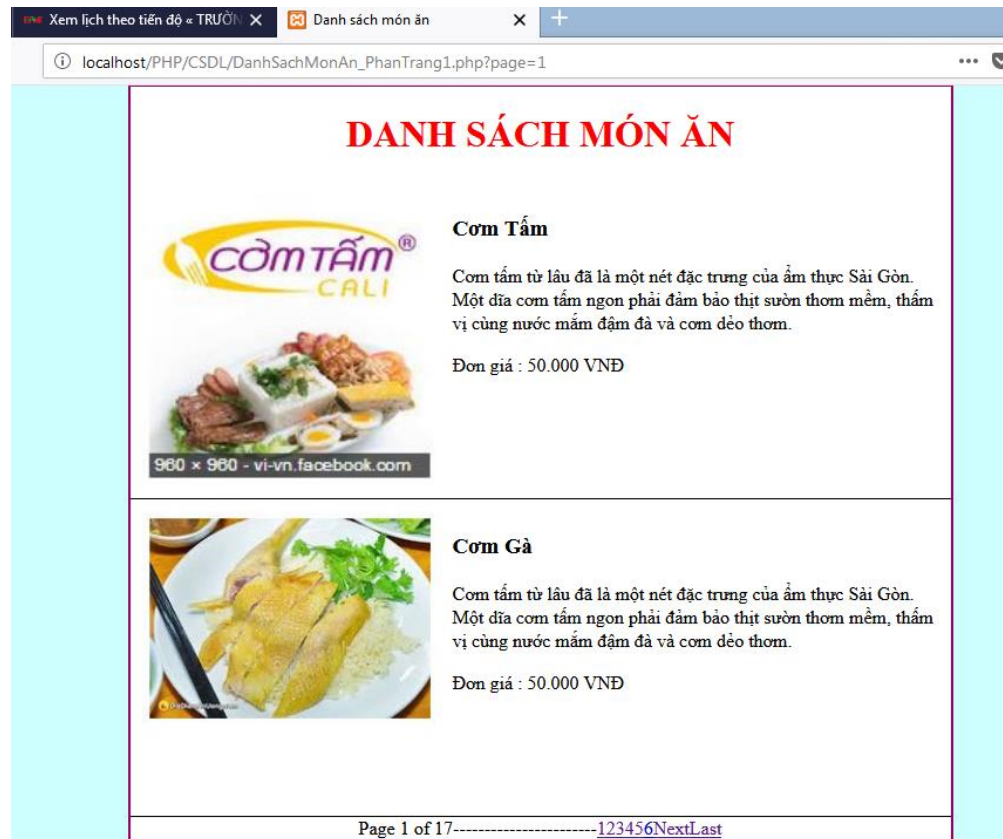
PHÂN TRANG

3. Các bước phân trang **Hiển thị link phân trang**

```
56 // hiển thị các link bao gồm trang hiện tại và link trang hiển thị
57 // (trái và phải) bắt đầu từ $start, kết thúc là $end
58 // $start và $end được tính trong pagination.php
59 for($pages = $start ; $pages <= $end ; $pages++)
60 {
61     if($pages == $curr_page)
62     {
63         echo '<a href="'. $page_url . '?page=' . $pages . '>' . $pages . '</a>';
64     }
65     else
66     {
67         echo '<a href="'. $page_url . '?page=' . $pages . '>' . $pages . '</a>';
68     }
69 }
70 // nếu trang hiện tại < tổng số trang
71 if($curr_page < $total_pages )
72 {
73     // thì hiển thị nút 'Next'
74     echo '<a href="'. $page_url . '?page=' . ($curr_page+1) . '>Next</a>';
75 }
76 // nếu trang hiện tại + số link muốn hiển thị (ở đây là + với số link bên phải) > tổng số trang
77 if(($curr_page + $num_links) < $total_pages )
78 {
79     // thì hiển thị nút 'Last'
80     echo '<a href="'. $page_url . '?page=' . $total_pages . '>Last</a>';
81 }
```

PHÂN TRANG

3. Các bước phân trang Hiển thị link phân trang



Page 1 of 17-----123456NextLast

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

Thiết kế một trang PHP có form cho phép thêm khách hàng vào table khách hàng các bước thực hiện như sau

