# **CÁC THÀNH PHẦN CƠ BẢN TRONG PHP (09)**

## 1. Từ khóa và kí hiệu trong PHP

### **1.1 Từ khóa**

Trong PHP có vài chục từ khóa và hằng số được định nghĩa trước, không được đặt tên hằng số trùng với những từ khóa này. Các từ khóa được viết theo một quy tắc nhất định để tạo ra chương trình PHP.

Ví dụ từ khóa for để tạo lệnh vòng lặp, từ khóa if để tạo ra cấu trúc dẽ nhánh, include để gộp file mã nguồn ...

Danh sách đầy đủ các từ khóa: \_\_halt\_compiler, abstract, and, array, as, break, callable, case, catch, class, clone, const, continue, declare, default, die, do, echo, else, elseif, empty, enddeclare, endfor, endforeach, endif, endswitch, endwhile, eval, exit, extends, final, for, foreach, function, global, goto, if, implements, include, include\_once, instanceof, insteadof, interface, isset, list, namespace, new, or, print, private, protected, public, require, require\_once, return, static, switch, throw, trait, try, unset, use, var, while, xor

### **1.2 Kí hiệu**

Để viết mã PHP sử dụng các ký tự - ký hiệu sau:

Các ký tự chữ và số: A ... Z, a .. z, 0 .. 9 kể cả các ký tự Unicode được dùng để đặt tên các biến, hàm ...

<?php

$họ\_và\_tên = "Nguyễn Văn A";

echo $họ\_và\_tên;

?>

Các ký hiệu toán học: +, -, \*, /, =, >, <, (, )

<?php

$ketqua = (5 + 2.2) / 3;

echo $ketqua;

?>

Các ký hiệu đặc biệt như: ., [], $, ký hiệu xuống dòng \n, ký hiệu tab \t

<?php

$ho = "Nguyễn Văn";

$ten = "A";

$tendaydu =$ho . " " . $ten;

echo $tendaydu;

?>

## 2. Các kiểu dữ liệu sơ cấp

Các kiểu dữ liệu nguyên tố định nghĩa trước trong PHP gồm có: boolean, integer, float (double), string

### **2.1. Kiểu số nguyên**

Kiểu số nguyên biểu diễn các số trong tập N = {..-2, -1, 0, 1, 2 ...}

Giá trị lớn nhất, nhỏ nhất nó biểu diễn được ở dạng số nguyên tùy thuộc vào hệ thống đang chạy PHP (32 hay 64 bit), kiểm tra bằng hằng số PHP\_INT\_MAX và PHP\_INT\_MIN

<?php

echo "Số nguyên lớn nhất: " . PHP\_INT\_MAX;

echo "\n";

echo "Số nguyên nhỏ nhất: " . PHP\_INT\_MIN;

?>

### **2.2. Kiểu dấu phảy chấm động (số thực)**

Biểu diễn số thực (có phần nguyên và phần thập phân sau dấu chấm), nhập một số thực sử dụng theo cách:

* Số thập phân: 1.20
* Số dạng khoa học: 2E3, 2E-3

### **2.3. Kiểu chuỗi ký tự (sâu ký tự, text)**

Một chuỗi ký tự là tập hợp các ký tự để biểu diễn một văn bản. Một ký tự mặc định là chiếm một byte bộ nhớ (một ký tự ngoài 256 ký tự chuẩn, Unicode chiếm 2byte)

Biểu diễn chuỗi ký tự hoặc dùng cắp nháy đơn "" hoặc nháy kép ''

### **2.4. Kiểu logic**

Kiểu này để thực hiện các phép toán liên quan đến logic, nó nhận giá trị true hoặc falsemang ý nghĩa đúng, sai.

<?php

$ketqua = 5 > 4;

var\_dump($ketqua);

?>

## 3. Biến, hằng, biểu thức

### **3.1. Phân loại, khai báo và sử dụng biến**

Biến là cách lưu trữ thông tin trong chương trình PHP. Những điều cơ bản về biến của PHP gồm có:

