

# CHƯƠNG 4

# HÀM

# NỘI DUNG

- 4.1. GIỚI THIỆU VỀ HÀM**
- 4.2. CÁC HÀM XÂY DỰNG SẴN**
- 4.3. HÀM TỰ TẠO**
- 4.4. CÁCH GỌI HÀM**
- 4.5. CÁCH SỬ DỤNG LẠI MÃ PHP**

# GIỚI THIỆU VỀ HÀM

- ◆ Khái niệm hàm
- ◆ Phân loại hàm

# Khái niệm hàm

- ❑ Hàm dùng để thực hiện một công việc nào đó. Chẳng hạn như: lấy ngày tháng của hệ thống, kết nối tới CSDL, thông báo lỗi,...
- ❑ Tất cả các hàm trong php đều có dạng cơ bản sau:

**Tên\_hàm(*Đối số*)**

Trong đó:

- **Đối số:** Tùy thuộc vào các hàm khác nhau mà có thể có một hoặc nhiều đối số hoặc không có đối số.
  - Cần nắm được kết quả trả về của hàm.
- ❑ Hàm chỉ được thực thi khi có lời gọi hàm.
  - ❑ Có thể gọi hàm ở bất kỳ đâu trong trang web.

# Phân loại hàm

## ☐ Hàm trong php được chia thành hai loại

- Các hàm xây dựng sẵn: Là các hàm dùng để thực hiện các chức năng cơ bản của php.
- Các hàm do người dùng tự định nghĩa: được xây dựng trong quá trình viết code.

# CÁC HÀM XÂY DỰNG SẴN

- ◆ Các hàm ngày giờ
- ◆ Các hàm thao tác với file

# Các hàm ngày giờ

- Dùng để lấy hoặc thiết lập ngày tháng trên Server.
- Một số hàm ngày giờ cơ bản:
  - Hàm date()
  - Hàm time()
  - Hàm getdate()
  - Hàm checkdate()
  - Hàm mktime()

# Hàm date()

- Trả về ngày, giờ dựa trên ngày giờ của Server theo định dạng chỉ ra.
- Cú pháp: **date (\$format, [\$timestamp = 'time()'])**
- Trong đó:

**\$format:** là định dạng mà hàm này sẽ trả về

**\$timestamp:** là thời gian truyền vào (kiểu INT), mặc định sẽ lấy thời gian hiện tại (chính là hàm time()).

# Hàm date()

## ☐ Một số ký tự định dạng:

- **a**: am hoặc pm
- **A**: AM hoặc PM
- **B**: mã giờ Internet
- **d**: ngày của tháng (01 ... 31)
- **D**: ngày của tuần (Mon ... Sun)
- **F**: Tháng có tên đầy đủ (January ... December)
- **g**: giờ (1 ... 12)
- **G**: giờ (0 ... 23)
- **h**: giờ (01 ... 12)
- **H**: giờ (00 ... 23)
- **i**: phút (00 ... 59)

# Hàm date()

- ◆ **j**: ngày của tháng (1 ... 31)
- ◆ **l**: ngày của tuần (Monday ... Sunday)
- ◆ **L**: kiểm tra năm nhuận (1: nếu năm nhuận, 0: ngược lại)
- ◆ **m**: tháng (01 ... 12)
- ◆ **M**: tháng (Jan ... Dec)
- ◆ **n**: tháng (1 ... 12)
- ◆ **s**: giây (00 ... 59)
- ◆ **S**: hậu tố thứ tự của tiếng anh (st, nd, rd, th)
- ◆ **t**: số ngày của tháng (28 ... 30)
- ◆ **T**: thiết lập timezone của máy
- ◆ **w**: ngày của tuần dạng số (0: Sunday, ..., 6: Saturday)
- ◆ **Y**: năm 4 số (2009)
- ◆ **y**: năm 2 số (09)
- ◆ **z**: ngày của năm (0 ... 365)

# Hàm date()

Ví dụ:

Lấy thời gian hiện tại theo định dạng ngày/tháng/năm - giờ:phút:giây

Vậy mã code PHP sẽ là:

```
echo date('d/m/Y - H:i:s');
```

# Hàm date()

Ví dụ:

```
<?php  
$nextWeek = time() + (7*24*60*60);  
echo "Now : " . date("Y-m-d") . "<br>";  
echo "Next Week: " . date("Y-m-d", $nextWeek);  
?>
```

# Hàm time()

- ◆ Trả về số giây.

