1. PHP làm việc thế nào

-Thông thường khi người dùng gọi thông tin dữ liệu từ trang website, thì máy chủ dử liệu sẽ gọi PHP Engine để đưa ra kết quả mà người dùng mong muốn và yêu cầu qua trang PHP và thường trải qua 4 giai đoạn giao tiếp như sau để đưa ra kết quả:

. Lexing

Lexing đây là một quá trình để chuyển một đoạn mã của PHP trở thành một chuỗi các token khác nhau có gắn giá trị riêng để để tạo các lexer từ file khai báo sẵn có trước.

. Parsing

Đây là giai đoạn phân tích parser thông qua file grammar BNF với Bison và nhận các luồng token từ lexer (như bước 1) để thực hiện công việc: Xem và kiểm tra tính hợp lệ của token được tiếp nhận và trả về có khớp với nhau hay không quy tắc ngữ pháp trong tập tin ngữ pháp BNF được xác định trước.

. Compiling

Compiling chính là giai đoạn biên dịch và nó sử dụng AST đưa ra các mã để tác dụng bằng cách thông qua cây phương pháp đệ quy chuẩn và tối ưu hóa câu lệnh và gọi hàm ra.

. Interpreter

Interpreter là một giai đoạn gọi là thông dịch mã yêu cầu. Thông thường mã tác vụ chạy bằng Zen Engine với khoảng cách rất ngắn với kết quả đưa ra như khi sử dụng PHP echo mà thôi.

2. File cấu hình php.ini

-Khi muốn thay đổi bất kỳ một cấu hình nào đó trong PHP như tăng bộ nhớ sử dụng, thay đổi đường dẫn lưu tạm thời, bật chức năng hiển thị lỗi,... thì sẽ cần can thiệp vào tập tin php.ini.

-Lưu ý: nếu sử dụng các dịch vụ Shared Host thì có thể nhiều nhà cung cấp không cho phép tự thiết lập php.ini mà sẽ có một tập tin khác tương tự tên là user.ini, hoặc là không có gì cả, còn nếu dùng localhost thì sẽ có quyền sửa nội dung php.ini

Cấu trúc tập tin php.ini

Trong tập tin này, nó sẽ chia ra làm hai phần chính gồm:

Phần ghi chú: Từ dòng 1 đến 195. Nội dung trong đây là để chúng ta đọc sơ bộ xem trong file này có các thiết lập gì và ý nghĩa, giá trị của nó.

Phần thiết lập: Từ dòng 196 trở xuống sẽ là các thiết lập. Các thiết lập có ký tự ; (còn gọi là ký tự comment) nghĩa là thiết lập đó đang sử dụng giá trị mặc định. Nếu bạn muốn đổi thiết lập thì hãy xoá dấu ; trước nó và sửa giá trị theo ý muốn.

Cũng nên lưu ý rằng, các thiết lập trong php.ini sẽ có dạng:

[code]tên\_thiết\_lập = giá trị[/code]

Giá trị có 3 kiểu:

Kiểu số kèm đơn vị: 500M, 100K, 1G (500MB, 100KB, 1GB). Hoặc cũng có vài thiết lập chỉ sử dụng số.

Kiểu boolean: On hoặc Off, hoặc có thể là 0 và 1.

Kiểu tên giá trị: Một vài thiết lập có hỗ trợ giá trị là các tên thiết lập.

-Các thiết lập thông dụng

Tất cả các thiết lập trong php.ini có thể chúng ta không cần thuộc nó mà chỉ đụng vào khi cần thôi, nên hiểu qua các thiết lập mà sẽ thường xuyên đụng vào nhất.

disable\_functions

Thiết lập này sẽ có chức năng vô hiệu hoá một số hàm có sẵn trong PHP vì lý do bảo mật. Tạm thời ở đây chúng ta sẽ thử vô hiệu hoá các hàm nguy hiểm như exec(), system(), passthru(), symlink() bằng cách khai báo giá trị chodisable\_functions như sau.

php.ini-disable\_function

expose\_php

Tuỳ chọn này sẽ giúp ẩn thông tin về PHP trên HTTP Header của website nếu không muốn cho ai đó biết rằng mình đang sử dụng PHP. Nếu muốn ẩn đi, chỉ cần sửa giá trị thành Off.

max\_execution\_time

Đây là tuỳ chọn rất phổ biến, nó sẽ giúp bạn gia hạn thời gian thực thi một kịch bản PHP nào đó nếu nó cần nhiều thời gian hơn 30 giây. Nếu thời gian thực hiện script quá giá trị trong này nó sẽ báo lỗi “Fatal error: Maximum execution time of {giây} seconds”. Hiện tại bạn có thể nâng lên thành 600, tức 600 giây.

