

CHUỖI TRONG PHP

A – Lý Thuyết

1 – Kiểu Dữ Liệu Chuỗi (String)

Kết quả:

String là chuỗi các ký tự có chiều dài vô hạn. Chúng chứa các chữ cái, ký tự ASCII, ký tự số. Nội dung của String phải được đánh dấu nơi bắt đầu và kết thúc của chúng.

Có hai cách thường dùng :

1. **Single Quotes (nháy đơn):**

```
$var = 'Nhất Nghệ';  
print 'Nhất Nghệ';
```

2. **Double Quotes (nháy đôi):**

```
$var = "Nhất Nghệ";  
print "Nhất Nghệ";
```

Hai cách này tương tự nhau nhưng chúng cũng khác nhau, hãy chạy thử đoạn code sau:

<?php

```
$var = "TRUNG TÂM ĐÀO TẠO MẠNG MÁY TÍNH NHẬT NGHỆ ";  
print "$var NHATNGHE<br>";           // Dùng nháy kép  
print '$var NHATNGHE <br>';           #Dùng nháy đơn  
print $var . ' NHATNGHE';
```

?>

Để đưa dấu nháy đơn hay kép vào trong chuỗi, cần đặt ký tự \ ngay sau nháy, VD:

```
$var = " Nhất Nghệ "; // Đúng  
$var = 'Nhất Nghệ'; // Sai  
$var = 'Nhất Nghệ' ; // Đúng  
$x = "mmm\"oo'oo\n";  
$x = 'mmm"oo\'oo\n';  
$name = "shlomi";  
$animals = array("dog" => "azit" , "cat" => "mitzi");  
echo "hello ! My name is $name and my dog's name is ${animals['dog']}";
```

Một số ký tự đặc biệt đi sau ký tự escaped

- \n linefeed – xuống dòng
- \r carriage return – quay về đầu dòng
- \t horizontal tab - TAB

- \\ backslash
- \\$ dollar sign
- " double quote
- \[0-7]{1,3}
- \x[0-9A-Fa-f]{1,2} ký tự thập lục

Ngoài hai cách sử dụng nháy đơn và nháy kép cho chuỗi, PHP còn cung cấp một phương pháp khởi tạo biến chuỗi có nội dung lớn (nhiều dòng) gọi là **heredoc**. Thí dụ:

```
$myStr = <<< End_Of_Quote
    Trung tâm đào tạo mạng máy tính Nhất Nghệ.
    105 Bà Huyện Thanh Quan.
End_Of_Quote;
echo $myStr;
```

Trung tâm đào tạo mạng máy tính Nhất Nghệ.
105 Bà Huyện Thanh Quan.

Các chú ý: Trước và sau <<< phải có khoảng trắng. Cho phép sử dụng nháy đơn và nháy kép trong chuỗi.

```
$dialogue = <<<No_More
    "It's not going to happen!" she fumed. He raised an eyebrow. "Want to bet?"
No_More;
echo $dialogue;
```

"It's not going to happen!" she fumed. He raised an eyebrow. "Want to bet?"

2 – In chuỗi

Dùng phát biểu **echo** cho phép in nhiều giá trị ra trang web. Xem một số thí dụ sau đây:
Có thể sử dụng () khi in một giá trị, thí dụ hai phát biểu sau `echo "Printy";` hay `echo("Printy");` là như nhau.

Xuất một lúc nhiều giá trị thì không được dùng cặp dấu ngoặc. Thí dụ:

```
echo "First", "second", "third"; // Đúng
echo("Hello", "world"); // Sai
```

Dùng hàm `print()` cũng in chuỗi nhưng chỉ cho phép xuất một giá trị. Hàm trả về True nếu in chuỗi thành công, ngược lại trả về False. Thí dụ:

```
if (! print("Hello, world")) {
    die("you're not listening to me!");
} Hello, world
```

Dùng hàm `printf()` cũng in chuỗi nhưng chỉ cho phép định dạng giá trị in ra. Hàm trả về True nếu in chuỗi thành công, ngược lại trả về False. Thí dụ:

* In số thập phân có hai chữ số lẻ 27.45

```
printf("%.2f", 27.452);
```

- In số thập phân và số thập lục.

```
printf("The hex value of %d is %x", 214, 214); The hex value of 214 is d6
```

- Thêm số 0 vào các vị trí tổng khi in số nguyên

```
printf("Bond. James Bond. %03d.", 7); Bond. James Bond. 007.
```

- Định dạng ngày tháng khi in:

```
printf("%02d/%02d/%04d", $month, $day, $year); 12/28/2010
```

