



# Ngôn ngữ PHP

**ThS. Nguyễn Duy Hải**  
**PGĐ TT CNTT-ĐHSP Hà Nội**

# Giới thiệu

---

- ✓ PHP=PHP: Hypertext Preprocessor. Tên gốc là Personal Home Page
- ✓ Là ngôn ngữ để viết các trang web động
- ✓ Bộ biên dịch PHP là phần mềm mã nguồn mở, có thể chạy cùng với các web server Apache, IIS,...
- ✓ Thường kết hợp với hệ quản trị CSDL MySQL



# Nhúng PHP vào HTML

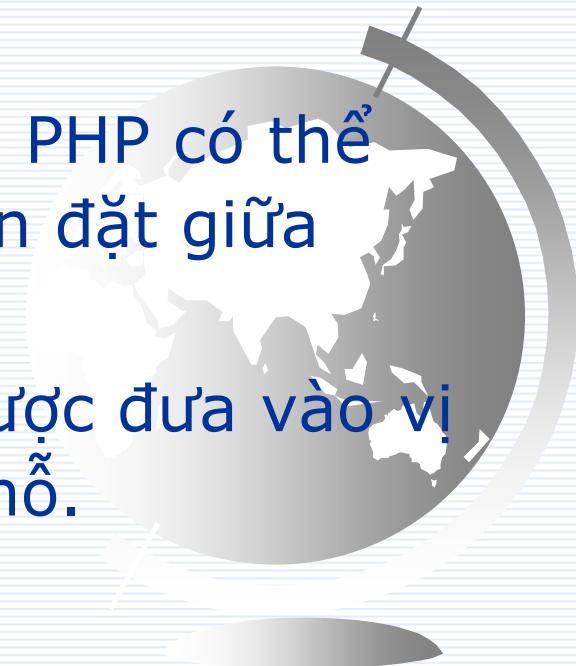
- ✓ Có thể nhúng mã PHP vào mọi vị trí trong trang HTML.
- ✓ Đoạn mã PHP được đặt giữa: `<?php ... ?>`:

`<?php`

//Đoạn lệnh PHP ở đây

`?>`

- ✓ Một cấu trúc lệnh thông thường của PHP có thể được tách làm nhiều phần, mỗi phần đặt giữa `<?php...?>`
- ✓ Kết quả do đoạn lệnh PHP đưa ra được đưa vào vị trí mà đoạn lệnh PHP đang chiếm chỗ.



# Đặc điểm PHP

---

- ✓ Có khả năng hướng đối tượng
- ✓ Thông dịch
- ✓ Phân biệt chữ hoa/chữ thường
- ✓ Mỗi lệnh kết thúc bởi chấm phẩy (;)





# Cú pháp ngôn ngữ

# Chú thích trong PHP

---

✓ // Dòng chú thích

✓ # Dòng chú thích

✓ /\*

Đoạn chú thích trên nhiều dòng

\*/



- ✓ Phân biệt chữ hoa, chữ thường
- ✓ Bắt đầu bằng dấu đô la (\$), tiếp ngay sau \$ là tên biển.
- ✓ Tên biển bao gồm chữ cái, chữ số, dấu gạch nối (\_) và phải bắt đầu bằng chữ cái hoặc dấu gạch nối.
- ✓ Ví dụ: \$a, \$b, ...
- ✓ Biển không cần khai báo (được tự động khai báo vào lần gán giá trị đầu tiên).
- ✓ Việc sử dụng biển chưa khởi tạo sẽ gây lỗi.



# Biến động (biến biến)

✓ Cho phép sử dụng giá trị của biến làm tên biến khác.

