

# **Web tĩnh và web động**

**Giảng viên:**

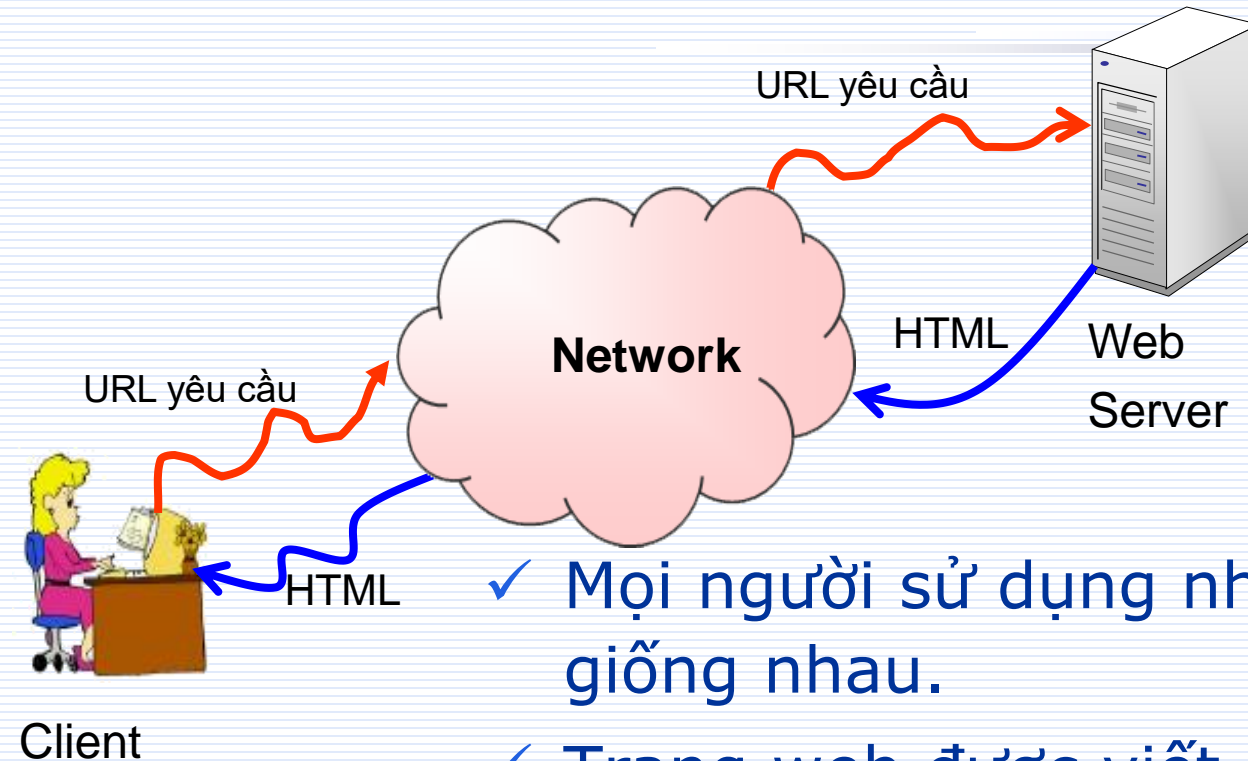
**Ths. Hoàng Anh Đức**

**- Mobile: 0986999984**

**- Email: [ducha.hung@gmail.com](mailto:ducha.hung@gmail.com)**

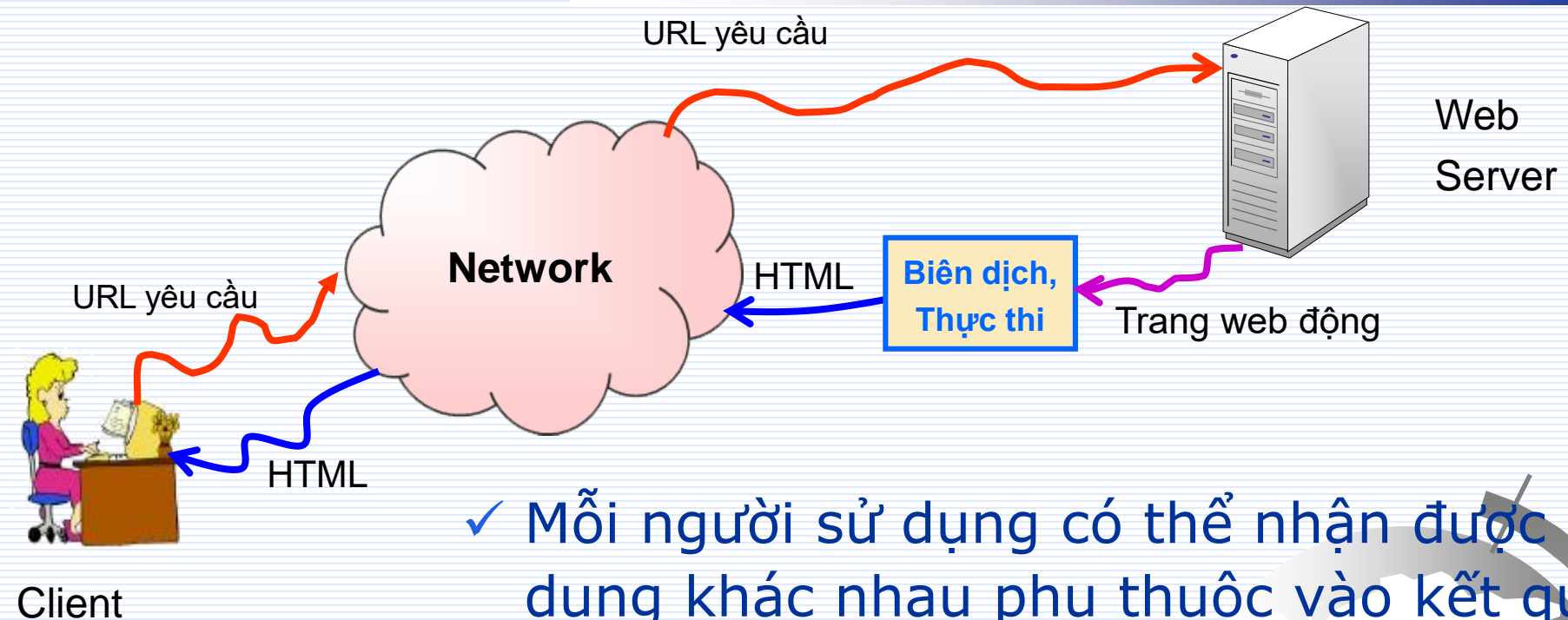
**Bộ môn Công nghệ phần mềm, Phòng 703, Nhà C12 tầng**

# Trang web tĩnh



- ✓ Mọi người sử dụng nhận được kết quả giống nhau.
- ✓ Trang web được viết bằng HTML, chỉ thay đổi khi có sự thay đổi của người xây dựng
- ✓ Khả năng tương tác yếu
- ✓ Webserver hoạt động giống 1 file server.

# Trang web động



- ✓ Mỗi người sử dụng có thể nhận được nội dung khác nhau phụ thuộc vào kết quả chạy chương trình.
- ✓ Trang web viết bằng HTML + Ngôn ngữ lập trình phía server. Có thể được thay đổi bởi người sử dụng
- ✓ Khả năng tương tác mạnh

# Một số công nghệ viết web động

---

## ✓ Động phía client:

- JavaScript, VBScript được chạy ở client.
- Applet (Chạy chương trình Java)
- Flash



# Một số công nghệ viết web động

---

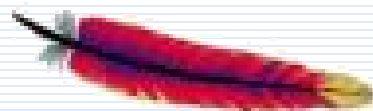
