BÀI 1:
Giới thiệu về PHP

Mục tiêu bài học

- Giới thiệu về PHP: Lịch sử, các đặc điểm
- Biến và hằng trong PHP
- Cú pháp của PHP
- Các kiểu dữ liệu trong PHP
- Các cấu trúc điều kiện trong PHP
  - Cấu trúc if else
  - Cấu trúc switch case

Giới thiệu về PHP

- **PHP** là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát.
- PHP rất thích hợp với việc phát triển ứng dụng web và có thể dễ dàng nhúng vào trang HTML.
- Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.
- PHP được tích hợp với nhiều hệ quản trị CSDL phổ biến, bao gồm MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.
- PHP đã có tới 7 phiên bản. Bản mới nhất hiện nay là PHP 7.

Đặc điểm sử dụng của PHP

- PHP cho phép tạo ra các đoạn mã kịch bản và nhúng vào các trang web để xử lý yêu cầu của client, cũng như hiển thị kết quả.
- PHP cho phép thao tác với các tập tin của hệ thống, VD ta có thể tạo, mở, đọc/ghi và đóng các tập tin.
- PHP cho phép xử lý các form nhập liệu, ta có thể thực hiện các thao tác submit form, nhập và xử lý dữ liệu trên form, v.v…
- PHP cung cấp khả năng kết nối và thao tác với dữ liệu trong các hệ quản trị cơ sở dữ liệu phổ biến. Ta có thể thêm, sửa, xóa, truy vấn dữ liệu một cách dễ dàng thông qua PHP.
- Cho phép truy cập và thao tác với cookies và quản lý session

Các đặc điểm của PHP

- Mã nguồn tử, có thể download và sử dụng miễn phí.
- Cross platform
- Tương thích với nhiều loại server
- Hỗ trợ nhiều loại CSDL thông dụng
- Cú pháp đơn giản, dễ học, dễ phát triển
- Có cộng đồng đông đảo hỗ trợ

Ví dụ "Hello World" trong PHP

- Ví dụ:

```
<html>

   <head>
      <title>Hello World</title>
   </head>

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

</html>
```

- Kết quả:

```
Hello, World!
```

Các kiểu dữ liệu được hỗ trợ bởi PHP

- PHP hỗ trợ các kiểu dữ liệu sau:
  - Integer: Cho phép lưu trữ các số nguyên 32 bits, có giá trị trong khoảng -2,147,483,648 và 2,147,483,647
  - Float: Cho phép lưu trữ các giá trị số thực 64 bits.
  - String: Gồm một tập hợp các ký tự 8-bit.
  - Boolean: Bao gồm 2 giá trị literal: 'true' và 'false'
  - Array: Cho phép lưu trữ một tập hợp các cặp keys và values.
  - Object: Là một thể hiện của một class.
  - Null: Chỉ có một giá trị null.

Comment trong PHP

- Comment là những dòng chú thích, dùng để giải thích ý nghĩa của các đoạn lệnh trong chương trình. Comment sẽ không được xử lý.
- PHP hỗ trợ 2 loại comment:
  - **Single-line comments: Comment trên một dòng**

```
// This is a comment too. Each style comments only
print "An example with single line comments";
```

  - **Multi-lines comments: Comment trên nhiều dòng**

```
<?
   /* This is a comment with multiline
      Author : Mohammad Mohtashim
      Purpose: Multiline Comments Demo
      Subject: PHP
   */

   print "An example with multi line comments";
?>
```

Cú pháp PHP

- Mọi dòng lệnh trong PHP sẽ được bọc trong cặp dấu **<?php ?>**
- PHP phân biệt chữ hoa-thường
- Các câu lệnh trong các biểu thức sẽ được kết thúc bởi dấu chấm phẩy (;)
- Để in kết quả của chương trình, ta sử dụng lệnh echo.
- PHP là một ngôn ngữ không quy định chặt chẽ về kiểu (Loosely Typed Language).
  - PHP sẽ tự động chuyển đổi giá trị được lưu trong các biến thành kiểu dữ liệu phù hợp, tùy thuộc vào giá trị của biến.

Biến trong PHP

- Trong PHP, biến được bắt đầu với một ký tự $, sau đó là tên biến.
- Để gán một chuỗi văn bản cho biến, ta đặt giá trị trong cặp dấu nháy đơn hoặc nháy kép.
- PHP phân biệt chữ hoa-thường (case-sensitive).
- Trong PHP, các biến có thể được khai báo ở bất cứ đâu trong các đoạn script.
- Phạm vi (scope) của biến là phần kịch bản mà biến đó có thể được tham chiếu/sử dụng.