memory\_limit

Mặc định PHP sẽ được chỉ định sử dụng 128MB bộ nhớ của hệ thống cho việc xử lý kịch bản trong PHP. Tuy nhiên với các mã nguồn lớn, có thể nó sẽ cần thêm bộ nhớ nên bạn sẽ chỉnh ở đây.

error\_reporting

Mặc định PHP sẽ hiển thị tất cả các lỗi bao gồm lỗi nghiêm trọng (Fatal Error), cảnh báo (Warning), ghi chú (Notice),…nhưng nếu bạn muốn tuỳ chọn hiển thị các lỗi nhất định thì sẽ điều chỉnh giá trị ở đây.

display\_errors

Tuỳ chọn hiển thị lỗi ra ngoài website. Khi học lập trình hoặc khi đang viết code cho dự án, bạn nên bật bằng cách sửa giá trị thành On để thấy ngay lỗi thay vì trang trắng.

log\_errors

Tuỳ chọn này sẽ lưu lỗi vào các tập tin log, hay còn gọi là error\_log đó.

file\_uploads

Tuỳ chỉnh cho phép upload tập tin thông qua PHP.

upload\_max\_filesize

Dung lượng tối đa mỗi tập tin khi upload. Ngoài ra, bạn cũng phải nâng dung lượng của tuỳ chọnpost\_max\_size lên nhé.

có thể ấn File >> Save để lưu lại hoặc Ctrl + S.

Để kiểm tra lại giá trị thiết lập ta có thể tạo ra một tập tin mới ở thư mục project tên là info.php, sau đó copy đoạn này vào.

[code lang=”php”]&lt;?php phpinfo(); ?&gt;[/code]

Sau đó bạn truy cập vào đường dẫn http://domain.c9.io/info.php để xem thông tin thiết lập PHP.

Cách tạo tệp php.ini mới trong cPanel:

Đăng nhập vào cPanel.

Trong mục Files, nhấp vào biểu tượng File Manager.

. Chọn Home Directory và click Go.

. Nhấp vào nút New File ở đầu trang.

. Trong trường có tên New File Name, nhập php.ini và nhấn Create New File.

. Nhấp vào file có tên php.ini.

. Chọn nút Edit ở đầu trang và nhấp vào nút Edit trong pop-up box.

Tìm file php.ini với file php có hàm phpinfo()

Cấu trúc lệnh:

# vi /var/www/html/info.php

<?php

phpinfo();

?>

được sử dụng để gọi ra thông tin về các giá trị, thông số thiết lập PHP đang sử dụng với:

var/www/html là vị trí chứa source code web mặc định trên Apache

3. PHP Tags

Mọi thứ bên trong cặp thẻ mở và thẻ đóng đều được trình phân tích cú pháp php diễn giải. Php đã giới thiệu ba loại thẻ mở và thẻ đóng.

a.Standard tags

<?php ?>

b.Short echo tag

<?= ?>

here, <?= is equivalent meaning of "<?php echo"

c.Short tag

<? ?>

Thẻ tiêu chuẩn và thẻ echo ngắn luôn có sẵn.

Nhưng thẻ ngắn có thể bị vô hiệu hóa thông qua chỉ thị tệp cấu hình php.ini short\_open\_tag hoặc bị tắt theo mặc định nếu PHP được xây dựng với cấu hình thẻ --disable-short-tags.

Nếu một tệp chỉ chứa mã PHP nên bỏ qua thẻ đóng PHP ở cuối tệp.

Điều này ngăn chặn khoảng trắng ngẫu nhiên hoặc các dòng mới được thêm vào sau thẻ đóng PHP

4. PHP Statements và Whitespace

-Câu Lệnh echo

Câu lệnh echo được sử dụng để hiển thị kết quả của một hoặc nhiều chuỗi ký tự, biến hoặc biểu thức toán học....

vd:

<?php

echo "Xin Chào", "<br>";

$name = "PHP";

echo $name, "<br>";

echo 1 + 2;

?>

-Câu Lệnh print

Câu Lệnh print được dùng để hiển thị giá trị của một giá trị dạng chuỗi, biến hoặc kết quả của biểu thức...

<?php

$greet = "Xin Chào";

print $greet;

?>

Tương tự như echo thì print cũng không phải là hàm do đó khi sử dụng câu lệnh print chúng ta không cần đặt giá trị cần in bên trong cặp dấu ngoặc ().

