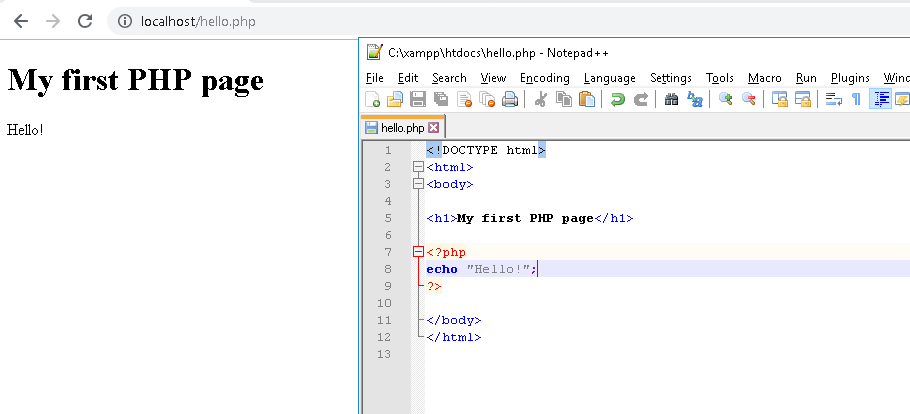
Yêu cầu bài thực hành:

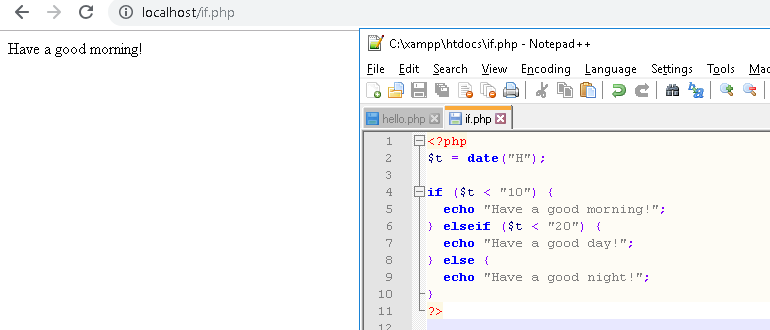
**Đinh Thái Hợp  
B2003785  
Bài 2**

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

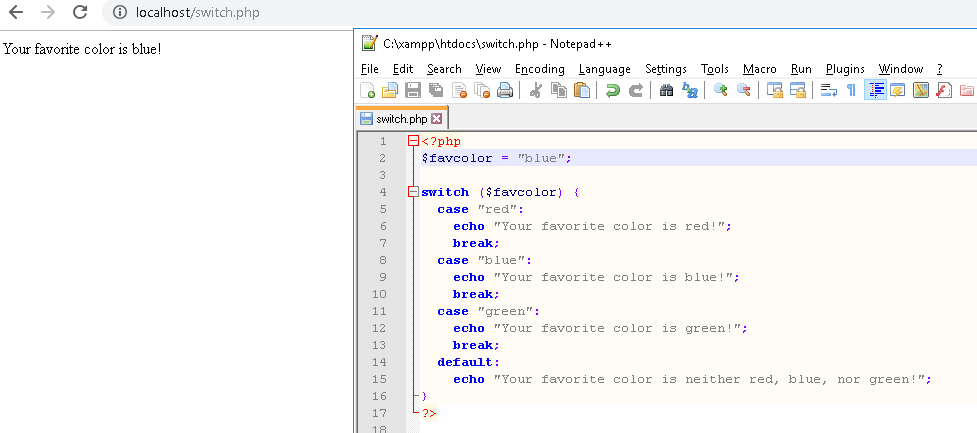
*Hello.php*



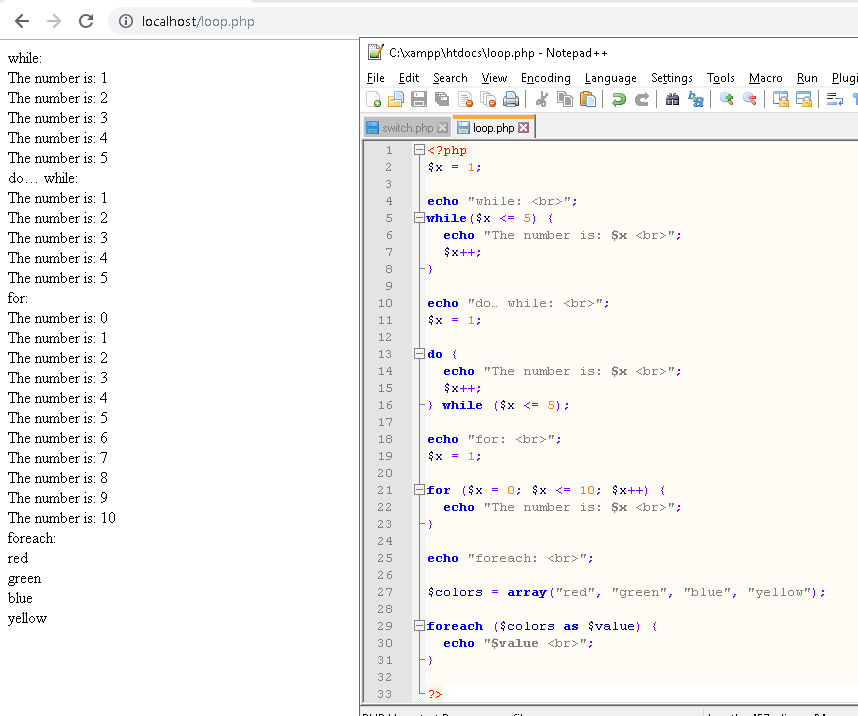
*If.php*



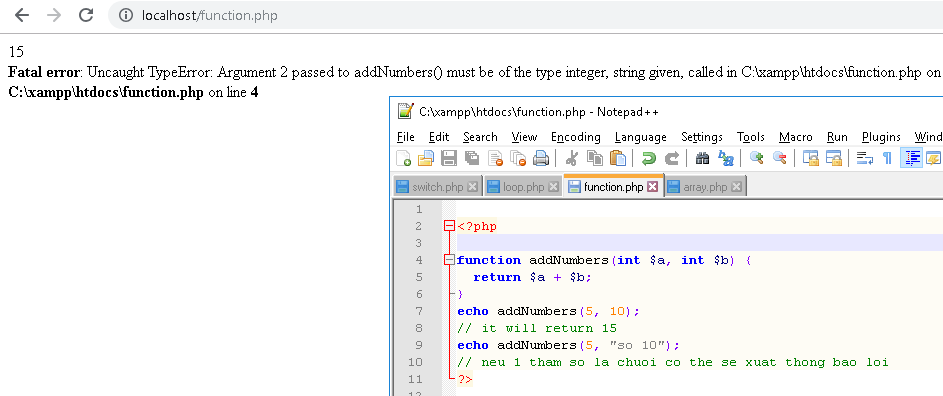
*Switch.php*



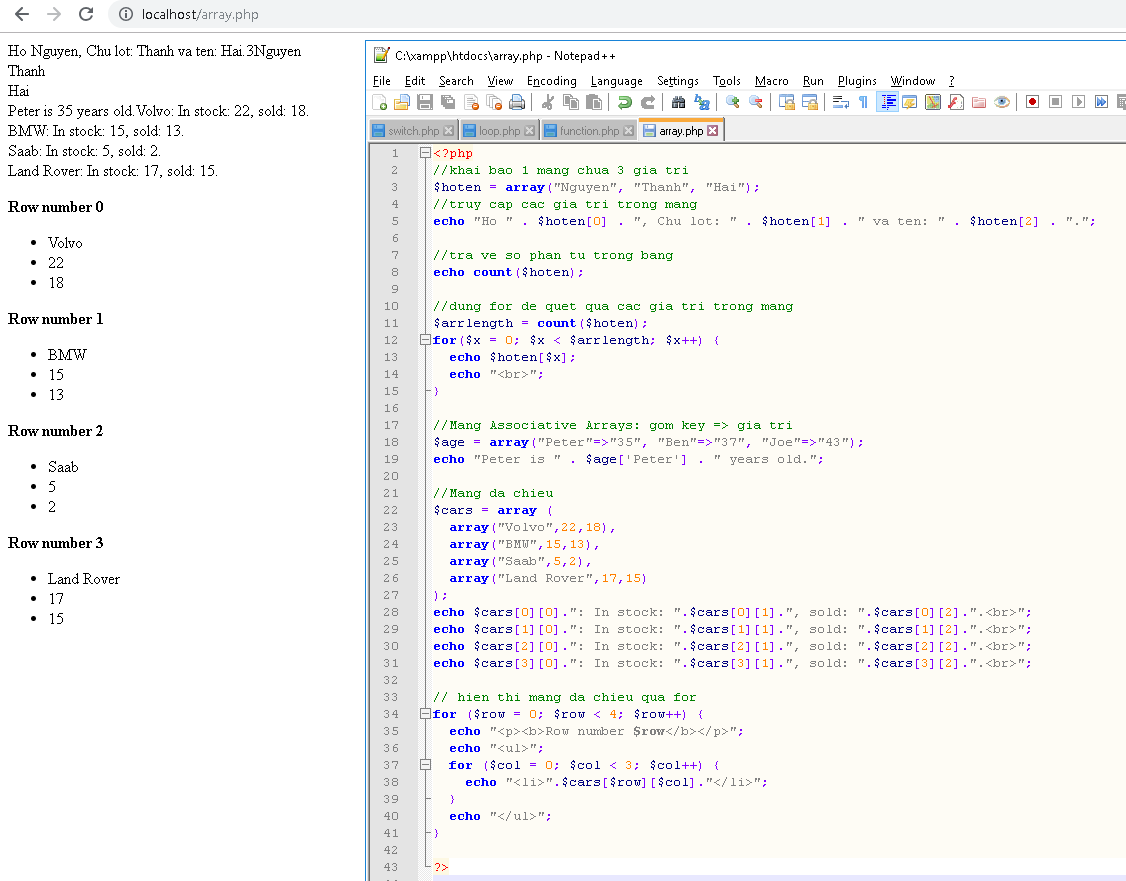
*Loop.php*



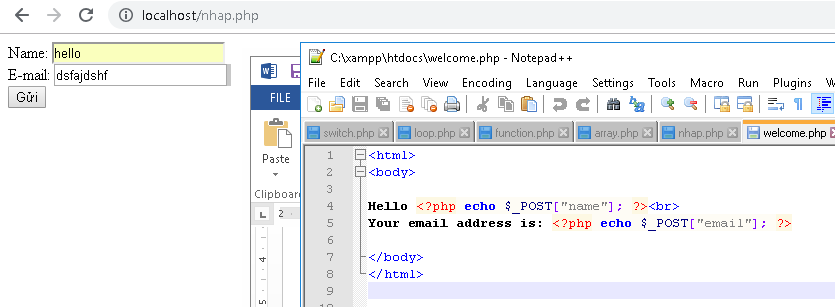
