
Bài 1

Tổng quan về Ứng dụng Web

Module: **ADVANCED PROGRAMMING WITH PHP 2.0**

Mục tiêu



- Trình bày được nội dung, yêu cầu, lịch trình và kết quả của môn học APP-PHP
- Trình bày được các thành phần của một hệ thống web
- Tạo được ứng dụng PHP chạy trên Apache Web Server
- Xử lý dữ liệu form trong PHP với các phương thức GET và POST
- Sử dụng được cú pháp PHP để thao tác với biến
- Sử dụng được cú pháp PHP để thao tác với cấu trúc điều kiện
- Sử dụng được cú pháp PHP để thao tác với cấu trúc lặp



ADVANCED PROGRAMMING WITH PHP 2.0

ADVANCED PROGRAMMING WITH PHP

2.0



- **Mục đích:** Kết thúc module học viên thành thạo được các kỹ thuật lập trình với ngôn ngữ PHP và xây dựng được các ứng dụng theo mô hình Lập trình Hướng đối tượng, thiết kế được website, xây dựng được các ứng dụng web sử dụng ngôn ngữ lập trình PHP. Kết thúc module, học viên đủ năng lực để học thêm các web framework của ngôn ngữ PHP.
- **Thời gian:** 24 bài
- **Đánh giá:**
 - Thi thực hành và lý thuyết cuối module, điểm đạt: 75%
 - Bảng đánh giá kỹ năng theo chuẩn đầu ra
- **Yêu cầu:**
 - Phần mềm PHPStorm

ADVANCED PROGRAMMING WITH PHP

2.0



- **Tài liệu học tập:**
- CodeGymX: [BC-PHP-APP] Advanced Programming with PHP 2.0
 - Source code mẫu trên kênh Github của CodeGym
 - Các tài liệu tham chiếu bên ngoài
- **Tài liệu tham khảo:**
 - Sách Joel Murach & Ray Harris - Lập trình cơ bản PHP và MySQL
 - Sách Joel Murach & Ray Harris - Lập trình nâng cao PHP và MySQL
 - Trang web Laravel.com
 - [ebook] PHP Advanced and Object-Oriented Programming

Thảo luận

Web và các giao thức

Static website

Dynamic website

Web và Website

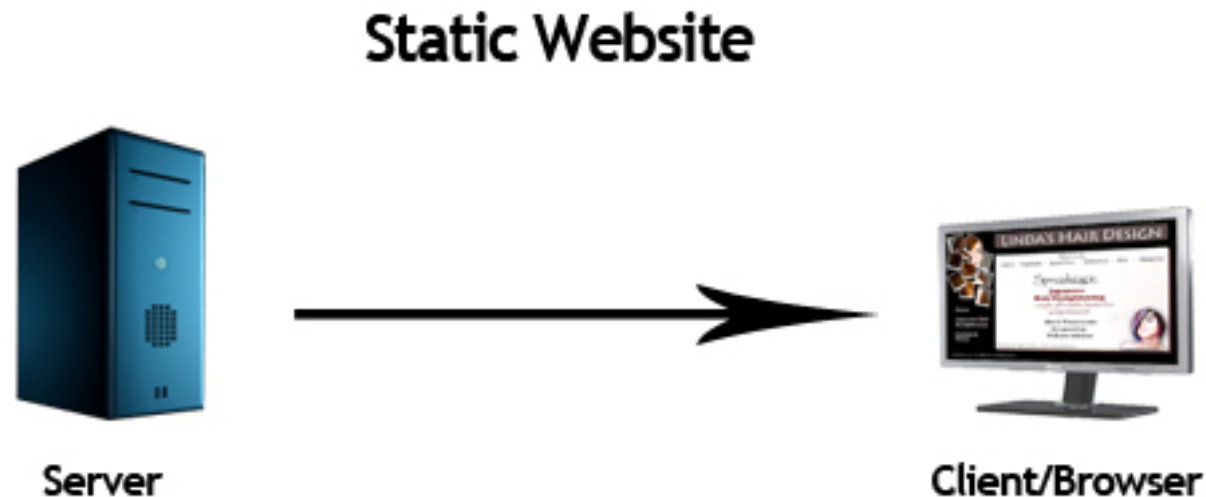


- Web là một không gian trên Internet mà ở đó các tài liệu và các tài nguyên khác được chia sẻ dưới dạng các URL
- Website là tập hợp các trang tài liệu và tài nguyên có liên quan đến nhau, thường được truy cập thông qua một tên miền
- Các trang tài liệu có thể chứa văn bản, hình ảnh, âm thanh, video...
- Mỗi website có địa chỉ cụ thể trên internet gọi là URL
- Website được lưu trữ trên một hoặc nhiều máy chủ (Server)
- Website được quản lý bởi cá nhân, công ty hoặc một tổ chức
- Có hai loại website cơ bản là web động (dynamic web) và web tĩnh (static web)

Static Web



- Website tĩnh là các trang web mà nội dung của nó không thay đổi
- Những trang web trong website tĩnh được viết bằng mã HTML hoặc thêm CSS, JavaScript để thêm các hiệu ứng
- Không cần sử dụng các ngôn ngữ lập trình web như Java, PHP, ... hoặc thiết kế cơ sở dữ liệu.

