

LẬP TRÌNH WEB

Đà Nẵng tháng .../20...

Giảng viên: Ths. Võ Ngọc Đạt

Email: vongocdatit@gmail.com





BÀI 2

TỔNG QUAN VỀ PHP

NỘI DUNG BÀI 2

- Giới thiệu về php
- Cú pháp php
- Biến và hằng
- Các kiểu dữ liệu trong php
- Các phép toán
- Các cấu trúc điều khiển

ĐẶC ĐIỂM CỦA FILE PHP

- ☐ Các file PHP trả về kết quả cho trình duyệt là một trang thuần HTML
- ☐ Các file PHP có thể chứa văn bản (Text), các thẻ HTML (HTML tags), Javascript, Css và các đoạn mã kịch bản máy chủ (Server Script)
- ☐ Các file PHP có phần mở rộng là: **.php**

NHỮNG ĐIỂM MỚI TRONG PHP 7

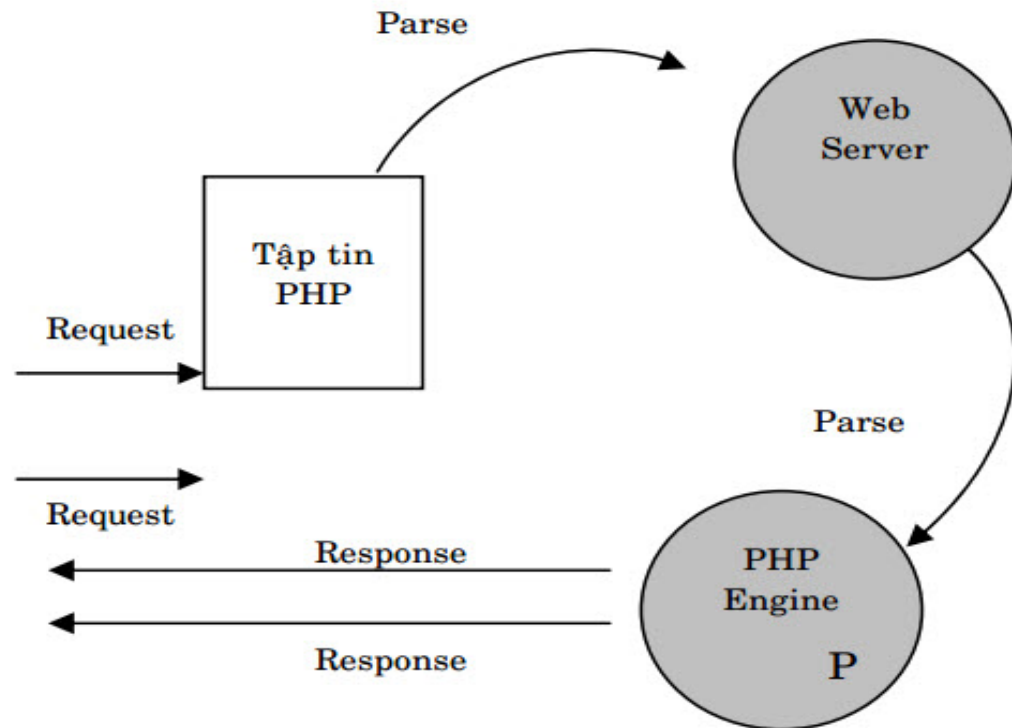
- **Sử dụng Zend Engine hoàn toàn mới:** PHP 7 có một phiên bản engine hoàn toàn mới có tên gọi là PHP#NG (Next Generation), bổ sung thêm một mô hình đối tượng mở rộng và nâng cao hiệu suất thực thi đáng kể cho ngôn ngữ này.
- **Tốc độ nhanh gấp hai lần:** bằng cách sử dụng PHP 7 không chỉ giúp code thực thi nhanh hơn mà cũng sẽ cần ít máy chủ hơn để phục vụ cùng một số lượng user.
- **Quản lý lỗi dễ dàng hơn:** PHP7 bổ sung các Engine Exceptions để bắt lỗi và ngoại lệ.