Bước 1 : Thiết kế form

THÊM KHÁCH HÀNG MỚI

Tên Khách Hàng	
Email	
Địa chỉ	
Điện thoại	
Hình	Chọn tập tin... Chưa chọn tập tin.
Ghi chú	
SAVE	

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

Bước 2 : Yêu cầu kiểm tra tính hợp lệ của dữ liệu nhập bằng JS ()

THÊM KHÁCH HÀNG MỚI

Tên Khách Hàng	
Email	
Địa chỉ	
Điện thoại	
Hình	Chọn tập tin... Chưa chọn tập tin.
Ghi chú	
SAVE	

```
function kiemTra()  
{  
    var ten_KH = document.getElementById("th_tenKH");  
    if(ten_KH.value == "")  
    {  
        alert("Nhập tên Khách hàng");  
        ten_KH.focus();  
        return false;  
    }  
    return true;  
}
```

.....

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

Bước 3 : Kiểm tra nút SAVE có được nhấn hay không, nếu được nhấn thì lấy thông tin trên form gán vào các biến

```
12 // ma_khach_hang, ten_khach_hang, email, dia_chi, dien_thoai, hinh, ghi_chu
13 if(isset($_POST["btn_SAVE"]))
14 {
15     // Lấy thông tin trong form gán vào các biến
16     $ma_khach_hang =NULL;
17     $ten_khach_hang = $_POST["ten_KH"];
18     $email=$_POST["email_KH"];
19     $dia_chi= $_POST["diaChi_KH"];
20     $dien_thoai = $_POST["dienThoai_KH"];
21     $hinh = $_FILES["hinh"] ["error"]==0 ? $_FILES["hinh"] ["name"] : "";
22     $ghi_chu = $_POST["ghiChu"];
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

Bước 4 : Xây dựng câu lệnh thêm

```
24      // Xây dựng câu lệnh thêm
25      $sql = "INSERT INTO khách_hang VALUES(?, ?, ?, ?, ?, ?, ?)";
26      // Khai báo mảng tham số truyền các giá trị nhập từ form
27      $param = array($ma_khach_hang, $ten_khach_hang, $email, $dia_chi,
28      // Thực thi câu lệnh
29      $sta = $pdo->prepare($sql);
30      $kq = $sta->execute($param);
31
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

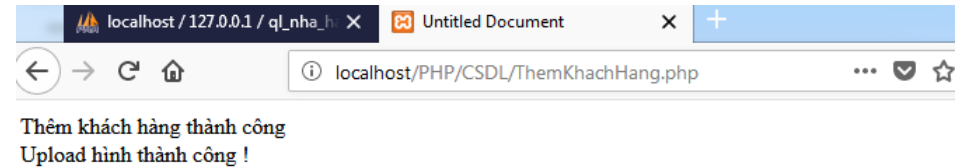
Bước 5 : Kiểm tra quá trình thêm

```
32     if($kq)
33     {
34         echo "Thêm khách hàng thành công";
35         // Di chuyển hình về server trong thư mục image
36         if ($hinh != "")
37         {
38             $kt = move_uploaded_file($_FILES["hinh"]["tmp_name"],"image/$hinh");
39             if ($kt)
40             {
41                 echo "<br>Upload hình thành công !";
42             }
43             else
44             {
45                 echo "<br> Upload hình không thành công !";
46             }
47         }
48     }
49     else
50     {
51         echo "Thêm bị lỗi , xem lại";
52     }
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

1. Thêm

Bước 6 : Chạy thử



THÊM KHÁCH HÀNG MỚI

Tên Khách Hàng	Trương Mạnh Hùng
Email	hungtm@gmail.com
Địa chỉ	12345789 Go Dau
Điện thoại	0123456789
Hình	Chọn tập tin... Penguins.jpg
Ghi chú	reisure
SAVE	

Server: 127.0.0.1 » Database: ql_nha_hang » Table: khách_hang "Khách hàng"

	ma_khach_hang	ten_khach_hang	email	dia_chi	dien_thoai	hinh_hinh	ghi_chu
☐ Edit ☐ Copy ☐ Delete	8	Nguyễn Linh Giang	giangnl@gmail.com	1254 Gò Dầu	093568745	H8.jpg	
☐ Edit ☐ Copy ☐ Delete	9	Trương Thị Thu Hiền	hienttt@gmail.com	54 Lương Thế Vinh	01234587	H9.jpg	
☐ Edit ☐ Copy ☐ Delete	10	Trương Thị Thanh Xuân	xuanttt@gmail.com	99 Trường Chinh	0124897333	H10.jpg	
☐ Edit ☐ Copy ☐ Delete	11	Trương Mạnh Hùng	hungtm@gmail.com	123456 GỖ đầu	0123456789	Penguins.jpg	news customer

☐ Check all With selected:

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

2. Xóa

Thiết kế một trang PHP có form cho phép xóa một khách hàng trong bảng khách hàng các bước thực hiện như sau

Bước 1 : Thiết kế form

XÓA KHÁCH HÀNG	
Tên Khách hàng	Trương Mạnh Hùng
DELETE	

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

2. Xóa

Bước 2 : Yêu cầu kiểm tra tính hợp lệ của dữ liệu cần xóa bằng JS ()

XÓA KHÁCH HÀNG

Tên Khách hàng	Trương Mạnh Hùng
DELETE	

Không được để trống

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

2. Xóa

Bước 3 : Kiểm tra nút DELETE có được nhấn không ? Nếu được nhấn thì lấy thông tin trên form làm tham số cho câu truy vấn delete.

