

# Tổng quan về PHP



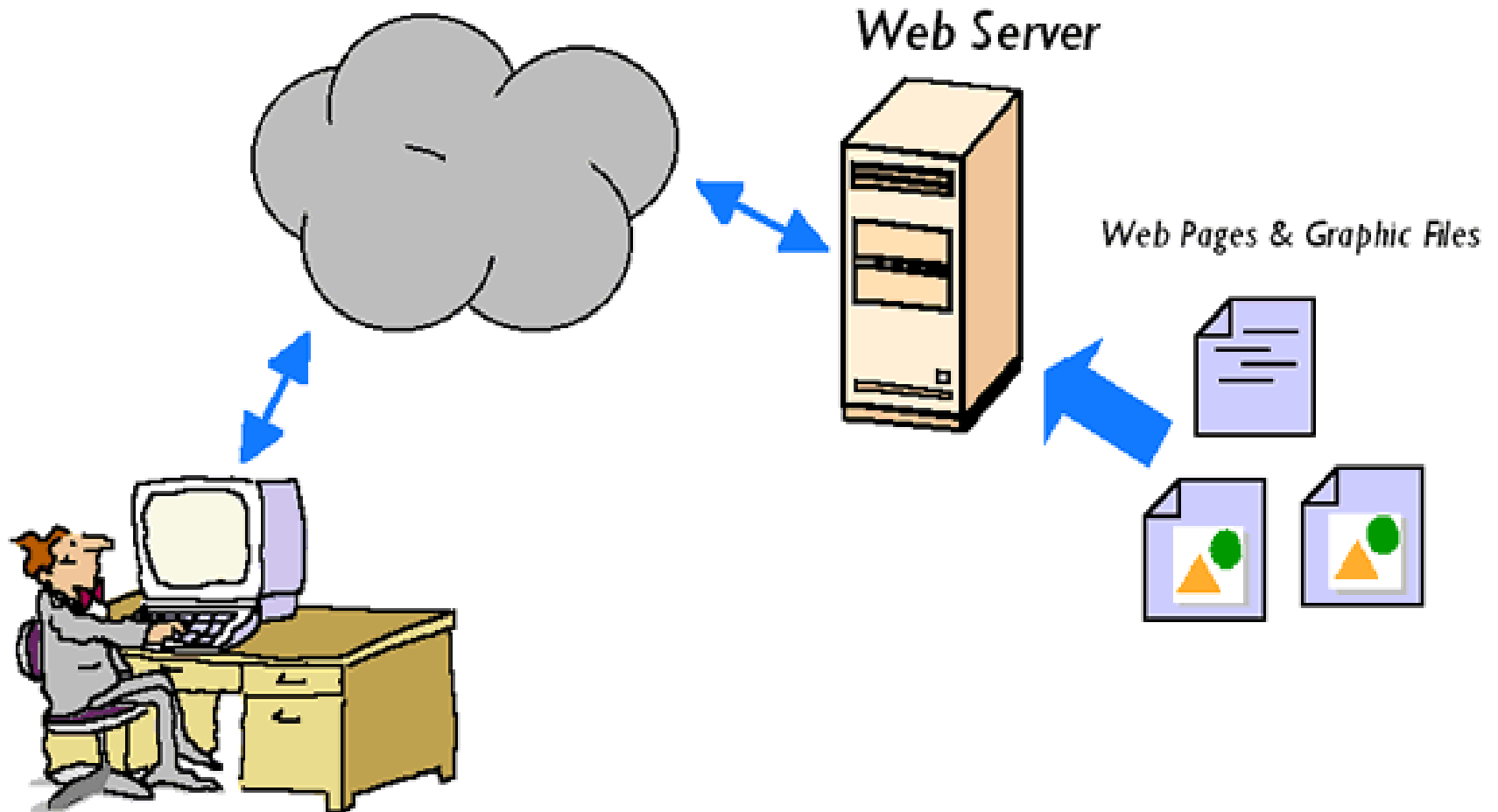


# Nội dung

---

- Nội dung
  - Lịch sử
    - Web Server
    - PHP
  - Lập trình cơ bản với PHP
    - Khai báo PHP,
    - Biến, hằng
    - Kiểu dữ liệu
    - Toán tử
    - Lệnh điều khiển
    - Hàm và khai báo
    - Thư viện hàm trong PHP
    - ...

# Khái niệm về Web Server





# Khái niệm Web Server

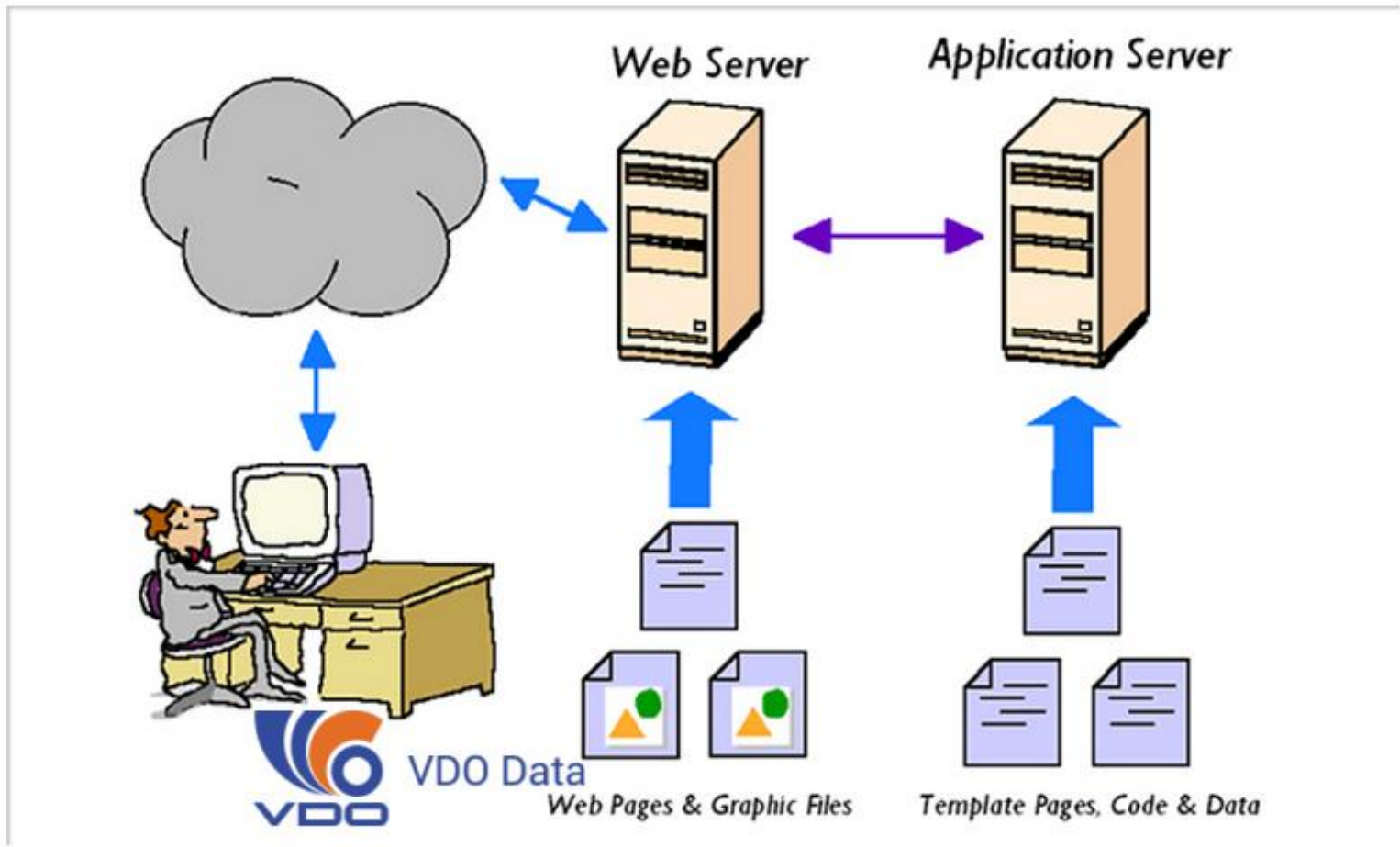
---

- **Web Server** – một máy tính lưu trữ các file thành phần của một WebSite (các tài liệu HTML, các file ảnh, CSS và các file JavaScript) và có thể phân phối chúng tới thiết bị của người dùng cuối thông qua kết nối mạng Internet với giao thức HTTP và có thể truy cập tới thông qua tên miền hoặc địa chỉ.
- **Web Server** – đều hiểu và chạy được các file \*.htm và \*.html, tuy nhiên mỗi **Web Server** lại phục vụ một số kiểu file chuyên biệt chẳng hạn như IIS của Microsoft dành cho \*.asp, \*.aspx còn Apache dành cho \*.php.
- **Web Server** – khi nhận yêu cầu truy cập các thông tin, Web Server Software sẽ xử lý yêu cầu và gửi lại những thông tin (HTML) mà bạn mong muốn.