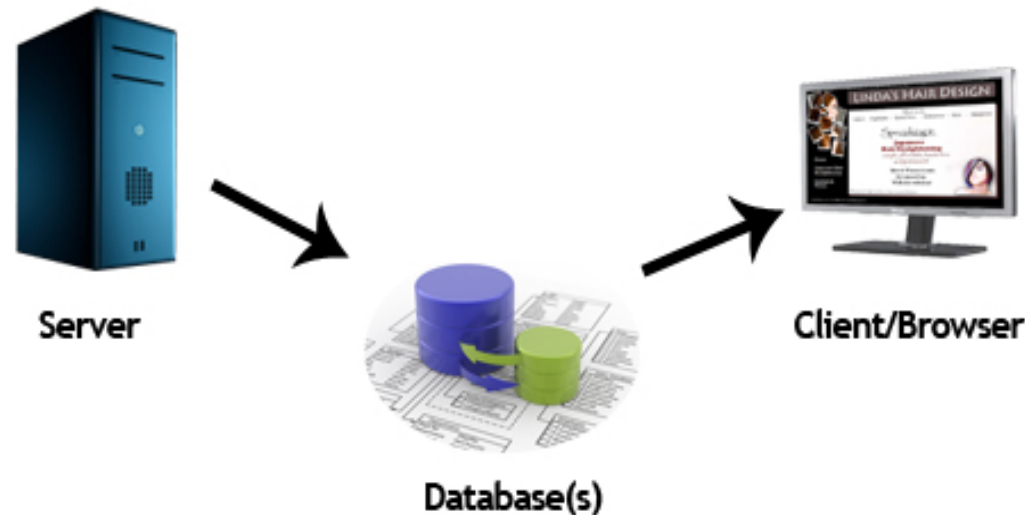


Dynamic Web



- Website động là website có nội dung thay đổi
- Sự thay đổi có thể là tùy theo thời gian, tùy theo người dùng, tùy theo ngữ cảnh
- Thường sử dụng đến các ngôn ngữ phía server (server-side), chẳng hạn như Java, PHP, Python, C#... và cơ sở dữ liệu

Dynamic Website





Protocol



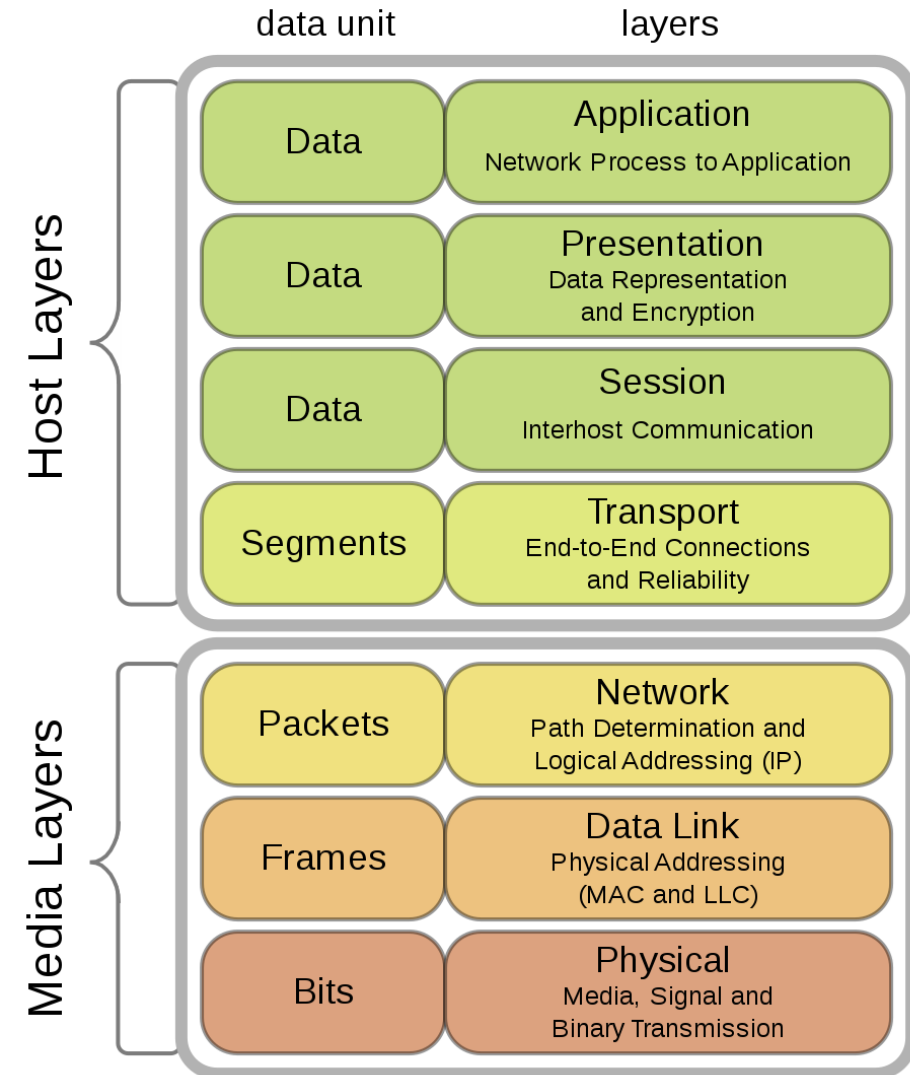
Giao thức (Protocol)