NHỮNG ĐIỂM MỚI TRONG PHP 7

- **Hỗ trợ các hệ thống Windows 64-Bit:** nhất quán hỗ trợ 64-bit nghĩa là cả các số nguyên 64-bit và các file lớn sẽ được hỗ trợ, cho phép tự tin chạy ngôn ngữ này trên hệ điều hành Windows 64-bit trong tương lai.
- **Bổ sung thêm các toán tử mới**
- **Cho phép khai báo kiểu chính xác:** PHP 7 giới thiệu các kiểu khai báo và trả về cho các kiểu dữ liệu.
- **Bổ sung thêm các class Anonymous (class vô danh)**
- **Thuận tiện khi Imports từ cùng Namespace**
- **Loại bỏ các chức năng cũ không còn được hỗ trợ**

QUÁ TRÌNH THÔNG DỊCH TRANG PHP

- ❑ Php là mã kịch bản trình chủ được chạy trên nền php Engine, cùng với ứng dụng Web Server để quản lý chúng.
- ❑ Khi trang php được gọi, Web Server triệu gọi php Engine để thông dịch, dịch trang php và trả về kết quả cho người sử dụng là một trang thuần HTML



Quá trình thông dịch trang php

CÚ PHÁP PHP

- ❑ Có thể nhúng các lệnh của php vào trang HTML
- ❑ Đoạn mã php luôn được bắt đầu và kết thúc bởi cặp thẻ theo cú pháp:

<?php

các lệnh của php;

?>

- ❑ Đoạn mã php có thể đặt bất kỳ đâu trong tài liệu
- ❑ Thông thường một trang php, bao gồm các thẻ HTML như một trang HTML nhưng có thêm các đoạn mã php.

CÚ PHÁP PHP

Lưu ý:

- ❑ Các file php phải có phần mở rộng là **.php**. Nếu phần mở rộng là .html thì đoạn mã php sẽ không được thực thi.
- ❑ Có thể viết các câu chú thích cho đoạn mã php. Có hai cách viết là:

/ chú thích nhiều dòng lệnh*/*

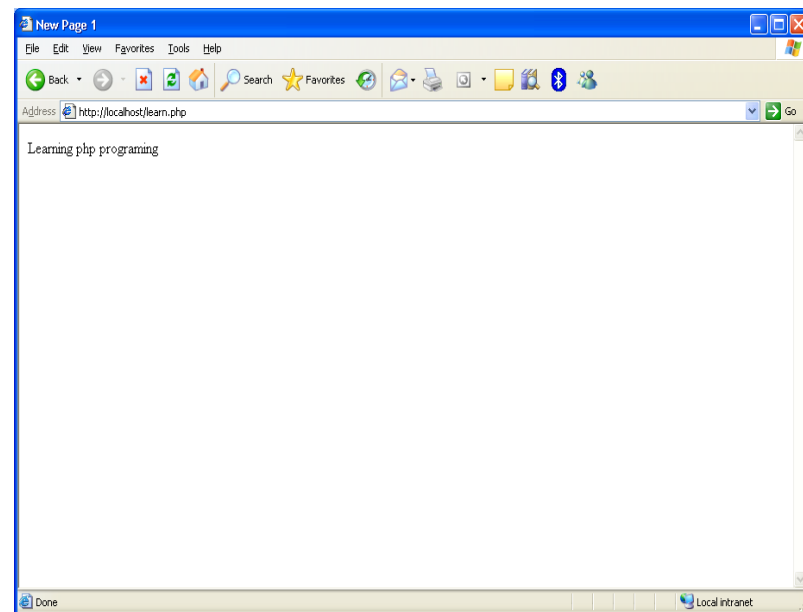
hoặc

// chú thích một dòng lệnh

CÚ PHÁP PHP

Ví dụ: Đoạn mã php hiển thị câu “Learning php programing” lên trình duyệt như sau:

```
<html>
<head>
    <title>Demo php</title>
</head>
<body>
<?php
    echo “Learning php programing”;
?>
</body>
</html>
```