# Web Server vs Application Server

## Top server phổ biến

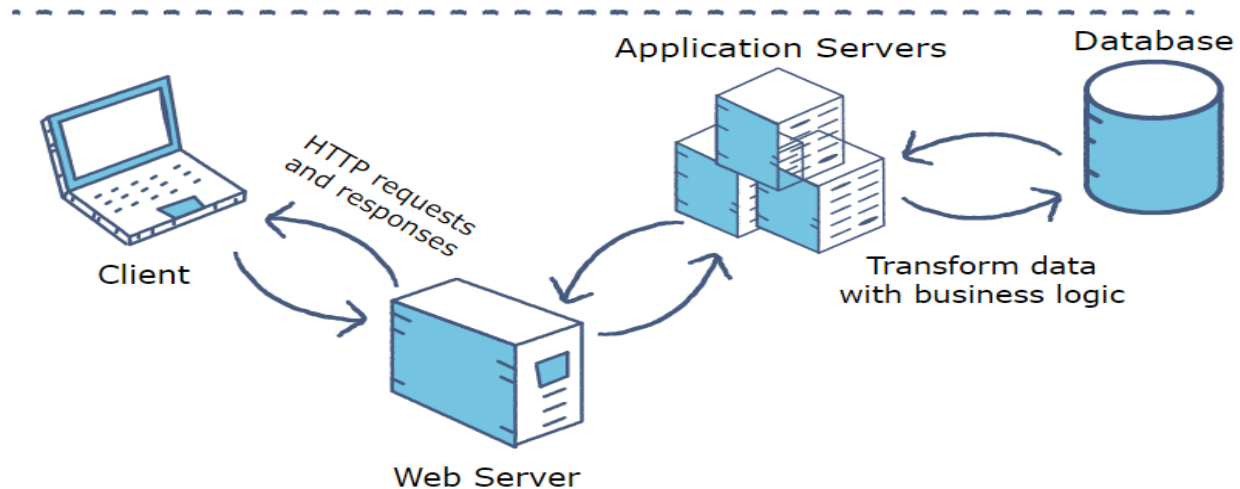
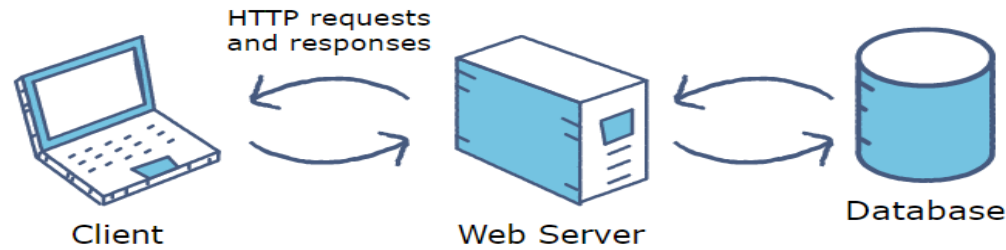
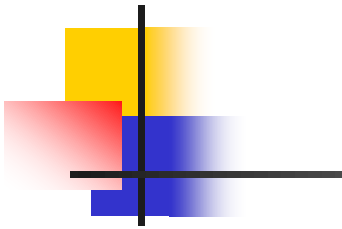
- **Web Server:** LiteSpeed, Nginx, Apache, IIS
- **Application Server:** Apache Tomcat, Oracle WebLogic, JBoss





# Web Server vs Application Server

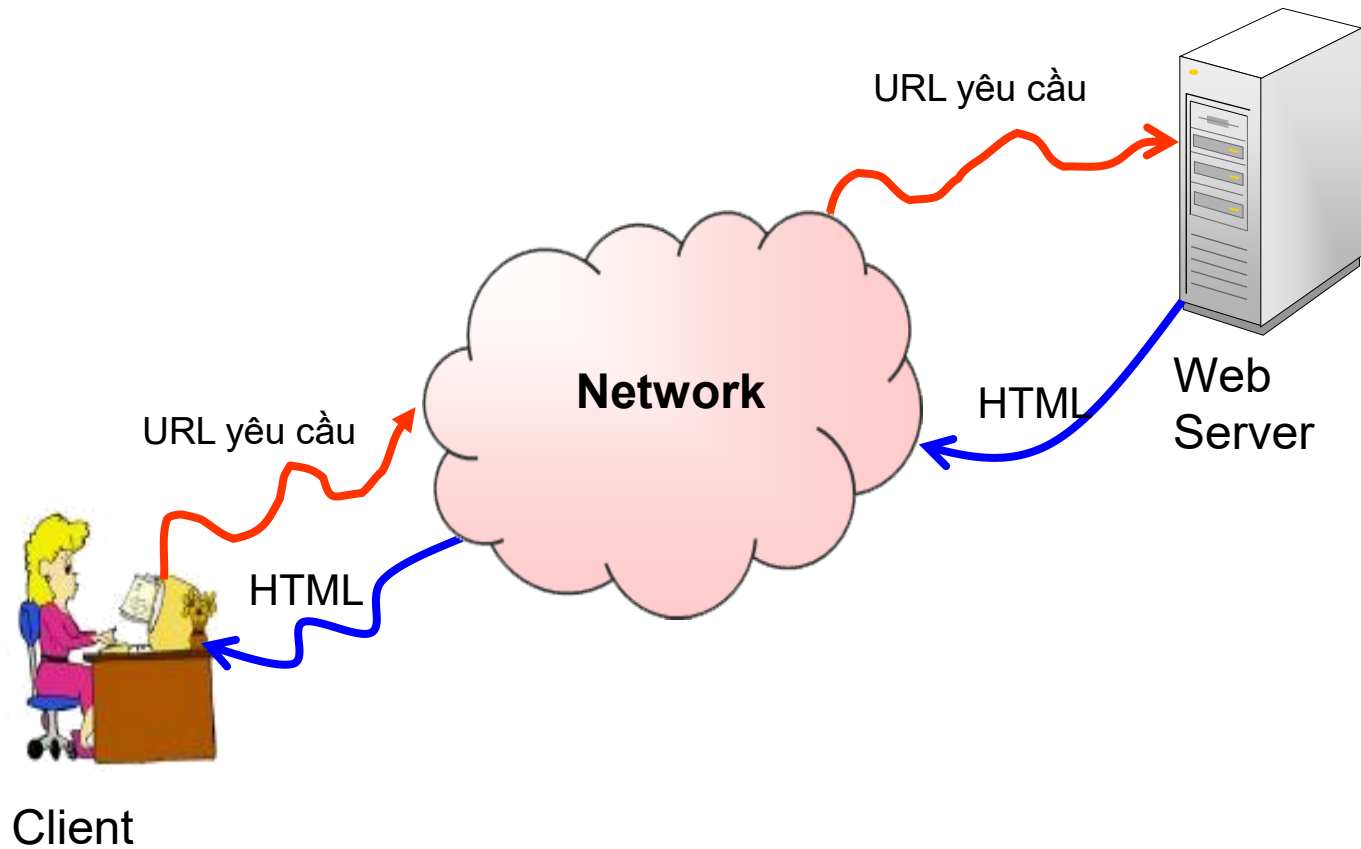
- Web server là một hệ thống máy tính dùng để lưu trữ, xử lý và phân phối nội dung website đến client. Theo mặc định, nó chỉ xử lý nội dung HTML tĩnh sau:
  - Documents
  - Images
  - Videos
  - Fonts
- Web server chỉ chấp nhận request qua giao thức HTTP/HTTPS.
- Application server là một framework chịu trách nhiệm cung cấp nội dung động của website cho client.
- Application server cung cấp cho client khả năng tương tác với hệ thống bằng những nghiệp vụ logic. Từ đó, application server sẽ chuyển đổi dữ liệu này thành nội dung động và thực thi các functions. Nội dung động có thể bao gồm:
  - Kết quả phiên giao dịch
  - Hỗ trợ quyết định
  - Phân tích thời gian thực



- **Web Server** sẽ nhận request trực tiếp từ client, xử lý và phản hồi nội dung tĩnh HTML về client. Nếu trong request có yêu cầu nội dung động, nó sẽ chuyển tiếp yêu cầu dạng này **đến Application Server**. Sau khi kích hoạt các functions trong mã nguồn, tương tác và nhận kết quả từ database, application server sẽ phản hồi về cho web server. Web server lúc này có nhiệm vụ chuyển đổi nội dung động nhận được thành dạng mã HTML và phản hồi về client.

