**1, Hàm là gì?**

-Hàm là một hoặc nhiều đoạn mã được viết ra để thực thi một hoặc nhiều hành động khi gọi nó, hàm có khả năng gọi đi gọi lại được. VD như thay vì tính tổng của 2 số nhất định chúng ta có thể sử dụng hàm để tính tổng của 2 số bất kỳ mà chúng ta muốn.

(Nghe hơi khó hiểu đúng không? Mình sẽ giải thích ở phần giới qua các ví dụ)

**2, Cấu trúc hàm.**

-Cú Pháp:

function tenHam($param)

{

// code

}

**Trong đó:**

* **function**: là từ khóa bắt buộc để khai báo hàm trong PHP.
* **tenHam**: là tên của Hàm các bạn muốn đặt nó cũng bao gồm các ràng buộc như đặt [tên biến](http://toidicode.com/bien-va-hang-trong-php-54.html).
* **$param**: là các tham số mà các bạn muốn truyền vào trong hàm(không giới hạn số lượng tham số truyền vào).

-**Chú ý**: Trong PHP không ràng buộc hàm có giá trị trả về với không có giá trị trả về nên có thể chứa return hoặc không.

**VD1**: Hàm tính tổng có giá trị trả về và không có giá trị trả về.

function tinhTong($a,$b)

{

return $a + $b;

}

hoặc

function tinhTong($a,$b)

{

echo $a+$b;

}

**Giá trị mặc định cho biến truyền vào.**

-Để set tham số mặc định cho biến truyền vào hàm chúng ta sử dụng theo cú pháp:

function tenham($thamso = 'giatri')

{

}

**Trong đó: $thamso**là các tham số các bạn muốn truyền vào khi gọi và **giatri** là giá trị mặc định của tham số đó.

**VD2**:

function readName($name = 'world')

{

echo 'hello' . $name;

}

- Ở ví dụ trên mình xây dựng một hàm **readName** để đọc tên của người và nếu như khi gọi hàm **readName** mà chúng ta không truyền tham số **$name** cho nó thì mặc định nó sẽ in ra '*hello world*', và nếu như có truyền tham số thì nó sẽ in ra hello + tham số đó.

**3, Gọi hàm.**

-Phía trên mình mới chỉ hướng dẫn các bạn xây dựng hàm thôi và để sử dụng được hàm đó chúng ta cần phải gọi hàm. Để gọi hàm trong PHP thì các bạn làm theo cú pháp sau:

tenHam();

//hoặc

tenHam($param);

**Trong đó:**

* **tenHam**: là tên của hàm các bạn muốn gọi.
* **$param**: là tham số các bạn muốn truyền vào hàm (nếu lúc khai báo hàm có yêu cầu truyền tham số).

Để cho dễ hiểu hơn thì các bạn xem VD sau nhé:

<?php

//khởi tạo hàm readName và thiết lập tham số mặc định

function readName($name = 'world')

{

echo 'hello' . $name;

}

//gọi hàm không truyền tham số

readName(); // kết quả: hello world

//gọi hàm có truyền tham số

readName('Tài'); //kết quả: hello Tài

//khởi tạo biến a

$a = 'Toidicode';

//gọi hàm

readName($a);//kết quả: hello toidicode

?>

**4, Các ví dụ khác.**

**Hàm với câu lệnh rẽ nhánh**

**Đề bài**: viết hàm kiểm tra số nếu số lớn hơn 10 thì in ra '*lớn hơn 10*' và nhỏ hơn hoặc bằng 10 thì in ra '*nhỏ hơn bằng 10*':

<?php

//khởi tạo hàm checkNumber có tham số truyền vào và đồng thời xét tham số mặc định cho nó = 0

function checkNumber($number = 0)

{

// nếu số truyền vào lớn hơn 10

if ($number > 10) {

echo 'lớn hơn';

} // nếu số truyền vào nhỏ hơn 10

else {

echo 'nhỏ hơn hoặc bằng';

}

}

// gọi hàm

//gọi hàm không truyền tham số

checkNumber(); //kết quả nhỏ hơn hoặc bằng

//gọi hàm có truyền tham số

checkNumber(11);// kết quả lớn hơn

**Hàm với vòng lặp**

**Đề bài**: viết hàm in ra số từ 1 đến số n, với n là tham số tùy chỉnh.

<?php

//khởi tạo hàm loopNumber có tham số truyền vào và đồng thời xét tham số mặc định cho nó = 0

function loopNumber($number = 0)

{

for ($i = 0; $i <= $number; $i++) {

echo $i;

}

}

//gọi hàm

//gọi hàm không truyền tham số

loopNumber(); // kết quả: 0;

// gọi hàm truyền tham số

loopNumber(10); //kết quả: 012345678910

**Chú ý**: Hai ví dụ trên mình mới chỉ làm ở mức độ basic nhất, chưa bao gồm ràng buộc chặt chẽ tham số truyền vào.

**5, Kiểm tra hàm đã tồn tại.**

-Trong thực tế khi xây dựng các dự án với PHP hướng thủ tục thì số lượng các hàm sẽ không dừng ở con số 5,10,20,... mà nó sẽ lớn hơn rất là nhiều, như vậy thì chuyện trùng lặp hàm là điều không thể tránh khỏi đối với một lập trình viên không chuyên nghiệp hoặc một lý do nào khác khiến việc trùng lặp hàm xảy ra. Chính vì đều đó trong PHP đã cung cấp cho chúng ta một hàm function\_exists() để giải quyết vấn đề đó.

**Cú Pháp**

function\_exists('functionName');

-Trong đó: functionName là tên của hàm các bạn kiểm tra và hàm này sẽ trả về giá trị TRUE nếu hàm đã tồn tại và  ngược lại FALSE nếu chưa tồn tại.

**VD3**:

<?php

if (!function\_exists('loopNumber')) {

function loopNumber($number = 0)

{

for ($i = 0; $i <= $number; $i++) {

echo $i;

}

}

}