```
11  if(isset($_POST["btn_Delete"]))
12  {
13      // Lấy thông tin trong form gán vào các biến
14      $ten_khach_hang = $_POST["ten_KH"];
15
16      //Xây dựng câu lệnh xóa
17      $sql = " DELETE FROM khach_hang WHERE ten_khach_hang = '$ten_khach_hang' ";
18      // Thực thi câu lệnh
19      $sta = $pdo->prepare($sql);
20      $kq = $sta->execute();
```

2. Xóa

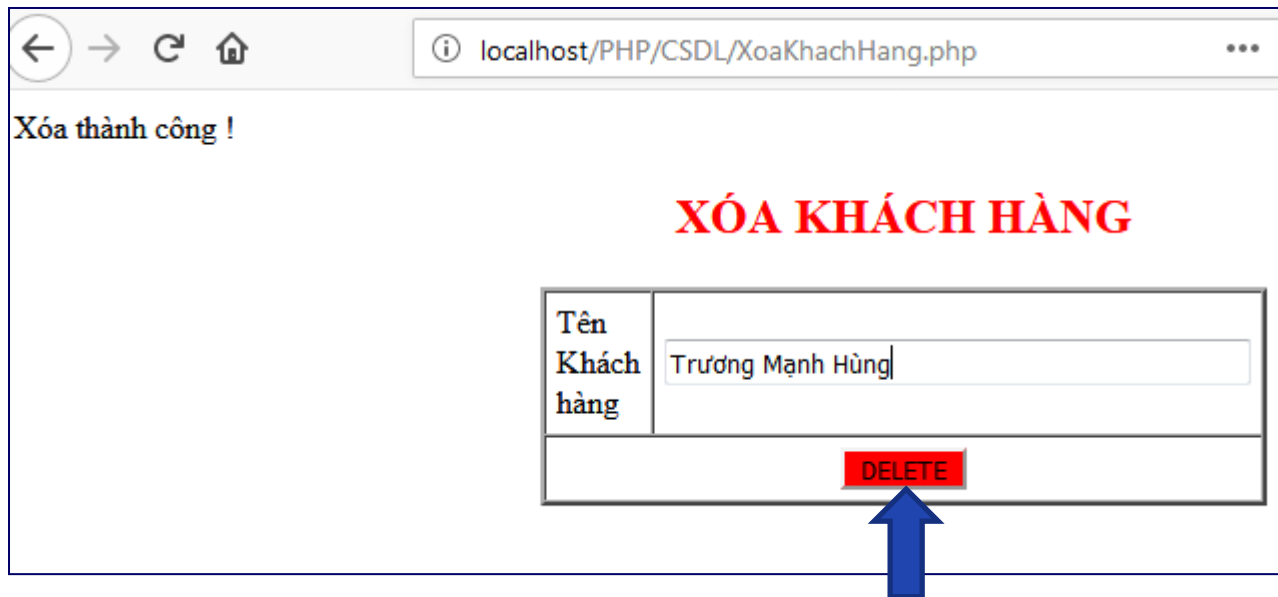
Bước 4 : Kiểm tra quá trình xóa