## ✓ Động phía server:

- CGI: Common Gateway Interface
- Java Server Pages (JSP)
- ASP, ASP.NET: Microsoft
  - Viết bằng VBScript, JavaScript chạy phía server.
  - Sử dụng web server IIS.
- PHP: Mã nguồn mở
  - Ngôn ngữ lập trình PHP, chạy phía server.
  - Webserver: Apache, IIS
  - Bộ biên dịch: PHP
- Perl
- Python



# Cài đặt và cấu hình các phần mềm

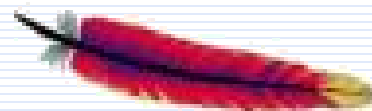
- ✓ Web server: Chọn 1 trong 2
  - IIS (Internet Information Service) – tích hợp trong Windows.
  - **Apache**: <http://www.apache.org>
- ✓ Biên dịch: PHP: <http://www.php.net>
- ✓ Hệ quản trị CSDL: MySQL  
[www.mysql.com](http://www.mysql.com)
- ✓ Hỗ trợ quản lý CSDL MySql
  - MySql Control
  - **PHPMyAdmin (web)**
  - MySQL Front



# Cài đặt và cấu hình các phần mềm

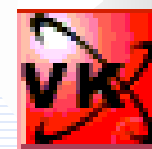
## ✓ Hỗ trợ soạn thảo:

- NotePad++, EditPlus, Zen Studio
- PHPEdit, PHPExpert
- Macromedia Dreamweaver
- Netbean IDE



## ✓ Phần mềm khác:

- Bộ gõ: **Unikey**, Vietkey bản đầy đủ
- Adobe Photoshop
- Xara3D...



