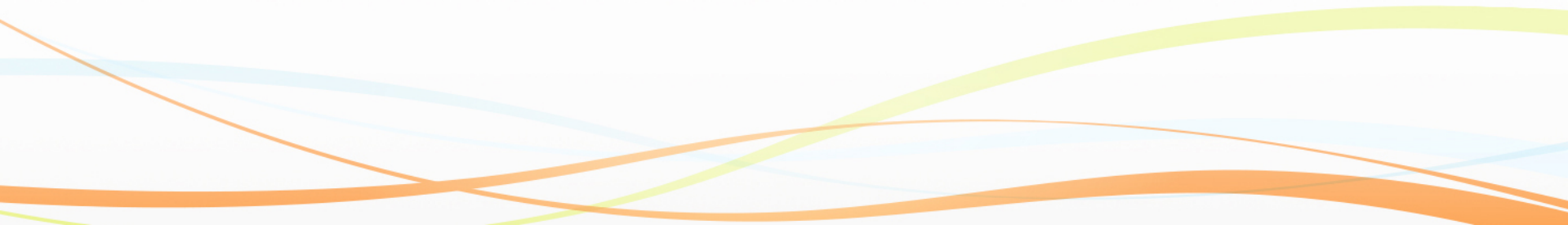




CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

Nội dung

- Giới thiệu về PHP.
 - Cơ chế hoạt động của webserver.
 - Cú pháp và các quy ước trong PHP.
 - Lập trình hướng đối tượng trong PHP
- 

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

- PHP (Hypertext Preprocessor) – Là ngôn ngữ lập trình phía server, được thiết kế để xây dựng và phát triển ứng dụng web
- Ra đời vào năm 1995 bởi Rasmus Lerdorf
- 1997: PHP 2 ra đời, hỗ trợ CSDL, Upload File, mảng ...
- 1998: PHP 3, hỗ trợ giao thức email, đa hệ điều hành. bộ phân tích mã PHP (parser) của Zeev Suraski và Andi Gutmans
- 2000: PHP 4, Trở thành một thành phần độc lập cho các webserver, thêm tính năng bảo mật. Parse đổi tên thành Zend Engine

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

- 2005: PHP 5 ra đời, Bổ sung Zend Engine II hỗ trợ lập trình HĐT, XML, SOAP cho Web Services, SQLite
- PHP 7 đang được phát triển
- Download php tại: <http://php.net>

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

- PHP 7 là ngôn ngữ thông dịch
- Tập tin PHP có phần mở rộng là .php
- Cú pháp tương tự như C & Perl