*Function.php*



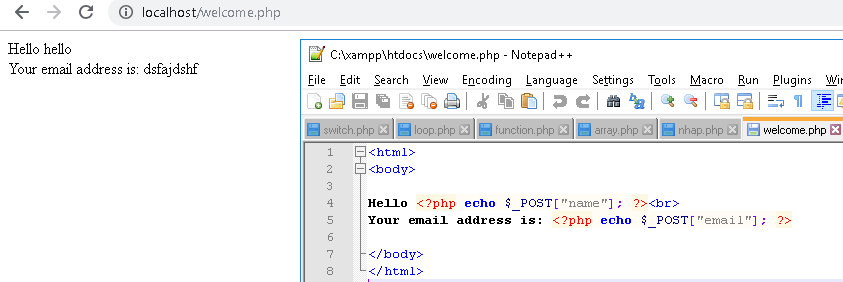
*Array.php*



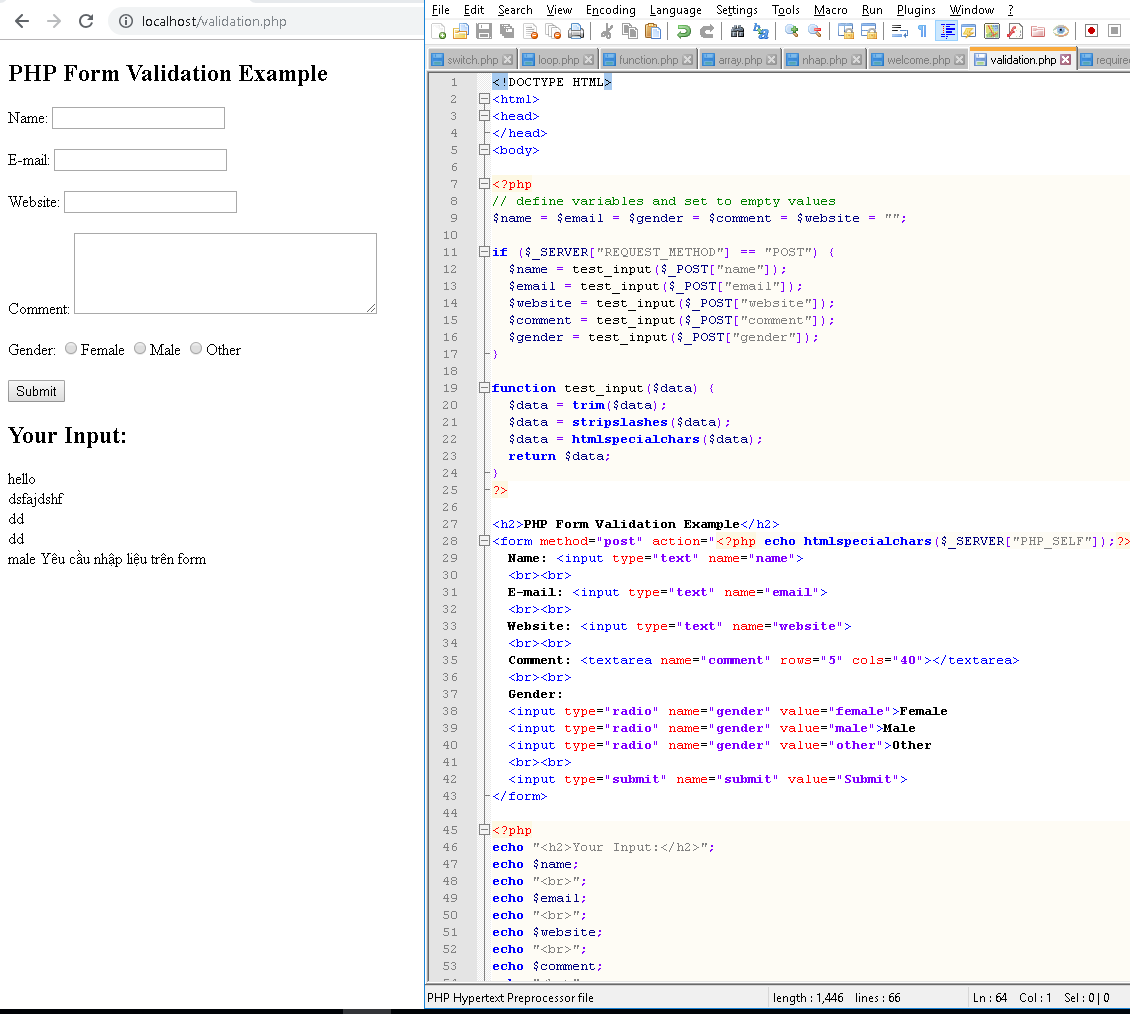
*Nhap.php*



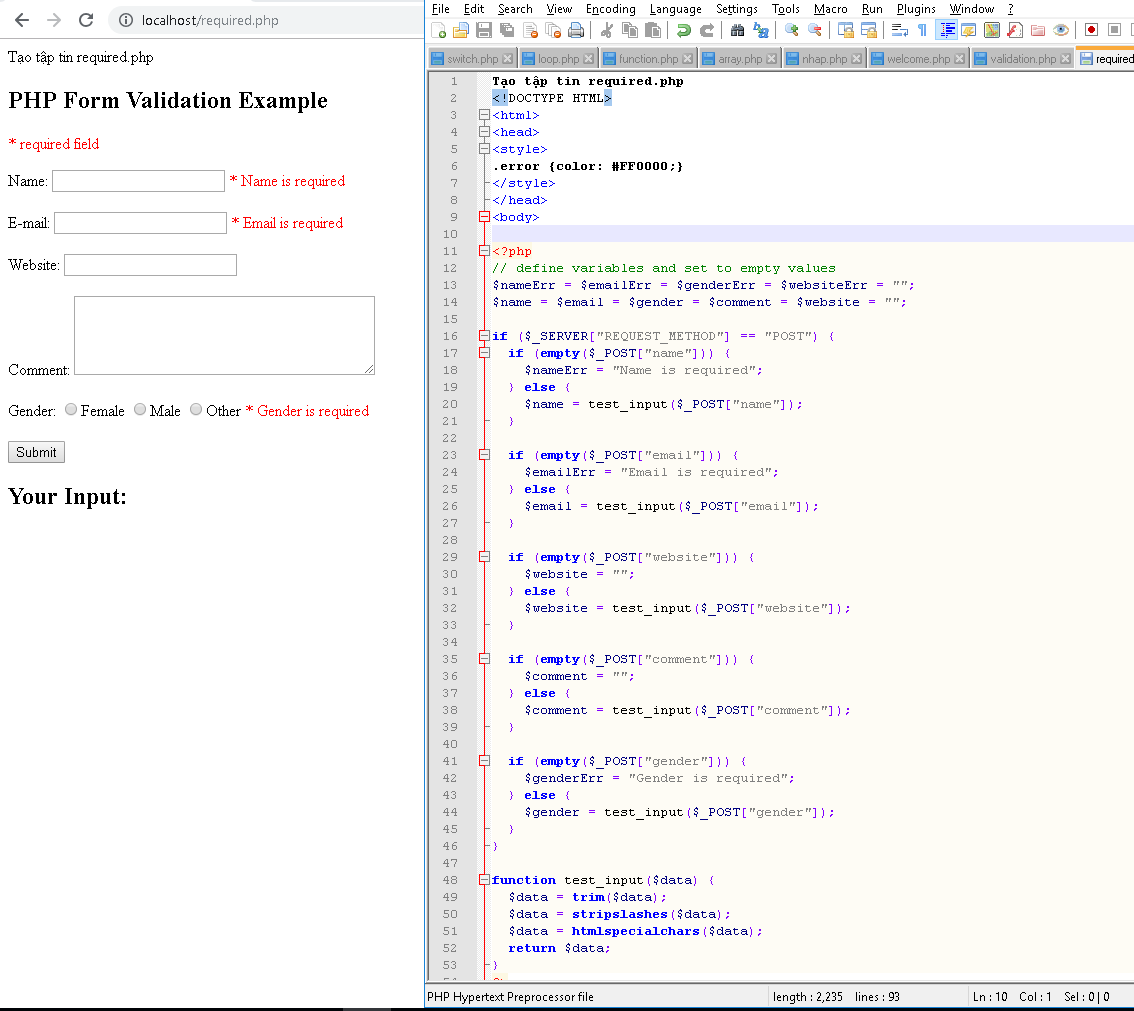
*Welcome.php*



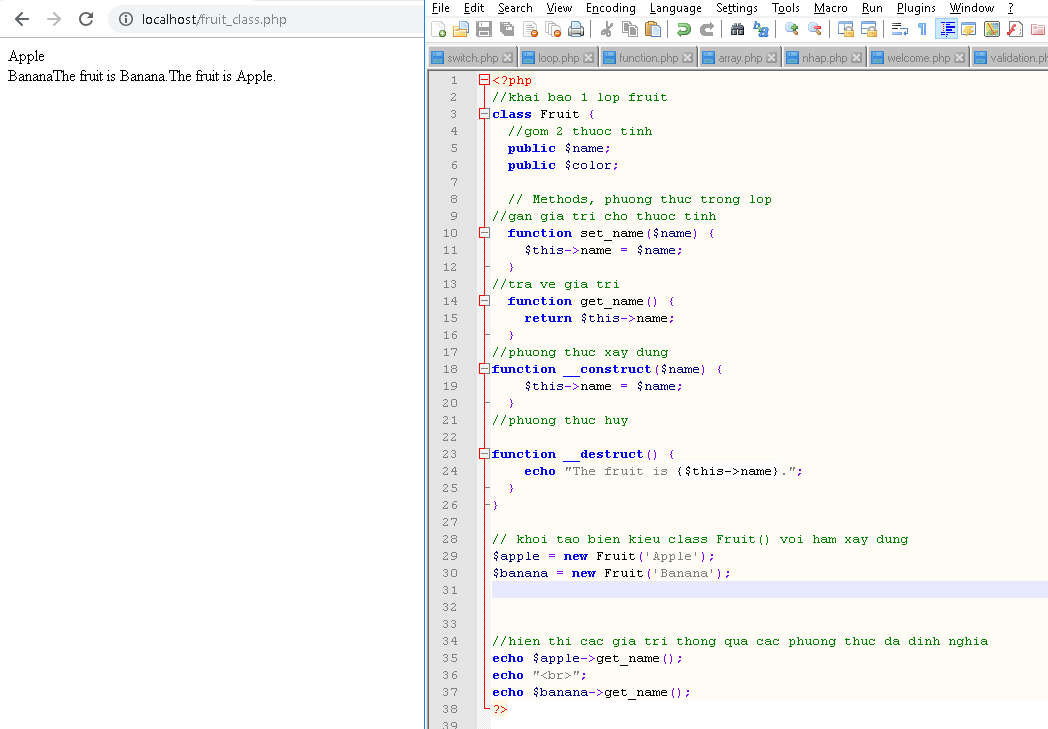
*Validation.php*



*Required.php*



*Fruit\_class.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:

$x ;

Gán trị cho biến:

$x = 5;

<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>

<?php  
echo "Today is " . date("Y/m/d") . "<br>";  
echo "Today is " . date("Y.m.d") . "<br>";  
echo "Today is " . date("Y-m-d") . "<br>";  
echo "Today is " . date("l");  
?>

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 thể bắt đầu bằng số

Tên biến chỉ có thể chứa các ký tự chữ-số và dấu gạch dưới (Az, 0-9 và \_)

Tên biến có phân biệt chữ hoa chữ thường ( $agevà $AGElà hai biến khác nhau)

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

Bắt đầu

<?php

Kết thúc

?>

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

/\* \*/; //

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

Có 3 loại khối lệnh để thực hiện vòng lặp: for, while, do while.