# Phân loại Web Server

- Static Web Server







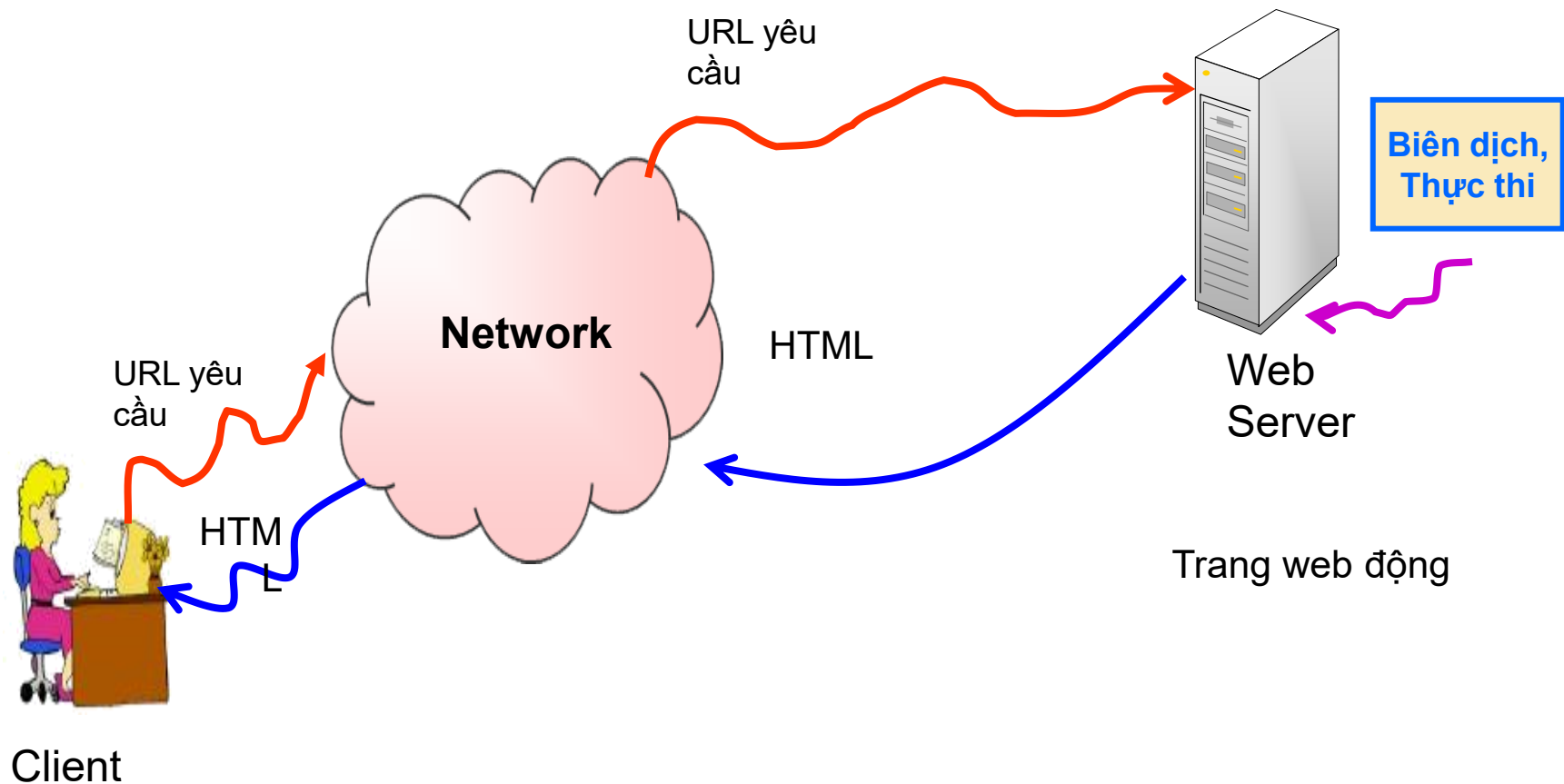
# Phân loại Web Server

---

- 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 do WebServer hoạt động giống như File Server.
  - Ưu điểm:
    - Truy cập nhanh
    - Chi phí đầu tư thấp
    - Ít dùng tài nguyên máy chủ
- ➔ Không còn được sử dụng nhiều hoặc chỉ dùng làm Web site giới thiệu công ty.

# Phân loại Web Server

## ■ Dynamic Web Server





# Phân loại Web Server

---

- 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 Client/Server và có thể được thay đổi bởi người sử dụng.
- Ưu điểm:
  - Cho phép người dùng tương tác với Website.
  - Cho phép quản lý nội dung thông qua CSDL.

Xây dựng các Website thương mại điện tử, tin tức, ...



# Phân loại Web Server

---

- Một số công nghệ sử dụng cho Dynamic Web Server
  - Dynamic Client Side:
    - JavaScript
    - VBScript
    - Flash
    - Applet
  - Dynamic Server Side:
    - JSP: Java Server Pages – Sun Microsystems
    - ASP, ASP.NET: Active Server Pages – Microsoft
    - **PHP**: LAMP – Liên minh mã nguồn mở
    - PERL: Practical Extraction and Reporting Language
    - CGI: Common Gateway Interface



# DATABASE SERVER

---

MySQL	Cơ sở dữ liệu mã nguồn mở, sẵn dùng cho tất cả các hệ điều hành phổ biến
Oracle	Máy chủ cơ sở dữ liệu của Oracle, sẵn dùng cho tất cả các hệ điều hành phổ biến
DB2	Máy chủ cơ sở dữ liệu của IBM, sẵn dùng cho tất cả các hệ điều hành phổ biến
MS SQL	Máy chủ cơ sở dữ liệu của Microsoft, chỉ chạy trên HĐH Windows