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

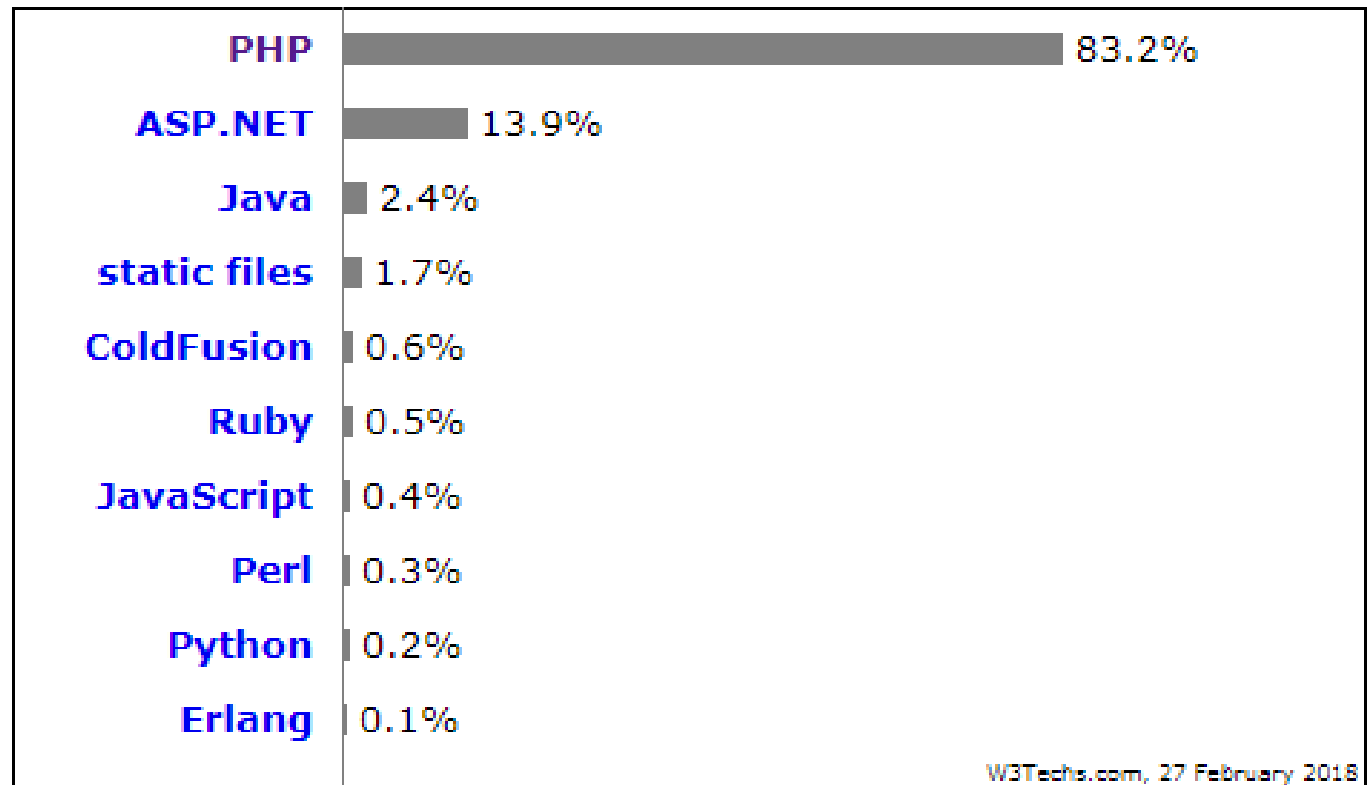
► Giới thiệu về PHP

- Đơn giản
- Chạy trên hệ thống máy chủ
- Tốc độ xử lý nhanh, dễ sử dụng
- Miễn phí, có cộng đồng hỗ trợ rộng lớn
- Thực thi trên tất cả các hệ điều hành
- Làm việc được với nhiều CSDL khác nhau
MySQL, Oracle, MS SQL ...
- Các framework: Joomla, Drupal, Zend...
- Nhiều editor hỗ trợ: PHP Designer,
DreamWeaver, NetBeans, PHPStorm...

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

- Usage of server-side programming languages for websites



W3Techs.com, 27 February 2018

Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

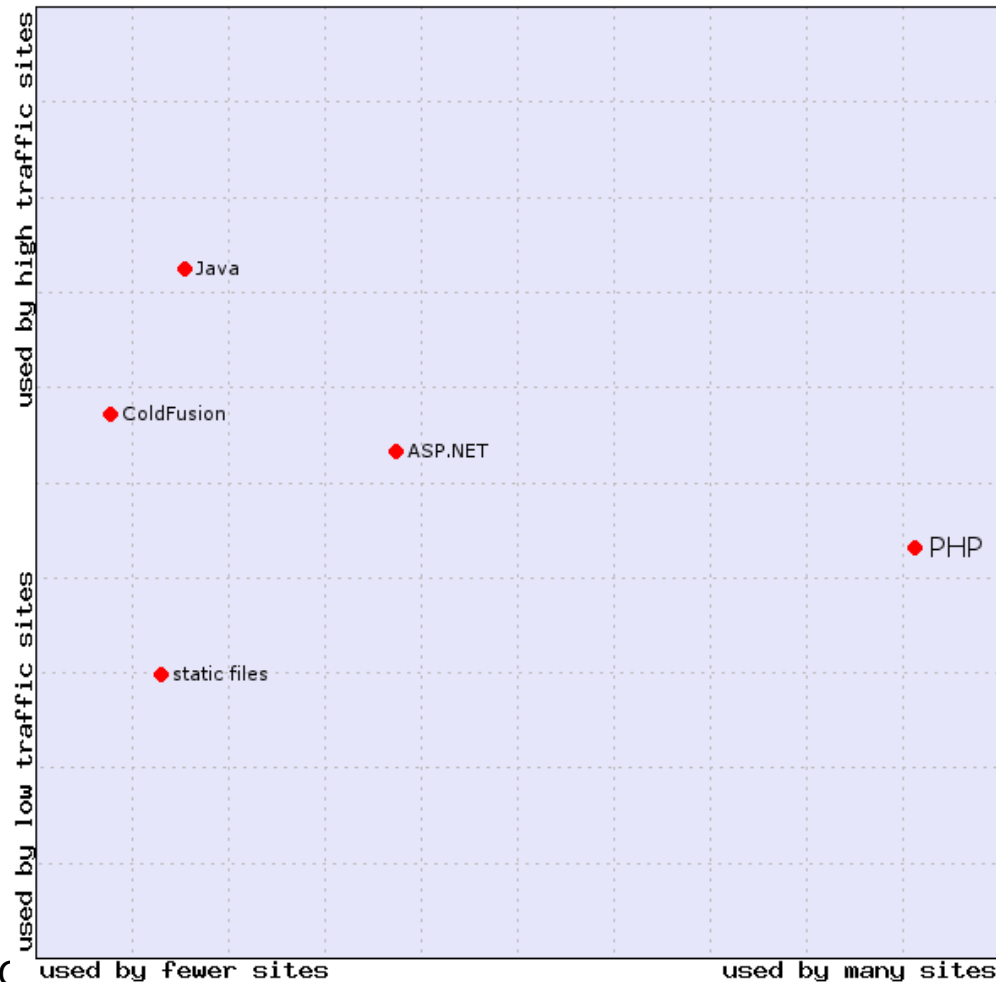
- This report shows the historical trends in the usage of server-side languages since February 2017.

	2017 1 Feb	2017 1 Mar	2017 1 Apr	2017 1 May	2017 1 Jun	2017 1 Jul	2017 1 Aug	2017 1 Sep	2017 1 Oct	2017 1 Nov	2017 1 Dec	2018 1 Jan	2018 1 Feb	2018 27 Feb
PHP	82.4%	82.5%	82.6%	82.6%	82.6%	82.7%	82.7%	82.8%	82.8%	82.9%	83.0%	83.1%	83.1%	83.2%
ASP.NET	15.3%	15.2%	15.1%	15.1%	15.1%	15.0%	14.9%	14.8%	14.5%	14.3%	14.2%	14.1%	14.1%	13.9%
Java	2.7%	2.7%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.5%	2.5%	2.5%	2.4%	2.4%
static files	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.7%
ColdFusion	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%
Ruby	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.5%	0.5%	0.5%	0.5%	0.6%	0.6%	0.5%	0.5%
JavaScript	0.3%	0.3%	0.3%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%
Perl	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%
Python	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%
Erlang	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

