HỌC PHẦN "LẬP TRÌNH WEB VỚI PHP"

(Web programming with PHP)

GVGD: ThS. Trần Mạnh Đông

PHP cơ bản

NỘI DUNG (3)

- Xuất dữ liệu ra trình duyệt với:
  - Lệnh **echo**
  - Lệnh **print**
- Xuất dữ liệu ra trình duyệt với các hàm
  - Hàm **print_r()**
  - Hàm **var_dump()**
  - Hàm **var_export()**

Lệnh echo trong PHP

- **echo:** là một câu lệnh của ngôn ngữ lập trình PHP, không phải là một hàm
- Lệnh **echo** được sử dụng để in chuỗi, chuỗi nhiều dòng, biến, mảng,…
- Có thể được sử dụng có hoặc không có dấu ngoặc đơn: echo() hoặc echo …;
- Echo không trả về bất kỳ giá trị nào

Lệnh **echo** trong PHP…

- Ví dụ: In chuỗi ra trình duyệt

```
<?php
   echo "Hello World!";
?>
```

Lệnh echo trong PHP…

- Ví dụ: in chuỗi nhiều dòng

```
<?php
echo "Bạn học các ngôn ngữ lập trình sau!
    Ngôn ngữ lập trình PHP
    Ngôn ngữ lập trình Java
    Ngôn ngữ lập trình C sharp
    Ngôn ngữ lập trình Python";
?>
```

Lệnh echo trong PHP…

- Ví dụ: in ký tự đặc biệt

```
<?php
	echo "In dấu \"ngoặc kép\".";
?>
```

Lệnh echo trong PHP…

- Ví dụ: In giá trị của Biến

```
<?php
   $subject="Lập trình Web với PHP!";
   echo "Môn học: $subject";
?>
```

Lệnh **print** trong PHP

- **print:** là một lệnh của ngôn ngữ lập trình PHP, không phải là một hàm
- Lệnh **print** được sử dụng để in dữ liệu ra trình duyệt
- Khác nhau giữa **echo** và **print**:
  - **echo**: chấp nhận một danh sách tham số, cách nhau bởi dấu phẩy, và không giới hạn bằng đóng mở ngoặc nhọn. echo cũng không trả lại bất kì giá trị nào
  - **print**: chỉ chấp nhận 1 tham số duy nhất. Và nó cũng không thể dùng như là biến hàm được. print chỉ có thể in ra dạng xâu, và chậm hơn so với echo trong việc này
  - **echo** là nhẹ và nhanh hơn **print**

Hàm print_r()

- Hàm **print_r()** giúp chúng ta xuất *dữ liệu mảng (array)* phục vụ cho quá trình kiểm tra trong xử lý chương trình

```
print_r(variable, return);
//hoặc
print_r(array $data);
```

- Cú pháp:
  - *variable* là biến muốn in thông tin. *variable* có thể có kiểu dữ liệu bất kỳ.
  - *return* là tham số quyết định kiểu trả về của hàm print_r() là string hoặc bool. Mặc định return=false
  - Kiểu dữ liệu trả về của hàm **print_r()** là **string** hoặc **bool**. Nếu **$return=false** thì hàm **print_r()** có kiểu dữ liệu trả về là **bool** (trả về **true** hay giá trị **1**) và sẽ in ra thông tin của biến **$value**. Nếu **$return=true** thì hàm **print_r()** có kiểu dữ liệu trả về là **string** chứa thông tin của biến **$value** và không in ra thông tin của biến **$value**
  - Input: $data là mảng cần xuất dữ liệu lên màn hình. Output: Dữ liệu hiển thị theo cấu trúc mảng
  - Sử dụng kết hợp với thẻ<pre> để in mảng rõ hơn mảng, đối tượng.