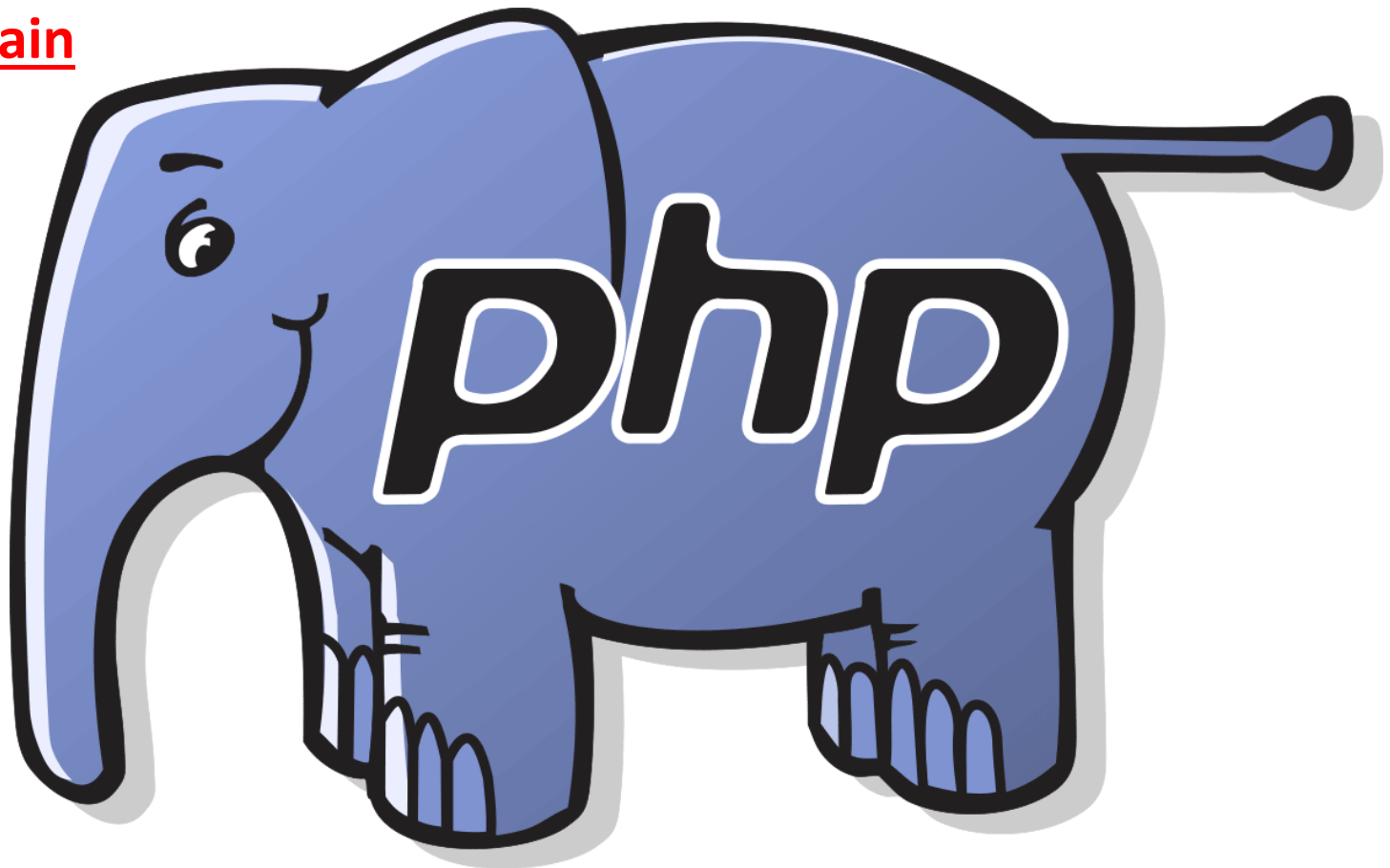


# Ngôn ngữ thông dụng phía máy chủ

PHP	Thường được sử dụng với Apache nhưng hiện cũng được IIS hỗ trợ, xử lý các file có đuôi .php.
ASP.NET	Sử dụng bởi Microsoft IIS. ASP.NET sử dụng tên đuôi .aspx và làm việc chủ yếu với các ứng dụng được viết trên C# hoặc Visual Basic
Pearl	Được phát triển cho mục đích xử lý văn bản bằng dòng lệnh của UNIX và viết các ứng dụng web. Perl sử dụng đuôi .pl
Python	Được dùng để phát triển nhiều loại ứng dụng trong đó có ứng dụng web. Python được sử dụng chủ yếu với Apache và có đuôi là .py.
JSP	Cần thêm một máy chủ ứng dụng như Tomcat server, được phát triển miễn phí bởi Công ty Apache Software Foundation. Các trang JSPs sử dụng đuôi .jsp và chủ yếu làm việc với servlet được viết trên ngôn ngữ Java

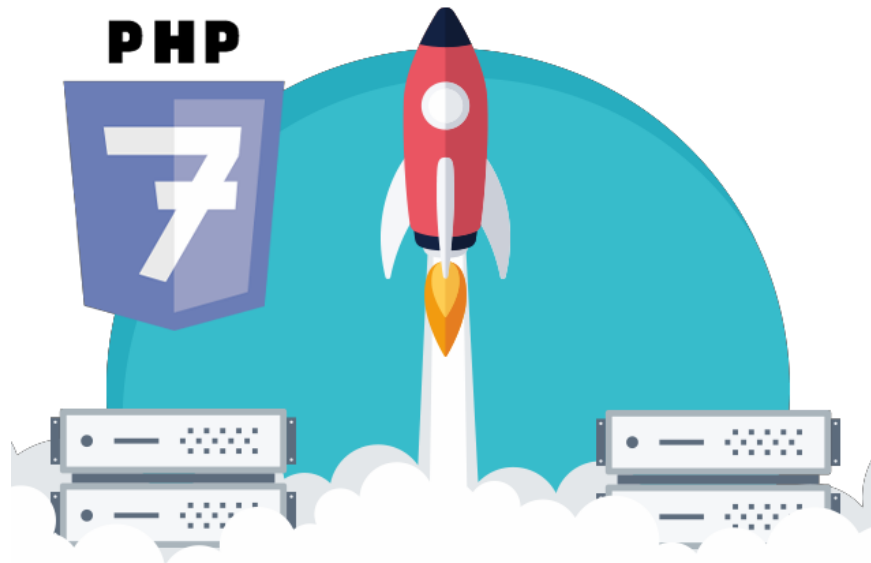
# Tổng quan về PHP

- Timeline
- Main



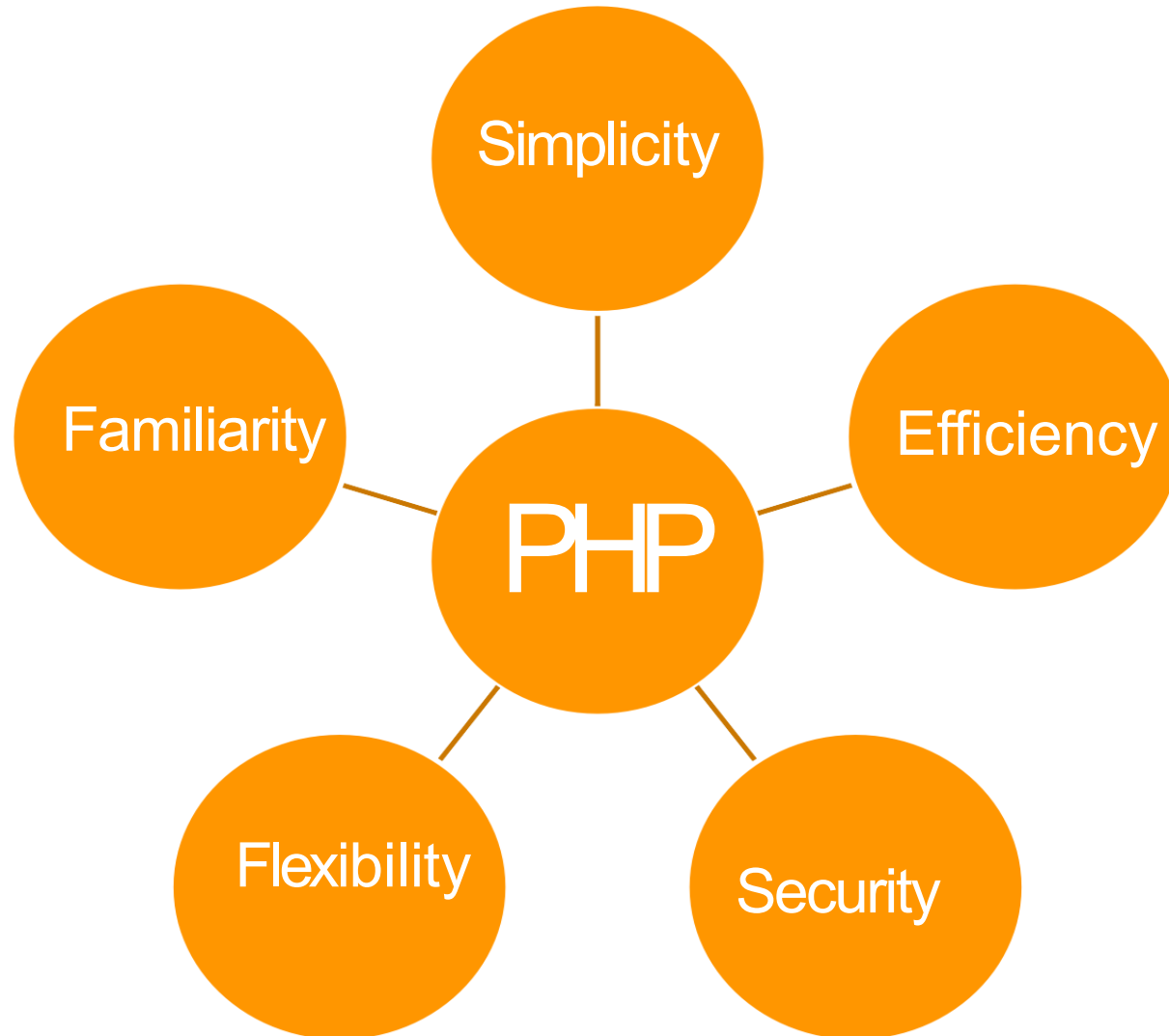
# Tổng quan về PHP

- ❑ PHP: Hypertext Preprocessor
- ❑ Ngôn ngữ lập trình kịch bản
- ❑ Phát triển các ứng dụng viết cho máy chủ
- ❑ Mã nguồn mở
- ❑ Nhúng vào các trang HTML





