

TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI

Chương 2

PHP căn bản

Nguyễn thị Quỳnh Hoa - Khoa CNTT- ĐH Sư phạm Hà Nội

NỘI DUNG

- Chú thích, hằng, biến trong PHP
- Các kiểu dữ liệu
- Các kiểu toán tử
- Cấu trúc điều khiển
- Hàm và cách sử dụng hàm trong PHP
- Phạm vi biến
- Làm việc với file trong PHP

Chú thích trong PHP

- `<?PHP`
- `//`Dòng chú thích
- `#`Dòng chú thích
- `/*`Chú thích nhiều dòng
Chú thích nhiều dòng

.....

`*/`

`?>`

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

Ví dụ



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

Kiểu số (INT)


- Kích thước của kiểu INT là 32 bit nên có dải biểu diễn :
 -2^{31} đến $2^{31} - 1$

```
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 số thực (Float)

- Từ $1.7E-308$ đến $1.7E+308$
- Ví dụ : `$a = 0.17;`
- `$b= 12.38;`

Kiểu chuỗi (String)

- Giới hạn trong dấu nháy đơn ‘ ‘ hoặc nháy kép “ “
- Dùng dấu nháy đơn khi chuỗi dữ liệu không chứa các ký tự đặc biệt và không có nhu cầu sử dụng các ký tự chuỗi đặc biệt
- Sử dụng dấu nháy đôi khi chuỗi cần sử dụng các ký tự đặc biệt hoặc muốn khai báo tên biến vào mà ko cần nói chuỗi
- Ký tự đặc biệt :
 - \n: In chuỗi với một dòng
 - \t : In chuỗi với một tab
 - \r : Trở về đầu dòng
- Ví dụ : 

Nối chuỗi

- Trong PHP, khi bạn cần nối chuỗi dữ liệu với một biến, hàm hoặc hằng thì sử dụng dấu chấm (.) hoặc dấu (,) để nối lại với nhau

- Ví dụ

```
<?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

Kiểu Logic (Booleans)

- Có 2 trạng thái : Hoặc **True** hoặc **False**

Kiểu mảng (Array)

- 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

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

Các hàm sắp xếp mảng

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

Ép kiểu và kiểm tra kiểu

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

Ép dữ liệu sang kiểu INT

Cú pháp : (int) \$Tên_biến

- Hàm **is_int(\$Tên_biến)** 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
?>
```

- Ví dụ :
- Kết quả :

Câu hỏi?

- 127.0.0.1 và <http://localhost> ?
- Những thành phần cần thiết nào tạo nên một trang web động?

Các toán tử

1. Toán tử gán
2. Toán tử số học
3. Toán tử so sánh
4. Toán tử logic
5. Toán tử kết hợp

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	$a\%b$

Ví dụ :
Kết quả

Toán tử kết hợp (Tăng giảm)

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

Ví dụ :
Kết quả

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>></code>	So sánh lớn hơn	<code>\$a>\$b</code>
So sánh nhỏ	<code><</code>	So sánh nhỏ hơn	<code>\$a<\$b</code>
So sánh lớn hơn hoặc bằng	<code>>=</code>	Lớn hơn hoặc bằng	<code>\$a>=\$b</code>
So sánh nhỏ hơn hoặc bằng	<code><=</code>	Nhỏ hơn hoặc bằng	<code>\$a<=\$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
```

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);
```

ĐỘ ƯU TIÊN TOÁN TỬ' LUẬN LÝ

- Theo thứ tự sau : NOT->AND->OR
- Ví dụ Xét biểu thức sau có giá trị là gì?
- $10 > 7 \&\& !(-4 > 9) || 100 == 100$
- Viết đoạn code trên

ĐỘ ƯU TIÊN CÁC TOÁN TỬ

- Bảng thứ tự ưu tiên của toán tử số học

Loại toán tử	Toán tử	Tính kết hợp
Một ngôi	$-$, $++$, $--$	Phải sang trái
Hai ngôi	\wedge	Trái sang phải
	$*$, $/$, $\%$	
	$+$, $-$	
	$=$	Phải sang trái

- Ví dụ : $-29*5-66$
- $==$ và $===$ khác nhau ntn?

Cấu trúc điều khiển

- Cấu trúc rẽ nhánh
- Cấu trúc vòng lặp

Câu điều kiện?

- Viết cho bất kỳ ngôn ngữ nào
- Thực hiện những hành động khác nhau trong những điều kiện khác nhau

CẤU TRÚC Rẽ NHÁNH TRONG PHP

- Câu điều kiện IF... ELSE
- Câu điều kiện SWITCH

CÂU ĐK IF

- IF
- IF... ELSE
- IF...ELSE IF...ELSE

IF....

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

- Kết quả
- Bài tập :
 1. Viết đoạn mã PHP xem một số có phải là số chẵn hay ko
 2. Kiểm tra xem biến nhập vào có phải kiểu string hay ko

IF...ELSE

- Bài tập : Viết đoạn mã PHP xem một số có phải là số chẵn hay ko
- Ví dụ
- Kết quả
- Ví dụ : Kiểm tra xem số nhập vào là số âm hay không
- Ví dụ : Kiểm tra mối quan hệ của 2 số nhập vào

SWITCH

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

Ví dụ :

Viết CT nhập vào một số, dùng lệnh rẽ nhánh switch kiểm tra số đó nếu:

- Bằng 0 thì xuất dòng lệnh “Số không”
- Bằng 2 thì xuất dòng lệnh “Số hai”
- Các số còn lại thì xuất dòng lệnh “Không tìm thấy”

SWITCH

