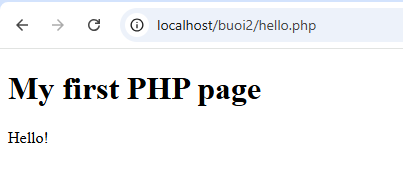
**BÀI TẬP THỰC HÀNH 02**

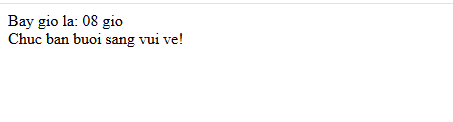
**LÀM QUEN VỚI NGÔN NGỮ PHP**

1. *Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.*

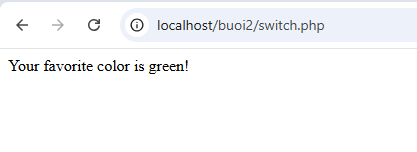
* **Tạo tập tin hello.php hiển thị chữ “Hello!”**



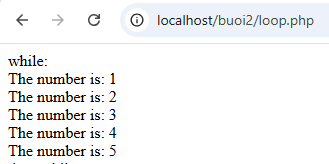
* **Tạo cấu trúc if, elseif và else xét biết t có kiểu giờ hiện hành, và xét theo giờ hiện để hiển thị thông báo**



* **Tạo cấu trúc switch xét biến favcolor có chứa giá trị gì để hiển thị thông báo**



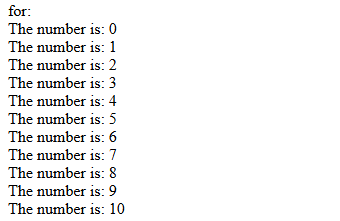
* **Tạo file loop.php viết các lệnh và quan sát kết quả**
* **Lệnh while:** biến x lặp từ 1->5 và in ra màn hình



* **Lệnh do..while:** biến x lặp từ 1->5 và in ra màn hình



* **Lệnh for:** biến x lặp từ 0->10 và in ra màn hình



* **Lệnh foreach:** lặp qua từng phần tử của mảng colors và in ra màn hình



* **Tạo tập tin function.php với lệnh để trả về kết quả cộng 2 biến**

*GHI CHÚ:* ***declare(strict\_types=1)*** *- buộc mọi thứ phải được sử dụng theo cách đã định.*

* **Không sử dụng strict với bắt đầu bằng ký tự *“so 10”***



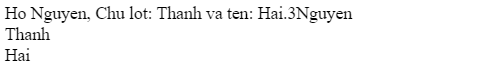
* **Không sử dụng strict với ký tự bắt đầu bằng số hoặc khoảng trắng *“10 so”***

****

* **Sử dụng strict**



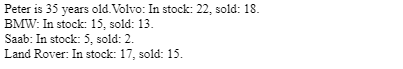
* **Mảng:**
* **Sử dụng mảng và in giá trị**