Từ khóa:

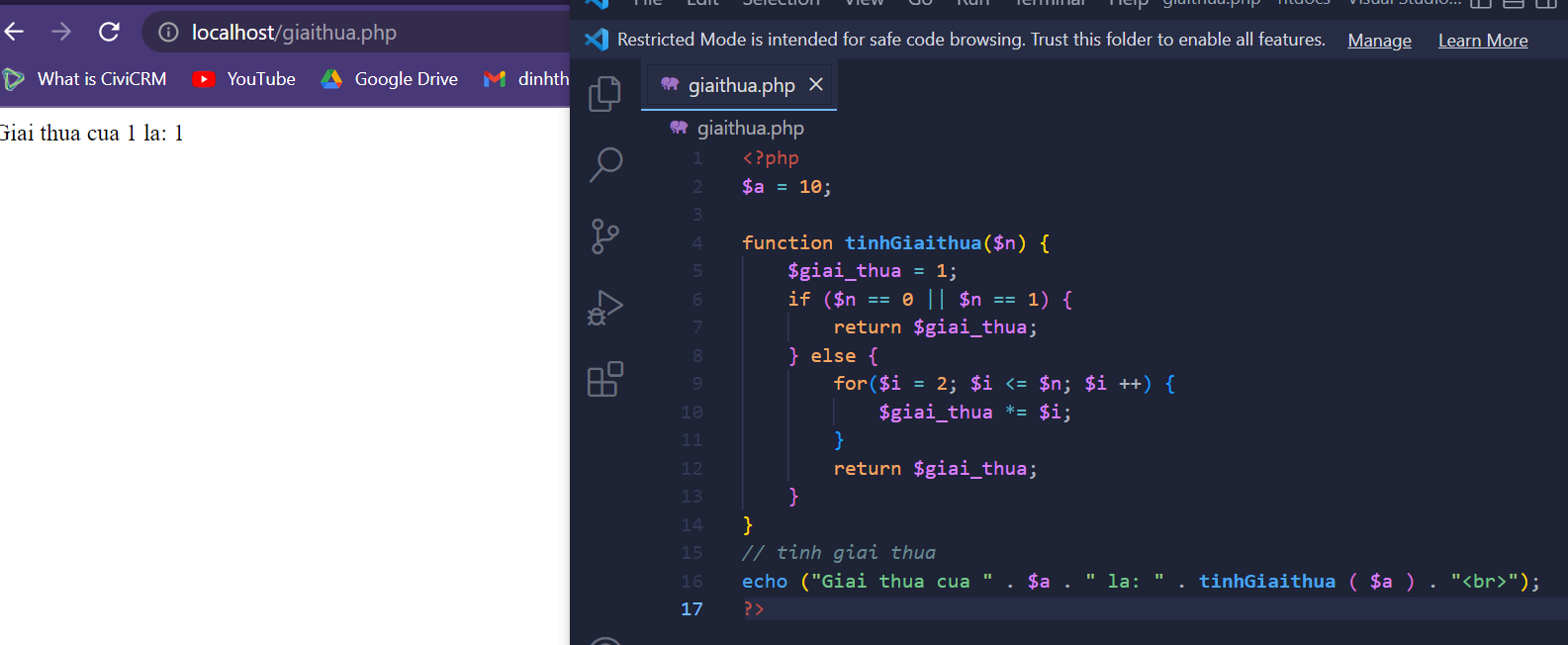
**break** được dùng để thoát dòng lặp, kết thúc vòng lặp hoặc cấu trúc **switch**.

**continue** thường được sử dụng trong vòng lặp **for** để chuyển đến bước cuối cùng trong 1 lần lặp, các lệnh trong thân vòng lặp nằm phía dưới lệnh continue sẽ bị bỏ qua ở lần lặp hiện tại.

