

# CÔNG NGHỆ WEB



## *Chương 2 : PHP cơ bản*

hoantq@hnue.edu.vn

# CÔNG NGHỆ WEB



Quy tắc về cú pháp

Toán tử

Định danh

Cấu trúc điều khiển

“

*Quy tắc về cú pháp*

## Quy tắc dấu

- Dấu cho phép viết mã PHP:
  - `<?php ?>`
  - `<? ?>`
- Dấu kết thúc lệnh trong PHP: **;**
- Dấu tạo ghi chú 1 dòng trong PHP: **//**
- Dấu tạo ghi chú nhiều dòng trong PHP: **/\*...\*/**
- Dấu tạo chuỗi trong PHP: **'...'** hoặc **"..."**
- Dấu nối chuỗi trong PHP: **,** hoặc **.**
- Dấu khai báo biến trong PHP: **\$**

## Hằng trong PHP

### 1) Hằng có sẵn trong PHP :

```
1 <?php
2     TRUE , FALSE ; // có giá trị tương ứng là 1 và 0
3     PHP_VERSION ; // cho biết phiên bản PHP trong server của bạn
4     PHP_OS ;      // cho biết server của bạn là linux hay window
5 ?>
```

### 2) Hằng do chúng ta tự định nghĩa :

**Cú pháp : define('tên hằng','giá trị');**

```
1 <?php
2     define("NAME","PHPANDMYSQL.NET"); // định nghĩa hằng có tên là NAME
3     echo NAME; // xuất ra màn hình "PHPANDMYSQL.NET"
4 ?>
```

## Biến trong PHP

### Cú pháp : \$Tên\_biến

#### Lưu ý :

- Biến phải bắt đầu bằng dấu \$ sau đó là các ký tự chữ cái hoặc dấu gạch dưới
- Không được chứa dấu cách trong biến
- Không được đặt tên biến bắt đầu bằng chữ số
- Biến phân biệt chữ hoa chữ thường
- Biến không được trùng với hàm có sẵn trong PHP

## Biến trong PHP

### Ví dụ

```
12 <b>KHAI BÁO BIẾN<br></b>
13 <?php
14 $so_nguyen = 1;
15 $so_thuc=12.34;
16 $chu = "Bòm";
17 echo $so_nguyen;
18 echo "<br>".$chu."<br>";
19 echo $so_thuc."<br>";
20 $so_nguyen=5;
21 echo $so_nguyen."<br>";
22 echo '$chu có $so_nguyen viên kẹo';
23 echo "<br>";
24 echo "$chu có $so_nguyen viên kẹo";
25 // $12var=12;
26 // Biến ko được bắt đầu bằng chữ số
27 /* $c=012;
28 echo "<br>".$c;
29 $d=0x12;
30 echo "<br>".$d;
31 */
32 ?>
```

## KHAI BÁO BIẾN

1

Bòm

12.34

5

\$chu có \$so\_nguyen viên kẹo

Bòm có 5 viên kẹo

## Biến trong PHP

- Biến động (biến biến): Cho phép sử dụng giá trị của biến làm tên biến khác

```
<?php
    $bien1="name";
    $$bien1="Loan";
    echo $name;
?>
```

## Ví dụ

**Biến động**  
Loan



## Các kiểu dữ liệu

- ▶ Kiểu dữ liệu đơn : Có 4 kiểu
  - ▶ Số nguyên
  - ▶ Số thực
  - ▶ Chuỗi
  - ▶ Logic
- ▶ Kiểu dữ liệu đa giá trị : Có 2 kiểu
  - ▶ Mảng
  - ▶ Đối tượng
- ▶ Kiểu dữ liệu tài nguyên : Sử dụng khi tương tác dữ liệu
- ▶ Kiểu dữ liệu rỗng : NULL

Integer

Double

Boolean

Null

String

Array

Object

## Kiểu số và logic

- ▶ Kích thước của kiểu INT là 32 bit nên có dải biểu diễn :