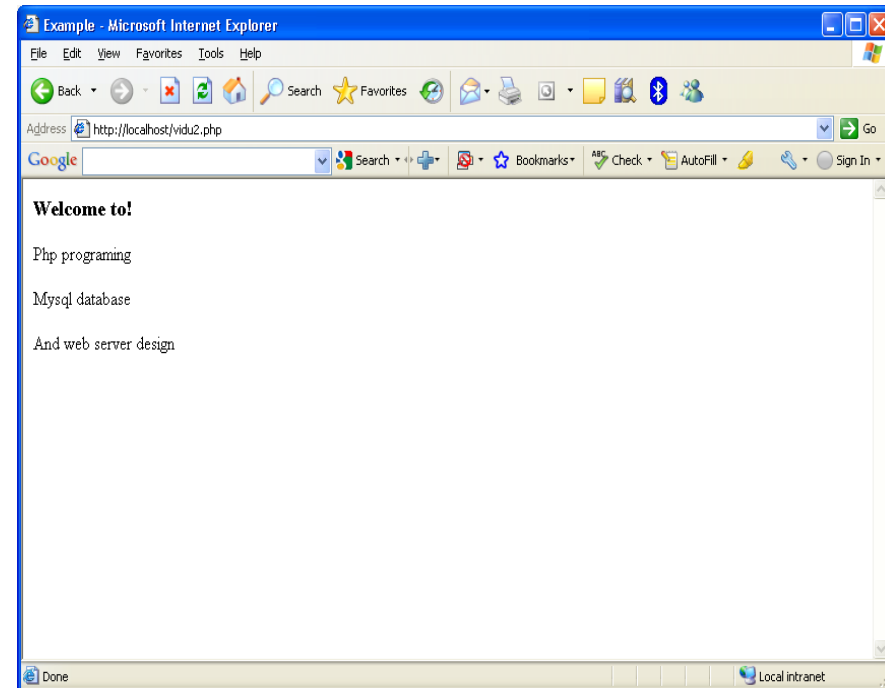
Kết quả của ví dụ trên

- ☐ Mỗi câu lệnh trong php được kết thúc bằng dấu (;)
- ☐ Có hai câu lệnh cơ bản, dùng để hiển thị các câu text ra browser là : *echo* và *print*

CÚ PHÁP PHP

Ví dụ: Trang vidu2.php như sau:

```
<html>
<head>
<title>Example</title>
</head>
<body>
  <h3> Welcome to!</h3>
  <?php
    echo "<p>Php programing</p>";
    print "<p>Mysql database</p>";
  ?>
  <p>And web server design</p>
</body>
</html>
```