✓ VD:

```
$a = "hello";
```

```
$$a = "world"; // $hello = "world"
```





# Kiểu dữ liệu

---

- ✓ PHP hỗ trợ 4 kiểu
  - Số
  - Chuỗi
  - Logic
  - Mạng & đối tượng



✓ Số nguyên từ  $-2^{31}$  đến  $2^{31}-1$

– Hệ thập phân: VD:  $a = 16;$

– Hệ 16 (hexa): VD:  $a=0x10;$

– Hệ 8 (bát phân): VD:  $020;$

✓ Số thực (thập phân): từ  $1.7E-308$  đến  $1.7E+308$

– Biểu diễn:  $a = 0.017$

– Dạng khoa học:  $a = 17.0E-03$



# Kiểu chuỗi

- ✓ Giới hạn bởi nháy đơn (') hoặc kép (")
- ✓ Chuỗi đặt trong nháy kép bị thay thế và xử lý ký tự thoát. Trong nháy đơn thì không.
- ✓ Ví dụ:

```
$a = "Hello";
```

```
$b = "$a world"; //tương đương $b="Hello world"
```

```
$c = '$a world'; //$c='$a world' (không thay đổi)
```



# Kiểu chuỗi (tiếp)

- ✓ Để làm rõ các biến trong chuỗi, cần bao biến vào giữa cặp ngoặc nhọn { }

```
$a = "He";
```

```
$b = "$allo"; //lỗi vì PHP hiểu là $allo
```

```
$c = "{$a}llo"; //đúng ($c = "Hello")
```

- ✓ Biến mảng, biến biến, biến trong tên có dấu gạch nối hoặc dấu chấm khi viết trong chuỗi bắt buộc phải bao lại.



# Kiểu chuỗi (tiếp)

## ✓ Ký tự thoát: \

- Sử dụng để viết các ký tự đặc biệt trong chuỗi
- VD:

Cần có chuỗi: Người ta nói "PHP rất tốt"

```
$a = "Người ta nói "PHP rất tốt""; //Sai
```

```
$a = "Người ta nói \"PHP rất tốt\""; //Đúng
```

- Một số ký tự phải sử dụng ký tự thoát: \$, \, "
- Ngoài ra:
  - \n: Xuống dòng
  - \r: trở về đầu dòng
  - \t: dấu tab
  - ...



# Kiểu chuỗi (tiếp)

## ✓ Kiểu tài liệu (heredoc):

- Cho phép viết 1 chuỗi trên nhiều dòng.
- Không cần sử dụng ký tự thoát:
- Cách viết:

```
$biến = <<<Ký_hiệu  
    nội dung trên nhiều dòng  
Ký_hiệu;
```

- **Chú ý:** Ký\_hiệu phải được viết ở ký tự đầu tiên của dòng
- Ví dụ:

```
$a = <<<EOD
```

Đây là chuỗi nằm trên nhiều dòng sử dụng cú pháp kiểu tài liệu 'heredoc'

```
EOD;
```



# Kiểu logic

---

- ✓ Có 2 trạng thái: `true` và `false`



# Kiểu mảng

- ✓ Cho phép chứa nhiều giá trị dữ liệu được đánh chỉ mục bằng số hay chuỗi.
  - ✓ Để truy cập vào 1 phần tử, đặt chỉ số trong [ ]
  - ✓ Ví dụ:
- ✓ Có thể sử dụng chỉ số không liên tiếp
  - ✓ Có thể thêm phần tử vào cuối mảng bằng cách bỏ chỉ số:

```
$a[0] = "Xin";
```

```
$a[1] = "Chào";
```

```
$a[0] = "Xin";
```

```
$a[] = "Chào"; // $a[1]
```

```
$a[] = "Bạn"; // $a[2]
```





# Kiểu mảng (tiếp)

## ✓ Chỉ số chuỗi:

```
$a["ten"] = "Hồng";
```

```
$a["mau"] = "Đỏ";
```

## ✓ Mảng nhiều chiều

– Được coi là mảng của mảng

– Ví dụ:

```
$a[0][0] = "TT";
```

```
$a[0][1] = "Họ tên";
```

```
$a[1][0] = 1;
```

```
$a[1][1] = "Nguyen Van A";
```