$-2^{31}$  đến  $2^{31} - 1$

- ▶ Kiểu float (số thực) từ  $1.7E-308$  đến  $1.7E+308$ 
  - ▶ Ví dụ : \$a = 0.17;
  - ▶ \$b = 12.38;
- ▶ Kiểu Boolean có 2 trạng thái :  
Hoặc **True** hoặc **False**

```
1 <?php
2 $a= 123; //Số thập phân
3 $b=-123; //Số âm
4 $c= 012; //Số bát phân
5 $d= 0x12; //Số thập lục phân
6 echo $a.'<br>';
7 echo $b.'<br>';
8 echo $c.'<br>';
9 echo $d.'<br>';
10 ?>
```

## Kiểu chuỗi (String)

- ▶ Giới hạn trong dấu
  - ▶ nháy đơn ' ' (chuỗi dữ liệu không và không có nhu cầu sử dụng các ký tự chuỗi đặc biệt)
  - ▶ hoặc nháy kép " " (chuỗi cần sử dụng các ký tự đặc biệt hoặc muốn khai báo tên biến vào mà không cần nối chuỗi)

### Ví dụ

#### Biến kiểu String

Các bạn gái Khoa \$x rất xinh đẹp

Các bạn gái Khoa CNTT rất xinh đẹp

```
$x='CNTT';  
$chuoil='Các bạn gái Khoa $x rất xinh đẹp';  
$chuoil2="Các bạn gái Khoa $x rất xinh đẹp";  
echo $chuoil."<br>";  
echo $chuoil2."<br>";
```

```
echo $chuoil."<br>";
```

## Nối chuỗi

- Sử dụng dấu chấm (.) hoặc dấu (,) để nối lại với nhau

- Ký tự đặc biệt :

- \n: In chuỗi với một dòng
- \t: In chuỗi với một tab
- \r: Trở về đầu dòng

```
<?php
$text1="Nguyễn Quỳnh Tâm. ";
$text2=" Lê Ngọc Tú";
echo $text1.$text2."<br>";
echo $text1,$text2;
?>
```

### Ví dụ

#### Nối chuỗi

Nguyễn Quỳnh Tâm. Lê Ngọc Tú  
Nguyễn Quỳnh Tâm. Lê Ngọc Tú

## Chuỗi nhiều dòng

### Cú pháp Here Document (Heredoc) trong PHP

```
1 <?php
2 $tuoim = <<<EOD
3 Các lĩnh vực chính của công nghệ thông tin bao gồm quá trình tiếp thu, xử lý, lưu trữ và phổ
  biến hóa âm thanh, phim ảnh, văn bản và thông tin số bởi các vi điện tử dựa trên sự kết
  hợp giữa máy tính và truyền thông. Một vài lĩnh vực hiện đại và nổi bật của công nghệ
  thông tin như: các tiêu chuẩn Web thế hệ tiếp theo, sinh tin, điện toán đám mây, hệ
  thống thông tin toàn cầu, trí thức quy mô lớn và nhiều lĩnh vực khác. Các nghiên cứu
  phát triển chủ yếu trong ngành khoa học máy tính.
4 EOD;
5 echo $tuoim;
6 ?>
```

Ép kiểu dữ liệu

Cú pháp :`(int)` \$Tên\_biến

Ký hiệu	Ý nghĩa kiểu
<code>(int)</code> , <code>(integer)</code>	Số nguyên
<code>(real)</code> , <code>(double)</code> , <code>(float)</code>	Số thập phân
<code>(string)</code>	Chuỗi
<code>(array)</code>	Mảng
<code>(object)</code>	Đối tượng
<code>(bool)</code> , <code>(boolean)</code>	Logic
<code>(unset)</code>	NULL, tương tự như gọi <code>unset()</code>