Kết quả của ví dụ trên

BIẾN VÀ HẲNG TRONG PHP

- ☐ Biến
- ☐ Hằng

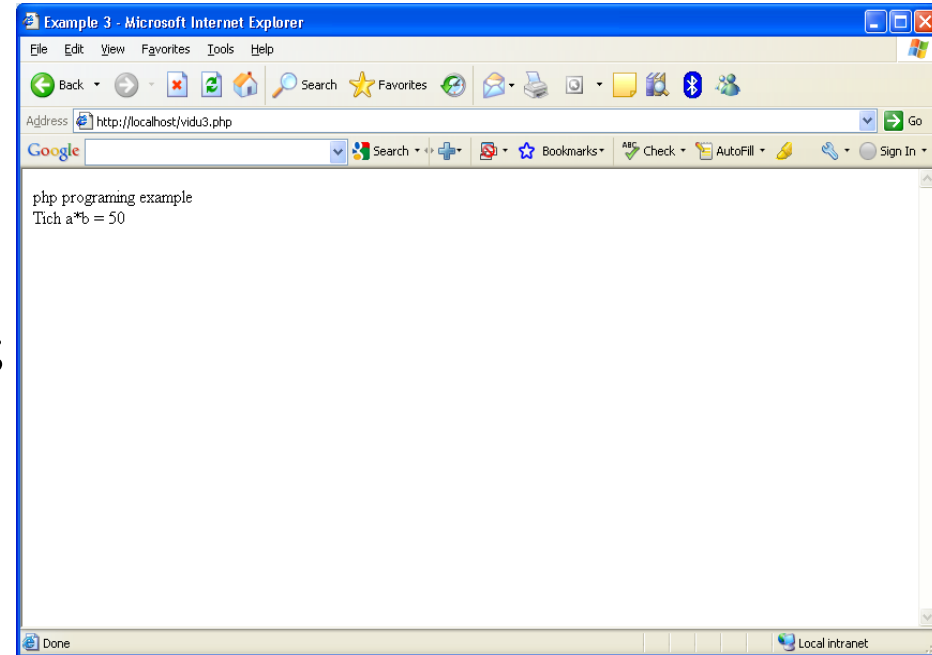
BIẾN

- ❑ Biến dùng để lưu giá trị như: xâu, số, ký tự, mảng,...
- ❑ Tất cả các biến trong php đều bắt đầu bằng ký hiệu: \$
- ❑ Biến được khai báo tự động khi sử dụng (gán giá trị), theo cú pháp: **\$Tên_biến = Giá_trị;**
- ❑ Php là ngôn ngữ không định kiểu, nghĩa là không cần khai báo kiểu cho biến. Php sẽ chuyển kiểu của biến một cách tự động tùy thuộc vào giá trị của nó.
- ❑ **Lưu ý:** Quy tắc đặt tên biến trong php giống như trong C, C++, ...

BIÊN

Ví dụ: Ta có trang **vidu3.php** như sau:

```
<html>
<head>
<title>Example 3</title>
</head>
<?php
    $a = "php programing example";
    echo "$a <br>";
    $a = 5; $b = 10;
    $c = $a*$b;
    echo "Tich a*b = $c";
?>
</body>
</html>
```



Kết quả của ví dụ trên

BIẾN

❑ Phạm vi của biến:

- ❑ Nếu biến được khai báo trong Script thì có phạm vi trong toàn Script (**global variable**)
- ❑ Nếu biến được khai báo trong một hàm nào đó thì chỉ có tác dụng trong hàm đó (**local variable**)
- ❑ Muốn sử dụng biến toàn cục trong một hàm thì phải khai báo từ khóa **global** hoặc **\$GLOBALS['tên']**
- ❑ Nếu muốn biến không mất giá trị khi thoát khỏi hàm và giữ nguyên giá trị khi hàm được gọi nhiều lần (**static variable**)

```
<?php
$x = 10;
function test_variable()
{
    $x = 0;
    echo "Gia tri cua bien x trong ham: " . $x . "<br>";
}
test_variable();
echo "Gia tri cua bien x ngoai ham: " . $x;
?>
```

```
<?php
$x = 10;
function test_variable()
{
    global $x;
    echo "Gia tri cua bien x trong ham: " . $x . "<br>";
}
test_variable();
echo "Gia tri cua bien x ngoai ham: " . $x;
?>
```

```
<?php
function test_variable()
{
    static $x = 0;
    $x++;
    echo $x . "<br>";
}
test_variable();
test_variable();
test_variable();
?>
```

BIẾN

- ❑ Một số hàm kiểm tra/ loại bỏ biến:
 - ❑ Sử dụng hàm **isset(Tên_biến)** để kiểm tra biến đó có tồn tại hay không? Kết quả trả về kiểu Boolean.
 - ❑ Sử dụng hàm **unset(Tên_biến)** để loại bỏ biến đang tồn tại ra khỏi trạng thái thực thi
 - ❑ Sử dụng hàm **empty(Tên_biến)** để kiểm tra biến có tồn tại hoặc rỗng không ? Kết quả trả về kiểu Boolean.