# Kiểu mảng (tiếp)

## ✓ Khởi tạo mảng bằng hàm `array()`

### - Chỉ số mặc định:

```
$a=array("xin", "chào", "bạn");
```

*/\*tương đương với*

```
$a["0"]="xin";
```

```
$a["1"]="chào";
```

```
$a["2"]="bạn";*/
```

### - Chỉ số mặc định với chỉ số đầu khác 0:

```
$a=array(5=>"xin", "chào", "bạn");
```

*/\*tương đương với*

```
$a[5]="xin";
```

```
$a["1"]="chào";
```

```
$a["2"]="bạn";*/
```



# Kiểu mảng (tiếp)

## ✓ Khởi tạo mảng bằng hàm `array()` (tiếp)

– Chỉ số chuỗi:

```
$a=array("ten"=>"Hồng", "mau"=>"Đỏ");
```

/\*Tương đương với:

```
$a["ten"] = "Hồng";
```

```
$a["mau"] = "Đỏ";*/
```

– Mảng nhiều chiều:

```
$a = array(  
    array("TT", "Họ tên"),  
    array(1,"Nguyễn Văn A")  
);
```



# Ép kiểu và kiểm tra kiểu

## ✓ Sử dụng cú pháp tương tự C/C++

```
$x = "123abc"; // $x  
    là chuỗi  
$x = (int) "123abc";  
// $x là số  
    nguyên=123
```

## ✓ Các kiểu (bảng)

## ✓ Kiểm tra kiểu:

```
gettype(),  
is_bool(),  
is_long(), ...
```

| Ký hiệu                      | Ý nghĩa kiểu                      |
|------------------------------|-----------------------------------|
| (int), (integer)             | Số nguyên                         |
| (real), (double),<br>(float) | Số thập phân                      |
| (string)                     | Chuỗi                             |
| (array)                      | Mảng                              |
| (object)                     | Đối tượng                         |
| (bool), (boolean)            | Logic                             |
| (unset)                      | NULL, tương tự<br>như gọi unset() |

## ✓ Kiểu số:

– Kết hợp: +, -, \*, /, %

– Tăng giảm: ++, --

VD: `$a++`; `$a--`; `++$a`; `--$a`;

## ✓ Chuỗi: Toán tử chấm (.)

## ✓ Logic: AND (&&), OR (||), XOR, !

## ✓ Gán: =, +=, -=, .=

## ✓ So sánh: ==, !=, <, >, <=, >=





# Các cấu trúc điều khiển

# Rẽ nhánh **if**

```
if (BTLG)  
    lệnh;
```

```
if (BTLG)  
    lệnh1;  
else  
    lệnh2;
```

```
if (BTLG1)  
    lệnh1;  
elseif (BTLG2)  
    lệnh2;  
elseif (BTLG3)  
    lệnh3;  
else  
    lệnhkhác;
```

```
$biến= BTLG?Giá_trị1:Giá_trị2
```

# switch

```
switch ($biến) {  
    case (giá_trị1):  
        các_lệnh1;  
    case (giá_trị2):  
        các_lệnh2;  
    default:  
        các_lệnh_khác;  
}
```





# Lặp không xác định

---

```
while (BTLG)  
    lệnh;
```

```
do {  
    các_lệnh  
} while (BTLG);
```



# Lặp xác định

```
for (lệnh_khởi_tạo; BTLG; lệnh2)  
    lệnh;
```

```
foreach ($biến_mảng as $giá_trị)  
    lệnh;
```

```
foreach ($biến_mảng as $khóa=>$giá_trị)  
    lệnh;
```

# break và continue

---

- ✓ **break**: Kết thúc khối lệnh `for`, `while`, `do-while` hoặc `switch`
- ✓ **continue**: bỏ qua phần sau `continue` trong vòng lặp, chuyển sang vòng lặp tiếp.