## Kiểm tra kiểu dữ liệu

- Hàm **is\_int(\$Tên\_bi\_ến)** kiểm tra xem một biến có phải là kiểu INT hay không

```
<?php
    $tuoi='23';//Biến tuoi là kiểu string có
    giá trị là '23'
    $tuoi=(int)$tuoi;//Lúc này biến tuoi là
    một int có giá trị là 23
?>
```



## Toán tử

## Toán tử

- Toán tử số học
- Toán tử bit
- Toán tử gán
- Toán tử logic
- Toán tử tăng giảm
- Toán tử so sánh



## Toán tử gán và toán tử số học

### ▸ Toán tử gán

```
<?php
$name="Mã Đức Chung";
$x=10;|
?>
```

### ▸ Toán tử số học

Tên	Ký hiệu	Mô tả	Ví dụ
Phép cộng	+	Cộng hai số	$\$a + \$b$
Phép trừ	-	Trừ hai số	$\$a - \$b$
Phép nhân	*	Nhân hai số	$\$a * \$b$
Phép chia	/	Chia hai số	$\$a / \$b$
Phép chia lấy dư	%	Chia lấy dư	$\$a \% \$b$

## Toán tử kết hợp (tăng giảm)

++	$\$a++ \Rightarrow \$a = \$a + 1;$
--	$\$a-- \Rightarrow \$a = \$a - 1;$
+=	$\$a += \$b \Rightarrow \$a = \$a + \$b;$
-=	$\$a -= \$b \Rightarrow \$a = \$a - \$b;$
*=	$\$a *= \$b \Rightarrow \$a = \$a * \$b;$
/=	$\$a /= \$b \Rightarrow \$a = \$a / \$b;$

dụ :

## Toán tử quan hệ

Tên	Ký hiệu	Mô tả	Ví dụ
So sánh bằng	<code>==</code>	Hai số bằng nhau	<code>\$a==\$b</code>
So sánh khác	<code>!=</code>	Hai số khác nhau	<code>\$a!=\$b</code>
So sánh lớn hơn	<code>&gt;</code>	So sánh lớn hơn	<code>\$a&gt;\$b</code>
So sánh nhỏ hơn	<code>&lt;</code>	So sánh nhỏ hơn	<code>\$a&lt;\$b</code>
So sánh lớn hơn hoặc bằng	<code>&gt;=</code>	Lớn hơn hoặc bằng	<code>\$a&gt;=\$b</code>
So sánh nhỏ hơn hoặc bằng	<code>&lt;=</code>	Nhỏ hơn hoặc bằng	<code>\$a&lt;=\$b</code>

## Toán tử quan hệ

```
1 $a = 12; // Biến $a kiểu INT có giá trị = 12
2 $t = ($a == 12); // Biến $t có giá trị là TRUE vì biểu thức (12 == 12) đúng
3 $t = ($a > 12); // Biến $t có giá trị là FALSE vì biểu thức (12 > 12) sai
4 $t = ($a >= 12); // Biến $t có giá trị TRUE vì biểu thức (12 >= 12) đúng
5 $t = ($a != 12); // Biến $t có giá trị FALSE vì biểu thức (12 != 12) sai
```

## Độ ưu tiên của các toán tử

Loại	Toán tử	Thứ tự ưu tiên
Unary	! ++ --	Phải sang trái
Tính nhân	* / %	Trái sang phải
Tính cộng	+ -	Trái sang phải
Quan hệ	< <= > >=	Trái sang phải
Tính bằng	== !=	Trái sang phải
Logic AND	&&	Trái sang phải
Logic OR		Trái sang phải
Điều kiện	?:	Phải sang trái
Gán	= += -= *= /= %=	Phải sang trái

## Toán tử luận lý

Tên	Ký hiệu	Mô tả	Ví dụ
Phép và	&& hoặc and	Cả hai vế phải thỏa mãn điều kiện	$a > b$ && $a > c$
Phép hoặc	hoặc or	Một trong hai thỏa mãn điều kiện là được	$a > b$    $a > c$
Phủ định	!	Phủ định một điều kiện, giá trị nào đó	$a \neq b$

```
1 $a = 100;
2 $b = 200;
3 $tong = $a + $b;
4 $check = ($a < $b) && ($tong > 200);
```