- Tập các quy ước/chuẩn mực
- Cách thức giao tiếp giữa các bên
- Ví dụ: HTTP, FTP, SMTP...
- Giao thức này có thể được xây dựng dựa trên giao thức khác
- Nếu không tuân thủ giao thức thì các bên sẽ không “hiểu” được nhau

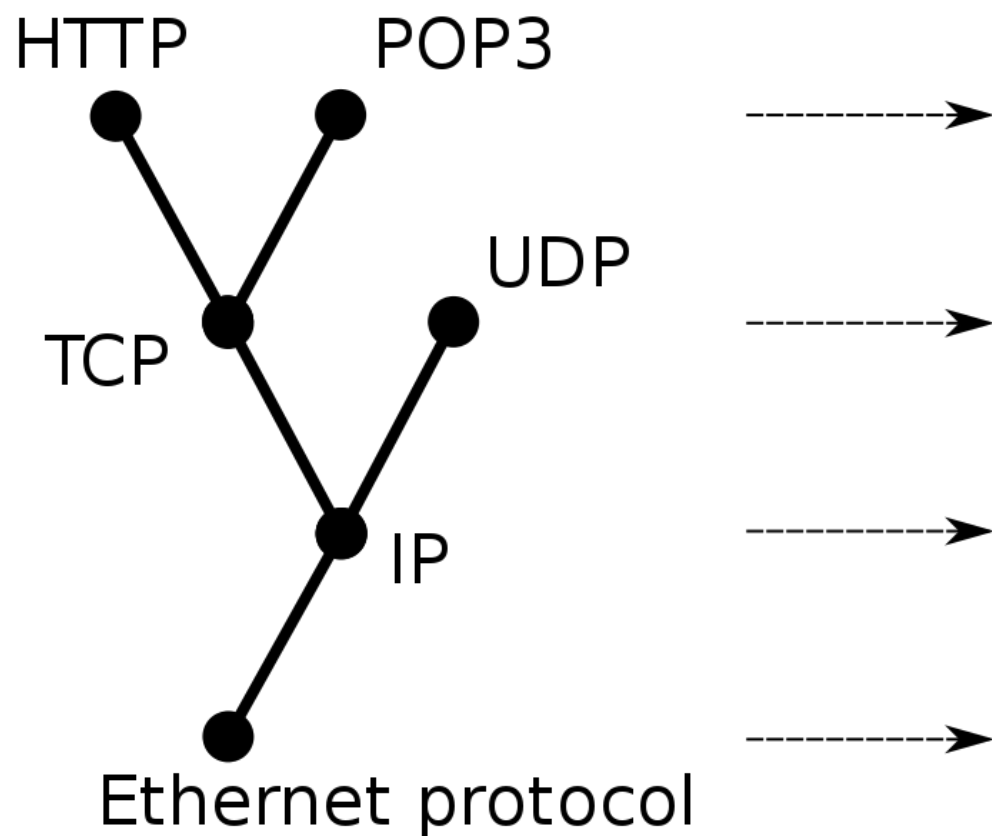
Tầng cấp của mẫu hình OSI



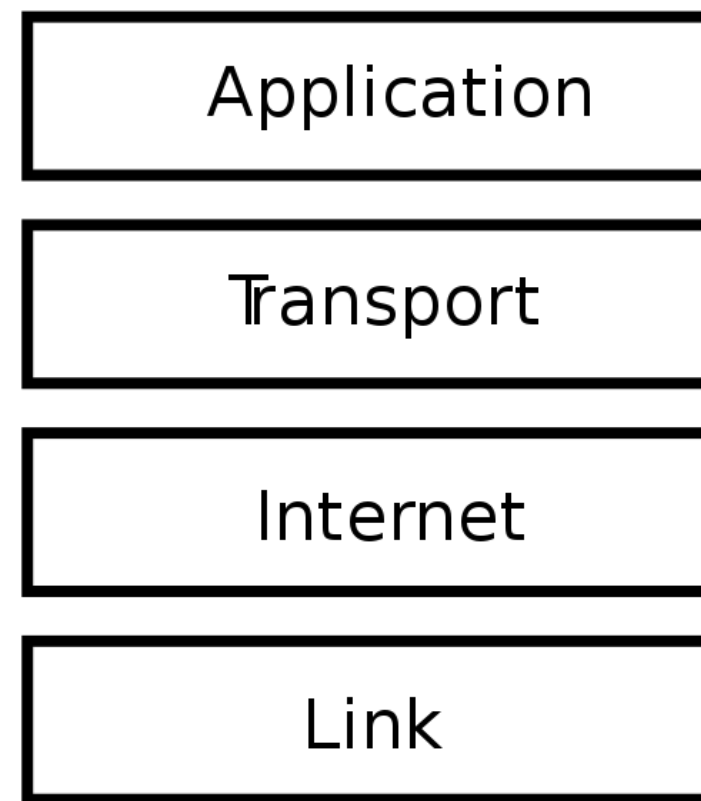
- Tầng 1: Tầng vật lý (Physical Layer)
- Tầng 2: Tầng liên kết dữ liệu (Data-Link Layer)
- Tầng 3: Tầng mạng (Network Layer)
- Tầng 4: Tầng giao vận (Transport Layer)
- Tầng 5: Tầng phiên (Session layer)
- Tầng 6: Tầng trình diễn (Presentation layer)
- Tầng 7: Tầng ứng dụng (Application layer)