# Các đặc trưng của PHP





# Tổng quan về PHP

---

- **PHP** - **Personal Home Page** hay **PHP Hypertext Preprocessor**, là một ngôn ngữ lập trình kịch bản được chạy ở phía Server nhằm sinh ra mã HTML trên Client.
- **PHP** đã trải qua rất nhiều phiên bản và được tối ưu hóa cho các ứng dụng Web, với cách viết mã rõ ràng, tốc độ nhanh, dễ học nên **PHP** đã trở thành một ngôn ngữ lập trình Web rất phổ biến và được ưa chuộng.
- Ngôn ngữ, các thư viện, tài liệu gốc của **PHP** được xây dựng bởi cộng đồng và có sự đóng góp rất lớn của **Zend Inc.** nên đã tạo ra một môi trường chuyên nghiệp cho phép phát triển ứng dụng ở quy mô lớn.



# Tổng quan về PHP – Lịch sử

---

- **PHP** được phát triển từ một sản phẩm có tên là **PHP/FI - Personal Home Page Forms Interpreter** do Rasmus Lerdorf tạo ra năm 1994 với các chức năng rộng hơn, Rasmus đã viết ra một bộ thực thi bằng C để có thể truy vấn tới CSDL và giúp cho người sử dụng phát triển các ứng dụng Web đơn giản.
- **1997: PHP/FI 2.0**, lần viết lại của phiên bản C, đã thu hút được hàng ngàn người sử dụng trên toàn thế giới với xấp xỉ 50.000 tên miền đã được ghi nhận là có cài đặt nó, chiếm khoảng 1% số tên miền có trên mạng Internet.



# Tổng quan về PHP – Lịch sử

---

- **1997: PHP 3.0 - PHP: Hypertext Preprocessor**, được Andi Gutmans và Zeev Suraski đưa ra sau khi viết lại hoàn toàn bộ mã nguồn trước đó nhằm hỗ trợ cộng đồng phát triển các ứng dụng thương mại điện tử.
- **PHP 3.0** đã được cài đặt cho xấp xỉ 10% số máy chủ Web có trên mạng Internet thời kỳ này.
- **2000: PHP 4.0** – được viết lại các Engine nhằm tăng tốc độ xử lý cũng như tích hợp các thư viện, công cụ được đưa ra và chiếm 20% số máy chủ Web.
- **2005: PHP 5.0** – hỗ trợ lập trình hướng đối tượng, tích hợp các xử lý XML, nâng cấp Zend Engine.
- Hiện nay, đã có các phiên bản 6.0 và 7.0, 8.0



# Tổng quan về PHP – Đặc điểm

- **PHP** – là ngôn ngữ kịch bản (Scripting Language), một nhánh của ngôn ngữ lập trình mà trong đó tập tin chứa mã lệnh viết bằng ngôn ngữ kịch bản có thể được chạy (thực thi) trực tiếp trên máy mà không cần phải chuyển sang một định dạng khác thông qua các Engine.
  - PHP Engine phân tích các đoạn mã nằm trong những dấu giới hạn của nó. Bất cứ mã nào nằm ngoài những dấu giới hạn đều được xuất ra trực tiếp không thông qua xử lý.
    - `<?php ... ?>`, `<?....?>`
    - `<script language="php"> ... </script>`
- **PHP** – dựa trên các quy tắc của C và Java để xây dựng nên quy tắc của ngôn ngữ ➔ đơn giản và dễ học.

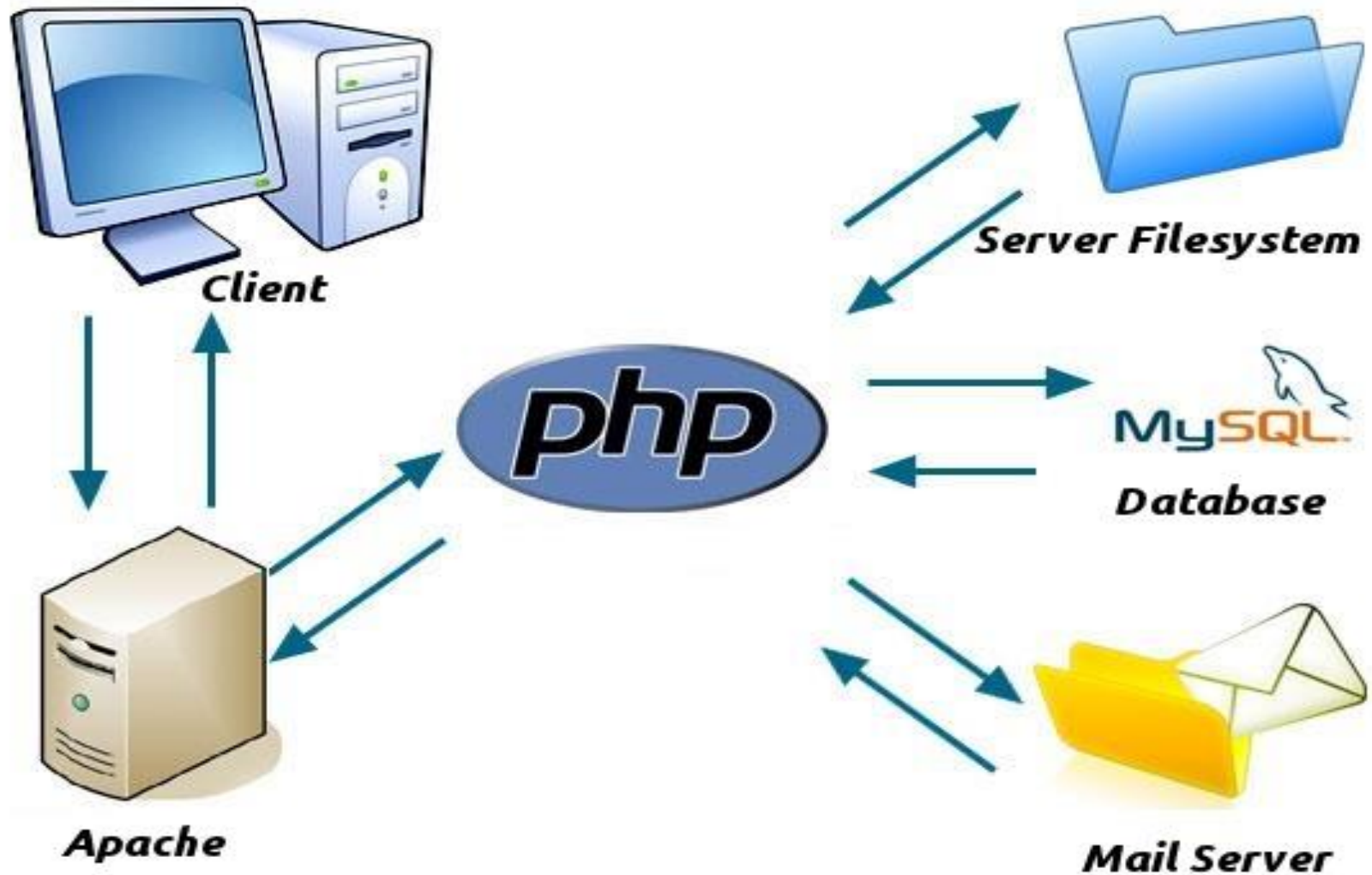


# Tổng quan về PHP – Đặc điểm

---

- **PHP** – là ngôn ngữ lập trình mã nguồn mở và điều này có nghĩa là bạn có thể sử dụng nó hoàn toàn miễn phí.
- **PHP** – hỗ trợ bởi cộng đồng có thể chạy trên nhiều nền tảng hệ điều hành khác nhau như Windows, Mac OS và Linux.
- **PHP** – được hỗ trợ bởi một cộng đồng lớn với đầy đủ tài liệu để tham khảo và phát triển các ứng dụng Web Server thương mại.
  - <https://www.w3schools.com/php/default.asp>
  - <https://www.tutorialspoint.com/php/index.htm>
  - PHP Notes for Professionals - <https://goalkicker.com/>