****

* **Sử dụng mảng kiểu “gom key => giá trị”**

****

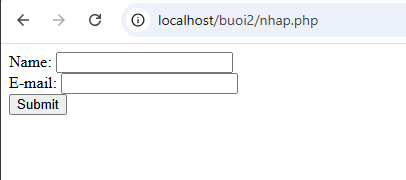
* **Sử dụng mảng đa chiều**

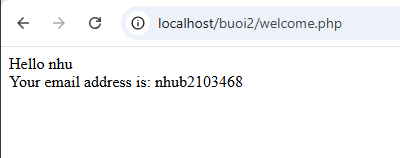
****

* **Hiển thị mảng đa chiều qua for**

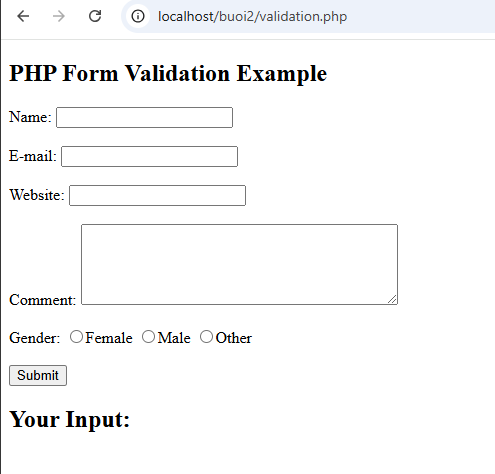
****

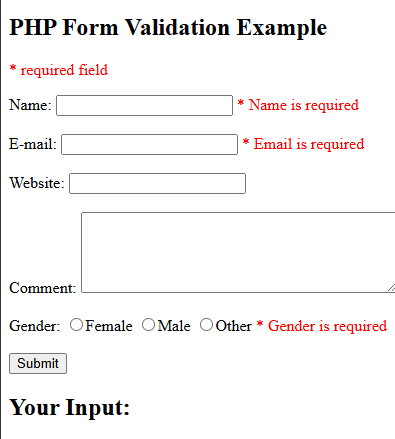
* **Xử lý với form:**



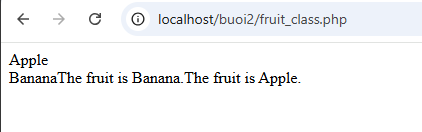


* **Kiểm tra dữ liệu trên form:**





* **Hướng đối tượng (OOP) trong PHP:**



1. ***Hãy cho biết cách khai báo biến, và gán trị cho biến trong PHP. Liệt kê các chú ý khi đặt tên biến. Cho các ví dụ về khai báo biến, và gán giá trị kiểu chuỗi, kiểu số, ngày tháng.***

* **Khai báo biến:**
* Sử dụng dấu đô-la ($) để bắt đầu đặt tên biến.
* Tên biến phải bắt đầu bằng một chữ cái hoặc dấu gạch dưới (\_).
* Tên biến có thể chứa các chữ cái, số và dấu gạch dưới.
* Tên biến phải phân biệt chữ hoa và chữ thường.
* **Ví dụ:** *$tenBien; $\_bienDauGachDuoi; $so1; $chuoi;*
* **Gán trị cho biến:** Sử dụng toán tử gán (=) để gán giá trị cho biến.
* **Ví dụ:** *$so1 = 10; $chuoi = "Hello"; $ngayThang = "2024-01-22";*
* **Chú ý khi đặt tên biến:**
* Tránh sử dụng các từ khóa dành riêng của PHP (ví dụ: if, else, while, etc.).
* Sử dụng tên biến mô tả và có ý nghĩa với giá trị mà nó đang lưu trữ.
* Tuân theo quy tắc chữ hoa và chữ thường để tạo tên biến dễ đọc và hiểu.
* **Ví dụ khai báo biến:** *$soLuongSanPham; $tenNguoiDung;$ngaySinh;*
* **Gán giá trị cho biến:**
* **Kiểu chuỗi:** *$soDienThoai = "123456789";*
* **Kiểu số:** *$soLuong = 20;*
* **Kiểu ngày, tháng:***$ngayNhapHang = "2024-01-22";*

1. ***Liệt kê ký hiệu để biết bắt đầu và kết thúc khối lệnh PHP.***

* Ký hiệu bắt đầu khối lệnh PHP: **<?php**
* Ký hiệu kết thúc khối lệnh PHP: **?>**
* Ví dụ:

<?php

// Bắt đầu khối lệnh PHP

$tenBien = "Hello, PHP!";

echo $tenBien;

// Kết thúc khối lệnh PHP

**?>**

1. ***Liệt kê các ký hiệu có thể được dùng để tạo comment (ghi chú) trong khối lệnh PHP.***

* Các ký hiệu dùng tạo comment:
* Comment 1 dòng: //
* Comment nhiều dòng: /\* khối lệnh \*/
* Ví dụ:

**//Comment một dòng:**

<?php

// Đây là một comment trên một dòng

$tenBien = "Hello, PHP!";

echo $tenBien;

?>

**------------------------------------------------------**

**//Comment nhiều dòng:**

<?php

/\* Đây là một comment

trên nhiều dòng \*/

$tenBien = "Hello, PHP!";

echo $tenBien;