Ví dụ: Xét đoạn mã sau đây:

```
<?php
    $a = 10;
    echo "empty : " . empty($a) . "</br>";
    echo "isset : " . isset($a) . "</br>";
    unset($a);
    echo "isset : " . isset($a) . "</br>";
?>
```


BIẾN

- ☐ Ngoài ra, để kiểm tra kiểu dữ liệu của biến có thể sử dụng các hàm sau:
 - ☐ `is_array()`
 - ☐ `is_double()`
 - ☐ `is_float()`
 - ☐ `is_long()`
 - ☐ `is_int()`
 - ☐ `is_string()`
 - ☐ `is_object()`

HẲNG

- ❑ Hằng trong php được khai báo giống như các ngôn ngữ C, C++
- ❑ Tên hằng thường được viết bằng chữ hoa
- ❑ Cú pháp khai báo hằng như sau:

define (“Tên_hằng”, giá_trị);

Ví dụ: Ta có khai báo hằng $MAX = 100$ như sau:

define (“MAX”, 100);

CÁC KIỂU DỮ LIỆU

- ❑ Php hỗ trợ 6 kiểu dữ liệu như sau:
 - ❑ **Boolean**: kiểu true hay false
 - ❑ **Integer**: sử dụng cho giá trị có kiểu dữ liệu là số nguyên
 - ❑ **Float, Double**: sử dụng cho giá trị có kiểu dữ liệu là số thực
 - ❑ **String**: sử dụng cho các giá trị có kiểu dữ liệu là chuỗi và ký tự
 - ❑ **Array**: sử dụng cho các giá trị có kiểu dữ liệu là mảng
 - ❑ **Object**: sử dụng cho các giá trị có kiểu dữ liệu là đối tượng của lớp

CÁC PHÉP TOÁN

- ☐ Phép gán
- ☐ Các phép toán số học
- ☐ Các phép toán quan hệ
- ☐ Các phép toán logic
- ☐ Phép toán trên chuỗi
- ☐ Các phép toán tự tăng giảm
- ☐ Biểu thức điều kiện

PHÉP GÁN

- ❑ Phép gán là phép toán cơ bản của mọi ngôn ngữ lập trình.
- ❑ Phép gán đơn: cú pháp: \$Tên_biến = Giá_trị;
- ❑ Phép gán mở rộng: \$Tên_biến **pt** = Giá_trị;
- ❑ Trong đó: **pt** có thể là: +, -, *, /, %

CÁC PHÉP TOÁN SỐ HỌC

- ❑ Phép toán số học một ngôi: -, ++, --
- ❑ Phép toán số học hai ngôi:

Toán tử	Tên	Ví dụ
+	Cộng	$\$a + \b
-	Trừ	$\$a - \b
*	Nhân	$\$a * \b
/	Chia	$\$a / \b
%	Chia lấy dư	$\$a \% \b

CÁC PHÉP TOÁN TỰ TĂNG GIẢM

- ❑ Để tăng (hoặc giảm) giá trị của một biến lên (xuống) một đơn vị có thể sử dụng phép toán tự tăng ++ và tự giảm --
- ❑ Có hai cách viết phép toán tự tăng giảm:
 - ❑ ++\$Tên_biến (hoặc --\$Tên_biến)
 - ❑ \$Tên_biến++ (hoặc \$Tên_biến--)
- ❑ Lưu ý: cần phân biệt hai cách viết trên.

CÁC PHÉP TOÁN QUAN HỆ

- ❑ Các phép toán quan hệ trả về kết quả là **true** hoặc **false**

Toán tử	Tên	Ví dụ
<	Nhỏ hơn	$3 < 5$
<=	Nhỏ hơn hoặc bằng	$A \leq b$
>	Lớn hơn	$\$a > \b
>=	Lớn hơn hoặc bằng	$\$a \geq \b
= =	Bằng	$123 = \text{"123"}$
= = =	Bằng và cùng kiểu dữ liệu	$123 == \text{"123"}$
!=	Khác	$123 != \text{"123"}$
!= =	Khác kiểu dữ liệu	$123 !== \text{"123"}$
<>	Khác	$123 <> \text{"123"}$

CÁC PHÉP TOÁN LOGIC

- Giống như các phép toán quan hệ các phép toán logic trả về kết quả là true hoặc false