<https://blog.luyencode.net/lenh-break-va-continue-trong-c/>

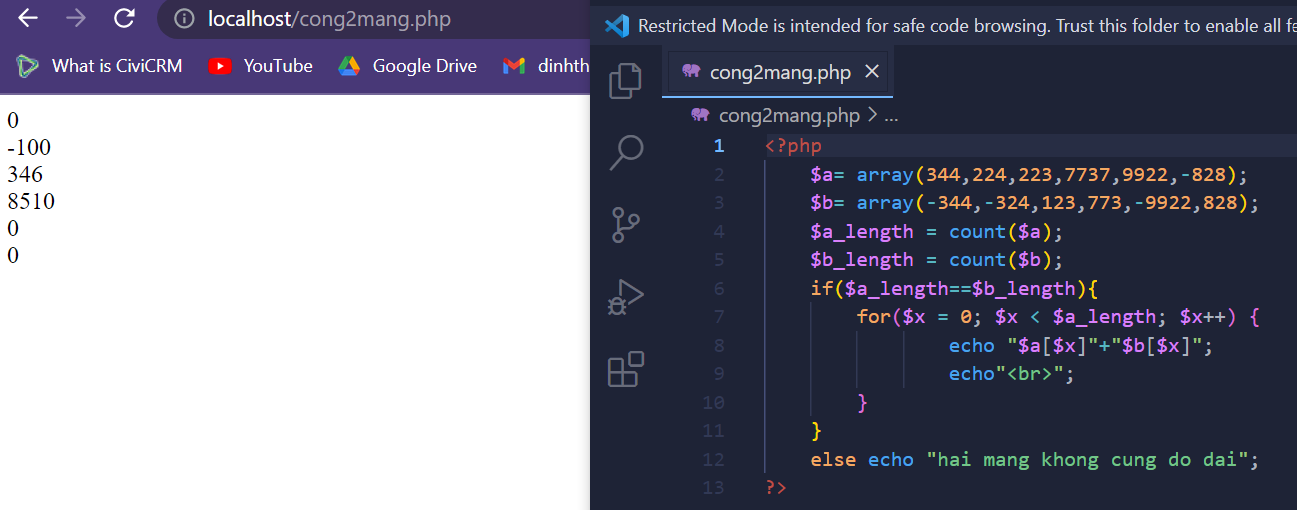
https://cpp.daynhauhoc.com/3/3-tu-khoa-break-va-continue/

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



https://viettuts.vn/bai-tap-php/tinh-giai-thua-trong-php

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.



1. Chạy ví dụ tập tin validation.php, dựa vào link , bạn hãy giải thích những bước diễn ra quá trình này.

Điều đầu tiên chúng ta sẽ làm là chuyển tất cả các biến thông qua hàm htmlspecialchars () của PHP.

Khi chúng ta sử dụng hàm htmlspecialchars (); sau đó nếu người dùng cố gắng gửi nội dung sau trong trường văn bản:

<script> location.href ('http://www.hacked.com') </script>

- điều này sẽ không được thực thi, vì nó sẽ được lưu dưới dạng mã thoát HTML, như thế này:

& lt; script & gt; location.href ('http://www.hacked.com') & lt; / script & gt;

Mã hiện an toàn để được hiển thị trên một trang hoặc bên trong e-mail.

Chúng tôi cũng sẽ thực hiện hai việc nữa khi người dùng gửi biểu mẫu:

Loại bỏ các ký tự không cần thiết (thêm khoảng trắng, tab, dòng mới) khỏi dữ liệu đầu vào của người dùng (với hàm trim () PHP)

Xóa dấu gạch chéo ngược (\) khỏi dữ liệu đầu vào của người dùng (với hàm dấu gạch chéo ngược () trong PHP)

Bước tiếp theo là tạo một hàm sẽ thực hiện tất cả việc kiểm tra cho chúng ta (điều này thuận tiện hơn nhiều so với việc viết đi viết lại cùng một đoạn mã).

Chúng tôi sẽ đặt tên cho hàm test\_input ().

Bây giờ, chúng ta có thể kiểm tra từng biến $ \_POST bằng hàm test\_input ()