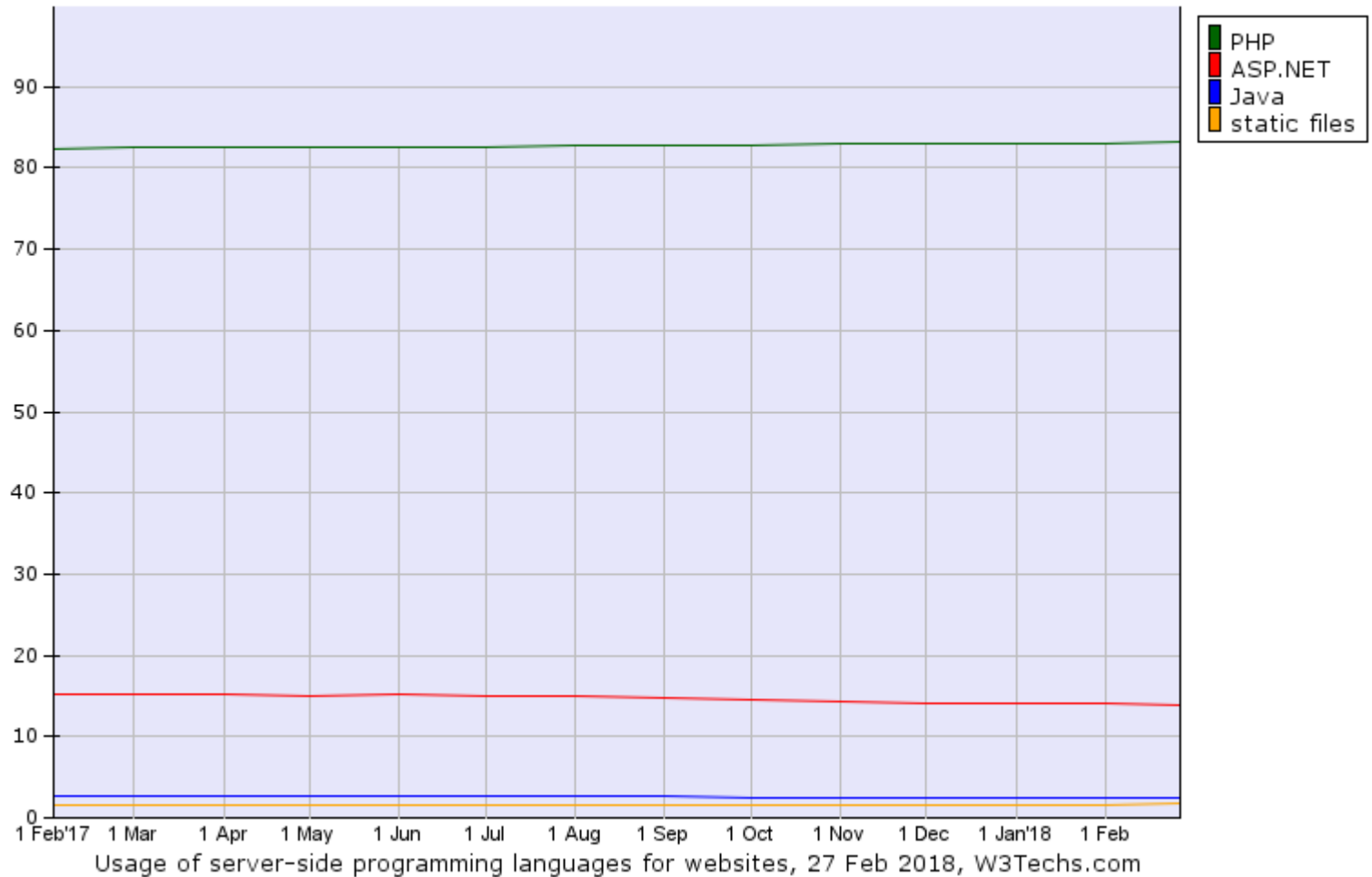
► Giới thiệu về PHP

PHP Market Position, 27 Feb 2018, W3Techs.com



CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP



CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

Websites	Popularity (unique visitors per month) ^[1]	Front- end (Client- side)	Back-end (Server-side)	Database
Google.com ^[2]	1,600,000,000	JavaScript	C, C++, Go, ^[3] Java, Python	Bigtable, ^[4] MariaDB ^[5]
Facebook.com	1,100,000,000	JavaScript	Hack, PHP (HHVM), Python, C++, Java, Erlang, D, ^[6] Xhp, ^[7] Haskell ^[8]	MariaDB, MySQL, ^[9] HBase Cassandra ^[10]
YouTube.com	1,100,000,000	JavaScript	C, C++, Python, Java, ^[11] Go ^[12]	Vitess, BigTable, MariaDB ^[5] ^[13]
Yahoo	750,000,000	JavaScript	PHP	MySQL, PostgreSQL, ^[14] VB.NET
Amazon.com	500,000,000	JavaScript	Java, C++, Perl ^[16]	Oracle Database ^[17]
Wikipedia.org	475,000,000	JavaScript	PHP, <u>Hack</u>	MySQL ^[citation needed] , MariaDB ^[18]
Twitter.com	290,000,000	JavaScript	C++, Java, Scala, Ruby ^[19]	MySQL ^[20]
Bing	285,000,000	JavaScript	ASP.NET	Microsoft SQL Server
eBay.com	285,000,000	JavaScript	Java, ^[21] JavaScript, ^[22] Scala ^[23]	Oracle Database
MSN.com	280,000,000	JavaScript	ASP.NET	Microsoft SQL Server
Microsoft	270,000,000	JavaScript	ASP.NET	Microsoft SQL Server
Linkedin.com	260,000,000	JavaScript	Java, JavaScript, ^[24] Scala	Voldemort ^[25]