## ✓ Khái báo hàm

```
function tên_hàm(ts1, ts2,..., tsn) {  
    //các lệnh trong thân hàm  
}
```

- ✓ Để trả ra giá trị cho hàm ta sử dụng lệnh  
`return biểu_thức;`
- ✓ Để thoát khỏi hàm:
  - Sử dụng `return`
  - `exit;`



# Hàm tự tạo (tt)

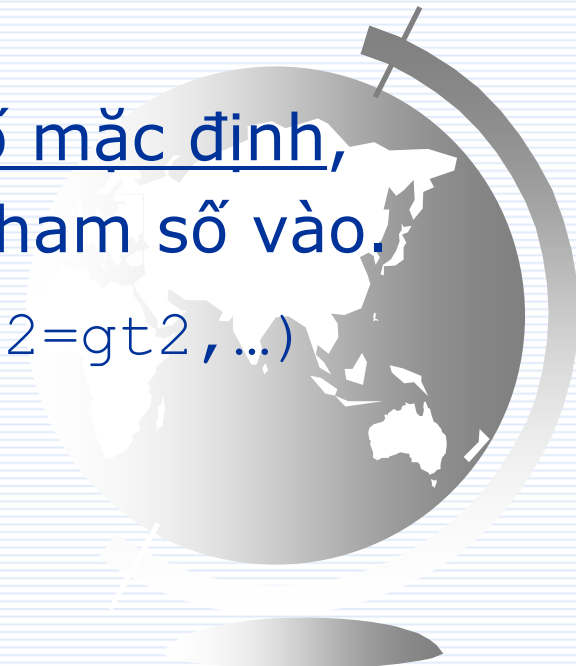
## ✓ Gọi hàm:

```
tên_hàm(gt1, gt2,... , gtn);
```

## ✓ Chú ý:

- Nếu hàm không có tham số, khi khai báo hàm, gọi hàm vẫn phải giữ lại ()
- Để khai báo hàm với các tham số mặc định, khi khai báo ta đưa ngay giá trị tham số vào.

```
function tên_hàm(ts1=gt1, ts2=gt2,...)
```



# Phạm vi biến

✓ Phạm vi biến là phạm vi ở đó biến xác định.

Trong PHP, biến có 3 phạm vi:

- Local variables: biến cục bộ:
  - Khởi tạo trong hàm
  - có tác dụng từ khi khởi tạo đến hết hàm.
- Global variables: Biến toàn cục:
  - Khởi tạo ở ngoài hàm
  - Có tác dụng từ khi khởi tạo đến hết file trừ ở trong hàm.
  - Để đưa 1 biến toàn cục vào trong hàm, sử dụng khai báo **global** \$biến1, \$biến2,...; hoặc mảng **\$\_GLOBAL**
- Super Global Variables: Siêu toàn cục: Một số biến có sẵn của PHP: **\$\_SERVER**, **\$\_GLOBAL**...



# Biến tĩnh

- ✓ Khai báo trong hàm
- ✓ Giá trị được lưu trữ qua nhiều lần gọi hàm
- ✓ Chỉ được khởi tạo ở lần khai báo đầu tiên
- ✓ Để khai báo:

```
static $biến_tĩnh=giá trị;
```



# Chèn file

---

- ✓ **include** ("địa\_chi\_file");
- ✓ **require** ("địa\_chi\_file");
- ✓ **include\_once** ("địa\_chi\_file");
- ✓ **require\_once** ("địa\_chi\_file");







# Hướng đối tượng trong PHP

# Định nghĩa lớp

```
class tên_lớp{  
    //thânlớp  
}
```

- ✓ Thân lớp gồm có các khai báo dữ liệu, phương thức.
- ✓ Thân lớp phải được đặt trong 1 khối lệnh PHP duy nhất.
  - Các dữ liệu (biến) được khai báo bằng `var`
  - Các phương thức (hàm) khai báo như thông thường.