Mô hình TCP/IP



TCP/IP - model





Thảo luận

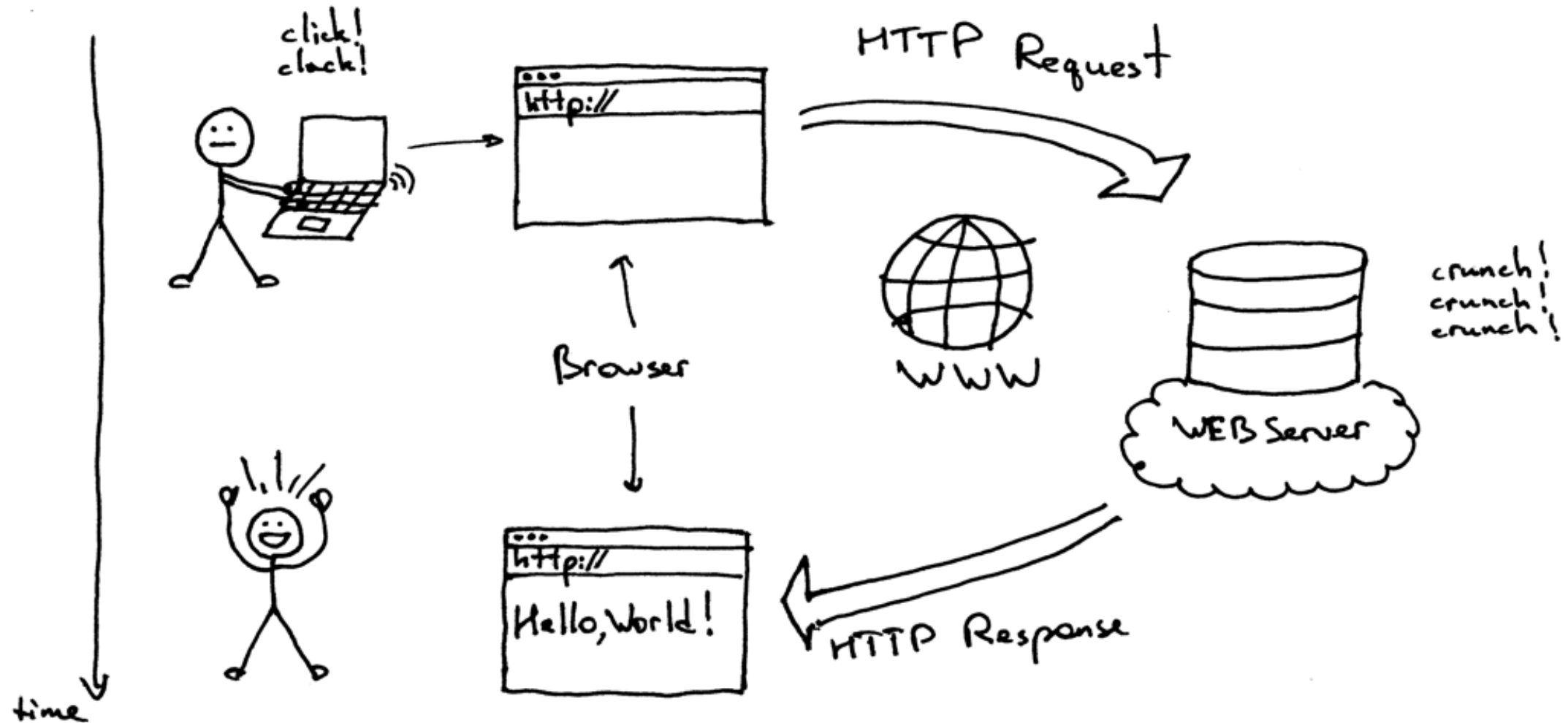
Giao thức HTTP

Giao thức HTTP



- HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản
- Là một trong năm giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client)
- Là giao thức Client/Server dùng cho World Wide Web-WWW
- HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP
- Tác giả của HTTP là Tim Berners-Lee

Mô hình Client - Server





HTTP Status Code

- Mã trạng thái HTTP được server phản hồi lại mỗi khi nhận được http request
- Status-Code là một số nguyên bao gồm 3 ký tự
 - Ký tự đầu tiên phân loại trạng thái
- Ký tự đầu gồm có: 1,2,3,4,5
 - 1xx: Thông tin
 - 2xx: Thành công
 - 3xx: Chuyển hướng
 - 4xx: Lỗi ở phía client
 - 5xx: Lỗi ở phía server

HTTP Status Code



1xx: Thông tin

100 (*Continue*)

101 (Switching protocols)

2xx: Thành công

200 (Successful)

201 (Created)

202 (Accepted)

203 (Non-authoritative information)

204 (No content)

205 (Reset content)

206 (Partial content)

3xx: Sự điều hướng lại

301 (Moved permanently)

302 (Moved temporarily)

304 (Not modified)

4xx: Lỗi Client

400 (Bad request)

401 (Not authorized)

403 (Forbidden)