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Giới thiệu về PHP

- Bộ mã nguồn được xây dựng và đóng gói → giúp các lập trình viên tiết kiệm thời gian, giải quyết các vấn đề về chuẩn chung ...
- Các framework PHP phổ biến:
 - Laravel (2011)
 - Symfony
 - CodeIgniter (2006)
 - CakePHP (2005)
 - Zend Framework

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cài đặt và cấu hình PHP

- Môi trường lập trình PHP thường gồm các phần mềm:
 - PHP: để xử lý mã lệnh PHP
 - MySQL/DB2/PostgreSQL...: để quản trị cơ sở dữ liệu
 - Apache/Microsoft IIS...: máy chủ web để chạy các ứng dụng PHP sau khi lập trình trên máy tính cá nhân

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cài đặt và cấu hình PHP

- Download Apache Server

<http://httpd.apache.org/download.cgi>

- Download PHP

<http://www.php.net/downloads.php>

- Download MySQL Database

<http://www.mysql.com/downloads/index.html>

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cài đặt và cấu hình PHP

- Tất cả các phần mềm trên được gói gọn trong gói phần mềm WAMP (Window) hoặc LAMP (Linux) hoặc XAMPP (Windows/Linux)
- Sử dụng Phần mềm soạn thảo: để viết mã cho ứng dụng PHP
- Phần mềm dò lỗi: để dò lỗi PHP trong quá trình phát triển ứng dụng

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cài đặt và cấu hình PHP

- Download WAMP tại <http://www.wampserver.com/en/>.
- Hoặc Download LAMP <http://lamphowto.com/>.
- Hoặc Download XAMPP <https://www.apachefriends.org/download.html>.

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cơ chế hoạt động của webserver

■ Webserver

- Máy tính mà trên đó cài đặt phần mềm phục vụ Web và chạy được các file *.htm và *.html,
- Phần mềm Web Server:
 - IIS của Microsoft dành cho *.asp, *.aspx...;
 - Apache dành cho *.php...;
 - Tomcat ..

