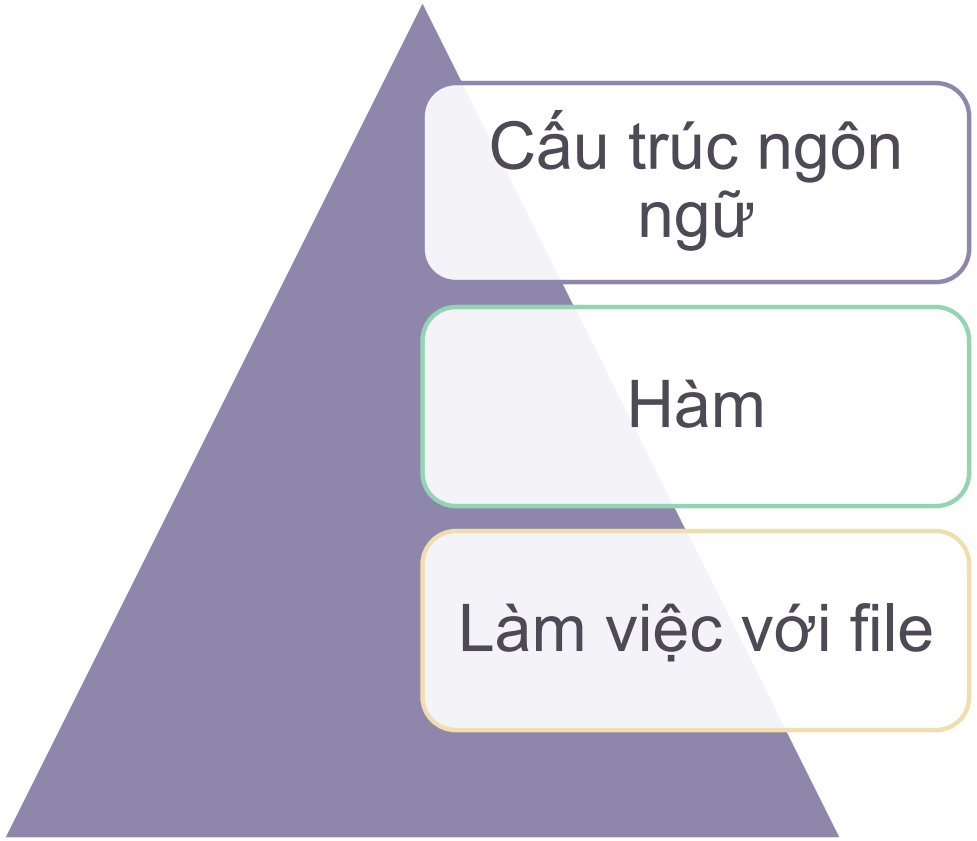
CÔNG NGHỆ WEB

Chương 2 : PHP cơ bản

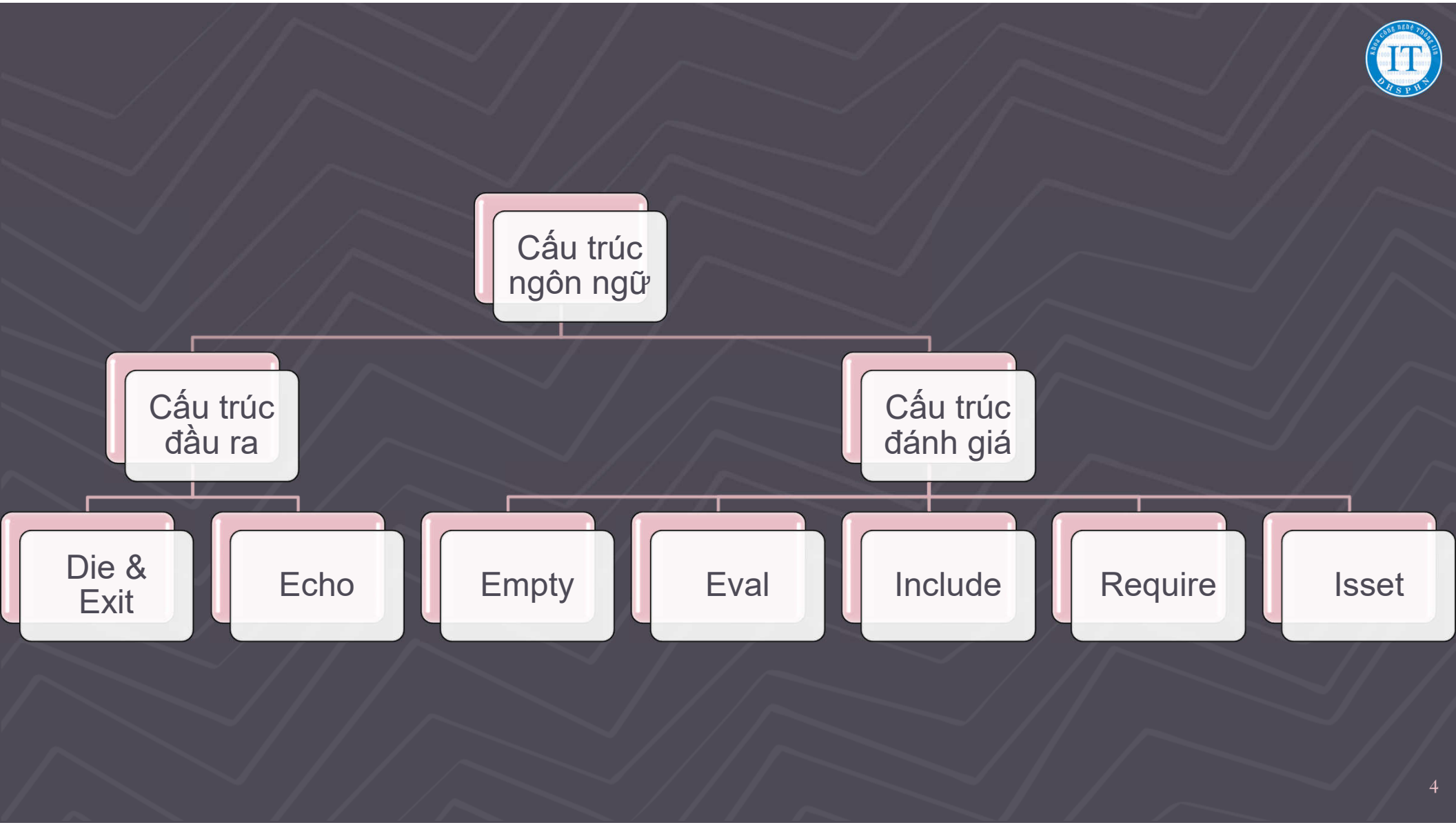
hoantq@hnue.edu.vn



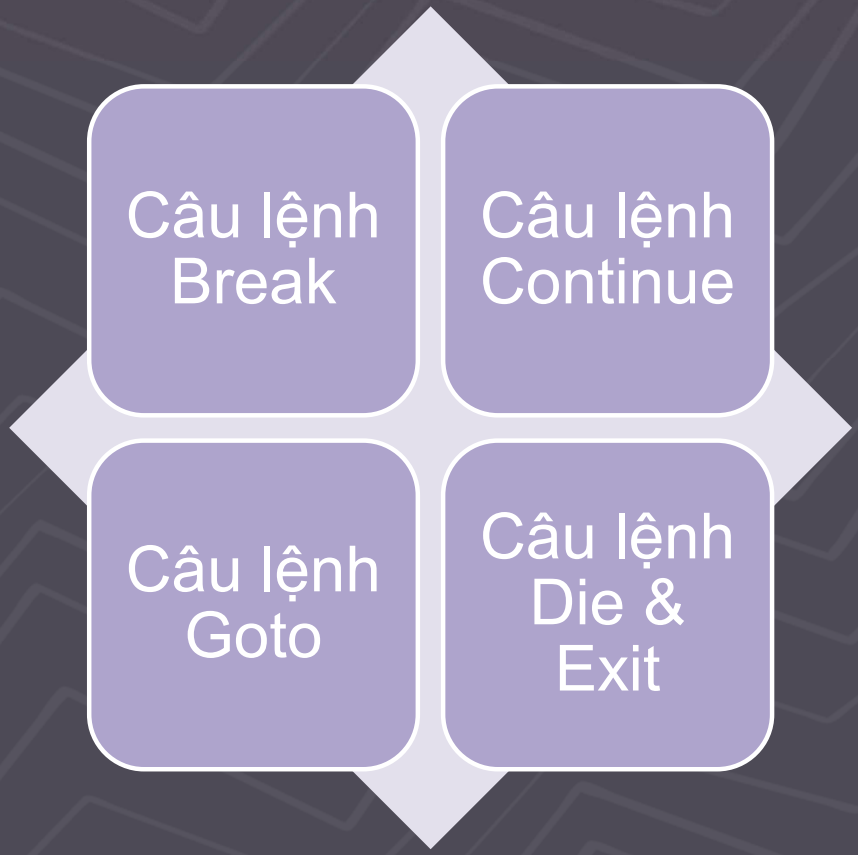
hoantq@hnue.edu.vn

“

Cấu trúc ngôn ngữ


Cấu trúc ngôn ngữ
Cấu trúc đầu ra
Cấu trúc đánh giá
Die & Exit
Echo
Empty
Eval
Include
Require
Isset

LỆNH BREAK, CONTINUE, GOTO, DIE, EXIT



BREAK

- Lệnh BREAK thường được dùng để thoát khỏi vòng lặp cho dù vòng lặp chưa kết thúc

```php
for ($i = 1; $i <= 100; $i++)
{
    echo $i . ' ';
    if ($i == 20)
    {
        break;
    }
}
```

CONTINUE

- Continue sẽ bỏ qua những đoạn code bên dưới nó và nhảy qua vòng lặp kế tiếp (ko thoát khỏi vòng lặp như lệnh break)

```php
for ($i = 1; $i <= 10; $i++)
{
    if ($i == 5)
    {
        continue;
    }
    echo $i . ' ';