# Web Server với PHP





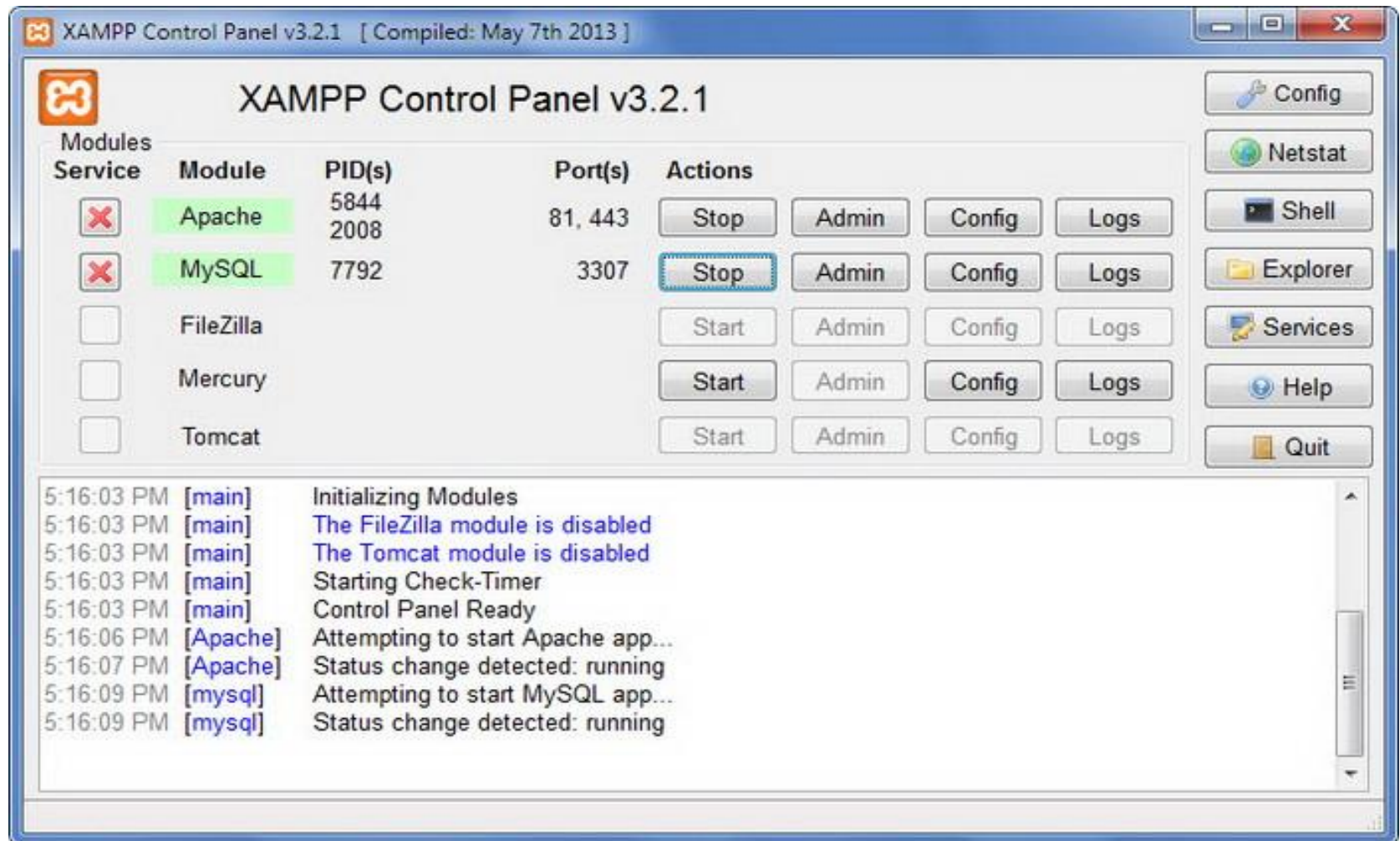
# Cài đặt và cấu hình PHP

---

- Cài đặt XAMPP:
  - Language: PHP
  - Database: MySQL
  - Web Server: Apache
- Cài đặt Notepad++ hoặc Sublime Text, Visual Code, Netbeans để hỗ trợ soạn thảo các trang php.
- Apache:
  - Các trang .php được đặt trong thư mục Web root hoặc thư mục con của Web root.
  - Truy cập thông qua Web Server – Apache thông qua HTML.
  - Có thể thay đổi các cấu hình thông qua XAMPP Control hoặc từ các file cấu hình:
    - Root
    - Database ...



# XAMPP





# Ví dụ về PHP

```
1 <html>
2   <head>
3     <title>Hello World</title>
4   </head>
5   <body>
6     <?php echo "Hello, World!";?>
7   </body>
8 </html>
```

<?php

echo "Chao mung ban den voi PHP";


?>



# CHUYÊN ĐỀ - Công nghệ phần mềm

---

PHP – Cơ sở của ngôn ngữ





# PHP – Các khái niệm cơ bản

## ■ Chú thích trong PHP

- Chú thích là những đoạn mã mà khi trình dịch gặp phải chúng thì trình dịch sẽ bỏ qua.
- Chú thích thường được dùng để ghi nhận lại những lời: giải thích, lưu ý, nhắc nhở, nhận xét , .... Ngoài ra, nó còn được dùng để vô hiệu hóa những đoạn mã với mục đích phục vụ cho việc kiểm tra sự thực thi của các đoạn mã khác
- Cách ghi chú thích trong PHP
  - Đặt cặp dấu gạch chéo // phía trước nội dung chú thích.
  - Dùng cặp dấu /\* \*/ để bao bọc nội dung chú thích.
- Ví dụ:
  - ```
/* Tài liệu học HTML  
   Tài liệu học CSS  
   Tài liệu học JavaScript */  
//Tài liệu học lập trình PHP  
echo "<p>Lập Trình Web 01</p>";
```



# PHP – Các khái niệm cơ bản

---

## ■ Từ khóa trong PHP

- Từ khóa là các từ được sử dụng bởi ngôn ngữ lập trình với một chức năng cụ thể nào đó.
- Trong PHP, các từ khóa được sử dụng:
  - Các từ khóa kiểu dữ liệu: Integer, Boolean, String, Float/Double, Object, Null, Resource, ...
  - Các từ khóa logic: true, false, and, or ...
  - Các từ khóa điều khiển: if, else, elseif, for, foreach, while, do ...
  - Các từ khóa thực thi: return, exit(), break, goto ...
  - Các từ khóa hướng đối tượng: class, interface, abstract, private, protected, public, extends, implements, new ...
  - Các từ khóa kiểu dữ liệu: array(), list(), ...
  - Các từ khóa kiểm soát lỗi: try, catch, finally, throw, ...
  - ...



# PHP – Các khái niệm cơ bản

## ■ Biến

- Phân biệt chữ hoa, chữ thường.
- Bắt đầu bằng dấu \$, tiếp ngay sau \$ là tên biến.
- Tên biến bao gồm chữ cái, chữ số, dấu gạch nối (\_) và phải bắt đầu bằng chữ cái hoặc dấu gạch nối.
- Ví dụ:
  - `$a, $b, $my_varibale ...`
- Biến không cần khai báo kiểu dữ liệu và được tự động xác định kiểu dữ liệu cho lần gán giá trị đầu tiên.
- Việc sử dụng biến chưa khởi tạo có thể gây lỗi.
- PHP hỗ trợ kiểu biến động – biến của biến
  - `$a = "hello";`
  - `$$a = "world";`
  - `echo "{$a} {$$a}" → hello world`



# PHP – Các khái niệm cơ bản

## ■ Hằng

- Đại lượng không đổi, thường là hằng toàn cục.
- Được khai báo thông qua hàm define()
- Cú pháp:
  - `bool define (name, value, case-insensitive)`
- Trả về true nếu thành công, ngược lại là false
- VD:

```
<?php
```

```
define ("COUNTRY", "Viet Nam");
```

```
echo COUNTRY; ➔ Viet Nam
```

```
echo country; ➔ Lỗi: không có hằng định nghĩa
```

```
define ("CITY", "Ha Noi", true);
```

```
echo city; ➔ Ha Noi
```

```
?>
```



