

PHP Quick reference

A. Cú Pháp Cơ Bản

1 Nhúng PHP

```
<p>HTML 1</p>
<?php echo 'HTML-output from PHP'; ?>
<p>HTML 2</p>
```

2 Các Biến

```
$name = "Anthony";
$pointerName = &$name;
$a = "bc";
$$a = "a"; //Biến tên bc có giá trị là "a"
Sử dụng với nhiều mảng: ${a}[1]
hoặc ${a}[1]
Hằng số: define("constantA", "value");
```

4 Các Hàm

Kiểm tra kiểu của dữ liệu: is_array, is_binary, is_bool, is_buffer, is_callable, is_double, is_float, is_int, is_integer, is_long, is_null, is_numeric, is_object, is_real, is_resource, is_scalar, is_string, is_unicode, get_resource_type, gettype.
Kiểm tra: empty, isset.
Cast: floatval, intval, strval, settype.
Đầu ra: var_dump, var_export, print_r.
Xóa: unset.
Tuần tự hóa: serialize, unserialize.

3 Các Biến Được Xác Định

\$GLOBALS: biến toàn cục
\$_SERVER: Lưu trữ thông tin về máy chủ và môi trường thực thi.
\$_GET: Lưu trữ các dữ liệu của phương thức GET Client gửi lên.
\$_POST: Lưu trữ các dữ liệu của phương thức POST Client gửi lên.
\$_FILES: Lưu trữ thông tin của tệp được tải lên.
\$_REQUEST: Lưu trữ các dữ liệu được gửi lên bao gồm cả phương thức GET và POST.
\$_argc: Lưu trữ số lượng đối số được truyền lên.
\$_argv: Lưu trữ mảng các đối số được truyền lên.
\$_SESSION: Chứa Session.
\$_ENV: Biến môi trường.
\$_COOKIE: Lưu trữ dữ liệu cookie nhận được từ trình duyệt người dùng.
\$php_errormsg: Biến lưu trữ thông báo lỗi cuối cùng.
\$_HTTP_RAW_POST_DATA: Lưu trữ dữ liệu thô của phương thức POST.
\$_HTTP_response_header: Lưu trữ 1 mảng các tiêu đề phản hồi từ HTTP

2 Kiểu Của Mảng

Mảng Số

```
// Sử dụng của mảng
$farben = array("red", "green");
// Gán giá trị
// Chỉ số bắt đầu từ 0
$farben[] = "red";
$farben[] = "green";
// Truy cập
Echo $color[1]; // Kết quả in ra "green"
```

Assoziatives Array

```
// Sử dụng của mảng
$sale = array("Jan" => 100,
              "Feb" => 120);
// Gán giá trị
$sales["Jan"] = 100;
$sales["Feb"] = 120;
// Truy cập
Echo $umsaetze["Jan"];
```

1 Khai Báo

Sử dụng array() để tạo mảng
Xây dựng mảng
array([mixed \$...]),
array(key => value, ...)
Hoặc \$array[key] = value;
\$arr[] = value;

B. Mảng

3 Các Hàm

array_flip: Đảo ngược lại key và value.
array_intersect: Tạo giao điểm các phần tử.
array_pop: Truy xuất phần tử cuối cùng.
array_push: Thêm phần tử vào cuối mảng.
array_rand: Lấy các phần tử ngẫu nhiên.
array_reverse: Đảo ngược thứ tự các phần tử.
array_search: Tìm kiếm một mảng và truy xuất các khóa.
array_shift: Truy xuất phần tử đầu tiên.
array_slice: Trích xuất một tập hợp con.
count: Đếm các phần tử
in_array: Kiểm tra sự xuất hiện của một giá trị.
shuffle: Phân tách các phần tử.

5 Bao Gồm Tập Lệnh

Các câu lệnh require("file.php") và include("file.php") có chức năng chèn thêm các tệp PHP. Khi có lỗi nếu sử dụng require thì sẽ tạo một cảnh báo và nếu sử dụng include thì sẽ tạo một lỗi nghiêm trọng. Các câu lệnh require_once và include_once chỉ thực hiện đưa vào cùng một tệp một lần.

C. Cấu Trúc Điều Khiển

1 Câu Lệnh Điều Khiển

Các câu lệnh điều khiển kiểm tra một biểu thức boolean và câu lệnh thực thi tùy thuộc vào đầu ra kiểm tra là đúng hay sai. Các biểu thức có thể kết hợp nhiều biểu thức với đầu && (và) hoặc || (hoặc).
Ví dụ:
\$expr = true;
if (expr) echo "singleStatementInNextRow";
\$number1 = 5;
\$number2 = 2;
if (\$number1 = 4){
 // statements
}
elseif (\$number1 >= 4 && \$number2 < 3){
 // statements
}
else {
 // statements
}

switch (\$animal) {
 case "Dog":
 // statements break;
 case "Cat":
 // statements break;
 default:
 // mặc định sẽ thực thi khi không có case nào đúng
}

3 Ngoại Lệ

Một ngoại lệ là một thông báo lỗi. Lớp cơ sở cho các trường hợp ngoại lệ đặc biệt thường là Exception. Một ngoại lệ được ném (kích hoạt) bởi throw. Một khối trong ngoại lệ là throw được xử lý bên trong khối try{} và ngoại lệ được xử lý trong khối catch{}.
Ví dụ:
// Throwing an exception
throw new Exception("Msg text", 123);
// Method call and exception handling
try {
 // Statements which may cause exception
} catch (MyException \$e) {
 echo \$e->getMessage(). " (".\$e->getCode().")";
} catch (Exception \$e) {
 echo \$e->getMessage(). " (".\$e->getCode().")";
}
// Declaring a specialized exception
class MyException extends Exception {
 public function __construct(\$message = "Msg text", \$code = 123) {
 \$this->message = \$message;
 \$this->code = \$code;
 }
 // Additional methods ... }

4 Báo Cáo Bổ Sung

Lệnh break dừng việc thực thi của vòng lặp. Lệnh continue bỏ qua việc thực hiện các câu lệnh và tiếp tục vòng lặp.
Ví dụ:
for (\$i = 0; \$i < 10; \$i++) {
 if (\$i == 2) continue;
 if (\$i % 2 == 0 || \$i == 5) {
 echo \$i."
";
 }
 if (\$i > 8) break;
}

2 Vòng Lặp

Các vòng lặp thực hiện các câu lệnh liên tục cho đến khi điều kiện là đúng.
Ví dụ:
\$number = 1;
while (\$number <= 5) {
 echo \$number."
";
 \$number++;
}

for (\$i = 0; \$i < 10; \$i++) {
 echo \$i."
";
}

Thực hiện các câu lệnh một lần và kiểm tra điều kiện ngay sau đó. Nếu điều kiện là đúng thì sẽ lặp lại thực thi câu lệnh cho đến khi điều kiện sai.
Ví dụ:
\$number = 10;
do {
 echo \$number."
";
 \$number++;
} while (\$number <= 5);
Duyệt qua các phần tử của mảng và thực hiện các câu lệnh cho mỗi lần lặp. Chỉ số của mảng sẽ được gán tạm thời cho biến.
Ví dụ:
\$celBodies = array("sun" => "yellow", "moon" => "pale", "stars" => "white");
foreach (\$celBodies as \$name => \$celBody) {
 echo \$name.": ".\$celBody."
";
}