Cú pháp: **time()**

Ví dụ: trả về thứ ngày hiện tại.

```
<?php  
    $t=time();  
    echo(date("D F d Y",$t));  
?>
```

# Hàm getdate()

- ◆ Trả về một mảng gồm các phần tử như sau:

[seconds] – Giây

[minutes] - phút

[hours] - giờ

[mday] – ngày của tháng

[wday] – ngày của tuần

[year] - năm

[yday] – ngày của năm

[weekday] – tên của thứ trong tuần

[month] – tên của tháng

Cú pháp: **getdate()**

# Hàm getdate()

Ví dụ:

```
<?php  
    $t = getdate(time());  
    echo $t['mday'] . '/' . $t['mon'] .  
        '/' . $t['year'];  
?>
```

# Hàm checkdate()

- ◆ Dùng để kiểm tra ngày, tháng, năm nào đó có hợp lệ không. Nếu có trả về true, ngược lại trả về false

Cú pháp: **checkdate(month, day, year)**

Ví dụ:

```
<?php  
    echo (checkdate(12, 31, 2019)) ;  
    echo (checkdate(2, 29, 2019)) ;  
    echo (checkdate(2, 29, 2019)) ;  
?>
```

# Hàm mktime()

- ◆ Dùng để lấy timestamp của thời gian được truyền vào.  
Cú pháp: **mktime(hour, minute, second, month, day, year)**

Ví dụ:

```
<?php  
$tomorrow=date ("d/m/Y",mktime (0,0,0,date ("m"),date ("d")  
+1,date ("Y")));  
echo "<br/>Ngày mai: " . $tomorrow;  
?>
```

# Các hàm thao tác với file

- ◆ Dùng để mở, đọc, ghi, dữ liệu vào file
- ◆ Một số hàm cơ bản:
  - ◆ Hàm **fopen(file\_name, format)**: dùng để mở file có tên `file_name` theo định dạng format (`r`, `r+`, `w`, `w+`, `a` ...)
  - ◆ Hàm **fclose(biến\_file)**: dùng để đóng file đang mở
  - ◆ Hàm **feof(biến\_file)**: dùng để kiểm tra xem đã kết thúc file hay chưa
  - ◆ Hàm **fgets(biến\_file)**: dùng để đọc theo từng dòng
  - ◆ Hàm **fgetc(biến\_file)**: đọc theo từng ký tự

# Các hàm thao tác với file

Mode	Điễn 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

# Các hàm thao tác với file

**Ví dụ:** Mở file với định dạng read only, in từng dòng và từng ký tự cho đến hết file

```
$file = fopen("thu.txt", "r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgets($file) . "<br />";
    echo fgetc($file);
}
fclose($file);
```

# Các hàm thao tác với file

## Ví dụ

```
$file=fopen("thu.txt","a") or exit ("co loi");
echo fwrite($file,"chào các bạn!");
$file=fopen("thu.txt","r") or exit("unable to open the file");
while (!feof($file))
{
    echo fgets($file)."<br/>";
    //echo fgetc($file)."<br/>";
}
fclose($file);
```

# CÁC HÀM TỰ TẠO

- ◆ **Cách xây dựng hàm trong php**
- ◆ **Một số lưu ý khi xây dựng hàm**

# Cách xây dựng hàm trong php

- Cú pháp:

```
function Ten_ham(ds tham số nếu có)
```

```
{
```

```
    thân hàm;
```

```
    [return giá trị;]
```

```
}
```

- PHP hỗ trợ truyền tham số vào hàm, truyền tham số dưới dạng tham chiếu (reference), truyền tham số với giá trị mặc định

# Cách xây dựng hàm trong php

- Ví dụ:

```
<?php  
function kiem_tra_so_chan($number)  
{  
    if ($number % 2 === 0) return true;  
    else return false;  
}  
  
$number = 12;  
if (kiem_tra_so_chan($number)) echo 'Số chẵn';  
else echo 'Số lẻ';  
}  
?>
```

# Cách xây dựng hàm trong php

- Ví dụ: Không dùng return

```
<?php  
function kiem_tra_so_chan($number)  
{  
    if ($number % 2 == 0) echo 'Số chẵn';  
    else echo 'Số lẻ';  
}  
$number = 12;  
kiem_tra_so_chan($number);  
?>
```

# Truyền tham số cho hàm

- Các tham số truyền vào hàm trong php có thể là các kiểu bất kỳ
- Số tham số truyền vào là không giới hạn
- Mỗi tham số cách nhau bởi dấu phẩy.
- **Ví dụ:**

```
function tinh tong($a, $b)
```

```
{
```

```
    return $a + $b;
```

```
}
```