}
```

BÀI TẬP

- **Ví dụ 1**: Lấy ra 5 số đầu tiên chia hết cho 7 trong dãy số từ 1 đến 100
- Ví dụ 2 : In một dãy số chẵn từ 1 đến 20 nhưng ko in ra số 4 và số 14.

GOTO

- Lệnh goto để nhảy đến một code nào đó

```php
$a = 12;
$b = 13;
$c = $a + $b;

echo $a;

goto label_end;

echo $b;

label_end;
```

DIE và EXIT

- Die và exit sẽ làm chương trình dừng ngay lập tức

```php
1  echo '123';
2
3  die(); // hoặc exit();
4  echo '456';
```

Hàm trong PHP

HÀM

- Hàm là tập hợp một hay nhiều câu lệnh được xây dựng để thực hiện một chức năng nào đó

```php
function func_name($vars)
{
    // các đoạn code
    return $val;
}
```

- Func_name: tên của hàm
- $vars là các biến truyền vào trong hàm
- Return$val là hàm sẽ trả về giá trị $val

Hàm

```
<?php
function xinchao(){
echo "Xin chào các bạn!.";
echo "Tôi là Lan."."<br>";
}
xinchao();
xinchao();

?>
```

Xin chào các bạn!.Tôi là Lan.
Xin chào các bạn!.Tôi là Lan.

Hàm

- Bài tập : Viết đoạn mã in ra 100 dòng "Đây là số n" với n từ 1 đến 100

```
function vd(){
    for ($i=0; $i <100 ; $i++) {
        echo "Đây là số".$i."<br/>";
    }
}
vd();
```

Hàm với tham số & Hàm với giá trị trả về

- Xét ví dụ sau :

```php
<?php
function add($a,$b){
    $c=$a+$b;
    return $c;
}
$d= add(5, 8);
echo $d.'<br/>';
function concatstr ($str1, $str2) {
    $str=$str1.$str2;
    return $str;
}
$s=concatstr("I'm a lecture.","My name is Hoa");
echo $s;
?>
```

Phạm vi của biến

Biến toàn cục	Biến cục bộ
Một biến được khai báo bên ngoài hàm	Một biến được khai báo bên trong hàm

Lưu ý: Biến cục bộ chỉ được truy cập bên trong hàm.