```
21         if ($kq)
22         {
23             echo "Xóa thành công !";
24         }
25     else
26     {
27         echo "Xóa thất bại, xem lại !";
28     }
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

2. Xóa

Bước 5 : Chạy thử



The screenshot shows a web browser window with the address bar displaying `localhost/PHP/CSDL/XoaKhachHang.php`. The page content includes a success message "Xóa thành công !" and a heading "XÓA KHÁCH HÀNG" in red. Below the heading is a form with a label "Tên Khách hàng" and a text input field containing "Trương Mạnh Hùng". At the bottom of the form is a red button labeled "DELETE". A blue arrow points to the "DELETE" button.

Xóa thành công !

XÓA KHÁCH HÀNG

Tên Khách hàng	Trương Mạnh Hùng
DELETE	

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

3. Cập nhật khách hàng

Bước 1 : Thiết kế form

localhost/PHP/CSDL/CapNhatKhachHang.php

CẬP NHẬT KHÁCH HÀNG

Chọn Khách hàng cần cập nhật TT					
TÊN KHÁCH HÀNG	EMAIL		ĐIỆN THOẠI	HÌNH	GHI CHÚ
		Nguyễn Hải Yến Nguyễn Hải Yến Trương Thị Khánh Uyên Nguyễn Văn Hòa Ngô Thị Nguyệt Đinh Duy Minh Nguyễn Chinh Tây Nguyễn Việt Thu Nguyễn Linh Giang Trương Thị Thu Hiền Trương Thị Thanh Xuân Trương Mạnh			
			UPDATE		

Bước 2 : Chọn nhân viên cần cập nhật từ list

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

3. Cập nhật khách hàng

Bước 3 : Kiểm tra nút UPDATE có được nhấn hay không, nếu được nhấn thì lấy thông tin trên form gán vào các biến

```
21      // Lấy thông tin trong form gán vào biến
22      $maKH = $_POST["ma_KH"];
23      $ten_khach_hang = $_POST["ten_KH"];
24      $email = $_POST["email_KH"];
25      $dia_chi = $_POST["diaChi_KH"];
26      $dien_thoai = $_POST["dienThoai_KH"];
27      $hinh = $_FILES["hinh"]["error"]==0 ? $_FILES["hinh"]["name"] : "";
28      $ghi_chu = $_POST["ghiChu"];
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

3. Cập nhật khách hàng

Bước 4 : Xây dựng câu lệnh cập nhật

```
30      //Xây dựng câu lệnh cập nhật
31      $sql ="UPDATE khách_hang
32              SET ten_khach_hang='$ten_khach_hang', email='$email', dia_chi='$dia_chi',
33                  dien_thoai = '$dien_thoai', ghi_chu = '$ghi_chu'
34              WHERE ma_khach_hang = '$maKH' ";
35
36      // Thực thi lệnh
37      $sta = $pdo->prepare($sql);
38      $kq = $sta->execute();
39
```

Bước 5 : Kiểm tra quá trình cập nhật