* Tất cả các biến của PHP đều có tên bắt đầu bằng ký tự dollar ($). Ví dụ $a, $name, $class ...
* Giá trị của biến là giá trị được gán sau cùng bằng toán tử gán =. Ví dụ $a = 5;
* Trong PHP có thể khai báo biến trước khi gán nhưng không bắt buộc. Nhưng nhớ là khi sử dụng trong biểu thức thì biến đã được khai báo hoặc đã được gán.
* Biến trong PHP không bị cố định sử dụng với một kiểu dữ liệu nào đó, nó có thể là số, ký tự, sâu ký tự. Không như C# khi khai báo biến là số nguyên mà gán cho biến đó sâu ký tự thì là lỗi, nhưng PHP không có cơ chế này. Điều này được cái linh hoạt nhưng lại dễ để lại bugs.
* Một biến được sử dụng trước khi gán giá trị cho nó thì nó có các giá trị mặc định.
* PHP có cơ chế tự động chuyển kiểu dữ liệu tùy theo tình huống sử dụng. Ví dụ biến $a = 5 ($a lưu trữ số nguyên 5) thì tùy cách sử dụng sau này có thể hiểu $a là số 5, $ là ký tự '5', $a là sâu ký tự: $b = $a+5; $c = $a.'10';

#### **Quy tắc đặt tên biến**

Ngoài ký hiệu $ cho biết sau đó là một tên biến, thì tên biến có các quy tắc sau:

* Bắt đầu bằng mọt ký tự chữ(A-z) hoặc bàng dấu \_ (gạch dưới): Ví dụ các biến lợp lệ $a, $Abc, $\_a; không hợp lệ như $1a;
* Không được sử dụng các ký tự đặc biệt:+,-,%,& ...

**Phân loại biến**

Phạm vị có thể hiệu là vùng mà biến có hiệu lực. PHP có 4 loại phạm vị sau:

* Các biến cục bộ
* Các tham số của hàm
* Các biến toàn cục
* Các biến tĩnh

Biến cục bộ

Biến khai báo bên trong một hàm là cục bộ, biến đó chỉ sử dụng trong nội bộ hàm. Nếu gán giá trị cho biến ở phạm vi bên ngoài khai báo hàm thì không có ý nghĩa gì cả.

<?php

$x = 100;

function abc() {

$x = 20;

echo "Biến x trong hàm là $x \n";

}

abc();

echo "Biến x ngoài hàm là $x \n";

?>

Biến toàn cục

Khác với biến cục bộ, biến toàn cục truy cập được từ nhiều nơi trong chương trình. Khai báo và sử dụng biến toàn cục thì khi khai báo dùng từ khóa GLOBAL.

<?php

$x = 100;

function abc() {

global $x;

echo "Biến x là $x \n";

}

abc();

?>

Biến tĩnh

Biến tĩnh (static), các biến thông thường khai báo trong hàm nó sẽ bị hủy khi hàm thi hành xong (kết thúc), tuy nhiên nếu khai báo là biến tĩnh thì biến này không bị hủy, có thể dùng lại ở lần gọi tiếp theo của hàm. Khai báo bằng từ khóa static trước tên biến.

<?php

function demso() {

static $so = 0;

$so = $so + 1;

echo $so . PHP\_EOL;

}

demso();

demso();

demso();

?>

Tham số hàm

Tham số hàm cũng là một loại biến cục bộ, trình bày chi tiết ở phần học về hàm.

### **3.2. Hằng trong PHP**

Hằng số là một tên hay định danh cho một giá trị đơn giản. Gọi là hằng số bởi vì giá trị ứng với tên hằng sau khi định nghĩa là không đổi trong quá trình thi hành script. Mặc định hằng số có phân biệt chữ hoa chữ thường. Thông thường viết code thì ta nên đặt tên hằng số bằng các ký tự viết hoa. Một hằng số phải bắt đầu bằng một ký tự chữ  (a-Z) hoặc gạch dưới (\_), theo sau không được chứa các ký tự đặc biệt.

Khai báo hằng số với cú pháp

define('ten\_hang\_so', giá-trị);

Giá trị gán cho hằng số có thể là số, chuỗi ...

<?php

define('MON', 'Thứ Hai');

define("TUE", 'Thứ Ba');

echo MON;

?>

**Sự khác nhau giữa hằng số và biến**

* Tên hằng số thi không cần bắt đầu bằng ký tự dollar $ như biến.
* Hằng số không thể sử dụng phép gán giá trị như biến (=), giá trị của nó phải được xác định duy nhất tại hàm define.
* Hằng số có thể truy cập bất cứ ở đâu mà không giới hạn phạm vị như biến.
* Ngay sau khi giá trị hằng thiết lập, nó không bị thay đổi hay định nghĩa lại nữa.

Một số hằng số định nghĩa sẵn trong PHP

| **Tên hằng** | **Mô tả** |
| --- | --- |
| \_\_LINE\_\_ | Dòng hiện tại của php file. |
| \_\_FILE\_\_ | Tên file đầy đủ của script đang thi hành. |
| \_\_DIR\_\_ | Tên thư mục đầy đủ của file PHP |
| \_\_FUNCTION\_\_ | Lấy tên của hàm đang chạy |
| PHP\_EOL | Ký tự xuống dòng |
| M\_PI | Số PI |

### **3.3. Biểu thức PHP**

Biểu thức là một sự kết hợp giữa các toán tử (operator) và các toán hạng (operand) theo đúng một trật tự nhất định.