Gọi hàm:

```
echo tinh tong($so1, $so2);
```

# Gán giá trị mặc định cho biến truyền vào

- Ví dụ:

```
function tinh tong($a, $b, $c = false)
```

```
{
```

```
    $tong = $a + $b;
```

```
    if ($c != false){
```

```
        $tong = $tong + $c;
```

```
}
```

```
    return $tong;
```

```
}
```

Gọi hàm: echo tinh tong(\$so1, \$so2);

Hoặc: echo tinh tong(\$so1, \$so2, 3);

# Biến toàn cục và biến cục bộ

- **Biến toàn cục:** là biến khai báo ở ngoài phạm vi hàm.
- **Biến cục bộ:** là biến khai báo ở trong phạm vi hàm.

**Chú ý:** Trong php để lấy giá trị biến toàn cục thì dùng lệnh **global \$tenbien;** để lấy

# Biến toàn cục và biến cục bộ

- **Ví dụ:**

```
$bien_toan_cuc = 12; // Biến toàn cục
function kiem_tra()
{
    $bien_cuc_bo = 13; // Biến cục bộ
    global $bien_toan_cuc; // Lấy biến toàn cục
    if ($bien_cuc_bo % $bien_toan_cuc){
        return true;
    }
    else{
        return false;
    }
}
```

# Các cách gọi hàm trong PHP

- Truyền bằng giá trị:
  - Mặc định tất cả các đối số truyền vào hàm đều là truyền bằng giá trị
  - Điều này có nghĩa là khi các đối số được truyền đến hàm được gọi, giá trị được truyền thông qua các biến tạm (tham số hình thức). Mọi thao tác chỉ thực hiện trên biến tạm này nên nó không hề tác động đến biến truyền vào.
  - Nếu tác động đến giá trị biến truyền vào thì sau khi thoát khỏi hàm giá trị đó không thay đổi.

# Các cách gọi hàm trong PHP

- Ví dụ:

```
$a = 1; // Biến  
function tang_len_1($a)  
{  
    return $a + 1;  
}
```

```
// Xuất giá trị trả về của hàm  
echo tang_len_1($a);
```

```
// Xuất giá trị của biến  
echo $a;
```

# Các cách gọi hàm trong PHP

- Truyền bằng tham chiếu:
  - Khi các đối số được truyền bằng giá trị thì giá trị của các đối số của hàm đang gọi không bị thay đổi. Tuy nhiên nếu muốn những giá trị đó thay đổi theo thì phải truyền tham biến vào hàm dạng tham chiếu.
  - Sử dụng dấu & trước đối số truyền vào.

# Các cách gọi hàm trong PHP

- Ví dụ:

```
$a = 1; // Biến
```

```
function tang_len_1(&$a)
```

```
{
```

```
    $a = $a + 1;
```

```
    return $a;
```

```
}
```

```
echo tang_len_1($a); // Xuất giá trị trả về của hàm
```

```
echo $a; // Xuất giá trị của biến
```

# Cách sử dụng lại mã php

- ◆ Sử dụng các khai báo sau:
  - ◆ Khai báo require(), include(): dùng để import nội dung một file PHP khác vào file PHP hiện tại.
  - ◆ Khai báo require\_once(), include\_once(): chỉ import đúng một lần nếu khai báo import nhiều lần.

# Khai báo require()

- ◆ Dùng để nhúng một trang này vào đoạn mã của một trang khác mà không cần phải viết lại mã.
- ◆ Hàm require() lấy tất cả text trong file đã chỉ định và sao chép chúng vào file có sử dụng hàm require.
- ◆ Nếu có bất kì vấn đề gì xảy ra trong quá trình nạp file thì hàm require() sinh ra một lỗi nghiêm trọng (Fatal Error) và ngăn chặn sự thực thi của script.

# Khai báo require()

## ◆ Ví dụ: Trang Bienluan.php

```
1 <html>
2 <head>
3 <title>trang 2</title>
4 </head>
5 <body>
6 <?php
7     $a=$_POST['t1'];
8     $b=$_POST['t2'];
9         if ($a==0)
10            if ($b!=0)
11                echo "PT vo nghiem";
12            else
13                echo "PT vo so nghiem";
14        else
15            echo "Nghiem PT: ". -$b/$a;
16    ?>
17 </body>
```