404 (Not found)

405 (Method not allowed)

406 (Not acceptable)

407 (Proxy authentication required)

408 (Request timeout)

409 (Conflict)

410 (Gone)

411 (Length required)

412 (Precondition failed)

413 (Request entity too large)

414 (Requested URI is too long)

416 (Requested range not satisfiable)

417 (Expectation failed)

5xx: Lỗi Server

500 (Internal server error)

501 (Not implemented)

502 (Bad gateway)

503 (Service unavailable)

504 (Gateway timeout)

505 (HTTP version not supported)

HTTP Status Code



HTTP Status Codes

Level 200 (Success)

200 : OK

201 : Created

203 : Non-Authoritative
Information

204 : No Content

Level 400

400 : Bad Request

401 : Unauthorized

403 : Forbidden

404 : Not Found

409 : Conflict

Level 500

500 : Internal Server Error

503 : Service Unavailable

501 : Not Implemented

504 : Gateway Timeout

599 : Network timeout

502 : Bad Gateway

HTTP Request



```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
licenseID=string&content=string&/paramsXML=string
```

HTTP Response



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

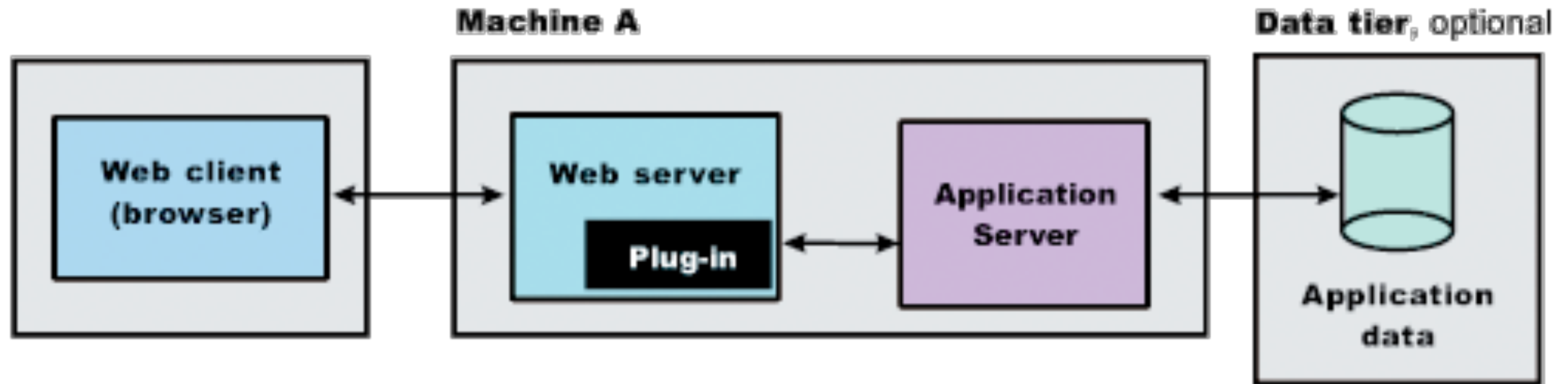
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