## Toán tử mảng

- Chú ý khi gộp 2 mảng có **key** giống nhau thì thứ tự toán hạng sẽ thay đổi kết quả

```
15 $array1=['a'=>"apple",'b'=>'ball','c'=>'Công nghệ thông tin'];
16 $array2=['c'=>'CNTT','d'=>'doll'];
17 print_r($array1+$array2);
18 /*([a]=>apple
19      [b]=>ball
20      [c]=>Công nghệ thông tin
21      [d]=>doll
22      )*/
23 print_r($array2+$array1);
24 /*([c]=>CNTT
25      [d]=>doll
26      [a]=>apple
27      [b]=>ball
28      )*/
```

## Toán tử mảng

- Trong **\$array4**, phần tử được gán các phần tử được chỉ định xuất hiện theo một **thứ tự** khác với **\$array3**; PHP lưu trữ chúng khác nhau.

```
30 $array3=['0'=>"apple",1=>'ball'];
31 $array4=["apple",'ball'];
32 var_dump($array3==$array4);//bool(true)
33 var_dump($array3=== $array4);//bool(false)
```



## Toán tử so sánh

- ▶ Toán tử so sánh logic trong PHP có thứ tự ưu tiên có sự khác biệt giữa các biến thể của **AND** và **&&**; **OR** và **||**; **XOR** and **^**. Các ký hiệu **&&**, **||** và **^** có thứ tự ưu tiên cao hơn và khu vực được đánh giá sớm hơn trong một biểu thức so với các ký hiệu **AND**, **OR** và **XOR** của chúng

```
14 <?php
15 $value1=10;
16 $value2=20;
17 $value3=false;
18 $value4=false;
19
20 if($value1===10 and $value2===10||$value3=== $value4){
21     echo "True";
22 }else{
23     echo "False";
24 }
```

- ▶ đánh giá ở bên phải của **||** thực hiện đầu tiên, sau đó **AND** ở bên trái thực hiện. Điều này có thể dẫn đến một đánh giá không thực tế do được ưu tiên

“

*Cấu trúc điều khiển*

# *Cấu trúc điều khiển*

Điều  
khiển

- If...ELSE
- SWITCH

Vòng  
lặp

- For
- While
- Foreach

# *Cấu trúc điều kiện IF*

IF

IF...  
ELSE

IF...ELSE  
IF...ELSE

```
<?php
    $x=29;
    if($x>=0){
        echo "Số dương";
    }
?>
```

## Switch

```
▷ SWITCH ($variable) {  
    case $value_1;  
        //chuỗi câu lệnh 1  
        break;  
    case $value_2:  
        //chuỗi câu lệnh 2  
        break;  
    default :  
        //chuỗi câu lệnh  
        break;  
}
```

# SWITCH



- ▶ Viết chương trình nhập kiểm tra số đó nếu số kiểu số nguyên và là:
- ▶ 0 thì xuất dòng lệnh “Số không”
- ▶ 2 thì xuất dòng lệnh “Số hai”
- ▶ Các số nguyên còn lại thì xuất dòng lệnh “Không tìm thấy”

```
<?php
    $x=28;
    Switch ($x)
    {
        case 0 :
            $ketqua="Số không";
            break;
        case 2 :
            $ketqua="Số hai";
            break;
        default :
            $ketqua="Không tìm thấy";
            break;
    }
    echo $ketqua;
?>
```

## Vòng lặp

- ▶ Vòng lặp
  - ▶ Là một mã lệnh trong đó chương trình được thực hiện lặp đi lặp lại nhiều lần cho đến khi thỏa mãn điều kiện nào đó
- ▶ **Các loại vòng lặp**
  - ▶ Vòng lặp for
  - ▶ Vòng lặp while và do while
  - ▶ Vòng lặp foreach