Mỗi toán hạng có thể là một hằng, một biến hoặc một biểu thức khác.

Trong trường hợp, biểu thức có nhiều toán tử, ta dùng cặp dấu ngoặc đơn () để chỉ định toán tử nào được thực hiện trước.

Ví dụ biểu thức nghiệm của phương trình bậc 2: (-$b + sqrt($delta)) / (2 \* $a)

**Toán tử số học**

| **Toán tử** | **Diễn tả** | **Ví dụ** |
| --- | --- | --- |
| + | Phép cộng hai số | $A + $B : 63 + 70 = 133 |
| - | Phép trừ | $A - $B : 63 - 70 = -7 |
| \* | Phép nhân | $A \* $B : 63 \* 70 = 4410 |
| / | Phép chia | $A / $B : 63 / 70 = 0.9 |
| ++ | Phép toán tăng thêm 1 vào biến | $A++ kết quả $A = 64 |
| -- | Phép toán giảm đi giá trị 1 | $B-- kết quả $B = 69 |

<?php

$a = 10;

$b = 20;

$c = $a + $b;

echo "Kết quả phép cộng: $c" . PHP\_EOL;

$c = $a - $b;

echo "Kết quả phép trừ: $c". PHP\_EOL;

$c = $a \* $b;

echo "Kết quả phép nhân: $c". PHP\_EOL;

$c = $a / $b;

echo "Kết quả phép chia: $c". PHP\_EOL;

$c = $a % $b;

echo "Phần dư phép chia: $c". PHP\_EOL;

$c = $a++;

echo "Thêm một vào $c: $a". PHP\_EOL;

$c = $a--;

echo "Giảm $a đi 1: $c". PHP\_EOL;

?>

**Toán tử so sánh**

Toán tử so sánh trả về kết quả kiểu logic bool (nhận giá trị true hoặc false)

| **Toán tử** | **Diễn tả** | **Ví dụ** |
| --- | --- | --- |
| == | So sánh bằng: true nếu hai toán hạng bằng nhau và falsenếu khác | $C == $D ≡ 50 == 27 ≡ false |
| != | So sánh khác:  nếu hai toán hạng có giá trị khác nhau thì có giá trị true, giá trị giống nhau thì là false | $C != $D ≡ 50 != 27 ≡ true |
| > | So sánh lớn hơn: true nếu toán hạng thứ nhất lớn hơn toán hạng thứ hai | $C > $D ≡ 50 > 27 ≡ true |
| >= | So sánh lớn hơn hoặc bằng: true nếu toán hạng thứ nhất lớn hơn hoặc bằng toán hạng thứ hai | $C >= $D ≡ 50 >= 27 ≡ true |
| < | So sánh nhỏ hơn: true nếu toán hạng thứ nhất nhỏ hơn toán hạng thứ hai | $C > $D ≡ 50 < 27 ≡ false |
| <= | So sánh nhỏ hơn hoặc bằng: true nếu toán hạng thứ nhất nhỏ hơn hoặc bằng toán hạng thứ hai | $C <= $D ≡ 50 <= 27 ≡ false |

<?php

$a = 10;

$b = 20;

if ($a > $b)

echo "$a lớn hơn $b";

else

echo "$a nhỏ hơn $b";

?>

**Toán sử logic**

| **Toán tử** | **Diễn tả** | **Ví dụ** |
| --- | --- | --- |
| and | Toán tử và: true nếu hai toán hạng có giá trị true | ($A and $B) ≡ (52 and 14) ≡ (true and true) ≡ true |
| or | Toán tử hoặc: true nếu một trong hai toán tử là true | ($A or $B) ≡ (52 or 14) ≡ (true or true) ≡ true |
| && | Toán tử và: true nếu hai toán hạng có giá trị true | ($A && $B) ≡ (52 && 14) ≡ (true && true) ≡ true |
| || | Toán tử hoặc: true nếu một trong hai toán tử là true | ($A || $B) ≡ (52 || 14) ≡ (true || true) ≡ true |
| ! | Toán NOT (phủ định): true nếu giá trị là false | !($A || $B) ≡ !(52 || 14) ≡ !(true || true) ≡ false |

<?php

$A = true;

$B = false;

$ketqua = $A && $B;

var\_dump($ketqua);

?>

## 4. Cấu trúc một chương trình

### **4.1. Thông dịch trong PHP**

Mã nguồn PHP không biên dịch thành mã máy khi phân phối, nó được một trình thông dịch PHP nạp mà biên dịch thành mã máy ở thời điểm thực thi. Chương trình thông dịch chạy PHP này được cài đặt ở phần chương I.