# Cài đặt và cấu hình các phần mềm

## ✓ WAMP Server

<http://www.wampserver.com/en/>

## ✓ XAMPP Server

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

## ✓ Tích hợp các phần mềm cần thiết:

- Apache
- PHP
- MySQL
- PHPmyadmin





# 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 (;)



A stylized graphic featuring a globe with a map of Vietnam highlighted in white. The globe is set against a light blue background with orange crosshairs. The text "Cú pháp ngôn ngữ" is overlaid on the globe.

# 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"
```



- ✓ PHP hỗ trợ 5 kiểu
  - Số (integer, float, double, real)
  - Chuỗi (string)
  - Logic
  - Mảng (array)
  - Đối tượng (object)





- ✓ 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";
```

```
$b = "$a11o"; //lỗi vì PHP hiểu là $a11o
```

```
$c = "{$a}11o"; //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:

```
$biên = <<<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`
- ✓ Kết quả hiển thị là 1 (`true`) và trống (`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:  

```
$a[0] = "Xin";  
$a[1] = "Chào";
```
- ✓ 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[] = "Chào"; // $a[1]  
$a[] = "Ban"; // $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[]="xin";
```

```
$a[]="chào";
```

```
$a[]="bạn"; */
```



# Kiểu mảng (tiếp)

– Chỉ số mặc định với chỉ số đầu khác 0:

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

/\*tương đương với

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

```
$a[]="chào";
```

```
$a[]="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), (float)	Số thập phân
(string)	Chuỗi
(array)	Mảng
(object)	Đối tượng
(bool), (boolean)	Logic
(unset)	NULL, tương tự 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: $==$ , $!=$ , $<$ , $>$ , $<=$ , $>=$



# XOR

A	B	A XOR B
T	T	F
T	F	T
F	T	T
F	F	F





# **Các cấu trúc điều khiển**

# Câu lệnh ghép

```
{  
    lệnh_1;  
    lệnh_2;  
    .....  
    lệnh_n;  
}
```





## Rẽ nhánh **if**

```
if (BTLG)  
    lệnh;
```

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

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

```
$biến= BTLG?Giá_trị_đúng:Giá_trị_sai;
```

# 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 $khoá=>$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.





# Phạm vi biến

- 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 **globals** `$biến1, $biến2, ...;` hoặc mảng **\$GLOBALS**
- Super Global Variables: Siêu toàn cục:  
Một số biến có sẵn của PHP: **\$\_SERVER**, **\$GLOBALS** . . .



# 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\_chỉ\_file");
- ✓ **require** ("địa\_chỉ\_file");
- ✓ **Điểm chung:** cho phép chèn một file PHP vào một file PHP hiện tại. Nếu file được chèn có lỗi thì chương trình sẽ hiển thị thông báo lỗi.
- ✓ **Khác nhau:**
  - Include: nếu file chèn vào có lỗi thì nó cảnh báo lỗi và tiếp tục thực hiện file cho tới cuối cùng
  - Require: nếu có lỗi thì nó báo lỗi và dừng tại đó, không thực hiện câu lệnh tiếp theo nữa



# Chèn file

---

✓ **include\_once** ("địa\_chỉ\_file");

✓ **require\_once** ("địa\_chỉ\_file");

Thông báo rằng file PHP chỉ được thêm một lần duy nhất và sẽ không chèn thêm nữa.



# Class

- ✓ Các lớp được khai báo thông qua từ khóa `class`, các thuộc tính khai báo dưới dạng các biến còn các phương thức được khai báo dưới dạng các hàm

```
class tên_lớp {  
    //danh_sách_các biến, hằng, lớp  
    //danh_sách_các_hàm  
}
```



# Class (tiếp)

- ✓ Khởi tạo một đối tượng thuộc một lớp với từ khóa ***new***

```
$tên_đối_tượng = new tên_lớp();
```

- ✓ Để gọi hàm trong lớp ta sử dụng toán tử ->

```
$tên_đối_tượng->tên_hàm();
```

- ✓ Hoặc dùng toán tử :: để gọi đến một hàm hay một biến (biến phải được khai báo *static*)

```
$tên_đối_tượng::tên_hàm();
```

```
$tên_đối_tượng::$tên_biến;
```



# Class (Tiếp)

## Ví dụ

```
class Counter{
    var $cong = 0;
    //var $khoitao= 0;
    function increment() {
        $this -> cong++;
    }
}
$aCounter = new Counter();
for($i=0;$i<10;$i++){
    $aCounter->increment();
    echo $aCounter->cong;
}
// hiển thị ra kết quả là 1-10

?>
```