# VÒNG LẶP FOR

```
1  for ($bien_dieu_khien; $bieu_thuc_dieu_kien; $bieu_thuc_thay_doi_bien_dieu_khien)
2  {
3      // lệnh
4  }
```

- \$bien\_dieu\_khien: gán giá trị ban đầu hoặc có sẵn
- \$bieu\_thuc\_dieu\_kien: xác định điều kiện thoát khỏi vòng lặp
- \$bieu\_thuc\_thay\_doi\_bien\_dieu\_khien: xác định biến điều khiển sẽ bị thay đổi ntn sau mỗi lần lặp



# VÒNG LẶP FOR

```
01  <?php
02  for ($x = 0; $x <= 20; $x++) {
03      echo "$x <br>";
04  }
05  ?>
```

- \$bien\_dieu\_khien: \$x=0
- \$bieu\_thuc\_dieu\_kien: \$x<=20
- \$bieu\_thuc\_thay\_doi\_bien\_dieu\_khien: \$x++

# VÒNG LẶP FOR LỒNG NHAU

```
1  for ($i = 1; $i < 10; $i++)  
2  {  
3      for ($j = 9; $j >= $i; $j--)  
4      {  
5          echo $j;  
6      }  
7  echo '<br/>';  
8  }
```

```
987654321  
98765432  
9876543  
987654  
98765  
9876  
987  
98  
9
```

# VÒNG LẶP FOR KẾT HỢP VỚI MẢNG

```
1 $sinhvien = array(  
2 'Nguyễn A',  
3 'Nguyễn B',  
4 'Nguyễn C',  
5 'Nguyễn D',  
6 'Nguyễn E',  
7 'Nguyễn F'  
8 );
```

```
1 echo $sinhvien[0];  
2 echo $sinhvien[1];  
3 echo $sinhvien[2];  
4 echo $sinhvien[3];  
5 echo $sinhvien[4];  
6 echo $sinhvien[5];
```

# CÔNG NGHỆ WEB



MẢNG

## Mảng

### ▸ KHAI BÁO :

```
$mang=array(key=>value);
```

### ▸ PHÂN LOẠI MẢNG :

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

## Mảng tuần tự

Là mảng có key là chữ số được bắt đầu bằng 0 và sắp xếp tăng dần

```
<?php
$ten = array ("Mai","Trang","Loan");
echo $ten[0]; //Lấy ra tên Mai
echo $ten[1]; //Lấy ra tên Trang
echo $ten[2]; //Lấy ra tên Loan
?>
```

Thêm phần tử vào mảng

```
<?php
$ten = array ("Mai","Trang","Loan");
$ten[]="Trúc"; //Được thêm vào vị
trí cuối cùng của mảng
echo $ten[3]; //Lấy ra Trúc
?>
```

## *Mảng không tuần tự*

Là mảng mà key của nó không sắp xếp theo thứ tự, nó là ký tự chữ

```
1 <?php
2 $thongtin=array("ten"=>"Mai Lan","tuổi"=>"22","giới tính"
   =>"Nữ","email"=>"mailan@gmail.com");
3 echo $thongtin["ten"];//Lấy ra Mai Lan
4 echo $thongtin["email"];//Lấy ra mailan@gmail.com
5 ?>
```

Thêm phần tử vào mảng

```
1 <?php
2 $thongtin=array("ten"=>"Mai Lan","tuổi"=>"22","giới tính"
   =>"Nữ","email"=>"mailan@gmail.com");
3 $thongtin["SĐT"]="0912423738";
4 echo $thongtin["SĐT"].|;
5 ?>
```

# Mảng đa chiều

Là một mảng lớn có nhiều mảng con nằm trong nó

```
<?php
$quanau=array(
    "xuất xứ"=>"Nhật",
    "Chất liệu"=>"Co giãn",
    "chi tiết"=>array(
        "màu"=>"xanh tím than",
        "size"=>"32",
        "giá"=>"500000")
    );
echo $quanau["chi tiết"]["màu"];|
?>
```



