|  |  |  |
| --- | --- | --- |
| TÌM HIỂU NGÔN NGỮ PHP | | |
| PHP Tutorial | Giới thiệu PHP | PHP là gì   * PHP là viết tắt của “PHP: Hypertext Preprocessor” * PHP là ngôn ngữ mô tả mã nguồn mở * Các tập lệnh PHP được thực thi trên máy chủ |
| Tệp PHP là gì ?   * Tệp PHP bao gồm văn bản ,HTML, CSS, JS và code PHP * Code PHP được thực thi trên máy chủ và kết quả được trả về trình duyệt dưới dạng HTML thuần túy * Tệp mở rộng “.php” |
| PHP có thể làm cái gì ?   * PHP có thể tạo nội dung trang động * PHP có thể tạo, mở, đọc, ghi, xóa và đóng tệp trên máy chủ * PHP có thể thu thập dữ liệu biểu mẫu * PHP có thể gửi và nhận cookie * PHP có thể thêm, xóa, sửa đổi dữ liệu trong cơ sở dữ liệu * PHP có thể được sử dụng để kiểm soát quyền truy cập của người dùng * PHP có thể mã hóa dữ liệu * Với PHP không bị giới hạn ở đầu ra HTML. Có thể xuất hình ảnh hoặc tệp PDF. Cũng có thể xuất bất kỳ văn bản nào, chẳng hạn như XHTML và XML. |
| Cú pháp PHP | Cú pháp PHP cơ bản   * Tập lệnh PHP có thể được đặt ở bất kỳ đâu trong tài liệu. * Một tập lệnh PHP bắt đầu bằng <?php và kết thúc bằng ?>:  |  | | --- | | <?php // Code PHP ?> |  * Phần mở rộng tệp mặc định cho các tệp PHP là ".php". * Một tệp PHP thường chứa các thẻ HTML và một số mã kịch bản PHP. * Các câu lệnh PHP kết thúc bằng dấu chấm phẩy (;).   Phân biệt chữ hoa chữ thường trong PHP   * Trong PHP, các từ khóa (ví dụ: if, other, while, echo, v.v.), các lớp, hàm và hàm do người dùng định nghĩa không phân biệt chữ hoa chữ thường. |
| PHP comment | * Comment trên dòng đơn dùng ký tự # hoặc //  |  | | --- | | <?php // This is a single-line comment  # This is also a single-line comment ?> |  * Comment trên nhiều dòng  |  | | --- | | <?php /\* This is a multiple-lines comment block that spans over multiple lines \*/ ?> | |
| PHP biến | Tạo (khai báo) các biến PHP   * Trong PHP, một biến bắt đầu bằng ký hiệu $, theo sau là tên của biến:  |  | | --- | | <?php $txt = "Hello world!"; $x = 5; $y = 10.5; ?> |  * Ghi chú: PHP không cần khai báo kiểu dữ liệu trước   Biến PHP  Rules for PHP variables:   * Một biến bắt đầu bằng dấu $, theo sau là tên của biến * Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới * Tên biến không được bắt đầu bằng số * Tên biến chỉ có thể chứa các ký tự chữ và số và dấu gạch dưới (A-z, 0-9 và \_ ) * Tên biến phân biệt chữ hoa chữ thường ($age và $AGE là hai biến khác nhau) |
| Phạm vi của biến | PHP có ba phạm vi biến khác nhau:   * local * global * static   Global and Local Scope   * Một biến được khai báo bên ngoài một hàm có GLOBAL SCOPE và chỉ có thể được truy cập bên ngoài một hàm:  |  | | --- | | <?php $x = 5; // global scope  function myTest() {   // Sử dụng biến bên trong hàm   echo "<p>Variable x inside function is: $x</p>"; } myTest();  echo "<p>Variable x outside function is: $x</p>"; ?> |  * Một biến được khai báo trong một hàm có LOCAL SCOPE và chỉ có thể được truy cập trong hàm đó:  |  | | --- | | <?php function myTest() {   $x = 5; // local scope   echo "<p>Variable x inside function is: $x</p>"; } myTest();  // báo lỗi  echo "<p>Variable x outside function is: $x</p>"; ?> |  * PHP Từ khóa global   + Từ khóa global được sử dụng để truy cập một biến toàn cục từ bên trong một hàm.  + Để làm điều này, hãy sử dụng từ khóa toàn cầu trước các biến (bên trong hàm):   |  | | --- | | <?php $x = 5; $y = 10;  function myTest() {   global $x, $y;   $y = $x + $y; }  myTest(); echo $y; // outputs 15 ?> |   + PHP cũng lưu trữ tất cả các biến toàn cục trong một mảng gọi là $GLOBALS[index]. Chỉ mục giữ tên của biến. Mảng này cũng có thể truy cập được từ bên trong các hàm và có thể được sử dụng để cập nhật các biến toàn cục một cách trực tiếp.   |  | | --- | | <?php $x = 5; $y = 10;  function myTest() {   $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y']; }  myTest(); echo $y; // outputs 15 ?> |  * PHP Từ khóa static   +Khi một chức năng được hoàn thành/thực thi, tất cả các biến của nó sẽ bị xóa. Tuy nhiên, đôi khi chúng ta muốn một biến local KHÔNG bị xóa.   |  | | --- | | <?php function myTest() {   static $x = 0;   echo $x;   $x++; }  myTest(); //output 0 myTest(); //output 1 myTest(); //output 2 ?> | |  | |
|  | PHP echo/print | * Câu lệnh echo và print PHP   + echo và print ít nhiều giống nhau. Cả hai đều được sử dụng để xuất dữ liệu ra màn hình  + Sự khác biệt là nhỏ: echo không có giá trị trả về trong khi print có giá trị trả về là 1 nên có thể sử dụng nó trong các biểu thức. |
|  | PHP Data type | * PHP string  |  | | --- | | <?php $x = "Hello world!";  ?> |  * PHP Integer  |  | | --- | | <?php $x = 5985;  $y = -123; ?> |  * PHP float  |  | | --- | | <?php $x = 10.365; var\_dump($x); ?> |  * PHP Boolean  |  | | --- | | <?php  $x = true; $y = false;  ?> |  * PHP Array  |  | | --- | | <?php $cars = array("Volvo","BMW","Toyota"); var\_dump($cars); ?> |  * PHP Object  |  | | --- | | <?php class Car {   public $color;   public $model;   public function \_\_construct($color, $model) {     $this->color = $color;     $this->model = $model;   }   public function message() {     return "My car is a " . $this->color . " " . $this->model . "!";   } }  $myCar = new Car("black", "Volvo"); echo $myCar -> message(); echo "<br>"; $myCar = new Car("red", "Toyota"); echo $myCar -> message(); ?> |  * PHP Null   Null là một kiểu dữ liệu đặc biệt chỉ có thể có một giá trị: NULL.  Nếu một biến được tạo mà không có giá trị, nó sẽ tự động được gán giá trị NULL.   |  | | --- | | <?php $x = "Hello world!"; $x = null; var\_dump($x); ?> | |
|  | PHP Strings | * PHP String Functions: PHP hỗ trợ nhiều hàm   Hàm strlen() trong PHP trả về độ dài của một chuỗi.   |  | | --- | | <?php echo strlen("Hello world!"); // outputs 12 ?> | |
|  | PHP Math | * PHP Math Functions: PHP hỗ trợ nhiều hàm toán học  1. pi() 2. min(), max() 3. abs() 4. … |
|  | PHP Constants | Các hằng số giống như các biến ngoại trừ một khi chúng được xác định thì chúng không thể thay đổi hoặc không được xác định.  Tên hằng hợp lệ bắt đầu bằng một chữ cái hoặc dấu gạch dưới (không có dấu $ trước tên hằng).   * Tạo PHP Constant   Để tạo hằng số sử dụng hàm define()  Syntax:   |  | | --- | | define(*name*, *value*, *case-insensitive*) |   Parameters:   * name: Chỉ định tên của hằng số * value: Chỉ định giá trị của hằng số * không phân biệt chữ hoa chữ thường: Chỉ định xem tên hằng có được phân biệt chữ hoa chữ thường hay không. Mặc định là sai   Example:   |  | | --- | | <?php define("GREETING", "Welcome to W3Schools.com!"); echo GREETING; //output Welcome to W3Schools.com! ?> | | <?php define("GREETING", "Welcome to W3Schools.com!", true); echo greeting; //output Welcome to W3Schools.com! ?> |  * PHP Constant Arrays  |  | | --- | | <?php define("cars", ["AlfaRomeo","BMW","Toyota"]); echo cars[0]; // output: AlfaRomeo ?> |  * Mặc định Constant Global Scope |