```
<?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;
?>
```

BÀI TẬP

- Viết đoạn code thể hiện thời khóa biểu của sv

VÒNG LẶP

- **Vòng lặp là gì?**

- 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];
```

VÒNG LẶP FOR KẾT HỢP VỚI MẢNG

```
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  }
```

VÒNG LẶP WHILE

```
1  while ($condition) {  
2      // dòng lệnh  
3  }
```

- \$condition là điều kiện để dừng vòng lặp

```
01  <?php  
02  
03  $x = 1;  
04  
05  while ( $x <= 20) {  
06      $x++;  
07      echo "$x <br>";  
08  }  
09  ?>
```

VÒNG LẶP WHILE

- Phân tích ví dụ sau :

```
1  $i = 0;  
2  $j = 10;  
3  while ($i < 100 && $j > 5){  
4      $i++;  
5      $j -= 2;  
6  }
```

VÒNG LẶP DO WHILE

```
1  do {  
2      // dòng lệnh  
3  } while ($condition);
```

- \$condition là điều kiện để dừng vòng lặp

```
1  $i = 1;  
2  do{  
3      echo $i;  
4      $i++;  
5  }while ($i <= 10);
```


VÒNG LẶP DO WHILE

- Viết đoạn mã in ra màn hình các số từ 100 đến 200 bằng 3 cách (3 vòng lặp)

```
1  for ($i = 100; $i <= 200; $i++){  
2      echo $i;  
3  }
```

```
1  $i = 100;  
2  while ($i <= 200){  
3      echo $i;  
4      $i++; // Tăng $i lên 1  
5  }
```

```
1  $i = 100;  
2  do {  
3      echo $i;  
4      $i++;  
5  } while ($i <= 200);
```

VÒNG LẶP WHILE DO LÒNG NHAU

```
1  $i = 1;
2  while ($i < 10)
3  {
4      $j = $i;
5      while ($j < 10)
6      {
7          echo $j;
8          $j++;
9      }
10     echo '
11     ';
12     $i++;
13 }
```

```
123456789
23456789
3456789
456789
56789
6789
789
89
9
```

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);
```

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

FOREACH

```
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;
}
```

FOREACH

```
//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

LỆNH BREAK, CONTINUE, GOTO, DIE, EXIT

- Câu lệnh Break
- Câu lệnh Continue
- Câu lệnh Goto
- Câu lệnh Die & Exit

FOREACH

- Lệnh BREAK thường được dùng để thoát khỏi vòng lặp cho dù vòng lặp chưa kết thúc

```
1  for ($i = 1; $i <= 100; $i++)  
2  {  
3      echo $i . ' ';  
4      if ($i == 20)  
5      {  
6          break;  
7      }  
8  }
```


CONTINUE

- Continue sẽ bỏ qua những đoạn code bên dưới nó và nhảy qua vòng lặp kế tiếp (ko thoát khỏi vòng lặp như lệnh break)

```
1  for ($i = 1; $i <= 10; $i++)
2  {
3      if ($i == 5)
4      {
5          continue;
6      }
7      echo $i . ' ';
8  }
```

GOTO

- Lệnh goto để nhảy đến một code nào đó

```
1  $a = 12;  
2  $b = 13;  
3  $c = $a + $b;  
4  
5  echo $a;  
6  
7  goto label_end;  
8  
9  echo $b;  
10  
11 label_end;
```

DIE và EXIT

- Die và exit sẽ làm chương trình dừng ngay lập tức

```
1  echo '123';  
2  
3  die(); // hoặc exit();  
4  echo '456';
```

HÀM

- Hàm là tập hợp một hay nhiều câu lệnh được xây dựng để thực hiện một chức năng nào đó

```
1 function func_name($vars)
2 {
3     // các đoạn code
4     return $val;
5 }
```

- Func_name: tên của hàm
- \$vars là các biến truyền vào trong hàm
- Return\$val là hàm sẽ trả về giá trị \$val

HÀM

```
<?php
function xin chào(){
echo "Xin chào các bạn!.";
echo "Tôi là Lan."."<br>";
}
xin chào();
xin chào();
?>
```

Xin chào các bạn!.Tôi là Lan.
Xin chào các bạn!.Tôi là Lan.