Biến trong PHP

- Các quy tắc khai báo biến trong PHP:
  - Biến được bắt đầu với ký tự $, sau đó là tên biến.
  - Tên biến phải bắt đầu với một ký tự hoặc một dấu gạch nối (underscore)
  - Tên biến không được bắt đầu với một chữ số
  - Tên biến chỉ có thể chứa các ký tự, các chữ số và dấu gạch nối (A-z, 0-9, and _)
  - Tên biến trong PHP là phân biệt chữ hoa-thường (Biến $age và $AGE được coi là hai biến khác nhau)

Hằng trong PHP

- Hằng (constant) là một tên hoặc một identifier có chứa một giá trị.
- Giá trị của hằng không thể thay đổi trong quá trình thực thi mã kịch bản. Mặc định hằng sẽ phân biệt chữ hoa-thường (case-sensitive).
- Theo quy ước, tên của hằng sẽ được viết in hoa.
- Tên của hằng được bắt đầu với một chữ cái hoặc một dấu gạch nối (underscore), theo sau là các ký tự, chữ số, hoặc dấu gạch nối. Khi ta đã định nghĩa một hằng, giá trị của hằng không thể thay đổi.
- Để định nghĩa hằng trong PHP, ta sử dụng hàm define(). Hàm này cho phép định nghĩa một hằng, và trả về giá trị của hằng đó, ta cần gán giá trị của hằng với một tên
- Không giống với tên biến, tên của hằng không cần phải có dấu ($) ở đầu.

Hằng trong PHP

- Phân biệt sự khác nhau giữa hằng và biến:
  - Tên: Tên hằng không cần phải bắt đầu với ký tự ($), nhưng tên biến thì phải có ký tự ($) ở đầu.
  - Hằng không thể được định nghĩa bằng một lệnh gán, ta cần định nghĩa hằng thông qua hàm define().
  - Hằng có thể được định nghĩa và truy cập ở bất cứ đâu mà không bị ràng buộc bởi các quy tắc phạm vi như biến.
  - Ngay khi định nghĩa hằng, ta cần phải khởi tạo giá trị cho hằng đó, giá trị của hằng là cố định, không thay đổi.
  - Ví dụ:

```
// Valid constant names
define("ONE",     "first thing");
define("TWO2",    "second thing");
define("THREE_3", "third thing");

// Invalid constant names
define("2TWO",    "second thing");
define("__THREE__", "third value");
```

Hằng trong PHP

- Ví dụ về hằng số:

```php
<?php
//dinh nghia hang so
define("AGE", "20");
define('ADDRESS', 'NewYork - Paris - Montreal');

echo 'Tuổi là: ' . AGE;
echo '<BR>Địa chỉ là: ' . ADDRESS;
?>
```

- Kết quả:

Demo

- Giáo viên demo về quá trình tạo file PHP, sau đó khai báo biến và hằng.

Toán tử trong PHP

- Toán tử (operator): Là những phép tính được thực hiện trên các toán hạng.
- Toán hạng (operand): Là những biến hoặc giá trị được sử dụng trong các toán tử.
- PHP hỗ trợ đầy đủ các loại toán tử phổ biến
  - Arithmetic Operators
  - Comparison Operators
  - Logical (or Relational) Operators
  - Assignment Operators
  - Conditional (or ternary) Operators

Toán tử trong PHP

- Toán tử đại số (Arithmetic Operators): Cho phép thực hiện các phép tính đại số.
- Các toán tử đại số mà PHP hỗ trợ bao gồm:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by de-numerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

Toán tử trong PHP

- Toán tử quan hệ(Relational Operators)
  - Cho phép thực hiện các phép so sánh giá trị giữa các toán hạng.
  - Kết quả trả về sẽ là một giá trị logic (true hoặc false)
- Các toán tử quan hệ mà PHP hỗ trợ bao gồm:

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Toán tử trong PHP

- Toán tử logic (Logical Operators)
  - Cho phép thực hiện các phép tính logic.
  - Kết quả trả về sẽ là một giá trị logic (true hoặc false)
- Các toán tử logic mà PHP hỗ trợ bao gồm:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero then condition becomes true.	(A && B) is true.
||	Called Logical OR Operator. If any of the two operands are non zero then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false.