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► web tĩnh



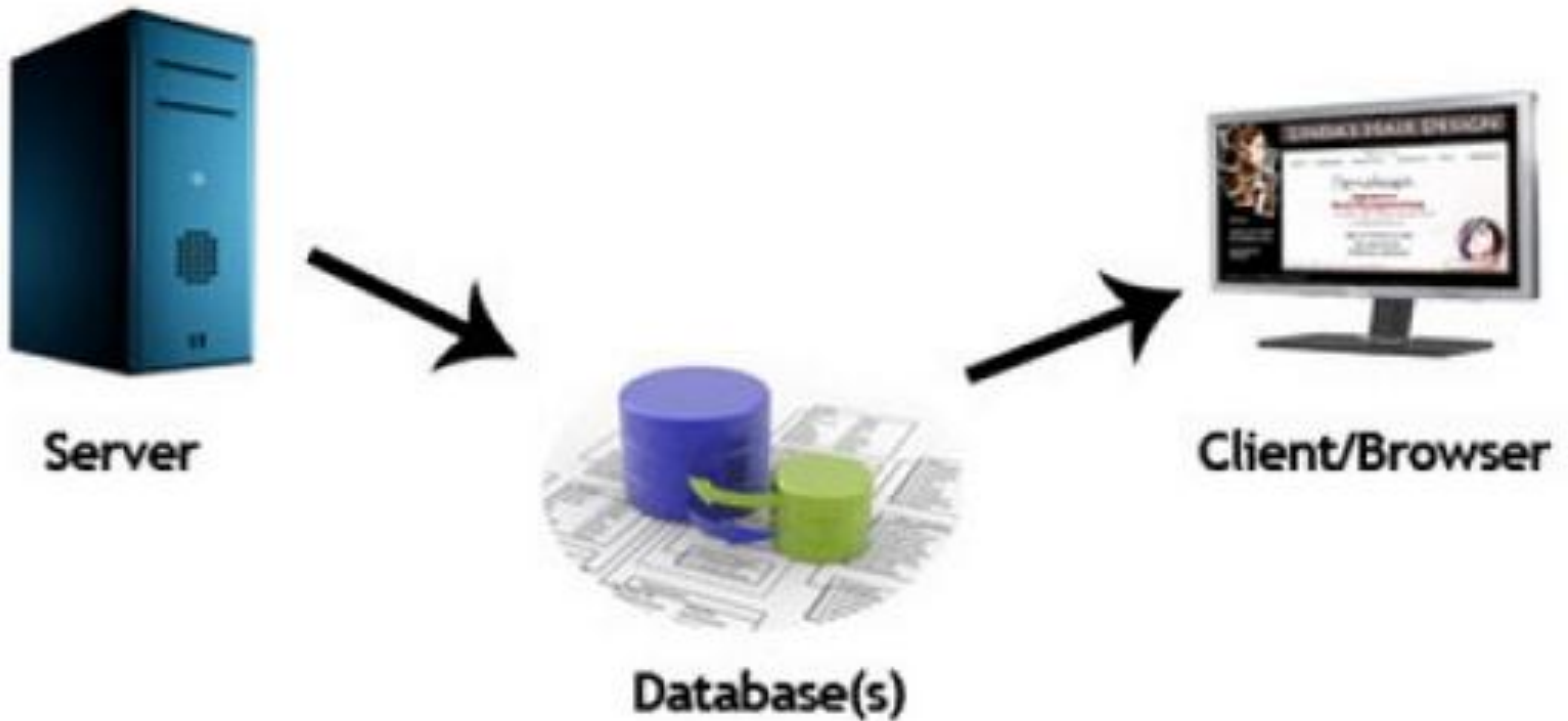
Server



Client/Browser

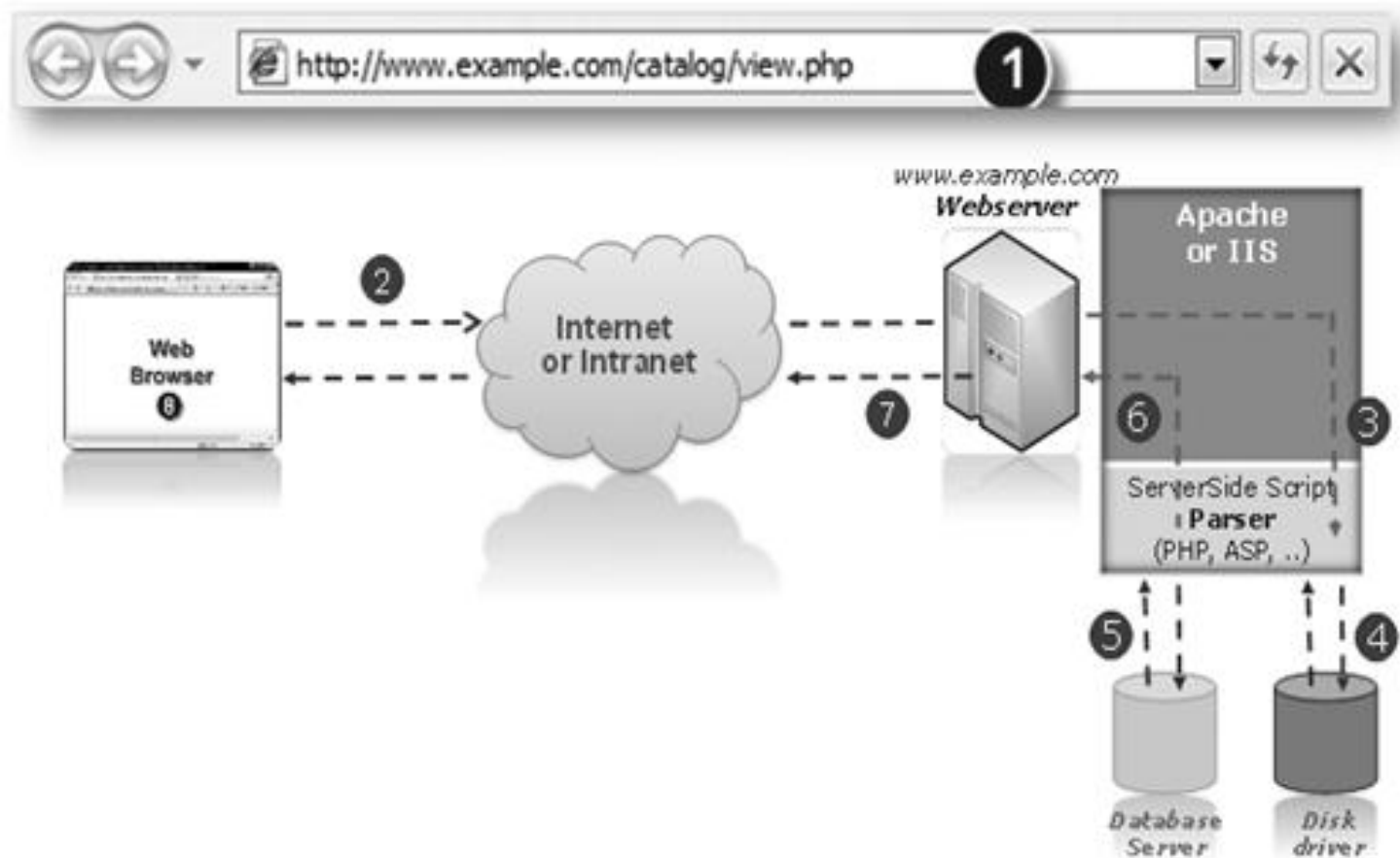
CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► web tĩnh



CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cơ chế hoạt động của webserver



Hình 2. 3 Cơ chế hoạt động của webserver

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- cấu trúc trang php
 - Có phần mở rộng là .php
 - Bao gồm: các thẻ (tag) HTML/CSS/Javascript
 - Code PHP