```
40      if($kq)
41      {
42          echo "Cập nhật khách hàng thành công";
43      }
44      else
45      {
46          echo "Cập nhật lỗi , xem lại";
47      }
48
```

THỰC HIỆN THÊM XÓA SỬA TRÊN CSDL

3. Cập nhật khách hàng

Bước 6 : Chạy thử

DL trước khi cập nhật

10	Trương Thị Thanh Xuân	xuanttt@gmail.com	99 Trường Chinh	0124897333	H10.jpg
11	Trương Mạnh	hungtm@gmail.com	123 Gò Dầu	0123456789	Penguins.jpg lobolo lobololobolo

CẬP NHẬT KHÁCH HÀNG

Chọn Khách hàng cần cập nhật TT		Trương Mạnh ▾			
TÊN KHÁCH HÀNG	EMAIL	ĐỊA CHỈ	ĐIỆN THOẠI	HÌNH	GHI CHÚ
Trương Mạnh Hùng	hungtm@gmail.com	123 Gò Dầu	0123456789	Chọn tập tin... Chưa chọn tập tin.	la ba la la ba la la ba la
UPDATE					

Sau cập nhật

10	Trương Thị Thanh Xuân	xuanttt@gmail.com	99 Trường Chinh	0124897333	H10.jpg
11	Trương Mạnh Hùng	hungtm@gmail.com	123 Gò Dầu	0123456789	Penguins.jpg la ba la la ba la la ba la

PHP EMAIL – SECURITY EMAIL

1. Giới thiệu

- Để gửi mail bằng PHP, cần phải có một web Server cho hệ thống quản lý Email gọi là Mail Server
- Khi cài đặt một Mail Server cần phải có một server và một trong các phần mềm Mail Server như: MS Exchange Server, Apache,...
- Khi cài đặt thành công mail server ta có thể cấu hình nhiều địa chỉ Email cho nhiều người sử dụng là nhân viên hay khách hàng,...
- Mỗi Email được gửi qua Internet cần phải có địa chỉ email người gửi và người nhận trong đó Email người gửi phải tồn tại trong

2. Phương thức gửi mail

- Cú pháp

Mail(to, subject, message, headers, parameters)

- Trong đó các tham số
 - **To**: Tham số bắt buộc, địa chỉ của người nhận mail
 - **Subject**: ts bắt buộc , chủ đề email (kg chứa ký tự xuống dòng)
 - **Message**: ts bắt buộc, là nội dung được gửi đi. Chú ý mỗi dòng được tách ra với một (\n) và một dòng không vượt quá 70 ký tự.
 - **Headers**: ts tùy chọn, là các phần bổ sung vào header như From, cc, Bcc. Chú ý các phần bổ sung được tách ra với (\r \n).
 - **Parameter**: ts tùy chọn , chỉ định tham số bổ sung cho chương trình send mail

PHP EMAIL – SECURITY EMAIL

3. Gửi Email dạng văn bản

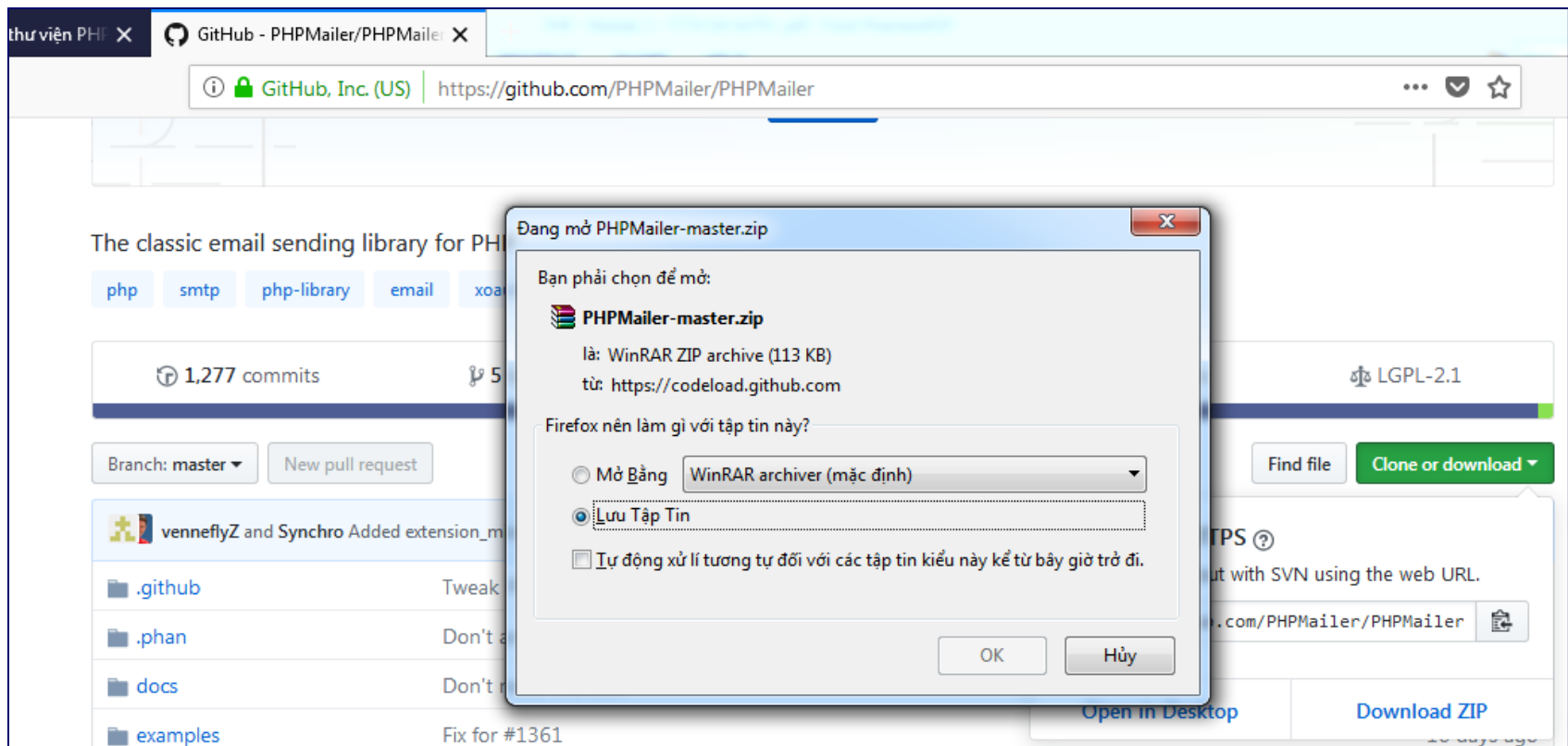
- Lần lượt khai báo và gán giá trị cho các biến \$to, \$subject, \$message, \$from, \$headers, sau đó sử dụng phương thức mail để gửi email
- Ví dụ:

```
7  <?php
8      $to = "yennh@cntp.edu.vn";
9      $subject = "Công tác CVHT";
10     $message =" Triển khai thông báo đến lớp chủ nhiệm -
11                 | về đăng ký học phần thay thế và tọa đàm sinh viên với nhà trường !";
12     $from= "yennh@ittc.edu.vn";
13     $headers = "FROM: ". $from;
14     mail($to,$subject,$message,$headers);
15     echo "Thu đã được gửi";
16  ?>
```

PHP EMAIL – SECURITY EMAIL

4. Thư viện PHP Mailer

- Tải PHP Mailer

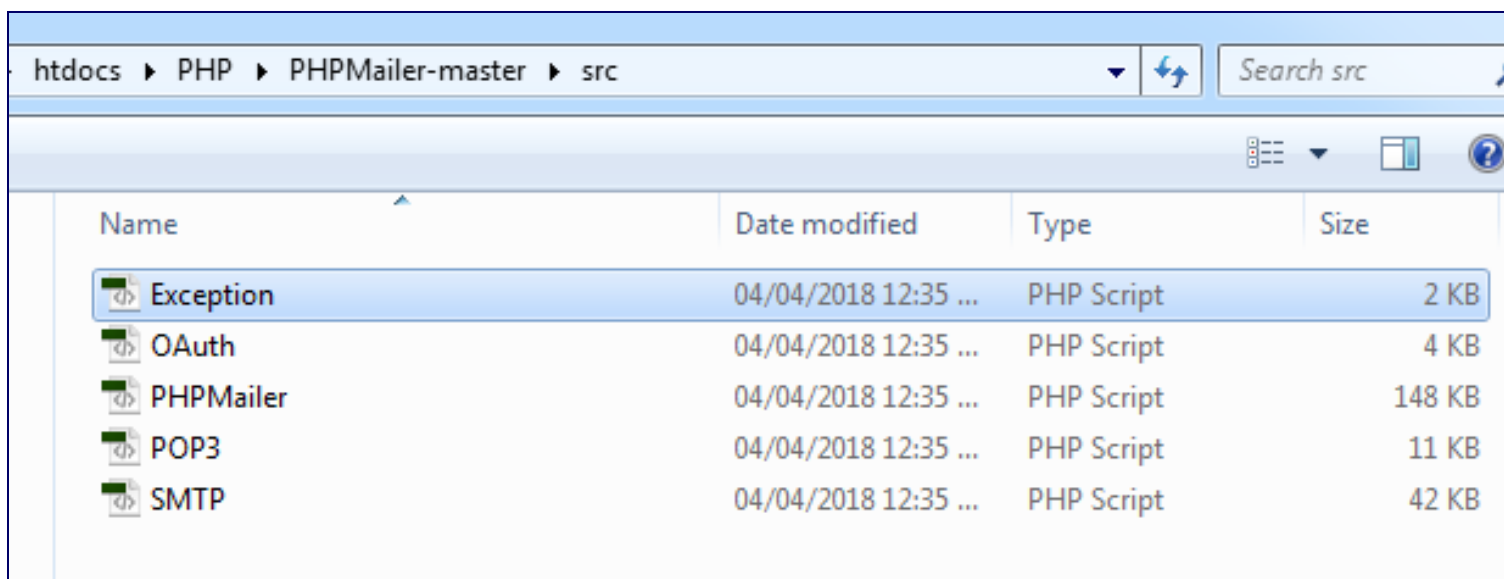


- Giải nén và copy folder vào htdoc

PHP EMAIL – SECURITY EMAIL

4. Thư viện PHP Mailer

- Tải PHP Mailer



Name	Date modified	Type	Size
Exception	04/04/2018 12:35 ...	PHP Script	2 KB
OAuth	04/04/2018 12:35 ...	PHP Script	4 KB
PHPMailer	04/04/2018 12:35 ...	PHP Script	148 KB
POP3	04/04/2018 12:35 ...	PHP Script	11 KB
SMTP	04/04/2018 12:35 ...	PHP Script	42 KB

- Giải nén và copy folder vào htdocs

PHP EMAIL – SECURITY EMAIL

4. Thư viện PHP Mailer

- Vào phần ví dụ trong trang

<https://github.com/PHPMailer/PHPMailer>

t/php/EMAIL/GoiMailSD_PHPMailer.php

GỎI MAIL SỬ DỤNG THƯ VIỆN PHP MAILER

Full Name	Trương Thị Khánh Uyên
Email	uyenttk@gmail.com
Subject	Chào hỏi
Content	Hi Mom , Hihí hí
SEND	

PHP EMAIL – SECURITY EMAIL

4. Thư viện PHP Mailer