Do PHP là ngôn ngữ thông dịch, do đó khi phân phối sản phẩm chính là phân phối mã nguồn PHP

### **4.2. Cấu trúc**

Một dự án (ứng dụng) PHP được chứa bên trong một thư mục, trong thư mục này có thể tạo ra các thư mục con, từ đó lưu các file mã nguồn với phần mở rộng là .php

Khi chương trình thông dịch nạp mã PHP, nó sẽ bắt đầu thực thi các đoạn mã chứa trong thẻ php

<?php

// mã PHP

?>

Mã nguồn PHP có thể chứa trên nhiều file .php khác nhau. Một file PHP này có thể gộp mã của file PHP khác bằng lệnh include "tenfile.php"

<?php

include "thuvien/thuvien1.php";

?>

**Chú thích cho Code PHP**

Khi viết code PHP, có nhiều lúc cần chú thích (comment) để sau này đọc lại dễ hiểu, các dòng chủ thích này không tác động gì đến mã PHP cả, bị bỏ qua (không hiện thị, không thi hành) khi mã được chạy. Có hai loại chú thích, chú thích một dòng và chú thích nhiều dòng.

*Chú thích một dòng*

Các ký tự trên một dòng đặt sau khí hiệu // hoặc ký hiệu #

<?php

// chú thích

# chú thích

echo "Hi";

?>

*Chú thích nhiều dòng*

Nội dung chú thích đặt trong cặp /\* ... \*/

<?php

/\*

Chú thích 1

Chú thích 2

\*/

echo "Hi";

?>

Các khoảng trắng không ảnh hưởng đến mã PHP

## 5. Câu lệnh PHP

### **5.1. Khái niệm về câu lệnh**

Một câu lệnh (statement) xác định một công việc mà chương trình phải thực hiện để xử lý dữ liệu đã được mô tả và khai báo. Các câu lệnh được ngăn cách với nhau bởi dấu chấm phẩy (;).

Ví dụ câu lệnh sau, gán một chuỗi ký tự vào biến tên $welcome

$welcome = 'Chào mừng đến với PHP';

**Khối lệnh trong PHP**

Các câu lệnh có thể cho vào một khối bằng cách sử dụng dấu { và }.

<?php

$pass = true;

if ($pass)

{

echo "Chúc mừng bạn";

echo "bạn đã hoàn thành bài kiểm tra";

}

?>

### **5.2. Lệnh gán**

Lệnh gán được dùng để gán giá trị, biểu thức vào một biến

Các toán tử gán gồm: **=, +=, -=, \*=, /=**

| **Toán tử** | **Diễn tả** | **Ví dụ** |
| --- | --- | --- |
| = | Toán tử gán: Gán biểu thức bên phải của = vào biến bên trái | $C = ($A + $B) kết quả $C bằng 66 |
| += | Toán tử công thêm: Công thêm vào biến bên trái += giá trị bên phải | $C = 11; $C += ($A + $B) kết quả $C bằng 77 |
| -= | Toán tử trừ bớt: Bớt đi giá trị biến bên trái của -= một lượng bằng biểu thức bên phải | $C = 60; $C -= ($A + $B) kết quả $C bằng -6 |
| \*= | Toán tử nhân với: $A \*= $B tương đương với $A = $A \* $B | $A \*= $B kết quả $A bằng 728 |
| /= | Toán tử chia cho: $A /= $B tương đương với $A = $A / $B | $A /= $B kết quả $A bằng 3.7142857142857 |

### **5.3. Nhập và xuất dữ liệu**

**Lệnh echo**

Lệnh echo để in ra nội dung dòng chữ, biến ... Nó là một phần của cấu trúc PHP nên không cần gọi bằng quy tắc hàm thông thường. Ví dụ in ra dòng chữ

<?php

echo "Lập trình PHP";

?>

**Hàm print** : tương tự echo cũng in ra màn hình dòng chữ

<?php

print("Lập trình PHP");

?>

**Lệnh readline** : đọc một dòng dữ liệu (đọc đến khi nhấn Enter)

<?php

echo "TÍNH CĂN BẬC 2" . PHP\_EOL;

$a = readline("Nhập số cần tính: ");

$can = sqrt($a);

echo "Căn của $a là $can";

?>

Có thể dùng hàm fgets(STDIN) để đọc dòng dữ liệu.