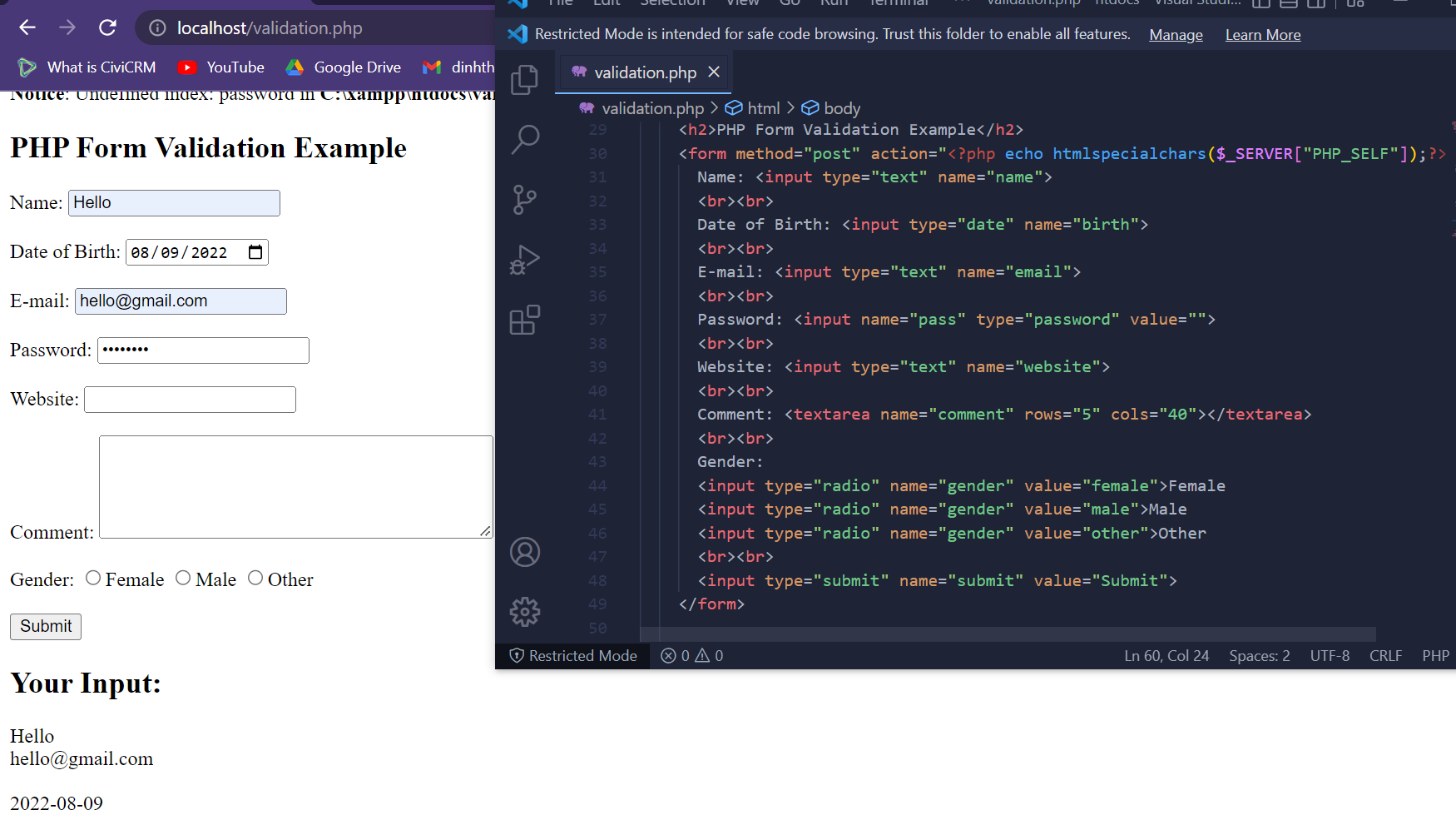
khi bắt đầu tập lệnh, chúng tôi kiểm tra xem biểu mẫu đã được gửi bằng cách sử dụng $ \_SERVER ["REQUEST\_METHOD"] hay chưa. Nếu REQUEST\_METHOD là POST, thì biểu mẫu đã được gửi - và nó sẽ được xác thực. Nếu nó chưa được gửi, hãy bỏ qua xác thực và hiển thị một biểu mẫu trống.

Tập lệnh hoạt động tốt ngay cả khi người dùng không nhập bất kỳ dữ liệu nào.

Bước tiếp theo là yêu cầu các trường nhập và tạo thông báo lỗi nếu cần.

https://www.w3schools.com/php/php\_form\_validation.asp

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.



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.

Chú ý:

* Các bạn nộp file word: Đặt tên file: <**mssv**>-<**hoten**>-<**bai2**>.docx nộp lên classroom, kèm với các file khác được yêu cầu. VD: **B123456-NguyenVanA-bai2.docx.** *Nếu đặt sai quy tắc, bài của bạn sẽ bị thất lạc và sẽ không được tính điểm.*
* Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi theo đúng thứ tự câu hỏi (nếu câu nào không trả lời được các bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời).
* Các bài bị phát hiện copy, sao chép từ các bạn khác sẽ không được tính điểm.
* Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.