?>

1. ***Cho biết có bao nhiêu loại khối lệnh để thực hiện vòng lặp. Cho biết sự khác biệt của foreach với các loại khối lệnh vòng lặp khác. Cho biết cách dùng của lệnh “continue” và lệnh “break” trong các vòng lặp.***

* **Các loại khối lệnh trong vòng lặp:** có 5 loại **[**for, while, do-while, foreach, foreach với key và value**]**
* **for:** Dùng để thực hiện một vòng lặp với điều kiện kiểm tra và sửa đổi biến đếm.
  + **Ví dụ:**

**for ($i = 0; $i < 5; $i++) {**

**// Thực hiện các công việc trong vòng lặp**

**echo $i;**

**}**

* **while:** Thực hiện một vòng lặp với điều kiện kiểm tra trước khi thực hiện mỗi lần lặp.
  + **Ví dụ:**

**$i = 0;**

**while ($i < 5) {**

**// Thực hiện các công việc trong vòng lặp**

**echo $i;**

**$i++;**

**}**

* **do-while:** Tương tự như while, nhưng kiểm tra điều kiện sau mỗi lần lặp.
  + **Ví dụ:**

**$i = 0;**

**do {**

**// Thực hiện các công việc trong vòng lặp**

**echo $i;**

**$i++;**

**} while ($i < 5);**

* **foreach:** Dùng để lặp qua mỗi phần tử của một mảng hoặc đối tượng.
  + **Ví dụ:**

**$array = [1, 2, 3, 4, 5];**

**foreach ($array as $value) {**

**// Thực hiện các công việc trong vòng lặp với mỗi phần tử**

**echo $value;**

**}**

* **foreach với key và value:** Cách sử dụng foreach để lặp qua mảng với cả key và value.
  + **Ví dụ:**

**$array = ['a' => 1, 'b' => 2, 'c' => 3];**

**foreach ($array as $key => $value) {**

**// Thực hiện các công việc trong vòng lặp với mỗi phần tử và key**

**echo "$key: $value";**

**}**

* **Sự khác biệt của foreach:** foreach được thiết kế đặc biệt để lặp qua mảng hoặc đối tượng một cách dễ dàng và tự động, giảm sự cần thiết của biến đếm và điều kiện lặp.
* **Cách dùng các lệnh:**
* **continue:**  Lệnh này dùng để bỏ qua phần còn lại của vòng lặp và chuyển đến lần lặp tiếp theo.
  + **Ví dụ:**

**for ($i = 0; $i < 5; $i++) {**

**if ($i == 2) {**

**continue; // Bỏ qua lần lặp khi $i bằng 2**

**}**

**echo $i;**

**}**

* **break:** Lệnh này dùng để kết thúc vòng lặp ngay lập tức khi được gọi.
  + **Ví dụ:**

**for ($i = 0; $i < 5; $i++) {**

**if ($i == 3) {**

**break; // Kết thúc vòng lặp khi $i bằng 3**

**}**

**echo $i;**

**}**

1. ***Viết 1 tập tin php tên giaithua.php, định nghĩa hàm giai thừa và chạy thử với 10!***

<?php

function tinhGiaiThua($n) {

if ($n == 0 || $n == 1)

return 1;

else

return $n \* tinhGiaiThua($n - 1);

}

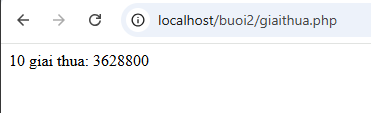
// Chạy thử với giá trị 10!