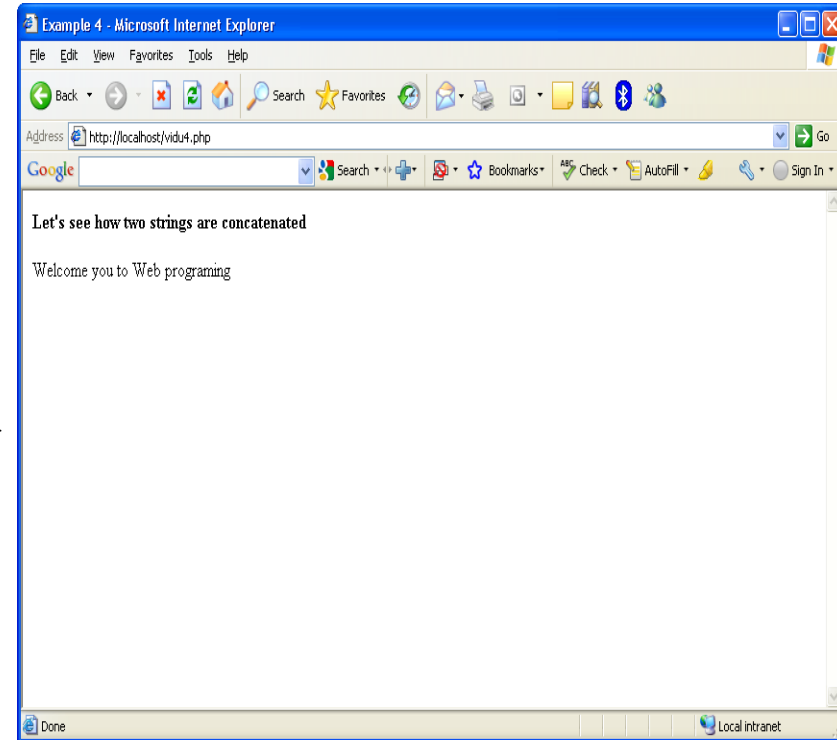
Toán tử	Tên	Ví dụ
&&	And (và)	$a \&\& b$
	Or (hoặc)	$a b$
!	Not (phủ định)	$!b$

PHÉP TOÁN VỀ CHUỖI

- ❑ **Phép cộng chuỗi:** Để cộng (ghép) hai chuỗi lại với nhau ta sử dụng dấu chấm (.)

Ví dụ: xét đoạn mã sau:

```
<html>
<head>
<title>Example 4</title>
</head>
<body>
<h4> Let's see how two strings are concatenated </h4>
<?php
    $st1 = "Welcome you to ";
    $st2 = "Web programing";
    echo $st1.$st2; ?>
</body>
</html>
```



Kết quả của ví dụ trên

TOÁN TỬ ĐIỀU KIỆN

□ Cú pháp:

Biến = Giá trị 1 > Giá trị 2 ? Giá trị 1 : Giá trị 2;

Ví dụ: In số lớn hơn

```
<?php
```

```
    $a = 10;
```

```
    $b = 20;
```

```
    echo $a > $b ? $a : $b;
```

```
?>
```

CÁC CẤU TRÚC ĐIỀU KHIỂN

- ☐ Cấu trúc rẽ nhánh
- ☐ Cấu trúc lặp

CẤU TRÚC RỄ NHÁNH

❑ Cấu trúc if

Cú pháp: *if (điều kiện) câu lệnh php;*

Ví dụ: In ra giá trị lớn nhất nếu $a > b$

```
<?php
```

```
    $a = 7; $b = 3;
```

```
    if ($a > $b) echo "Giá trị lớn nhất là: ".$a;
```

```
?>
```

CẤU TRÚC RỄ NHÁNH

□ Cấu trúc if ... else

Cú pháp: *if (điều kiện) công việc 1;*
else công việc 2;

Ví dụ: In ra giá trị lớn nhất

```
<?php
```

```
$a = 7; $b = 3;
```

```
if ($a > $b)
```

```
    echo "Giá trị lớn nhất là: ".$a;
```

```
else
```

```
    echo "Giá trị lớn nhất là: ".$b;
```

```
?>
```