HÀM

- Bài tập : Viết đoạn mã in ra 100 dòng “Đây là số n” với n từ 1 đến 100

```
function vd(){  
    for ($i=0; $i <100 ; $i++) {  
        echo "Đây là số".$i."<br/>";  
    }  
}  
vd();
```



HÀM VỚI THAM SỐ

- Bài toán : Xét ngày trong tuần
- Bài tập : Tạo ra một hàm cộng hai số (hoặc nhân hai số)



HÀM VỚI GIÁ TRỊ TRẢ VỀ

- Xét ví dụ sau :

```
<?php
function add($a,$b){
    $c=$a+$b;
    return $c;
}
$d= add(5, 8);
echo $d.'<br/>';
function concatstr ($str1, $str2) {
    $str=$str1.$str2;
    return $str;
}
$s=concatstr("I'm a lecture.", "My name is Hoa");
echo $s;
?>
```



PHẠM VI CỦA BIẾN

- Biến toàn cục
- Biến cục bộ

PHẠM VI CỦA BIẾN

- Ví dụ :

```
12      <?php
13      $x=10;
14      function baitoan(){
15          echo 'x = ' . $x;
16      }
17      baitoan();
18      ?>
```

Ví dụ

Phạm vi biến

Notice: Undefined variable: x in C:\xampp\htdocs\Ví dụ\JM\phamvibien.php on line 15
x =

PHẠM VI CỦA BIẾN

- Ví dụ :

```
12      <?php
13      $x=10;
14      function baitoan(){
15          global $x;
16          echo 'x = ' . $x;
17      }
18      baitoan();
19      ?>
```

Ví dụ

Phạm vi biến

x = 10

PHẠM VI CỦA BIẾN

- Ví dụ :

```
12 <?php
13 $x=10;
14 $y=11;
15 $z=12;
16 function baitoan(){
17     $y_cucbo=$GLOBALS['y'];
18     $z_cucbo=$GLOBALS['z'];
19     echo 'y='.$y_cucbo.'|',z='.$z_cucbo;
20 }
21 baitoan();
22 ?>
```

Ví dụ

Phạm vi biến
y=11,z=12

Làm việc với file

- Mở file
- Đọc file
- Ghi file
- Đóng file
- Một số hàm xử lý với file

MỞ, ĐÓNG FILE

```
open($path, $option)
```

- Trong đó \$path là đường dẫn file cần mở
- \$option là quyền cho phép thao tác trên file
- Đóng file sử dụng hàm fclose(\$fp)

MỞ FILE

Mode	Diễn giải
r	Read only
r+	Read + Write
w	Write only
w+	Write + Read. Nếu file này tồn tại thì nội dung cũ sẽ bị xóa đi và ghi lại nội dung mới, còn nếu file chưa tồn tại thì nó tạo file mới
a	Mở dưới dạng append dữ liệu, chỉ có write và nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới
a+	Mở dưới dạng append dữ liệu, bao gồm write và read. Nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới
b	Mở dưới dạng chế độ binary

ĐỌC FILE

- Đọc từng dòng `fgetc($fp)`
- Đọc từng ký tự `fgets($fp)`
- Đọc hết file `fread($fp, $size)`
- Trong đó, `$fp` là đối tượng lúc mở file
- `$size` là kích cỡ của file cần đọc
- Hàm `filesize($path)` để lấy kích cỡ của file cần đọc

GHI FILE

```
fwrite($fp, $content)
```

- \$fp là đối tượng trả về lúc mở file
- \$content là nội dung muốn ghi vào

CÁC HÀM XỬ LÝ FILE KHÁC

Hàm	Chức năng
File_exists(\$path)	Kiểm tra file có tồn tại không
Is_wriable(\$path)	Kiểm tra file có được cấp quyền ghi không
File_get_contents(\$path)	Lấy nội dung một file mà ko cần dùng hàm fread
File_put_content(\$path,\$noidung)	Ghi nội dung file mà ko cần dùng hàm fwrite
Rename(\$oldname,\$newname)	Đổi tên file
Copy(\$source,\$dest)	Copy file
Unlink(\$path)	Xóa file
Is_dir(\$filenam)	Kiểm tra một đườngf dẫn folder có tồn tại ko
Mkdir(\$path)	Tạo một folder mới

- \$path : đường dẫn đến file