

LAB 3: CẤU TRÚC LẶP VÀ HÀM

1. Mục tiêu

- Vận dụng được các cấu trúc lặp để giải quyết các bài toán cơ bản
- Định nghĩa hàm trong PHP
- Phân biệt các loại tham số.
- Phân biệt biến toàn cục và biến cục bộ

2. Tóm tắt lý thuyết

❖ Cấu trúc lặp

Cấu trúc while

while(điều kiện) // Trong khi điều kiện còn đúng thì thực hiện các lệnh bên trong

```
{  
    Khối lệnh  
}
```

Cấu trúc do ...while

do//thực hiện khối lệnh đến khi điều kiện sai thì dừng lại

```
{  
    Khối lệnh  
}while(điều_kiện);
```

Cấu trúc lặp For

for(biến đếm; điều kiện dừng; biểu thức được thực hiện sau mỗi lần lặp)

```
{  
    Khối lệnh  
}
```

Cấu trúc lặp foreach

foreach(\$tên_mảng as \$giá_trị)

```
{  
    Khối lệnh  
}
```

❖ Hàm

Định nghĩa hàm

```
function ten_ham( [các tham số])
```

```
{  
    Khởi lệnh  
    [return giá_trị_trả_về]  
}
```

Ví dụ:

```
function hello()  
{  
    echo "Hello world";  
}  
function cong($a,$b)  
{  
    return $a+$b;  
}
```

Sử dụng tham số mặc định: các tham số được gán giá trị mặc định tính từ phải sang. Nếu khi gọi hàm, ta không truyền tham số vào hàm, hàm sẽ lấy giá trị mặc định.

```
Function F($a, $b=2)  
{  
    Return $a*$b;  
}  
Echo F(3);// hiển thị 6
```

Truyền tham chiếu

```
function swap(&$a, &$b)  
{  
    $t=$a;  
    $a=$b;  
    $b=$t;  
}
```

❖ **Biến toàn cục và biến cục bộ**

- Biến toàn cục là biến được khai báo bên ngoài hàm. Nếu muốn sử dụng lại biến toàn cục bên trong hàm thì bên trong hàm, biến phải được khai báo lại với từ khóa global.
- Biến cục bộ là biến được khai báo bên trong 1 hàm. Biến này được tạo ra khi gọi hàm và sẽ bị hủy khi việc thực thi hàm kết thúc.

3. Ví dụ

- 3.1 Ví dụ lab3_1.php, sử dụng for, while và do ... while để tính $S = 1 + 2 + 3 + \dots + 100$
- 3.2 Ví dụ lab3_2.php định nghĩa 1 hàm trong PHP và phân biệt truyền tham số và tham trị
- 3.3 Ví dụ lab3_3.php giúp phân biệt biến cục bộ và biến toàn cục.
- 3.4 Ví dụ lab3_4.php: kết hợp vòng lặp và hàm .
- 3.5 Ví dụ lab3_5.php: Xây dựng 2 hàm BCC(\$n, \$color) và BanCo(\$size): để in ra Bảng cửu chương \$n với màu nền là \$color trong BCC(), và in ra bàn cờ dạng cờ vua với kích thước \$size.

4. Vận dụng

- 4.1 Sử dụng vòng lặp for để tính tổng các số chẵn từ 2 đến 100.
- 4.2 Sử dụng while hay do ...while để tìm n nhỏ nhất sao cho $1 + 2 + \dots + n > 1000$.
- 4.3 Trong ví dụ lab3_4.php sử dụng vòng lặp for. Hãy sử dụng do...while để thay cho for.
- 4.4 Thay đổi ví dụ lab3_5.php:
 - a. Hàm bảng cửu chương (BCC): sẽ nhận các tham số \$n, \$colorHead, \$color1, \$color2: \$colorHead: màu nền của dòng đầu tiên, \$color1: màu nền dòng lẻ, \$color2: màu nền các dòng chẵn. Các màu này đều có giá trị mặc định.
 - b. Các hàm này không xuất trực tiếp trong hàm mà tổng hợp thành một chuỗi và return chuỗi cần xuất.
 - c. Đưa các định nghĩa hàm này ra file function.php, và sử dụng nhóm câu lệnh include để load vào lab3_5.php.
 - d. Gọi các hàm vừa định nghĩa.

5. Nâng cao

- 5.1 Xuất n số nguyên tố đầu tiên.
- 5.2 Viết hàm kiểm tra một chuỗi có đối xứng không
Ví dụ:
 - abcba là chuỗi đối xứng
 - abcdab là chuỗi không đối xứng
- 5.3 Viết hàm tính tổng các chữ số có trong một chuỗi
Ví dụ có chuỗi ngay15thang7nam2015
→ tổng các chữ số = $1+5+7+2+0+1+5=21$

- 5.4 Viết hàm xuất ra hình chữ nhật rỗng có chiều dài là d và chiều rộng là r
Ví dụ d=6, r=4 có kết quả như sau:

```
* * * * * *  
*           *  
*           *  
* * * * * *
```

- 5.5 Viết hàm tính tổng các số có trong một chuỗi

Ví dụ có chuỗi ngay15thang7nam2015

→ tổng các chữ số = 15+7+2015=2037

- 5.6 Viết hàm loại bỏ các khoảng trắng dư thừa trong một chuỗi.