Hàm print_r()

- **Điểm khác nhau giữa echo, print và print_r trong PHP:**
  - **echo**: chấp nhận một danh sách tham số, cách nhau bởi dấu phẩy, và không giới hạn bằng đóng mở ngoặc nhọn. echo cũng không trả lại bất kì giá trị nào
  - **print**: chỉ chấp nhận 1 tham số duy nhất. Và nó cũng không thể dùng như là biến hàm được. print chỉ có thể in ra dạng xâu, và chậm hơn so với echo trong việc này
  - **print_r()**: là một hàm trong PHP. Đầu ra của nó là chi tiết của tham số truyền vào với kiểu của tham số đó.
    - Khi dùng hàm này, output được chứa ở buffer trong và có tham số trả lại. Nếu tham số trả lại là TRUE, print_r() sẽ in đầy đủ thông tin hơn so với việc chỉ đơn giản in ra giá trị của biến.
    - Dùng hàm này giúp chúng ta tránh được một số lỗi khi chạy, và nó khá giống hàm var_dump()

Hàm var_dump()

- Hàm **var_dump()** trong PHP là một công cụ mạnh mẽ dùng để kiểm tra và hiển thị thông tin về các biến, mảng, đối tượng hoặc biểu thức một cách chi tiết:
  - Hiển thị thông tin về giá trị *(với đối tượng thì nó gồm thuộc tính public, private, protected)* và kiểu dữ liệu, của một hoặc nhiều biến kể cả biểu thức
- Cú pháp:

```
var_dump($var, $vars ...);
//hoặc
var_dump($expression);
```

  - $var là biến cần xuất thông tin. $var có thể có kiểu dữ liệu bất kỳ
  - $vars là những biến khác cần xuất thông tin. $vars cũng có thể có kiểu dữ liệu bất kỳ
  - Nếu hàm nhận một đối số duy nhất. *$expression* có thể là một biến duy nhất hoặc một biểu thức chứa một số biến được phân tách bằng dấu cách thuộc bất kỳ loại nào
  - Hàm var_dump() không có kiểu dữ liệu trả về (void)

Hàm var_dump()...

- **Ưu điểm của hàm var_dump() trong PHP:**
  - Hiển thị thông tin chi tiết về biến và dữ liệu
  - Xác định lỗi kiểu dữ liệu, hỗ trợ gỡ lỗi dễ dàng
  - Kiểm tra dữ liệu từ form và yêu cầu: Khi chúng ta xử lý dữ liệu gửi từ HTML hoặc các yêu cầu HTTP, var_dump có thể giúp chúng ta kiểm tra được dữ liệu có truyền đúng và phù hợp hay không
  - Phân tích rõ hơn về dữ liệu đầu vào: Khi làm việc với dữ liệu từ nguồn bên ngoài như cơ sở dữ liệu hoặc API, var_dump cho phép chúng ta thấy rõ hơn về cấu trúc và kiểu dữ liệu
- **Một số lưu ý khi sử dụng hàm var_dump trong PHP**
  - var_dump có thể hiển thị thông tin nhạy cảm như mật khẩu hoặc dữ liệu cá nhân, vì vậy cần cẩn trọng khi sử dụng nó trong môi trường thật

Hàm var_export()

- Hàm ***var_export()*** giống với ***var_dump()***, nó kết xuất thông tin của một biến (hoặc biểu thức), nhưng có thêm thiết lập thông tin đó xuất ra <u>*output*</u> hay trả về chuỗi.
- Cú pháp:

```
var_export($var, $return);
```

  - $var là biến cần xuất thông tin. *$var* có thể có kiểu dữ liệu bất kỳ.
  - $return là tham số quyết định kiểu dữ liệu trả về của hàm *var_export().* Mặc định *$return=false*.
  - Nếu *$return=false* thì hàm *var_export()* xuất thông tin của biến *$var* và hàm *var_export()* trả về *NULL*. Nếu *$return=true* thì hàm *var_export()* trả về một chuỗi (string) chứa thông tin của biến *$var*.

Hàm var_export()…

- Lưu ý:
  - Hàm print_r() thường dùng để xuất mảng (array).
  - Hàm var_dump() được sử dụng khi muốn biết kiểu dữ liệu biến.
  - Hàm var_export() để hiển thị cấu trúc của các biến khác nhau trong PHP

Debug: Demo

- Demo echo, print
- Demo print_r()
- Demo var_dump()
- Demo var_export()