Toán tử trong PHP

- Toán tử gán (**Assignment** Operators)
  - Dùng để gán giá trị ở phía bên phải cho biến hoặc biểu thức ở phía bên trái.
- Các toán tử gán mà PHP hỗ trợ bao gồm:

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assign value of A + B into C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	C %= A is equivalent to C = C % A

Toán tử trong PHP

- Toán tử điều kiện (Conditional Operators)
  - Dùng để kiểm tra một điều kiện trong chương trình.
  - Biểu thức sẽ được đánh giá và trả về một giá trị true hoặc false. Nếu biểu thức trả về true, chương trình thực hiện lệnh thứ nhất. Nếu biểu thức trả về false, chương trình thực hiện lệnh thứ hai.
  - Cú pháp:

Operator	Description	Example
? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y

Toán tử trong PHP

- Ví dụ về toán tử điều kiện:

```php
<?php
$var = 5;
$result = ($var > 2 ? "true" : "false");
echo 'result = ' . $result;
?>
```

- Kết quả:

```
result = true
```

Toán tử trong PHP

- Thứ tự ưu tiên của các toán tử
  - Các toán tử có mức ưu tiên cao hơn sẽ được đánh giá trước.
  - Các toán tử nằm trong cặp dấu ngoặc () sẽ có mức ưu tiên cao nhất.
  - Thứ tự như sau:

Category	Operator	Associativity
Unary	! ++ --	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Logical AND	&&	Left to right
Logical OR	||	Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=	Right to left

Cấu trúc điều kiện

- Là những cấu trúc cho phép kiểm tra điều kiện trong chương trình, nếu điều kiện thỏa mãn thì mới thực thi thân của cấu trúc, nếu điều kiện không thỏa mãn thì bỏ qua.
- PHP hỗ trợ các cấu trúc điều kiện sau:
  - Cấu trúc if – thực thi các câu lệnh nếu điều kiện là true
  - Cấu trúc if...else – thực thi khối lệnh nếu điều kiện là true, và thực thi khối còn lại nếu điều kiện là false
  - Cấu trúc if...elseif....else – Dùng để kiểm tra nhiều điều kiện trong chương trình.
  - Cấu trúc switch – Đánh giá giá trị của một biến hoặc biểu thức, và thực thi một trong các nhánh .

Cấu trúc If

- Cấu trúc if đánh giá một điều kiện, và thực thi các câu lệnh nếu điều kiện là đúng (true).

```
if (condition) {
    code to be executed if condition is true;
}
```

- Ví dụ:

```php
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
}
?>
```

Cấu trúc If … else

- Cấu trúc if....else sẽ thực hiện khối if nếu điều kiện là true, hoặc thực thi khối else nếu điều kiện là false.

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

- Ví dụ:

```
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

Cấu trúc if...elseif....else

- Cấu trúc if....elseif...else dùng để kiểm tra nhiều điều kiện, nhiều nhánh trong chương trình.
- Đây là cú pháp mở rộng của cấu trúc if else.
- Cú pháp:

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

Cấu trúc if...elseif....else

- Ví dụ:

```php
<?php
$mark = 80;

if($mark >= 0 && $mark < 50) {
    echo 'Loại kém !';
} else if ($mark >= 50 && $mark < 70) {
    echo 'Loại trung bình !';
} else if($mark >= 70 && $mark < 80) {
    echo 'Loại khá !';
} else if($mark >= 80 && $mark <= 100) {
    echo 'Loại giỏi !';
} else {
    echo 'Điểm phải nằm trong khoảng 0 - 100 !';
}
?>
```

Cấu trúc Nested If

- Cấu trúc nested if là một cấu trúc if được đặt bên trong một cấu trúc if khác.
- Cấu trúc if bên trong bị phụ thuộc vào cấu trúc if bên ngoài. Nếu điều kiện của cấu trúc if bên ngoài thỏa mãn, thì cấu trúc if bên trong mới có cơ hội được đánh giá và thực thi.
- Ví dụ:

```
if (expression 1 )
{
if (expression 2 )
{
// statements 1
}
else
{
// Statements 2
}
}
else
{
if ( expression 2)
{
// Statements 3
}
else
{
// Statements 4
}
}
```

Cấu trúc switch

- Cấu trúc switch dùng để thực thi nhiều hành động khác nhau dựa trên các điều kiện khác nhau.
- Cơ chế xử lý:
  - Chương trình sẽ đánh giá giá trị của biến hoặc biểu thức trong cấu trúc switch.
  - Giá trị của biểu thức sẽ được lần lượt so sánh với giá trị của từng case trong cấu trúc.
  - Nếu giá trị của biểu thức khớp với giá trị của một case, khối lệnh của case đó sẽ được thực hiện.
  - Chú ý: Cần sử dụng lệnh break để kết thúc một case.
  - Nếu không có case nào khớp, khối default sẽ được thực thi.

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

Cấu trúc switch

- Ví dụ:

```php
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

Demo

- Giảng viên demo minh họa cấu trúc if else và switch case

Tổng kết

- Giới thiệu về PHP: Lịch sử, các đặc điểm
- Biến và hằng trong PHP
- Cú pháp của PHP
- Các kiểu dữ liệu trong PHP
- Các cấu trúc điều kiện trong PHP
  - Cấu trúc if else
  - Cấu trúc switch case