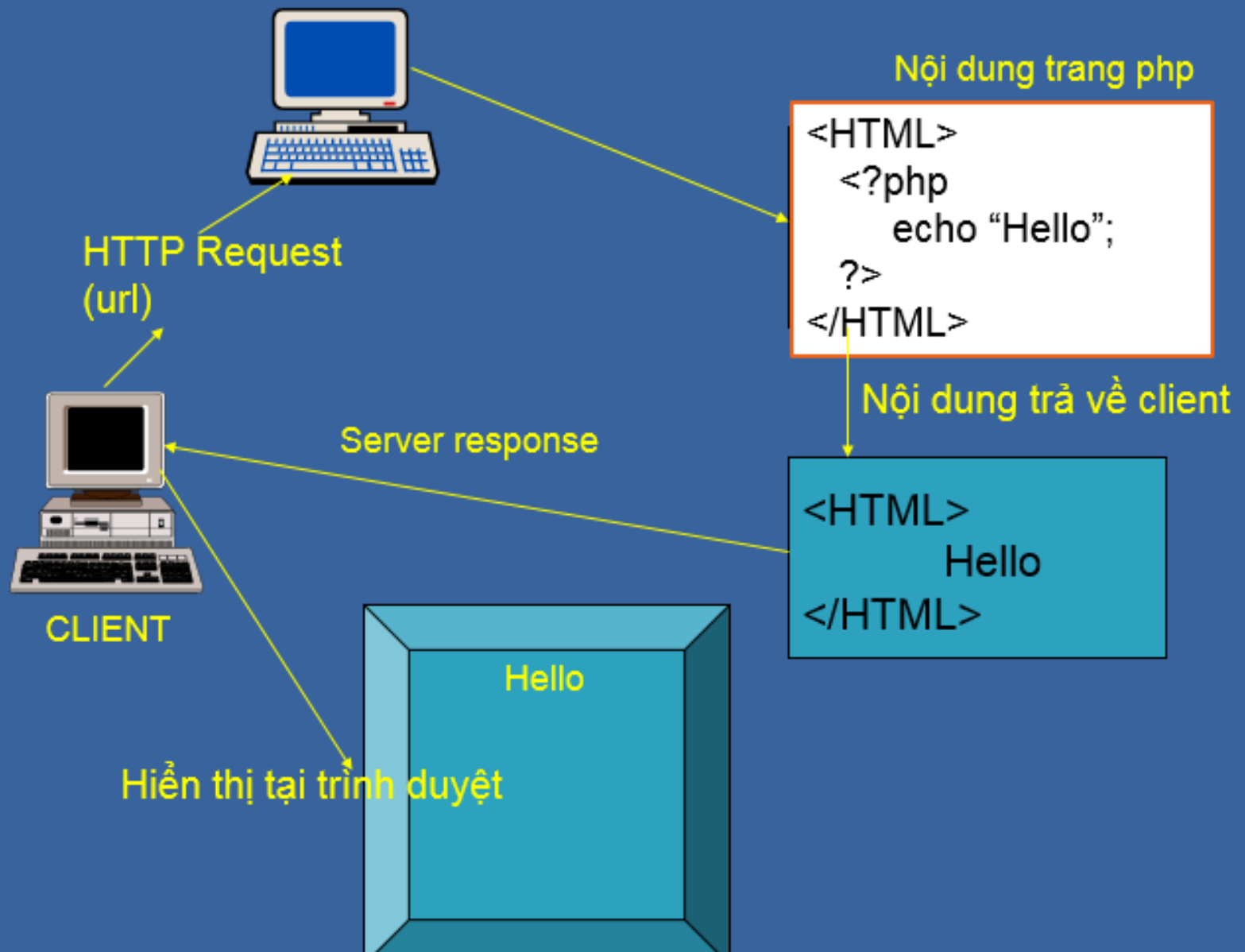
```
<html>

  <head>
    <title> hello world</title>
  </head>

  <body>
    <?php echo "Hello World!";?>
  </body>

</html>
```

WEB SERVER



CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- PHP có cấu trúc giống C / C++.
 - Tất cả các câu lệnh php kết thúc bởi dấu “;”
 - Khối (nhiều) lệnh được đặt trong cặp { }
 - Không phân biệt khoảng trắng, Tab, xuống dòng trong câu lệnh.

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- Mã lệnh php được đặt trong tag:

`<?php ?>`

hoặc

`<? ?>` set short_open_tag = On trong file php.ini

hoặc

`<SCRIPT Language='php'> </SCRIPT>`

hoặc

`<% %>`

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

■ Chú thích

// chú thích 1 dòng

chú thích 1 dòng

/*

Chú thích 1 đoạn

*/

<?php

// chu thich 1

echo "chu thích 1";

chu thich 2

echo "chú thích 2";

/* Đoạn chú thích

*/

echo "Đoạn chú thích"

;

?>

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- Các từ khóa
 - if,
 - else,
 - while,
 - echo...
 - tên class,
 - Tên functions
- không phân biệt chữ hoa hay chữ thường(trừ tên biến)

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Khai báo biến: \$tenbien
- Các biến được gán với toán tử =
Ví dụ: \$strName = "my Name";
- Khai báo hằng
define('ten_hang', 'gia_tri');
 - define: hàm tạo biến hằng
 - ten_hang: là tên biến hằng
 - gia_tri: giá trị của hằng

```
define('zipcode', '084');
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP:Biến

- Quy tắc đặt tên biến
 - Bắt đầu bằng \$, tiếp sau đó là ký tự hoặc _, sau đó có thể là ký tự, số hoặc _
 - Không đặt tên biến trùng với tên hàm
 - Không đặt tên biến bắt đầu bằng số, không có khoảng trắng trong tên biến
 - Nên khởi tạo giá trị ban đầu cho biến
- Tên biến phân biệt chữ hoa – thường

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

■ Phạm vi sử dụng biến

- Biến siêu toàn cục (global): có thể sử dụng ở mọi nơi, không thể định nghĩa bởi người dùng
 - `$GLOBALS`, `$_SERVER`, `$_GET`, `$_POST`, `$_SESSION`,
 - `$_COOKIE`, `$_REQUEST`, `$_ENV`, `$php_errormsg`.
- Biến toàn cục (không nằm trong hàm): được khai báo và sử dụng bên trong một script, mặc định là không thể sử dụng bên trong các hàm
- *Biến hàm (biến cục bộ)*: được khai báo và sử dụng cục bộ trong phạm vi hàm

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Phạm vi sử dụng biến