Web Server



- Web Server là một phần mềm/phần cứng để cung cấp các tài nguyên web
- Web Server tiếp nhận các request và trả về response
- Web Server sử dụng giao thức HTTP và một số giao thức liên quan khác
- Web Server có thể hoạt động độc lập hoặc có thể kết hợp với các phần mềm khác, chẳng hạn như phần mềm cân bằng tải, phần mềm bảo mật, phần mềm cache...

Hoạt động của Web Server

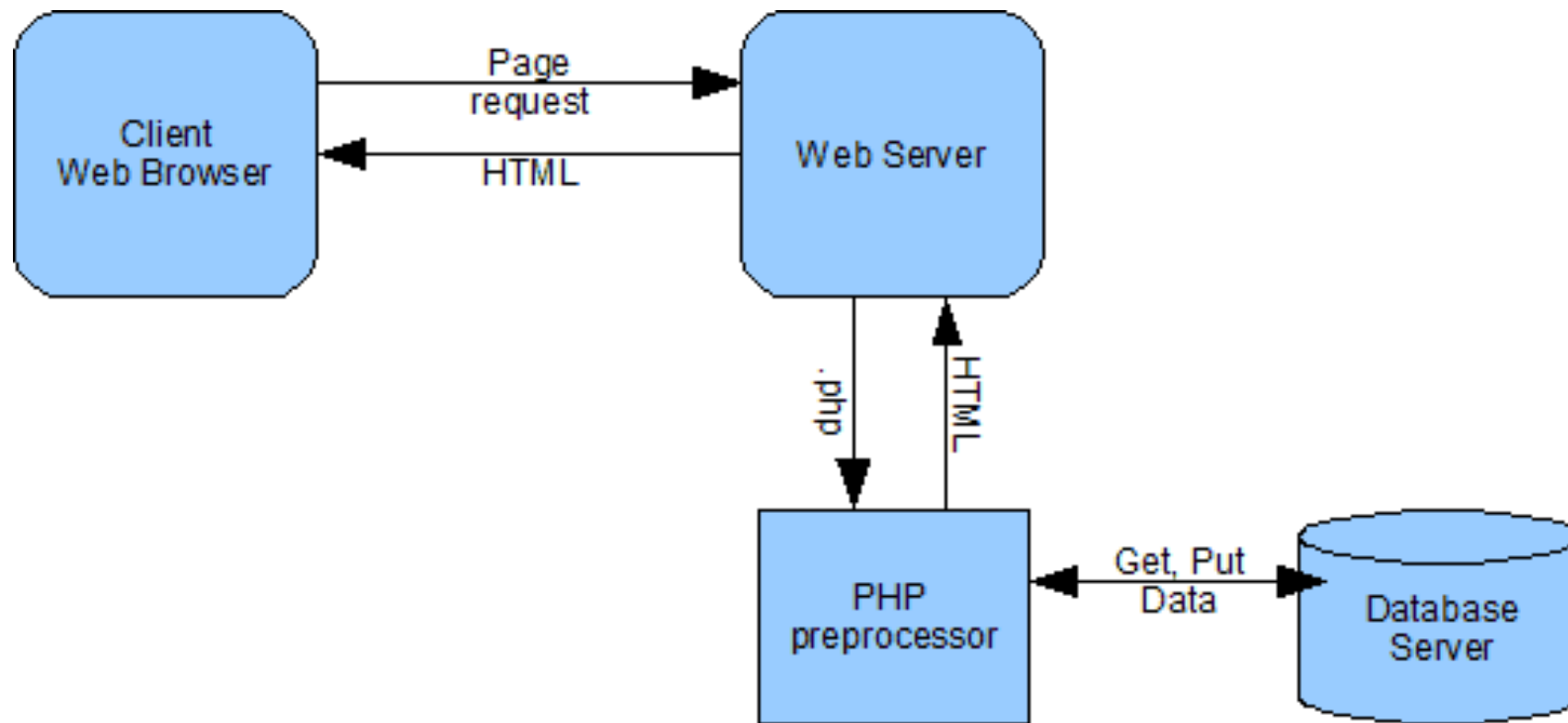




Thảo luận

Tổng quan PHP Web

Ứng dụng Web PHP



- LAMP là một bộ công nghệ được ưa chuộng và sử dụng phổ biến để xây dựng các ứng dụng web thông dụng
- Tên gọi LAMP xuất phát từ các chữ cái đầu của các công nghệ thành phần, bao gồm:
 - **Linux**: Hệ điều hành
 - **Apache**: Máy chủ dịch vụ Web
 - **MySQL**: Hệ quản trị CSDL
 - **PHP**: Ngôn ngữ server-side

- Linux là một họ các hệ điều mã nguồn mở
- Thường được sử dụng để cài đặt cho các máy chủ trong các doanh nghiệp
- Gọi Linux là một "họ" là bởi vì có rất nhiều hệ điều hành Linux:
 - Fedora
 - Debian
 - Ubuntu
 - CentOS
 - ...