-Khoảng trắng trong php

cách thêm khoảng trắng:

str\_repeat('&nbsp;', 5); // adds 5 spaces

or

<?

echo "\x20\x20\x20";

?>

or

echo str numpeat ("& nbsp;", 60);

-có thể sử dụng "\ n" cho một dòng mới và "\ t" cho một tab.

5. Comments

Comment trong PHP có thể được sử dụng để mô tả bất kỳ dòng code nào để các nhà phát triển khác có thể hiểu được mã một cách dễ dàng. Nó cũng có thể được sử dụng để ẩn bất kỳ dòng code hay khối code nào.

PHP hỗ trợ comment một dòng đơn và comment nhiều dòng.v

-PHP Comment: một dòng

Có hai cách để sử dụng comment một dòng đơn trong PHP.

// (comment dòng đơn kiểu C/C++)

# (Nhận xét dòng đơn kiểu Unix)

Ví dụ:

<?php

// đây là commnet dòng đơn kiểu C/C++

# đây là comment dòng đơn kiểu Unix Shell

echo "Ví dụ 1: Comment 1 dòng trong PHP.";

?>

-PHP Comment: nhiều dòng

Trong PHP, chúng ta cũng có thể comment nhiều dòng. Để làm như vậy, chúng ta cần phải để tất cả các dòng bên trong /\* comments \*/. Ví dụ đơn giản về comment nhiều dòng trong PHP.

<?php

/\*

Mọi thứ được đặt

trong nhận xét

sẽ không được hiển thị

trên trình duyệt;

\*/

echo "Ví dụ 2: Comment nhiều dòng trong PHP.";

?>

6. PHP Functions

-php có rất nhiều hàm có sẵn

vd: date(), md5(), str\_split(), count(), strlen(), substr(),str\_replace(), ...

-Người dùng cũng có thể tự tạo 1 hàm với cú pháp

function functionName() {

code to be executed;

}

hoặc hàm có đối số

function functionName($a) {

code to be executed;

return $a;

}

Sử dụng đối số chuyển qua tham chiếu để cập nhật một biến

<?php

function add\_five(&$value) {

$value += 5;

}

$num = 2;

add\_five($num);

echo $num;

?>

//result = 7.

7. Biến số

-Các lọai biến số

Một biến trong PHP là tên của vị trí bộ nhớ chứa dữ liệu. Biến là bộ nhớ tạm thời được sử dụng để lưu trữ dữ liệu tạm thời.

Trong PHP, một biến được khai báo sử dụng ký hiệu $ theo sau là tên biến.

Cú pháp khai báo một biến trong PHP được đưa ra dưới đây:

$ten\_bien = gia\_tri;

-Cách đặt tên và sử dụng biến số

Biến bắt đầu bằng ký hiệu $, theo sau là tên của biến.

Các biến PHP phải bắt đầu bằng chữ cái hoặc dấu gạch dưới.

Biến PHP không thể bắt đầu bằng số và ký tự đặc biệt.

Tên biến chỉ có thể chứa ký tự chữ và số và dấu gạch dưới (Az, 0-9 và \_).

Tên biến phân biệt chữ hoa chữ thường ($age và $AGE là hai biến khác nhau).

-PHP có ba phạm vi biến khác nhau:

địa phương (local).

toàn cầu (global).

tĩnh (static).

-Phạm vi toàn cầu và địa phương

<?php

$x = 5; // phạm vi toàn cầu

function myTest() {

// sử dụng biến $x bên trong hàm này sẽ bị lỗi

echo "<p>Biến x bên trong hàm là: $x</p>";

}

echo "<p>Biến x bên ngoài hàm là: $x</p>";

myTest();

?>

Kết quả:

Biến x bên ngoài hàm là: 5

Notice: Undefined variable: x in C:\xampp\htdocs\php\hang-so-magic.php on line 6

Biến x bên trong hàm là:

-Một biến được khai báo bên trong một hàm có một PHẠM VI ĐỊA PHƯƠNG và chỉ có thể được truy cập trong hàm đó:

<?php

function myTest() {

$x = 5; // phạm vi toàn cầu

// sử dụng biến $x bên trong hàm này sẽ bị lỗi

echo "<p>Biến x bên trong hàm là: $x</p>";

}

echo "<p>Biến x bên ngoài hàm là: $x</p>";

myTest();

?>

Kết quả:

Notice: Undefined variable: x in C:\xampp\htdocs\php\hang-so-magic.php on line 8

Biến x bên ngoài hàm là:

Biến x bên trong hàm là: 5

-Từ khóa global trong PHP

Từ khóa global được sử dụng để truy cập vào một biến toàn cầu từ bên trong một hàm

vd:

<?php

$x = 5;

$y = 10;

function myTest() {

global $x, $y;

$y = $x + $y;

}

myTest();

echo $y; // kết quả là 15

?>

-Từ khóa static trong PHP

Thông thường, khi một hàm được thực thi hoàn thành, tất cả các biến của nó sẽ bị xóa. Tuy nhiên, đôi khi chúng ta muốn một biến địa phương KHÔNG bị xóa. Chúng ta cần sử dụng nó cho một công việc khác.

vd:

<?php

function myTest() {

static $x = 0;

echo $x . "<br>";

$x++;

}

myTest(); //kq 0

myTest(); //kq 1

myTest(); //kq 2

?>

-Superglobal

Các biến toàn cầu trong PHP là:

$GLOBALS ->là một biến siêu toàn cục của PHP được sử dụng để truy cập các biến toàn cục từ bất kỳ đâu trong tập lệnh PHP (cũng từ bên trong các hàm hoặc phương thức).

$\_SERVER ->là một biến siêu toàn cầu của PHP chứa thông tin về tiêu đề, đường dẫn và vị trí tập lệnh.

$\_REQUEST ->là một siêu biến toàn cục PHP được sử dụng để thu thập dữ liệu sau khi gửi biểu mẫu HTML.

$\_POST ->là một biến siêu toàn cục PHP được sử dụng để thu thập dữ liệu biểu mẫu sau khi gửi biểu mẫu HTML với method = "post"

$\_GET ->là một biến siêu toàn cục PHP được sử dụng để thu thập dữ liệu biểu mẫu sau khi gửi biểu mẫu HTML với method = "get".$ \_GET cũng có thể thu thập dữ liệu được gửi trong URL.

$\_FILES -> lưu thông tin các file upload lên server

$\_ENV ->chứa một mảng kết hợp các biến truyền qua đoạn mã hiện tại thông qua môi trường phương thức

$\_COOKIE -> lưu trữ các thông tin ng truy cập website

$\_SESSION -> lưu trữ thông tin phiên làm việc của PHP

-Constant

8. Tóan tử của PHP

-Toán tử số học:

vd: +, -, \*, /, %

($a + $b)

-Toán tử gán:

vd:=, +=, -=, \*=, /=, %=

$a=0

$a +=1;

echo $a; //kết quả: 1

-Toán tử so sánh

vd: ==, ===, !=, <>, !==, <, >, >=, <=

<?php

$x = 25;

$y = 35;

$z = "25";

// Trả về kết quả và kiểu dữ liệu

var\_dump($x == $z); // Kết quả: boolean true

var\_dump($x === $z); // Kết quả: boolean false

var\_dump($x != $y); // Kết quả: boolean true

var\_dump($x !== $z); // Kết quả: boolean true

var\_dump($x < $y); // Kết quả: boolean true

var\_dump($x > $y); // Kết quả: boolean false

var\_dump($x <= $y); // Kết quả: boolean true

var\_dump($x >= $y); // Kết quả: boolean false

?>

-Các toán tử tăng và giảm

vd: ++$x, $x++, --$x, $x--

<?php

$x = 10;

echo ++$x; // Outputs: 11

echo $x; // Outputs: 11

$x = 10;

echo $x++; // Outputs: 10

// Vì trả về kết quả rồi mới tăng $x lên 1 nên:

echo $x; // Outputs: 11

$x = 10;

echo --$x; // Outputs: 9

echo $x; // Outputs: 9

$x = 10;

echo $x--; // Outputs: 10

// Vì trả về kết quả rồi mới giảm $x đi 1 nên:

echo $x; // Outputs: 9

?>

-Toán tử Logic

and, or, xor, &&, ||, !

-Toán tử chuỗi

vd: ., .=

<?php

$x = "Hello";

$y = " World!";

echo $x . $y; // Kết quả: Hello World!

$x .= $y;

echo $x; // Outputs: Hello World!

?>

9. Thứ tự toán tử trong PHP

-Loại unary: --, ++, !

-Loại tính nhân: \*, /, %

-Tính cộng: +, -

-Quan hệ: <, <=, >, >=

-Tính bằng: ==, !=

-LogicAND: &&

-LogicOR: ||

-Điều kiện: ?:

-Gán: %=, /=, \*=, -=, +=, =

10. Date, time

11. Tạo trang web động