|  | PHP Operators | * Toán tử số học  |  |  |  |  | | --- | --- | --- | --- | | Operator | Name | Example | Result | | + | Add | $x + $y | Tổng giữa x và y | | - | Subtraction | $x - $y | Trừ giữa x và y | | \* | Multiplication | $x \* $y | Nhân x và y | | / | Division | $x / $y | Thương số x và y | | % | Modulus | $x % $y | Số dư x chia y | | \*\* | Exponentiation | $x \*\* $y | X mũ y |  * PHP toán tử gắn  |  |  | | --- | --- | | Phép gắn | Miêu tả | | x = y | Toán hạng bên trái được đặt thành giá trị của biểu thức bên phải | | x += y | Gắn cộng | | x -= y | Trừ | | x \*= y | Nhân | | x /= y | Chia | | x %= y | Modulus |  * PHP toán tử so sánh  |  |  |  |  | | --- | --- | --- | --- | | Operator | Name | Example | Result | | == | Equal | $x==$y | Trả về true nếu x bằng y | | === | Identical | $x===$y | Trả về true nếu $x bằng $y và chúng cùng loại | | != | Not equal | $x != $y | Trả về true nếu $x không bằng $y | | <> | Not equal | $x <> $y | Trả về true nếu $x không bằng $y | | !== | Not identical | $x !== $y | Trả về true nếu $x không bằng $y hoặc chúng không cùng loại | | <, >, <=, >= |  |  |  | | <=> | Spaceship | $x <=> $y | Trả về một số nguyên nhỏ hơn, bằng hoặc lớn hơn 0, tùy thuộc vào việc $x nhỏ hơn, bằng hoặc lớn hơn $y. |  * PHP Toán tử tăng / giảm  |  |  |  | | --- | --- | --- | | **Operator** | **Name** | **Description** | | ++$x | tăng trước | Tăng $x lên một, sau đó trả về $x | | $x++ | tăng sau | Trả về $x, sau đó tăng $x lên một | | --$x, $x-- | Tương tự |  |  * Toán tử logic PHP  |  |  |  |  | | --- | --- | --- | --- | | **Operator** | Name | **Example** | **Result** | | And, && | And, And | $x and $y  , $x && $y | Đúng nếu cả $x và $y đều đúng | | Or, || | Or, Or | $x or $y  $x || $y | Đúng nếu $x hoặc $y là đúng | | xor | Xor | $x xor $y | Đúng nếu $x hoặc $y đều đúng, nhưng không phải cả hai | | ! | Not | !$x | Đúng nếu $x không đúng |  * PHP Toán tử String  |  |  |  |  | | --- | --- | --- | --- | | **Operator** | Name | **Example** | **Result** | | . | nối | $txt1 . $txt2 | Nối $txt1 và $txt2 | | .= | gán nối | $txt1 .= $txt2 | Nối thêm $txt2 vào $txt1 |  * Toán tử mảng PHP  |  |  |  |  | | --- | --- | --- | --- | | **Operator** | Name | **Example** | **Result** | | + | Hợp | $x + $y | Hợp của $x và $y | | == | Bằng | $x == $y | Trả về true nếu $x và $y có cùng cặp khóa/giá trị | | === | Identity | $x === $y | Trả về true nếu $x và $y có cùng cặp khóa/giá trị theo cùng thứ tự và cùng loại | | !=,<> | Không bằng | $x != $y,$x<>$y | Trả về true nếu $x không bằng $y | | !== | Non-identity | $x !== $y | Trả về true nếu $x không giống với $y |  * Toán tử gán có điều kiện PHP  |  |  |  |  | | --- | --- | --- | --- | | **Operator** | Name | **Example** | **Result** | | ?: | Ternary | $x = expr1? expr2 : expr3 | Trả về giá trị của $x. Giá trị của $x là expr2 nếu expr1 = TRUE. Giá trị của $x là expr3 nếu expr1 = FALSE | | ?? | Null hợp nhất | $x = expr1 ?? expr2 | Trả về giá trị của $x. Giá trị của $x là expr1 nếu expr1 tồn tại và không phải là NULL. Nếu expr1 không tồn tại hoặc là NULL, giá trị của $x là expr2. | |
|  | PHP Decision making | PHP có 3 loại:   * The if..else  |  | | --- | | if (condition) code to be executed if condition is true; else code to be executed if condition is false; |  * The elseif  |  | | --- | | if (condition) code to be executed if condition is true; elseif (condition) code to be executed if condition is true; else code to be executed if condition is false; |  * The switch .. case  |  | | --- | | switch(n){ case label1: // Code to be executed if n=label1 break; case label2: // Code to be executed if n=label2 break; ... default: // Code to be executed if n is different from all labels } | |