# *Các hàm sử dụng trong mảng*

<b>Sizepf(\$arr)</b>	In ra tổng số phần tử có bên trong mảng
<b>Array_values(\$arr)</b>	Tạo ra một mảng mới chứa toàn bộ giá trị của mảng đó
<b>Array_keys(\$arr)</b>	Bóc toàn bộ key trong mảng bỏ vào một mảng mới
<b>Each(\$arr)</b>	Để in ra một cặp key và giá trị của nó, thường dùng với vòng lặp while
<b>Array_reverse(\$arr)</b>	Đảo ngược thứ tự giá trị trong mảng
<b>Array_merge(\$arr1,\$arr2,...)</b>	Gộp các dữ liệu của hai hoặc nhiều mảng lại với nhau

## *Các hàm sắp xếp mảng*

<b>Sort()</b>	Xếp mảng theo thứ tự tăng dần
<b>Rsort()</b>	Xếp mảng theo thứ tự giảm dần
<b>Asort()</b>	Xếp mảng theo thứ tự tăng dần, dựa vào giá trị
<b>Ksort()</b>	Xếp mảng theo thứ tự tăng dần dựa vào key
<b>Arsort()</b>	Xếp mảng theo thứ tự giảm dần, dựa vào giá trị
<b>Krsort()</b>	Xếp mảng theo thứ tự giảm dần, dựa vào key

# VÒNG LẶP WHILE, DO WHILE TRONG TRUY XUẤT MẢNG

```
1 // Cho Danh Sách Năm
2 $nam = array(
3     1990,
4     1991,
5     1992,
6     1993,
7     1994,
8     1995
9 );
```

```
11 // Xuất theo cách thông thường
12 echo $nam[0];
13 echo $nam[1];
14 echo $nam[2];
15 echo $nam[3];
16 echo $nam[4];
17 echo $nam[5];
```

```
19 // Dùng while
20 $i = 0;
21 while ($i <= 5){
22     echo $nam[$i];
23     $i++; // Tăng biến $i
24 }
25
26 // Dùng do .. while
27 $i = 0;
28 do {
29     echo $nam[$i];
30     $i++;
31 }while ($i <=5);
```

# Lặp trong mảng

```
1 // Danh sách các năm
2 $nam = array(
3     1990,
4     1991,
5     1992,
6     1993,
7     1994,
8     1995
9 );
```

```
1 for ($i = 0; $i < 6; $i++){
2     echo $sinhvien[$i];
3 }
```

```
1 $count = count($sinhvien);
2 for ($i = 0; $i < $count; $i++){
3     echo $sinhvien[$i];
4 }
```

# FOREACH

```
1 | foreach ($array as $key => $value){  
2 |     // Các dòng lệnh  
3 | }
```

▶ Hoặc

```
1 | foreach ($array as $value){  
2 |     // Các dòng lệnh  
3 | }
```

- ▶ `$array` là mảng cần lặp
- ▶ `$key` là số chỉ mục
- ▶ `$value` là giá trị của phần tử ở vị trí `$key`

## Luyện tập

```
1 // Danh sách các năm
2 $nam = array(
3     1990,
4     1991,
5     1992,
6     1993,
7     1994,
8     1995
9 );
```

```
//Dùng foreach xuất ra các năm trong $nam
foreach ($nam as $key => $value){
    echo $value;
}
```

## Luyện tập

```
//Dùng foreach xuất ra các năm trong $nam  
foreach ($nam as $chimuc => $giatri){  
    echo $chimuc . ' => ' . $giatri;  
}
```

```
//Dùng foreach xuất ra các năm trong $nam  
foreach ($nam as $value){  
    echo $value;  
}
```

1	1990
2	1991
3	1992
4	1993
5	1994
6	1995