Phạm vi của biến

```php
<?php
$x=10;
function baitoan(){
    echo 'x = '.$x;
}
baitoan();
?>
```

Ví dụ

Phạm vi biến

Notice: Undefined variable: x in **C:\xampp\htdocs\Ví dụ\JM\phamvibien.php** on line **15**
x =

Phạm vi của biến

- Toán tử **global** đặt trước biến được khai báo trong hàm xem như là biến toàn cục

```php
<?php
$x=10;
function baitoan(){
    global $x;
    echo 'x = '.$x;
}
baitoan();
?>
```

Ví dụ

Phạm vi biến
x = 10

Phạm vi của biến

- Sử dụng biến môi trường GLOBALS, các chỉ số index là tên biến, tức là biến y. Và giá trị của biến toàn cục gán vào trong giá trị của biến cục bộ.

```php
<?php
$x=10;
$y=11;
$z=12;
function baitoan(){
    $y_cucbo=$GLOBALS['y'];
    $z_cucbo=$GLOBALS['z'];
    echo 'y='.$y_cucbo.',z='.
}
baitoan();
?>
```

Ví dụ

Phạm vi biến

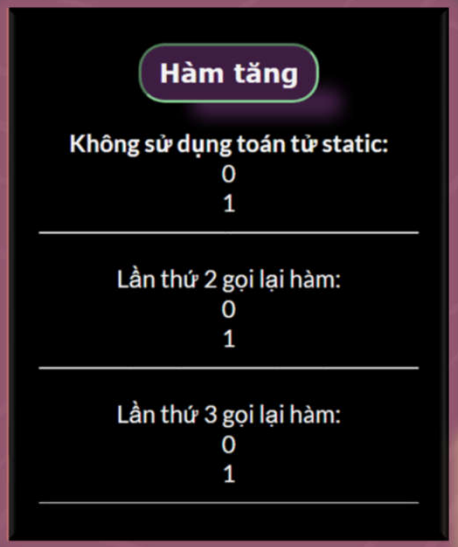
y=11,z=12

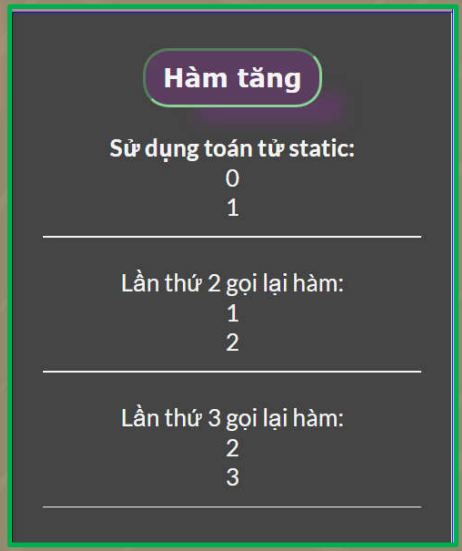
Phạm vi của biến

- Thông thường các biến sẽ xóa giá trị sau mỗi lần hàm được thực thi
- Sử dụng từ khóa **static** đặt trước biến được khai báo trong hàm
- Sau mỗi lần gọi hàm biến mang thông tin về giá trị của lần cuối cùng hàm được gọi

Phạm vi của biến

```
<div class="content__inner">
    <h3 class="link change">  Hàm tăng  </h3>
    <div><?php
        echo "<b>Không sử dụng toán tử static: </b><br>";
        increase1();
        echo "<hr class='hr'>";
        echo "Lần thứ 2 gọi lại hàm: <br>";
        increase1();
        echo "<hr class='hr'>";
        echo "Lần thứ 3 gọi lại hàm: <br>";
        increase1();
        echo "<hr class='hr'>";
    ?></div>
</div>
<div class="content__inner">
    <h3 class="link change">  Hàm tăng  </h3>
    <div ><?php
        echo "<b>Sử dụng toán tử static: </b><br>";
        increase2();
        echo "<hr class='hr'>";
        echo "Lần thứ 2 gọi lại hàm: <br>";
        increase2();
        echo "<hr class='hr'>";
        echo "Lần thứ 3 gọi lại hàm: <br>";
        increase2();
        echo "<hr class='hr'>";
    ?></div>
</div>
```