CẤU TRÚC RÊ NHÁNH

□ Cấu trúc if...else...if...else

Cú pháp: *if (điều kiện 1) công việc 1;*

else if (điều kiện 2) công việc 2;

...

else if (điều kiện n) công việc n;

else công việc n+1;

Ví dụ: In quý theo tháng

<?php

```
$thang=3;
if (($thang==1)||($thang==2)||($thang==3))
    echo "Quý 1";
else if (($thang==4)||($thang==5)||($thang==6))
    echo "Quý 2";
else if (($thang==7)||($thang==8)||($thang==9))
    echo "Quý 3";
else
    if (($thang==10)||($thang==11)||($thang==12))
        echo "Quý 4";
    else echo "Tháng không hợp lệ";
```

?>

CẤU TRÚC RỄ NHÁNH

□ Cấu trúc if lồng nhau

Cú pháp: *if(Điều kiện 1)* {

//Thực thi khi Điều kiện 1 đúng

if(Điều kiện 2){

//Thực thi khi điều kiện 1,2 đúng

}

}

Lưu ý:

- *Nếu cần thực thi nhiều câu lệnh thì cần đặt nó trong cặp dấu ngoặc móc { }*

CẤU TRÚC RỄ NHÁNH

❑ **Cấu trúc switch:** sử dụng khi có nhiều sự lựa chọn

Cú pháp:

switch (t)

```
{  
    case label 1: Lệnh thực thi nếu t=label 1; break;  
    case label 2: Lệnh thực thi nếu t=label 2; break;  
    ...  
    case label n: Lệnh thực thi nếu t=label n; break;  
    default: Lệnh thực thi nếu t khác khác từ 1 đến n;  
}
```

CẤU TRÚC RỄ NHÁNH

Ví dụ: Chọn chức năng

```
<?php
$kt= "insert";
switch($kt)
{
    case "edit":echo"sửa dữ liệu<br/>"; break;
    case "insert":echo"chèn dữ liệu<br/>"; break;
    case "delete":echo"xóa dữ liệu<br/>"; break;
    case "save":echo"lưu dữ liệu<br/>"; break;
    default:echo“Không có chức năng này<br/>”;
}
?>
```

CẤU TRÚC RÊ NHÁNH

Ví dụ: In số chẵn/số lẻ

```
<?php
```

```
$a=4;
```

```
switch($a)
```

```
{
```

```
    case $a%2==0: echo $a." là số chẵn"; break;
```

```
    case $a%2!=0: echo $a." là số lẻ"; break;
```

```
}
```

```
?>
```

CẤU TRÚC LẶP

❑ Cấu trúc for:

Cú pháp: **for** (*init*; *condition*; *increment*)
 {
 code to be executed;
 }

- *init* : Giá trị bắt đầu của biến đếm, dùng để khởi động của vòng lặp
- *condition* : Biểu thức điều kiện, là điều kiện để thực hiện vòng lặp
- *increment* : Cập nhật biến đếm, thay đổi biến đếm sau mỗi vòng lặp
- *code to be executed*: Mã lệnh thực thi

Ví dụ: In dãy số từ 0 đến 10

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "Số: $x" . "<br/>";  
}  
?>
```

CẤU TRÚC LẶP

❑ Cấu trúc while:

Cú pháp: **while** (*condition*)

```
{  
    code to be executed;  
}
```

- *condition*: Điều kiện để thực hiện vòng lặp
- *code to be executed*: Lệnh thực hiện

Ví dụ: Tính tổng các số từ 1 đến 10

```
<?php  
$i = 1;  
$total = 0;  
while($i <=10){  
    $total += $i;  
    $i++;  
}  
echo $total;  
?>
```

CẤU TRÚC LẶP

❑ Cấu trúc do ... while:

Cú pháp: **do**

{

code to be executed;

}

while (*condition*);

- *condition*: Điều kiện để thực hiện vòng lặp
- *code to be executed*: Lệnh thực hiện

Ví dụ: In dãy số từ 1 đến 10