|  | PHP Loop | * For- lặp qua một khối mã một số lần xác định.  |  | | --- | | for(initialization; condition; increment){ // Code to be executed } |  * While- lặp qua một khối mã cho đến khi điều kiện được đánh giá là đúng.  |  | | --- | | while (condition is true) { code to be executed; } |  * Do..while- khối mã được thực thi một lần và sau đó điều kiện được đánh giá. Nếu điều kiện là đúng, câu lệnh được lặp lại miễn là điều kiện được chỉ định là đúng.  |  | | --- | | do { code to be executed; } while (condition); |  * Foreach- lặp qua một khối mã cho mỗi phần tử trong một mảng.  |  | | --- | | foreach (array as value) { code to be executed; } | |
|  | PHP functions | |  | | --- | | function *functionName*() { *code to be executed*; } | |
| PHP Database | PHP Connect to MySQL | Trước khi chúng tôi có thể truy cập dữ liệu trong cơ sở dữ liệu MySQL, chúng tôi cần có thể kết nối với máy chủ:  MySQLi Object   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password";  // Create connection $conn = new mysqli($servername, $username, $password);  // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); } echo "Connected successfully"; ?> |   MySQLi Procedural   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password";  // Create connection $conn = mysqli\_connect($servername, $username, $password);  // Check connection if (!$conn) {   die("Connection failed: " . mysqli\_connect\_error()); } echo "Connected successfully"; ?> |   Close Connect   |  | | --- | | $conn->close(); // Object  mysqli\_close($conn); // Procedure | |
|  | PHP create DB | MySQL Object-Oriented   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password";  // Create connection $conn = new mysqli($servername, $username, $password); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  // Create database $sql = "CREATE DATABASE myDB"; if ($conn->query($sql) === TRUE) {   echo "Database created successfully"; } else {   echo "Error creating database: " . $conn->error; }  $conn->close(); ?> |  * MySQL Procedure  |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password";  // Create connection $conn = mysqli\_connect($servername, $username, $password); // Check connection if (!$conn) {   die("Connection failed: " . mysqli\_connect\_error()); }  // Create database $sql = "CREATE DATABASE myDB"; if (mysqli\_query($conn, $sql)) {   echo "Database created successfully"; } else {   echo "Error creating database: " . mysqli\_error($conn); }  mysqli\_close($conn); ?> | |
|  | PHP Create Table | MySQLi Object—Oriented   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  // sql to create table $sql = "CREATE TABLE MyGuests ( id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY, firstname VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT NULL, email VARCHAR(50), reg\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP )";  if ($conn->query($sql) === TRUE) {   echo "Table MyGuests created successfully"; } else {   echo "Error creating table: " . $conn->error; }  $conn->close(); ?> |   MySQL procedure   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = mysqli\_connect($servername, $username, $password, $dbname); // Check connection if (!$conn) {   die("Connection failed: " . mysqli\_connect\_error()); }  // sql to create table $sql = "CREATE TABLE MyGuests ( id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY, firstname VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT NULL, email VARCHAR(50), reg\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP )";  if (mysqli\_query($conn, $sql)) {   echo "Table MyGuests created successfully"; } else {   echo "Error creating table: " . mysqli\_error($conn); }  mysqli\_close($conn); ?> | |