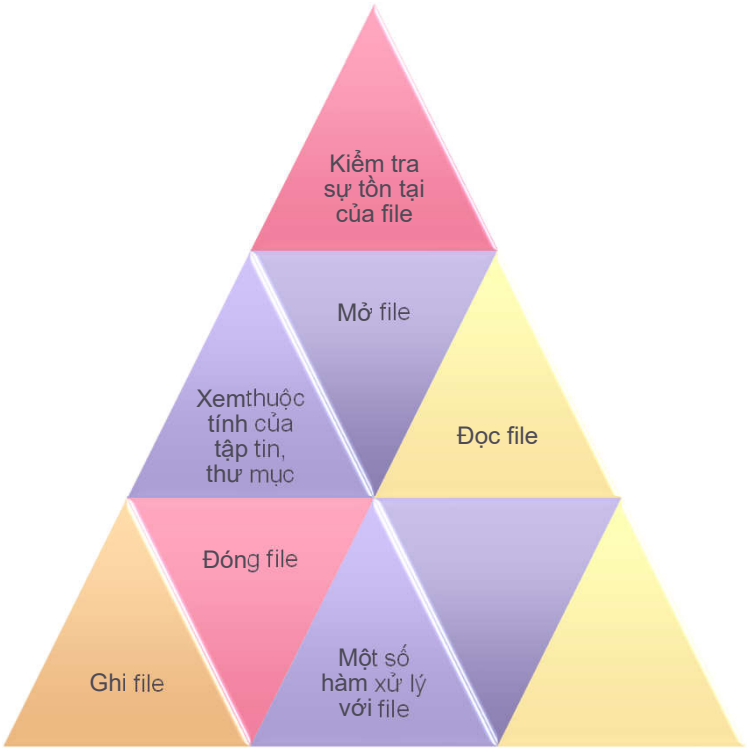




“

Làm việc với file

Làm việc với file



Xem một số thuộc tính cơ bản của tập tin, thư mục

Hàm	Chức năng
file_exist($filename)	Kiểm tra sự tồn tại (True/False)
Filetype($filename)	Trả về kiểu
Filesize($filename)	Trả về dung lượng
ls_readable($filename)	Kiểm tra quyền đọc
ls_writeable($filename)	Kiểm tra quyền ghi
ls_executable($filename)	Kiểm tra quyền thực thi

$filename : đường dẫn của thư mục/tập tin cần kiểm tra

Mở đóng file

```
open($path, $option)
```

- Trong đó $path là đường dẫn file cần mở
- $option là quyền cho phép thao tác trên file
- Đóng file sử dụng hàm fclose($fp)

Mở file

Mode	Diễn giải
r	Read only
r+	Read + Write
w	Write only
w+	Write + Read. Nếu file này tồn tại thì nội dung cũ sẽ bị xóa đi và ghi lại nội dung mới, còn nếu file chưa tồn tại thì nó tạo file mới
a	Mở dưới dạng append dữ liệu, chỉ có write và nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới
a+	Mở dưới dạng append dữ liệu, bao gồm write và read. Nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới
b	Mở dưới dạng chế độ binary

Đọc file

- Đọc từng dòng `fgetc($fp)`
- Đọc từng ký tự `fgets($fp)`
- Đọc hết file `fread($fp, $size)`
- Trong đó, $fp là đối tượng lúc mở file
- $size là kích cỡ của file cần đọc
- Hàm filesize($path) để lấy kích cỡ của file càn đọc

Ghi file

```
fwrite($fp, $content)
```

- $fp là đối tượng trả về lúc mở file
- $content là nội dung muốn ghi vào

Các hàm xử lý file khác

Hàm	Chức năng
File_exists($path)	Kiểm tra file có tồn tại không
Is_wriable($path)	Kiểm tra file có được cấp quyền ghi không
File_get_contents($path)	Lấy nội dung một file mà ko cần dùng hàm fread
File_put_content($path,$noidung)	Ghi nội dung file mà ko cần dùng hàm fwrite
Rename($oldname,$newname)	Đổi tên file
Copy($source,$dest)	Copy file
Unlink($path)	Xóa file
Is_dir($filenam)	Kiểm tra một đườngf dẫn folder có tồn tại ko
Mkdir($path)	Tạo một folder mới