$number = 10;

$giaithua = tinhGiaiThua($number);

echo "$number giai thua: $giaithua";

?>



1. ***Viết 1 tập tin cong2mang.php viết hàm xử lý để cộng được 2 mảng có cùng độ dài, nếu 2 mảng không cùng độ dài thì thông báo lỗi. Chạy thử hàm với 2 mảng sau: a={344,224,223,7737,9922,-828}, b={-344,-324,123,773,-9922,828} và cho biết kết quả hiển thị.***

<?php

function congHaiMang($a, $b) {

// Kiểm tra độ dài của hai mảng

$lenA = count($a);

$lenB = count($b);

if ($lenA != $lenB)

// Nếu độ dài không bằng nhau, thông báo lỗi

return "Lỗi: Hai mảng không có cùng độ dài.";

else {

// Nếu độ dài bằng nhau, thực hiện phép cộng

$ketQua = array();

for ($i = 0; $i < $lenA; $i++)

$ketQua[] = $a[$i] + $b[$i];

return $ketQua;

}

}

// Mảng đầu vào

$a = array(344, 224, 223, 7737, 9922, -828);

$b = array(-344, -324, 123, 773, -9922, 828);

// Gọi hàm và hiển thị kết quả

$result = congHaiMang($a, $b);

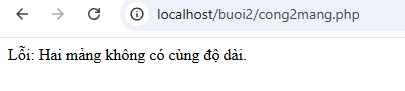
if (is\_array($result))

echo "Kết quả cộng hai mảng: " . implode(', ', $result);

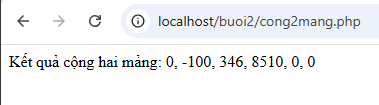
else

echo $result;

?>



* Khi số phần tử 2 mảng không bằng nhau



* Khi số phần tử 2 mảng bằng nhau

1. ***Chạy ví dụ tập tin validation.php, dựa vào link 2, bạn hãy giải thích những bước diễn ra quá trình này. [https://www.w3schools.com/php/php\_form\_validation.asp]***

**1. Biến và hàm khởi tạo:**

* Biến **$name, $email, $gender, $comment**, và **$website** được khởi tạo để lưu trữ dữ liệu nhập từ biểu mẫu.
* Hàm **test\_input** được định nghĩa để xử lý dữ liệu đầu vào. Nó sẽ thực hiện các bước sau đây:
* **trim**: Loại bỏ khoảng trắng ở đầu và cuối chuỗi.
* **stripslashes**: Loại bỏ các ký tự backslashes () từ chuỗi.
* **htmlspecialchars**: Chuyển đổi các ký tự đặc biệt thành các thực thể HTML.

**2. Kiểm tra và xử lý dữ liệu sau khi gửi biểu mẫu:**

* Sử dụng **$\_SERVER["REQUEST\_METHOD"]** để kiểm tra xem biểu mẫu đã được gửi đi (POST method) hay chưa.
* Nếu biểu mẫu đã được gửi đi, tiến hành lấy dữ liệu từ các trường của biểu mẫu và thực hiện hàm **test\_input** để xử lý dữ liệu.

**3. Biểu mẫu HTML:**

* Tạo một biểu mẫu HTML với các trường nhập dữ liệu như tên, email, website, comment, và gender.
* Sử dụng các thẻ **<input>, <textarea>, và <select>** để tạo các trường nhập dữ liệu và ô lựa chọn.
* Đặt thuộc tính **method="post"** để biểu mẫu gửi dữ liệu bằng phương thức POST.
* Đặt thuộc tính **action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>"** để biểu mẫu gửi dữ liệu đến chính trang hiện tại.

**4. Hiển thị kết quả:**

* Sau khi biểu mẫu được gửi đi, hiển thị thông tin được nhập vào trên trang.
* Sử dụng echo để hiển thị thông tin về tên, email, website, comment và giới tính.

1. ***Hãy sửa lại tập tin nhap.php, thêm 1 ô để nhập mật khẩu (các ký tự nhập vào sẽ biến thành dấu \*), và 1 ô để nhập ngày tháng năm sinh của bạn. Sau khi nhập xong hiển thị tên, email và mật khẩu, ngày tháng năm sinh đã nhập. Chụp màn hình sau khi bạn đã nhập dữ liệu của chính bạn, chụp màn hình hiển thị dữ liệu nhập.***

*nhapv2.php*

<!DOCTYPE HTML>

<html>

<body>

<form action="welcomev2.php" method="post">

Name: <input type="text" name="name"><br>

E-mail: <input type="text" name="email"><br>

Password: <input type="password" name="password"><br>

Date of Birth: <input type="date" name="dob"><br>

<input type="submit">

</form>

</body>

</html>

*welcomev2.php*

<!DOCTYPE HTML>

<html>

<body>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$name = test\_input($\_POST["name"]);