```
<?php
```

```
$i = 1;
```

```
do{
```

```
    echo $i . " ";
```

```
    $i++;
```

```
}while ($i <= 10);
```

```
?>
```

CẤU TRÚC LẶP

Ví dụ: Cấu trúc lặp lồng nhau

```
<?php
$a=0;
while($a<10)
{
    do
    {
        echo $a;
        $a++;
    }while($a<5);
    echo "<br/>";
    $a++;
}
?>
```

CẤU TRÚC LẶP

❑ **Cấu trúc foreach:** sử dụng khi lặp các phần tử của mảng.

Cú pháp: **foreach**(\$array as \$value)

{

code to be executed;

}

- *array: mảng*
- *value: phần tử thuộc mảng*
- *code to be executed: Lệnh thực thi*

CẤU TRÚC LẶP

Ví dụ: In danh sách thành phố

```
<?php
```

```
$tpho1 = array("HoChiMinh", "HaNoi", "HaiPhong", "DaNang");
```

```
$tpho2 = array("HCM" => "HoChiMinh", "HN" => "HaNoi", "HP" =>  
    "HaiPhong", "DN" => "DaNang");
```

```
foreach($tpho1 as $tp)
```

```
{
```

```
    echo $tp . "<br/>";
```

```
}
```

```
foreach($tpho2 as $chiso=>$giatri)
```

```
    echo $chiso . " => " . $giatri . "<br/>";
```

```
?>
```

BÀI THỰC HÀNH

1. Giải phương trình bậc 2 với các hệ số được gán sẵn

Giải phương trình bậc 2

Kết quả giải phương trình bậc 2 với $a = 1$, $b = 2$, $c = 1$

Phương trình nghiệm kép $x_1 = x_2 = -1$

BÀI THỰC HÀNH

2. Tổng kết, xếp loại:

- Gán điểm các môn Toán, Văn, Anh
- Tính và hiển thị điểm trung bình (Anh hệ số 2)
- Hiển thị xếp loại:
 - Điểm trung bình 8.5-10: Giỏi
 - Điểm trung bình 7.0-dưới 8.5: Khá
 - Điểm trung bình 5.0-dưới 7.0: Trung bình
 - Điểm trung bình dưới 5.0: Yếu

Tổng kết xếp loại

Tổng kết: 9.13
Xếp loại: Giỏi

BÀI THỰC HÀNH

3. Tính tiền:

- Gán mặt hàng, đơn giá, số lượng
- Tính tổng tiền = đơn giá * số lượng
- Tính thực trả: nếu tổng tiền ≥ 300000 thì giảm 10%

Tình tiền

Khách hàng mua mặt hàng vờ với đơn giá 7000 với số lượng 5

Tổng tiền: 35000

Thực trả: 35000

BÀI THỰC HÀNH

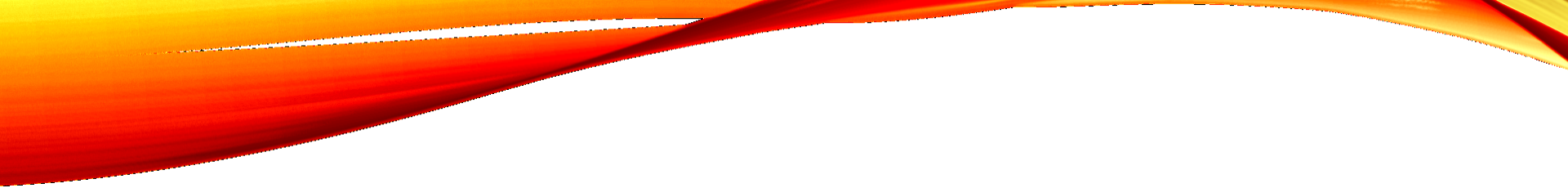
4. Tính giai thừa: $S = 1! + 2! + 3! + \dots + n!$ (n gán sẵn), sử dụng vòng lặp For, while, do while

Tính giai thừa

Dùng vòng lặp For: Giai thừa của 5 là 120

Dùng vòng lặp While: Giai thừa của 5 là 120

Dùng vòng lặp Do While: Giai thừa của 5 là 120



Q & A



THANK YOU!