|  | PHP insert | |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com')"; // MySQli Object-oriented if ($conn->query($sql) === TRUE) {   echo "New record created successfully"; } else {   echo "Error: " . $sql . "<br>" . $conn->error; }  $conn->close(); // MySQli Procedure  if (mysqli\_query($conn, $sql)) {   echo "New record created successfully"; } else {   echo "Error: " . $sql . "<br>" . mysqli\_error($conn); } mysqli\_close($conn); ?> | |
|  | PHP Get Last ID | Nếu thực hiện Insert hoặc UPDATE trên bảng có trường có thể lấy ngay ID của bản ghi được chèn/cập nhật cuối cùng.   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com')"; // MySQli Object-oriented  if ($conn->query($sql) === TRUE) {   $last\_id = $conn->insert\_id;   echo "New record created successfully. Last inserted ID is: " . $last\_id; } else {   echo "Error: " . $sql . "<br>" . $conn->error; }  $conn->close();  // MySQli Procedure  if (mysqli\_query($conn, $sql)) {   $last\_id = mysqli\_insert\_id($conn);   echo "New record created successfully. Last inserted ID is: " . $last\_id; } else {   echo "Error: " . $sql . "<br>" . mysqli\_error($conn); }  mysqli\_close($conn); ?> | |
|  | PHP Insert Multiple | Nhiều câu lệnh SQL phải được thực thi với hàm mysqli\_multi\_query().   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com');"; $sql .= "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('Mary', 'Moe', 'mary@example.com');"; $sql .= "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('Julie', 'Dooley', 'julie@example.com')"; // MySQli Object-oriented if ($conn->multi\_query($sql) === TRUE) {   echo "New records created successfully"; } else {   echo "Error: " . $sql . "<br>" . $conn->error; }  $conn->close();  // MySQli Procedure  if (mysqli\_multi\_query($conn, $sql)) {   echo "New records created successfully"; } else {   echo "Error: " . $sql . "<br>" . mysqli\_error($conn); }  mysqli\_close($conn); ?> | |
|  | PHP Select Data | |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "SELECT id, firstname, lastname FROM MyGuests"; $result = $conn->query($sql);  if ($result->num\_rows > 0) {   // output data of each row   while($row = $result->fetch\_assoc()) {     echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";   } } else {   echo "0 results"; } $conn->close(); ?> |   hàm num\_rows() kiểm tra xem có nhiều hơn 0 hàng được trả về hay không. Nếu có nhiều hơn 0 hàng được trả về, thì hàm fetch\_assoc() sẽ đặt tất cả kết quả vào một mảng kết hợp mà chúng ta có thể lặp qua. |
|  | PHP Delete Data | MySQLi Object-Oriented   |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  // sql to delete a record $sql = "DELETE FROM MyGuests WHERE id=3";  if ($conn->query($sql) === TRUE) {   echo "Record deleted successfully"; } else {   echo "Error deleting record: " . $conn->error; }  $conn->close(); ?> | |
|  | PHP Update Data | |  | | --- | | <?php $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB";  // Create connection $conn = new mysqli($servername, $username, $password, $dbname); // Check connection if ($conn->connect\_error) {   die("Connection failed: " . $conn->connect\_error); }  $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";  if ($conn->query($sql) === TRUE) {   echo "Record updated successfully"; } else {   echo "Error updating record: " . $conn->error; }  $conn->close(); ?> | |
| PHP Advanced | PHP include | Dùng để gộp file  Có 2 cách gộp  PHP include và PHP require  Chúng chỉ khác nhau khi file không được tìm thấy   * require sẽ tạo ra lỗi nghiêm trọng (E\_COMPILE\_ERROR) và dừng tập lệnh * include sẽ chỉ đưa ra cảnh báo (E\_WARNING) và tập lệnh sẽ tiếp tục |
|  | PHP sessions |  |