# PHP – Các khái niệm cơ bản

---

## ■ Kiểu dữ liệu

- PHP hỗ trợ 8 kiểu dữ liệu cơ bản
  - Integer
  - Float/Double
  - Boolean
  - String
  - Array
  - Object
  - Resource
  - NULL
- PHP không yêu cầu khai báo kiểu dữ liệu, nó sẽ được tự động xác định sau lệnh gán đầu tiên.
- PHP không hỗ trợ tự động ép kiểu dữ liệu, lập trình viên phải tự ép kiểu dữ liệu (giống C, Java)





# PHP – Các khái niệm cơ bản

---

## ■ Kiểu dữ liệu – Integer

- Miền xác định:  $-2^{31}$  đến  $2^{31}-1$
- Hỗ trợ các hệ cơ số khác nhau
  - Hệ 10 – Dec: `$x = 16`
  - Hệ 16 – Hex: `$a = 0xFF`
  - Hệ 8 – Oct: `$a = 020`
- Hàm `var_dump()` cho phép xác định kiểu dữ liệu và giá trị của nó.
  - `$a = 0xFF;`
  - `var_dump($a);`  
Kết quả: `int(255)`
- Một số hàm xác định kiểu dữ liệu:
  - `gettype($a)`
  - `is_bool($b) ...`



# PHP – Các khái niệm cơ bản

---

## ■ Kiểu dữ liệu – Float/Double

- Miền xác định: 1.7E-308 đến 1.7E+308
  - Hỗ trợ hai cách biểu diễn khác nhau
    - Cơ bản: `$x = 16.019`
    - Khoa học: `$a = 17.0E+2`
    - Hệ 8 – Oct: `$a = 020`
  - Ví dụ
    - `$a = 17.01E+02;`
    - `var_dump($a);`
- Kết quả: `float(1700)`



# PHP – Các khái niệm cơ bản

---

- **Kiểu dữ liệu – Boolean**

- Miền xác định: true hoặc false
- Hỗ trợ true/TRUE và false/FALSE
- Ví dụ
  - `$a = true;`
  - `var_dump($a);`Kết quả: `bool(true)`



# PHP – Các khái niệm cơ bản

## ■ Kiểu dữ liệu – String

- Biểu diễn chuỗi ký tự.
- Được giới hạn bởi nháy đơn (') hoặc nháy kép (")
- Chuỗi trong nháy kép cho phép thay thế và xử lý ký tự thoát (/) trong khi trong nháy đơn thì không xử lý.
- Các ký tự đặc biệt phải dùng đến ký tự thoát để PHP phân biệt: \*, \, \$, ...
  - \n: xuống dòng
  - \t: dấu tab
  - \r: trở về đầu dòng
- Ví dụ
  - `$a = "Hello";`
  - `$b = "$a world"; // $b="Hello world"`
  - `$c = '$a world'; // $c='$a world' (không thay đổi)`



# PHP – Các khái niệm cơ bản

---

## ■ Kiểu dữ liệu – String

- 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.



# PHP – Các khái niệm cơ bản

## ■ Kiểu dữ liệu – String – Chú ý

- Các ký tự đặc biệt: “ ‘ \$ , xuống dòng... : Sử dụng dấu \ ở trước (tùy thuộc vào việc chúng ta dùng kiểu chuỗi ra sao).
- Cách biểu diễn chuỗi:
  - Đặt chuỗi trong cặp dấu nháy kép
    - `$s="thu mot\" ti";`
    - `$s="My name's Sinh";`
  - Đặt chuỗi trong cặp dấu nháy đơn
    - `$s='thu mot "ti";`
    - `$s='My name\'s Sinh';`



# PHP – Các khái niệm cơ bản

---

## ■ Kiểu dữ liệu – String – Đưa biến vào chuỗi

### ■ Chỉ có 2 dạng chuỗi chấp nhận cách này:

- Chuỗi viết theo dạng cặp nháy kép
- Chuỗi dạng heredoc

### ■ Ví dụ

- `$x = "Hello";`
- `$a = "Say {$x} everybody!";`
- `echo $a;`

### ■ Nối chuỗi

- `$a = 1;`
- `$x = "hello"`
- `$b = $a." - Say ".$x." everybody!";`
- `echo $a;`



# PHP – Các khái niệm cơ bản

## ■ Kiểu dữ liệu – Mảng

- Biểu diễn tập dữ liệu có thể khác kiểu dữ liệu và được đánh chỉ số bằng số hoặc bằng chuỗi.
- Truy cập các phần tử thông qua chỉ số. Chỉ số mặc định được xác định từ 0.
- Khởi tạo thông qua hàm `array()`

### ■ Ví dụ

- `$a = array(1, 2, 5, 9, 10);`  
`echo $a[2]; ➔ 5`
- `$b = array("My", "name", "is", "Phuc");`  
`echo $b[1]; ➔ name`
- `$c = array("Peter"=>35, "Henry"=>15, "Ben"=>19);`  
`echo $c["Peter"]; ➔ 35`
- `$d = array(1, 11.10, "String", true);`  
`echo $d[2]; ➔ String`





# PHP – Các khái niệm cơ bản

## ■ Kiểu dữ liệu – Mảng

- PHP hỗ trợ cho phép thay đổi chỉ số đầu mặc định.

- Ví dụ

- `$a = array(10=>1, 2, 5, 9, 10);`  
`echo $a[0];` → Lỗi, không có phần tử chỉ số này  
`echo $a[10];` → 1

- Mảng nhiều chiều

- `$a = array(  
    array("TT", "Họ tên"),  
    array(1, "Nguyễn Văn An"),  
    array(2, "Trần Trọng Bình"),  
);`  
`echo $a[1][1];` → Nguyễn Văn An



# PHP – Các khái niệm cơ bản

## ■ Toán tử trong PHP

### ■ Kiểu số:

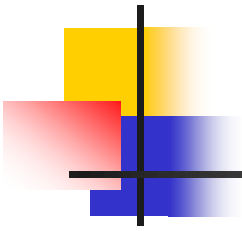
- Kết hợp: +, -, \*, /, %
- Tăng giảm: ++, --

VD: \$a++; \$a--; ++\$a; --\$a;

- Chuỗi: Toán tử chấm (.)
- Logic: AND (&&), OR (||), XOR, !
- Gán: =, +=, -=, .=
- So sánh: ==, ===, !=, !==, <, >, <=, >=

## ■ **PHP:** cung cấp các hàm toán học, hàm xử lý chuỗi, hàng xử lý mảng cho phép các lập trình viên sử dụng.

- <http://webcoban.vn/php/default.html>



# PHP – Các cấu trúc điều khiển





# PHP – Các cấu trúc điều khiển

## ■ Câu lệnh rẽ nhánh **if**

### ■ Cấu trúc:

```
■ if (condition){
    statements;
}
■ if (condition){
    statements;
} else {
    statements;
}
■ if (condition){
    statements;
} elseif (condition) {
    statements;
} else {
    statements;
}
```

## ■ Câu lệnh rẽ nhánh **switch**

### ■ Cấu trúc:

```
■ switch (biến){
    case label1:
        statements;
        break;
    case label2:
        statements;
        break;
    ...
    default:
        statements;
        break;
}
```



# PHP – Các cấu trúc điều khiển

## ■ while

### ■ Cấu trúc:

```
■ while(condition) {  
    statements;  
}
```

### ■ Ví dụ:

```
■ $x = 1;  
  while($x < 5) {  
      echo $x."\t";  
      $x++;  
  }
```

➔ 1 2 3 4

## ■ do ... while

### ■ Cấu trúc:

```
■ do {  
    statements;  
} while (condition);
```

### ■ Ví dụ:

```
■ $x = 5;  
  do {  
      echo $x."\t";  
      $x--;  
  } while ($x > 0);
```

➔ 5 4 3 2 1



# PHP – Các cấu trúc điều khiển

## ■ **for**

### ■ Cấu trúc:

```
■ for (initiation counter; condition; increment
      counter) {
    statements;
}
```

### ■ Ví dụ:

```
■ for ($x = 1; $x < 10; $x++) {
    echo $x."\t";
}
```

➔ 1 2 3 4 5 6 7 8 9



# PHP – Các cấu trúc điều khiển

## ■ foreach

- Được sử dụng khi duyệt mảng.