$email = test\_input($\_POST["email"]);

$password = test\_input($\_POST["password"]);

$dob = test\_input($\_POST["dob"]);

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

<h2>Your Input:</h2>

<p>Name: <?php echo $name; ?></p>

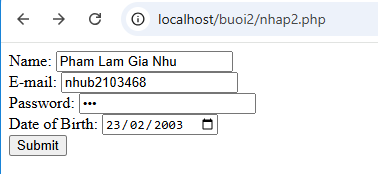
<p>Email: <?php echo $email; ?></p>

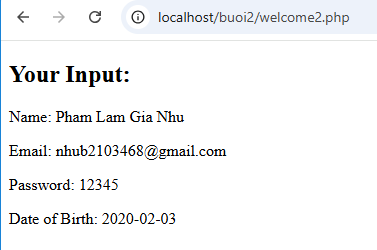
<p>Password: <?php echo $password; ?></p>

<p>Date of Birth: <?php echo $dob; ?></p>

</body>

</html>





1. ***Viết tập tin sinhvien\_class.php để khai báo 1 lớp sinhvien với các thuộc tính: mssv, hoten, ngaysinh với kiểu giá trị phù hợp. Viết các phương thức xây dựng, hủy, gán trị, trả về giá trị của các thuộc tính, viết hàm tính tuổi. Khai báo 1 biến sv có kiểu class sinhvien, gán các giá trị cần thiết với thông tin của bạn và thử tính tuổi dựa vào hàm vừa mới xây dựng trong class.***

<?php

class SinhVien {

**// Các thuộc tính của lớp**

public $mssv;

public $hoten;

public $ngaysinh;

**// Phương thức khởi tạo (constructor)**

public function \_\_construct($mssv, $hoten, $ngaysinh) {

$this->mssv = $mssv;

$this->hoten = $hoten;

$this->ngaysinh = $ngaysinh;

}

**// Phương thức hủy (destructor)**

public function \_\_destruct() {

echo " => Ket thuc!";

}

**// Phương thức gán giá trị cho mssv**

public function setMSSV($mssv) {

$this->mssv = $mssv;

}

**// Phương thức gán giá trị cho hoten**

public function setHoTen($hoten) {

$this->hoten = $hoten;

}

**// Phương thức gán giá trị cho ngaysinh**

public function setNgaySinh($ngaysinh) {

$this->ngaysinh = $ngaysinh;

}

**// Phương thức trả về giá trị của mssv**

public function getMSSV() {

return $this->mssv;

}

**// Phương thức trả về giá trị của hoten**

public function getHoTen() {

return $this->hoten;

}

**// Phương thức trả về giá trị của ngaysinh**

public function getNgaySinh() {

return $this->ngaysinh;

}

**// Phương thức tính tuổi**

public function tinhTuoi() {

$now = new DateTime();

$ngaysinh = new DateTime($this->ngaysinh);

$diff = $now->diff($ngaysinh);

return $diff->y;

}

}

**// Sử dụng lớp SinhVien**

$sv = new SinhVien("B2103468", "Phạm Lâm Gia Như", "2003-02-23");

**// Hiển thị thông tin sinh viên**

echo "MSSV: " . $sv->getMSSV() . "<br>";

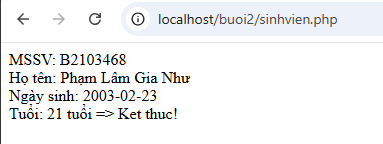
echo "Họ tên: " . $sv->getHoTen() . "<br>";

echo "Ngày sinh: " . $sv->getNgaySinh() . "<br>";

**// Tính tuổi và hiển thị**

echo "Tuổi: " . $sv->tinhTuoi() . " tuổi";

?>



**CÁC TÀI LIỆU THAM KHẢO**

**[1]** [**https://www.w3schools.com/php**](https://www.w3schools.com/php)