Apache Web Server

- Là phần mềm máy chủ dịch vụ web (Web Server)
- Web Server là nơi đón nhận các request (yêu cầu) của người dùng, xử lý và trả về kết quả
- Apache là một phần mềm mã nguồn mở và miễn phí
- Ngoài Apache thì cũng có một số Web Server khác:
 - Nginx
 - IIS
 - GWS
 - ...

- **MySQL** là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở
- Hệ quản trị CSDL có chức năng lưu trữ dữ liệu, bảo mật, cung cấp công cụ để truy xuất dữ liệu và nhiều tính năng khác.
- Ngoài MySQL thì còn có nhiều các hệ quản trị CSDL khác:
 - SQL Server
 - Oracle DB
 - PostgreSQL
 - MariaDB
 - ...

- PHP là một ngôn ngữ lập trình phía server (server-side)
- Chức năng của PHP là xử lý yêu cầu của người dùng và sinh ra kết quả để trả về
- Thường thì các trang PHP sẽ sinh ra mã HTML, CSS, Javascript...
- Ngoài PHP còn có nhiều ngôn ngữ khác làm được công việc tương tự, chẳng hạn:
 - Perl
 - Python
 - Java
 - ...

Tại sao nên sử dụng LAMP

- Các thành phần của LAMP đều là mã nguồn mở và miễn phí
- Cộng đồng sử dụng LAMP rất lớn
- Có nhiều các nền tảng (framework), thư viện (library), tool (công cụ)
- Các thành phần trong LAMP tương thích tốt với nhau và tuân theo các tiêu chuẩn của ngành
- Việc phát triển các ứng dụng LAMP là khá nhanh chóng và dễ dàng
- Một phần nguyên nhân của việc LAMP được sử dụng rộng rãi là yếu tố lịch sử

Cài đặt môi trường



- Cài đặt PHP
 - <http://php.net/>
- Cài đặt Apache
 - <https://httpd.apache.org/>
- File PHP có đuôi là .php



APACHE
HTTP SERVER



Nhúng mã PHP vào HTML



- Mã PHP được đặt trong cặp dấu:

```
<?php  
    //Code  
?>
```

```
<!DOCTYPE html>  
<html>  
    <body>  
  
        <?php  
            echo "My first PHP script!";  
        ?>  
  
    </body>  
</html>
```

Khai báo biến



- Biến trong PHP bắt đầu bằng dấu \$, tiếp theo sau là tên của biến.
- Tên biến phải bắt đầu bằng một ký tự hoặc là một dấu gạch dưới (_)
- Tên biến không thể bắt đầu bằng số
- Tên biến chỉ được bao gồm các ký tự chữ, ký tự số và dấu gạch dưới
- Tên biến phân biệt chữ hoa và chữ thường

```
<?php  
$txt = "Hello  
world!";  
$x = 5;  
$y = 10.5;  
?>
```



Câu lệnh echo

- Câu lệnh echo được sử dụng để in ra các chuỗi
- Cú pháp:

`echo(strings)`

- Ví dụ:

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

Lấy dữ liệu từ form



- Có thể lấy dữ liệu gửi lên từ form thông qua các biến \$_GET hoặc \$_POST tùy theo phương thức được quy định trong form

```
<form action="welcome.php" method="post">  
  Name: <input type="text" name="name">  
  <input type="submit">  
</form>
```

```
<?php  
  Welcome <?php echo $_POST["name"]; ?><br>  
?>
```

Thảo luận

Biến

Toán tử

Khai báo và sử dụng biến



- Biến bắt đầu bằng ký tự \$, tiếp theo là tên biến
- Tên biến phải bắt đầu bằng một ký tự chữ hoặc một dấu gạch dưới (_)
- Tên biến không được bắt đầu bằng ký tự số
- Tên biến chỉ được phép chứa ký tự chữ, ký tự số và dấu gạch dưới
- Tên biến là phân biệt chữ hoa và chữ thường

```
<?php  
$txt = "Hello  
world!";  
$x = 5;  
$y = 10.5;  
?>
```