```
8      require 'PHPMailer/src/Exception.php';
9      require 'PHPMailer/src/PHPMailer.php';
10     require 'PHPMailer/src/SMTP.php';
11
12
13     if (isset($_POST["btn_send"]))
14     {
15         $mail = new PHPMailer();
16         //Server settings
17         //$mail->SMTPDebug = 2;                // Enable verbose debug output
18         $mail->isSMTP();                       // Set mailer to use SMTP
19         $mail->Host = "smtp.gmail.com";        // Specify main and backup SMTP servers
20         $mail->SMTPAuth = true;               // Enable SMTP authentication
21
22         // Tk mail (có thực) này dùng để giả lập máy chủ SMTP của google
23         $mail->Username = "mnm06dhth@gmail.com"; // SMTP username
24         $mail->Password = "123456@123456";       // SMTP password
25
26         $mail->SMTPSecure = 'tls';              // Enable TLS encryption, `ssl` also accepted
27         $mail->Port = 465;                     // TCP port to connect to
28
29         $mail->Charset = "utf-8";
30     }
```

PHP EMAIL – SECURITY EMAIL

4. Thư viện PHP Mailer

```
31 //Recipients
32 $mail->setFrom($_POST["email"], $_POST["fullName"]); // Gởi dữ liệu từ form
33 $mail->addAddress("yennh@cntp.edu.vn", "Nguyễn Hải Yến"); // Add a recipient
34 /*$mail->addAddress('ellen@example.com'); // Name is optional
35 $mail->addReplyTo('info@example.com', 'Information');
36 $mail->addCC('cc@example.com');
37 $mail->addBCC('bcc@example.com');*/
38
39 //Attachments
40 /*$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments
41 $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name
42 */
43
44 //Content
45 $mail->isHTML(true); // Set email format to HTML
46 $mail->Subject = $_POST["subject"];
47 $mail->MsgHTML($_POST["content"]);
48 /*$mail->Body = 'This is the HTML message body <b>in bold!</b>';
49 $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';*/
50 $mail->send();
51 echo 'Message has been sent';
52 }
```

Q & A