# Khai báo require()

- ◆ Sau đó xây dựng trang PTBN.php và có sử dụng require() để nhúng trang Bienluan.php vào.

```
1 <html>
2 <head>
3 <title>Bail</title>
4 </head>
5 <body>
6 <table bgcolor="#0099FF">
7 <tr><td>
8 <form method="POST" action="trang11.php">
9   <p>Giai PT Bac Nhat </p>
10  <p>a: <input type="text" name="t1" size="20" value=""></p>
11  <p>b: <input type="text" name="t2" size="20" value=""></p>
12  <p><input type="submit" value="Submit" name="B1">&ampnbsp;
13  <input type="reset" value="Reset" name="B2"></p>
14 </form></td></tr>
15 <tr><td>
16 <?php require ("Bienluan.php") ; ?>
17 </td></tr>
18 </table>
19 </body>
20 </html>
```

# Khai báo include()

- ◆ Lấy tất cả text trong file đã chỉ định và sao chép chúng vào trong file có sử dụng hàm include
- ◆ Nếu có lỗi sẽ hiển thị cảnh báo và tiếp tục thực thi.

# Khai báo include()

## ◆ Ví dụ:

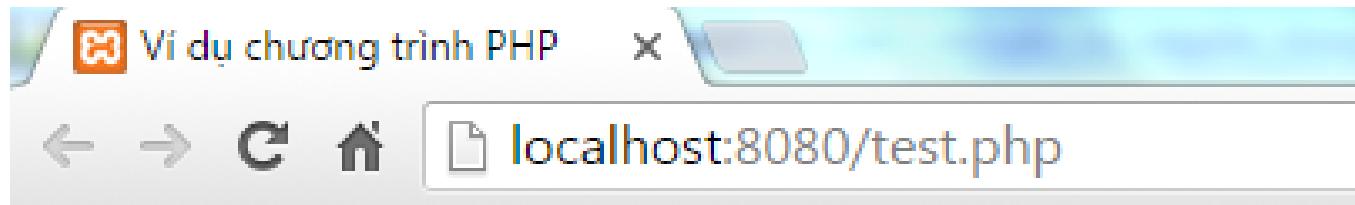
- ◆ Tạo một file là menu.php trong htdocs với nội dung sau

```
<a href="http://vietjack.com/index.jsp">Home</a> -  
<a href="http://vietjack.com/php">PHP</a> -  
<a href="http://vietjack.com/java">JAVA</a> -  
<a href="http://vietjack.com/html">HTML</a> <br />
```

- ◆ Tạo trang khác và gọi

```
<html>  
  <body>  
    <?php include("menu.php"); ?>  
    <p>Ví dụ minh họa cách include file trong PHP!</p>  
  </body>  
</html>
```

# Khai báo include()



Ví dụ minh họa cách include file trong PHP!

# Khai báo include()

## ◆ Ví dụ: Xét đoạn mã

```
21 <?php  
22     $a=$_POST['hsa'];  
23     if ($a==0)  
24         include ("giaiptbacnhat.php");  
25     else  
26         include ("giaiptbachai.php");  
27 ?>
```

# BÀI THỰC HÀNH

## 1. Tạo các file có tên như sau:

- Top.php
- Left.php
- Right.php
- Bottom.html

Sau đó, sử dụng require hoặc include để chèn vào trang index.php như sau:

# BÀI THỰC HÀNH

Top.php

Left.php

*Phần hiển thị nội dung  
chính*

right.php

Bottom.html

# BÀI THỰC HÀNH

## 2. Tạo 2 file có tên như sau:

- Giaiptbn.php
- Giaiptbh.php

Sau đó, sử dụng require hoặc include để chèn vào trang giainghiem.php.

## 3. Ngày tháng:

- Nhập vào xâu ngày tháng năm
- Tách xâu, kiểm tra và thông báo chính xác ngày, tháng, năm nhập vào. Nếu giá trị nhập vào không hợp lệ thì đưa ra thông báo.

# BÀI THỰC HÀNH

## 4. Tập tin

- Tạo một file .txt
- Đọc dữ liệu từ file và hiển thị ra màn hình
- Ghi nội dung mới vào file.

# BÀI THỰC HÀNH

## 5. Xây dựng hàm

- Tạo mảng 1 chiều có các giá trị 1, 5, 7, 2, 6, 10, 13, 15, 28, 30, 1
- Thực hiện các công việc sau bằng cách xây dựng các hàm:
  - Đếm và in danh sách các số chẵn
  - Đếm và in danh sách các số lẻ
  - Tính tổng các số chia hết cho 3
  - Tính tích các số không chia hết cho 3
  - In số lớn nhất và số nhỏ nhất
  - Sắp xếp mảng theo thứ tự tăng dần và in kết quả
  - In ra vị trí các phần tử trong mảng có giá trị chia hết cho 2.