```
<?php
```

```
$a = 1; // phạm vi toàn cục
```

```
function Test()
```

```
{
```

```
    echo $a; // phạm vi cục bộ
```

```
}
```

```
Test(); → không có
```

```
echo $a; → 1
```

```
?>
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Phạm vi sử dụng biến

```
<?php
$a = 1;
$b = 2;
function Sum()
{
    $_GLOBALS['b'] = $_GLOBALS['a'] + $_GLOBALS['b'];
}
Sum();
echo $b; → 3
?>
```

```
<?php
$a = 1;
$b = 2;
function Sum()
{
    global $a, $b;
    $b = $a + $b;
}
Sum();
echo $b; → 3
?>
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Phạm vi sử dụng biến:
 - Biến static: Không mất đi giá trị khi ra khỏi hàm

Khai báo:

`static $tên_biến;`

```
<?php
function Test()
{
    static $a = 0;
    echo $a;
    $a++;
}
Test(); → 0
Test(); → 1
Test(); → 2
```


CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Kiểm tra sự tồn tại của biến:

- Biến static: Không mất đi giá trị khi ra khỏi hàm

Hàm `isset($tenbien);`

```
<?php
```

```
$x=1;
```

```
if (isset($x)==true)
```

```
    echo "Có biến x";
```

```
else
```

```
    echo "Không có biến x ";
```

```
if (isset($y)==true)
```

```
    echo "Có biến y";
```

```
else
```

```
    echo "Không có biến y ";
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Biến

- Hủy 1 biến: Hàm **unset**(\$tenbien);

```
<?php
    $x=1;
    unset($x);
    if (isset($x)==true)
        echo "Có biến x";
    else
        echo "Không có biến x ";
?>
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- Xuất nội dung
 - echo “noi dung”;
 - print “noidung”;
 - printf(string str);

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- Xuất nội dung

```
echo "Hello world!<br>;
```

```
echo "Hello ", "Hutech";
```

```
print "Hello world!<br>;
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP

- Xuất nội dung

```
echo "Hello world!<br>";
```

```
echo "Hello ", "Hutech";
```

```
print "Hello world!<br>";
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

▪ Kiểu dữ liệu cơ sở:

- Boolean: có 1 trong 2 giá trị true/false
- Integer
- double
- String: không có giới hạn về kích thước
- Array
 - `$array_keys = array('a', 'b', 'c');`
 - Truy xuất mảng: `$array_keys[0]`
 - Gán giá trị cho mảng: `$array_keys[3] = 'f'`
- Object
 - `$myclass= new MyClass()`

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Ép kiểu:

- Ép kiểu chỉ định

```
$flag = 1;  
$flag = (bool)$flag ;  
// có thể dùng  
$flag = (boolean)$flag ;
```

- Ép kiểu ngầm định

```
$var = "100" + 15;
```

- settype(\$var, “kiểu dữ liệu”)

```
settype($a, “float”); //gán kiểu dữ liệu float cho biến a
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

▪ Ép kiểu:

\$var	(int)\$var	(bool)\$var	(string)\$var
null	0	false	""
true	1		"1"
false	0		""
"6 feet"	6	true	
"foo"	0	true	

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: Gồm 2 thành phần
 - **Key**: khóa để truy cập đến từng phần tử của mảng. Key có thể là chuỗi hoặc là số
 - **Value**: giá trị tương ứng với khóa

Key is integer

Key	Value
0	red
1	yellow
2	blue
3	green
4	black

Key is a string

Key	Value
Mary	44
John	54
Tom	36
Amy	28
Billy	63

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng:

- 1 Mảng tuần tự
- 2 Mảng không tuần tự
- 2 Mảng đa chiều

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – mảng 1 chiều, tuần tự
 - Key là số, bắt đầu từ 0, sắp xếp tăng dần

0	1	2	3	4	5	6	7	8	9
12	15	20	80	42	71	63	49	17	33

↑
Phần tử a[3]

```
$a = array(12,15,20,80,42,71,63,49,17,33);  
$a[3] = 10; // gán giá trị cho phần tử a[3]
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – mảng 1 chiều, không tuần tự
 - Key là chuỗi và sắp xếp không theo thứ tự

```
$phpSub=array("LyThuyet"=>"45",  
              "ThucHanh"=>"30",  
              "KiemTra"=>"40",  
              "Thi"=>"60",  
              );
```

```
echo $phpSub["KiemTra"];
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – mảng đa chiều
 - Là một mảng chứa nhiều mảng con bên trong

	Column 0	Column 1	Column 2	Column 3
Row 0	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row 1	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row 2	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

Multidimensional array

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – mảng đa chiều
 - Là một mảng chứa nhiều mảng con bên trong

```
$cars=array  
(  
    array("Volvo",100,96),  
    array("BMW",60,59),  
    array("Toyota",110,100)  
);
```


CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – Duyệt mảng
 - Duyệt mảng với vòng lặp foreach

`foreach (mảng as [khóa =>] giá trị)`

`Khối lệnh;`

- Duyệt mảng với vòng lặp for

```
$array_Number = array(1,2,3,4,5,6,7,8,9,10);  
for($i = 1; $i <10; $i++)  
{   echo $array_Number[$i]."<br>"; }
```


CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Kiểu dữ liệu

- Kiểu mảng: – Duyệt mảng
 - Duyệt mảng với con trỏ mảng
 - Con trỏ mảng trỏ vào phần tử đầu tiên khi mảng được tạo ra, dùng các hàm cung cấp sẵn để di chuyển đến phần tử cần truy xuất

```
$people = array("Peter", "Joe", "Glenn", "Cleveland");
```

```
echo current($people) . "<br>";  
echo next($people);
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Toán tử

Toán học	+ - * / % ++ --	
So sánh	< > <= >= != == === !==	
Luận lý	&& ?: ,	
Xử lý bit	! ~ << >> >>> AND OR XOR	
Gán	= += -= *= /= %= >>= <<= &= = ^= .=	
Ép kiểu	(kiểu dữ liệu)	(int) (double) (string)...

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

- Cấu trúc rẽ nhánh
- Cấu trúc chọn lựa
- Cấu trúc lặp

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ **Cấu trúc rẽ nhánh (if)**

- Là biểu thức dùng kiểm tra 1 sự kiện. Nếu chúng thỏa điều kiện đó thì sẽ thực thi một hành động. Ngược lại sẽ là một hành động khác.

– **Cú pháp: (if)**

```
if(Điều kiện)
{
    //khối lệnh
}
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trình bày

▪ Cấu trúc rẽ nhánh (if)

```
if (điều kiện_1)
{
    khối_lệnh_1
}
else if (điều kiện_2)
{
    khối_lệnh_2
}
else
{
    khối_lệnh_3
}
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước khiển

▪ Cấu trúc chọn lựa

```
switch (n)
{
    case label1:
        khởi lệnh nếu n=label1;
        break;
    case label2:
        khởi lệnh nếu n=label2;
        break;
    case label3:
        khởi lệnh nếu n=label3;
        break;
    ...
    default:
        khởi lệnh nếu n khác các
        el;
}
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ **Cấu trúc lặp:**

- Cấu trúc lặp cho phép thực hiện nhiều lần một khối lệnh của chương trình khi thỏa điều kiện
- Các cấu trúc: for, foreach, while, do ...while

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ Cấu trúc lặp for:

for (*giá trị khởi tạo; điều kiện lặp; giá trị lặp*)

{

lệnh lặp;

}

```
<?php
```

```
    $tong = 0;
```

```
    for($i=1; $i<=10;$i++)
```

```
    {
```

```
        $tong = $tong+ $i;
```

```
    }
```

```
    echo "Tổng các số từ 1 đến 10 là: ".$tong;
```

```
?>
```


CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ Cấu trúc lặp foreach:

- foreach thường được dùng để duyệt tập hợp (mảng).

```
foreach ($array as $value)
{
    khối lệnh;
}
```

```
<?php
    $giatri=array("a","b","c");
        foreach ($giatri as $value)
        {
            echo $value."<br>";
        }
?>
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ Cấu trúc lặp while:

```
while(điều kiện)
{
    khối lệnh;
}
```

```
$x = 1;
```

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

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

▪ do....while:

```
do
{
    Khóilệnh;
}while(điềukiện);
```

```
$x = 1;
```

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

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: cấu trúc điều khiển

- **Lệnh break và continue trong cấu trúc lặp**

- **Lệnh break:**

- Thoát khỏi cấu trúc dựa trên kết quả của biểu thức điều kiện.

Ví dụ: kiểm tra 1 số có phải là số nguyên tố không

- Lệnh continue: sẽ nhảy về đầu vòng lặp để kiểm tra giá trị của biểu thức điều kiện còn đúng hay không

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

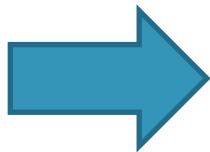
- Built-in function: hàm do php cung cấp
- User defined function: hàm do người dùng định nghĩa

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function:

```
function tên_hàm($arg_1, $arg_2, /* ..., */ $arg_n)
{
    statement here
    [return $retval;]
}
```



```
function sum($num1, $num2){
    return $num1 + $num2;
}
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function: gọi hàm

```
function sum($num1, $num2){  
    return $num1 + $num2;  
}
```

```
$tong = sum(1,5);  
echo ($ong."<br>");
```

```
$a =100;  
$b =100;  
$tong = sum($a,$b);  
echo ($ong."<br>");
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function: Gán giá trị mặc định cho tham số truyền vào

```
function display($mess='co loi xay ra'){  
    echo ($mess."<br>");  
}  
display();
```


CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function: Gọi hàm
 - Có thể truyền tham số theo tham chiếu
 - Cú pháp: viết ký hiệu '&' trước tham số

```
function sum(&$num1, $num2){  
    $num1++;  
    return $num1 + $num2;  
}
```

```
$a=1;  
$b=2;  
$tong=sum($a,$b);  
echo ($tong."<br>". $a);
```

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function: Gọi hàm
 - Truyền mảng tham số
 - Có thể truyền 1 mảng tham số với chiều dài của mảng thay đổi
 - Sử dụng các hàm được cung cấp sẵn
 - func_get_args(): trả về mảng chứa các tham số đầu vào của một hàm
 - Func_num_args(): trả về số lượng tham số được truyền vào
 - Func_get_arg(\$i): trả về tham số tại vị trí i của mảng tham số truyền vào

CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH PHP

► Cú pháp và các quy ước trong PHP: Hàm

- User defined function: Gọi hàm

```
function trungbinh($num){  
    $args = func_get_args();  
    $total =0;  
    for($i=0;$i< func_num_args();$i++){  
        $total+=func_get_arg($i);  
    }  
    return $total/func_num_args();  
}  
echo ( trungbinh(1,2,3,4,5) ."<br>" );
```