- Cấu trúc:

- ```
foreach ($array as $e_array) {  
    statements;  
}
```
  - ```
foreach ($array as $e_key->$e_value) {  
    statements;  
}
```

- Ví dụ:

- ```
$x = array(1, 3, 5, 9);  
foreach ($x as $e_x) {  
    echo $e_x."\t";  
}
```

➔ 1 3 5 9



# PHP – Các cấu trúc điều khiển

---

## ■ **break, continue, return**

### ■ **break:**

- Kết thúc khối lệnh for, foreach, while, do-while hoặc switch

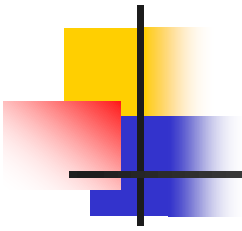
### ■ **continue:**

- Bỏ qua phần sau continue trong vòng lặp, chuyển sang vòng lặp tiếp theo.

### ■ **return:**

- Thoát khỏi hàm và trả về kết quả.





# PHP – Hàm và cơ chế sử dụng





# PHP – Hàm và cơ chế sử dụng

---

## ■ Hàm – function

- Hàm trong PHP cũng tương tự như khái niệm hàm trong các ngôn ngữ lập trình khác: tập hợp của một số các câu lệnh nhằm thực hiện một chức năng nào đó.
- Hàm nhằm mục đích tái sử dụng trong chương trình – không cần phải viết lại.
- Cấu trúc chung của một hàm bao gồm:
  - Tên hàm
  - Tham số đầu vào
  - Kết quả trả về

### Chú ý:

- PHP không yêu cầu khai báo kiểu dữ liệu nên trong định nghĩa hàm không cần có kiểu dữ liệu trả về.
- Tùy thuộc vào kết quả trả về thông qua return, hàm sẽ có kiểu dữ liệu tương ứng.



# PHP – Hàm và cơ chế sử dụng

## ■ Khai báo hàm

### ■ Đối với hàm không có tham số

```
■ function ten_ham() {  
    statements;  
    return $result;  
}
```

### ■ Đối với hàm có tham số

```
■ function ten_ham(ts_1, ts_2, ..., ts_n) {  
    statements;  
    return $result;  
}
```

### Chú ý:

#### ■ Để thoát khỏi hàm có thể sử dụng câu lệnh:

```
return;  
exit;
```



# PHP – Hàm và cơ chế sử dụng

## ■ Lời gọi hàm

- Giống như các ngôn ngữ lập trình khác, hàm được gọi thông qua tên và giá trị của các tham số của hàm
  - `ten_ham(gt_ts1, gt_ts2, ..., gt_ts_n);`
- Đối với hàm có giá trị trả về, có thể sử dụng biến để nhận kết quả trả về
  - `$result = ten_ham(gt_ts_1, gt_ts_2, ..., gt_ts_n);`

## Chú ý:

- Trong PHP hàm được xác định thông qua tên vì vậy không hỗ trợ hàm có cùng tên, khác tham số.
- Trong PHP hỗ trợ > 10.000 mặc định (build-in function)  
[https://www.tutorialspoint.com/php/php\\_function\\_reference.htm](https://www.tutorialspoint.com/php/php_function_reference.htm)
- Hàm có thể được khai báo giá trị mặc định cho các tham số.

```
function ten_ham(ts_1=1, ts_2='A', ..., ts_n)
```



# PHP – Hàm và cơ chế sử dụng

## ■ Ví dụ về cơ chế khai báo và sử dụng hàm

```
<?php
```

```
function addFunction($num1, $num2) {  
    $sum = $num1 + $num2;  
    return $sum;  
}  
$result = addFunction(10, 20);  
echo    "Returned value from the function :  
        $ result \n";  
$result = addFunction($result, $result);  
echo    "Returned value from the function :  
        $result \n";
```

```
?>
```

➔ Kết quả trả về:

```
Returned value from the function :30
```

```
Returned value from the function :60
```



# PHP – Hàm và cơ chế sử dụng

## ■ Cơ chế truyền tham số – tham trị

- Trong hàm PHP, khi sử dụng cơ chế truyền tham số tham trị thì giá trị của biến truyền vào sẽ không thay đổi sau khi ra khỏi lời gọi hàm.

```
<?php

function change_value($c_input){
    $c_input = $c_input + 10;
    echo "Modified value : {$c_input} \n";
}

$c_input = 10;
change_value($c_input);
echo "Original value : {$c_input} \n";

?>
```

➔ Kết quả trả về:

Modified value: 20

Original value: 10



# PHP – Hàm và cơ chế sử dụng

## ■ Cơ chế truyền tham số – tham chiếu

- Trong hàm PHP, khi sử dụng cơ chế truyền tham số tham chiếu thì giá trị của biến truyền vào sẽ thay đổi sau khi ra khỏi lời gọi hàm.

```
<?php  
    function change_value(&$c_input){  
        $c_input = $c_input + 10;  
        echo "Modified value : {$c_input} \n";  
    }  
    $c_input = 10;  
    change_value($c_input);  
    echo "Original value : {$c_input} \n";  
?>
```

➔ Kết quả trả về:

```
Modified value: 20  
Original value: 20
```



# PHP – Hàm và cơ chế sử dụng

## ■ Phạm vi hoạt động của biến trong PHP

- Phạm vi biến là phạm vi ở đó biến được 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 tệp **trừ ở trong hàm**.
    - Để đưa 1 biến toàn cục vào trong hàm, sử dụng khai báo  
`global $biến1, $biến2, ...;`  
`$GLOBALS`
  - Super global variables – biến siêu toàn cục:
    - Một số biến có sẵn của PHP: `$_SERVER`, `$GLOBALS`...





# PHP – Hàm và cơ chế sử dụng

## ■ Biến tĩnh

- Biến tĩnh được định nghĩa trong hàm và được sử dụng trong PHP để cho phép lưu giá trị thông qua nhiều lần gọi hàm.
- Biến tĩnh được khởi tạo ở lần khai báo đầu tiên.
- Khai báo biến tĩnh

```
static $COUNT = 0;
```

## ■ Ví dụ

```
function testFunction() {  
    static $COUNT = 0;  
    $COUNT++;  
    echo "Count: $COUNT \n";  
}  
testFunction();  
testFunction();
```

Kết quả: ?



# PHP – Kỹ thuật chèn file

---

- Dùng để import một file khác vào file hiện tại, lúc này file hiện tại có thể sử dụng mọi tài nguyên của file import
- **Các hàm**
  - `include("URL đến file")`
  - `require("URL Đến file")`
  - `include_once("URL Đến file")`
  - `require_once ("URL Đến file")`

## **Sự khác nhau giữa require và include:**

- Khi import một file bằng lệnh `require` thì nếu chương trình bị lỗi thì lập tức trình biên dịch sẽ dừng và xuất ra thông báo lỗi.
- Nếu sử dụng lệnh `include` thì đó chỉ là một cảnh báo nên chương trình vẫn chạy cho đến cuối chương trình.



# PHP – Điều hướng trang

---

- Dùng hàm header để điều hướng đến trang khác
  - Cú pháp và ví dụ
    - `header ($string, $replace = true, $http_response_code = null) {}`
    - `header('Location: http://www.freetuts.net/');`
    - `header('Location: http://www.freetuts.net/', true, 301);`
    - `header('Location: http://www.freetuts.net/', true, 404);`
- code là 301 là code khai báo chuyển hướng có điều kiện. Ví dụ khi thay đổi domain cho website thì khi người dùng vào domain cũ sẽ được chuyển hướng sang domain mới
- code 404: muốn chuyển hướng page not found



# PHP – Điều hướng trang

---

- Dùng hàm header để điều hướng đến trang khác
  - Cú pháp và ví dụ
    - `header ($string, $replace = true, $http_response_code = null) {}`
    - `header('Location: http://www.freetuts.net/');`
    - `header('Location: http://www.freetuts.net/', true, 301);`
    - `header('Location: http://www.freetuts.net/', true, 404);`
- code là 301 là code khai báo chuyển hướng có điều kiện. Ví dụ khi thay đổi domain cho website thì khi người dùng vào domain cũ sẽ được chuyển hướng sang domain mới
- code 404: muốn chuyển hướng page not found