Phạm vi của biến

- Biến có thể được khai báo bất cứ đâu
- Vị trí khai báo biến quy định phạm vi mà biến đó có thể được sử dụng
- Các phạm vi được quy định trong PHP là:
 - local (cục bộ): Sử dụng trong hàm
 - global (toàn cục): Sử dụng ngoài hàm
 - Có thể sử dụng từ khóa **global** để truy xuất biến toàn cục ở trong hàm
 - Có thể sử dụng mảng **\$_GLOBAL**
 - static (tĩnh): Giữ lại giá trị của biến local mà không hủy đi

Các kiểu dữ liệu



- PHP hỗ trợ các kiểu dữ liệu:
 - String (chuỗi)
 - Integer (số nguyên)
 - Float (số thập phân)
 - Boolean (Giá trị true/false)
 - Array (mảng)
 - Object (đối tượng)
 - NULL
 - Resource

```
<?php
$s = "Hello world!";
$i = 5985;
$f = 10.365;
$b = true;
$cars = array("Volvo", "BMW", "Toyota");
$herbie = new Car();
$n = null;
```


Các loại toán tử

- Arithmetic-Toán học: +, -, *, /, %
- Assignment-Gán: =, -=, +=, *=, /=, %=
- Comparison: ==, ===, !=, >=, <=, <, >, <>, !==
- Increment/Decrement: ++, --
- Logical-Logic: &&, ||, !
- String: Chuỗi

Toán tử	Tên	Ví dụ	Kết quả
.	Nối chuỗi	\$txt1 . \$txt2	Nối chuỗi \$txt1 với chuỗi \$txt2
.=	Nối chuỗi và gán	\$txt1 .= \$txt2	Thêm chuỗi \$txt2 vào cuối chuỗi \$txt1

- Array: Mảng

Thảo luận

Câu lệnh điều kiện

Câu lệnh lặp

Câu lệnh điều kiện



- PHP hỗ trợ các câu lệnh điều kiện:
 - **if**: Thực hiện câu lệnh nếu điều kiện là đúng
 - **if...else**: Thực hiện một câu lệnh nếu điều kiện là đúng, còn nếu điều kiện sai thì thực hiện một câu lệnh khác
 - **if...elseif....else**: Thực hiện các câu lệnh khác nhau với việc đánh giá nhiều hơn 2 điều kiện
 - **switch....case**: Chọn thực hiện một hoặc nhiều câu lệnh tùy thuộc vào điều kiện

Câu lệnh if



- Cú pháp

```
if (condition) {  
    code to be executed if condition is true;  
}
```

- Ví dụ:

```
<?php  
$t = date("H");  
if ($t < "20") {  
    echo "Have a good day!";  
}  
?>
```

Câu lệnh if...else



- Cú pháp:

```
if (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```

- Ví dụ:

```
<?php  
$t = date("H");  
if ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

Câu lệnh if...elseif...else



- Cú pháp:

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

- Ví dụ:

```
<?php  
$t = date("H");  
if ($t < "10") {  
    echo "Have a good morning!";  
} elseif ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

Câu lệnh switch...case: cú pháp



```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all labels;  
}
```

Câu lệnh switch...case: ví dụ



```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```




Thảo luận

Câu lệnh lặp

Câu lệnh lặp



- PHP hỗ trợ các câu lệnh lặp:
 - **while**: Kiểm tra điều kiện, nếu đúng thì thực hiện các câu lệnh, lặp lại các bước như thế.
 - **do...while**: Thực hiện các câu lệnh, kiểm tra điều kiện, nếu đúng thì lặp lại các bước như thế.
 - **for**: Thực hiện lặp lại các câu lệnh trong một số lần nhất định
 - **foreach**: Thực hiện lặp lại các câu lệnh đối với từng phần tử của một mảng

Câu lệnh while



- Cú pháp:

```
while (condition is true) {  
    code to be executed;  
}
```

- Ví dụ:

```
<?php  
$x = 1;  
  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

Câu lệnh do-while



- Cú pháp:

```
do {  
    code to be executed;  
} while (condition is true);
```

- Ví dụ:

```
<?php  
$x = 1;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>
```

Câu lệnh for



- Cú pháp:

```
for (init counter; test counter; increment counter) {  
    code to be executed;  
}
```

- Ví dụ:

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

Câu lệnh foreach



- Cú pháp:

```
foreach ($array as $value) {  
    code to be executed;  
}
```

- Ví dụ:

```
<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
?>
```



Tóm tắt bài học

- Web là một không gian trên Internet mà ở đó các tài liệu và các tài nguyên khác được chia sẻ dưới dạng các URL.
- Giao thức là tập các quy ước/chuẩn mực, quy định cách thức giao tiếp giữa các bên.
- Biến
- Toán tử
- Kiểu dữ liệu
- Câu lệnh điều kiện
- Câu lệnh lặp: for, while, do-while, foreach
- Tạo và thực thi ứng dụng PHP

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: ***Mảng và hàm trong PHP***