- Định dạng phần trăm:

```
printf("%.2f%% Complete", 2.1); 2.10% Complete
```

- Định số vị trí và số chữ số lẻ khi in số thập phân:

```
printf("You've spent $%.2f so far", 4.1); You've spent $ 4.10 so far
```

Hàm `sprintf` cũng cho phép định dạng chuỗi in ra nhưng không xuất kết quả ra trình duyệt mà ra bộ nhớ (lưu vào biến). Thí dụ:

```
$date = sprintf("%02d/%02d/%04d", $month, $day, $year);
```

3 – Truy xuất từng ký tự trong chuỗi

Hàm `strlen(Chuỗi)` sẽ trả về số ký tự trong chuỗi. Thí dụ:

```
$string = 'NHAT NGHE';  
echo $length = strlen($string); // $length = 9
```

Truy xuất từng ký tự trong chuỗi ta dùng cú pháp :

TênBiếnChuỗi { Số thứ tự của ký tự cần xử lý } ;

Chú ý : ký tự đầu tiên sẽ có số thứ tự là 0. Thí dụ:

```
$string = 'NHAT NGHE';  
for ($i=0; $i < strlen($string); $i++) {  
    printf("Kí tự thứ %d là %s<br>", $i, $string{$i} );  
}
```

Kí tự thứ 0 là N
Kí tự thứ 1 là H
Kí tự thứ 2 là A
Kí tự thứ 3 là T
Kí tự thứ 4 là
Kí tự thứ 5 là N
Kí tự thứ 6 là G

Kí tự thứ 7 là H

Kí tự thứ 8 là E

4 - Một số hàm xử lý chuỗi

```
$title = "      Programming PHP      \n";  
$str_1 = ltrim($title); // $str_1 is "Programming PHP      \n" Cắt bỏ khoảng trắng ở bên trái  
$str_2 = rtrim($title); // $str_2 is "      Programming PHP" Cắt bỏ khoảng trắng ở bên phải  
$str_3 = trim($title); // $str_3 is "Programming PHP" Cắt bỏ khoảng trắng ở hai bên
```

```
$record = "    Fred Flintstone\t35\tWilma      \n";  
$record = trim($record, " \r\n\t0\x0B"); // $record is "Fred Flintstone\t35\tWilma"
```

```
$string1 = "TRUNG tam";  
$string2 = "dao tao mang may tinh NHAT NGHE";  
print( strtolower($string1) ); // Kết quả là “trung tam” Chuyển tất cả thành chữ thường  
print( strtoupper($string1) ); // Kết quả là “TRUNG TAM” Chuyển tất cả thành chữ IN  
print( ucfirst($string2) ); // Kết quả là “Dao tao mang may tinh NHAT NGHE” Chuyển ký tự đầu  
câu thành chữ IN  
print( ucwords($string2) ); // Kết quả là “Dao Tao Mang May Tinh NHAT NGHE” Chuyển ký tự  
đầu từ thành chữ IN
```

5 - Một số hàm so sánh chuỗi

PHP có hai toán tử dùng để so sánh chuỗi.

Toán tử `==` chỉ so sánh nội dung không quan tâm đến kiểu dữ liệu (tự động chuyển kiểu).

Toán tử `===` so sánh cả nội dung lẫn kiểu dữ liệu. Thí dụ:

```
$o1 = 3; $o2 = "3";  
if ($o1 == $o2) {  
    echo("== returns true<br>"); }  
if ($o1 === $o2) {  
    echo("=== returns true<br>"); }
```

`== returns true`

Hàm `strcmp(str1, str2)`; so sánh hai chuỗi với nhau, Trả về 0 nếu hai chuỗi bằng nhau, trả về số nhỏ hơn 0 nếu str1 sắp thứ tự trước str2 và trả về số lớn hơn 0 nếu str2 xếp trước str1

Hàm `strcasecmp(str1, str2)`; chuyển đổi các chuỗi về chữ thường rồi mới so sánh. Kết quả trả về cũng như trên.

Tương tự hàm `strncmp(string_1, string_2, len)`; và `strncasecmp(string_1, string_2, len)`; so sánh len ký tự đầu tiên của chuỗi.

6 – Một số hàm xử lý chuỗi

Hàm **substr**(*string*, *start* [, *length*]); lấy ra chuỗi con bắt đầu từ vị trí *start* với *length* ký tự.

Thi dụ:

```
$name = "NHAT NGHE";  
$fluff = substr($name, 6, 4); // $fluff is "GHE"  
$sound = substr($name,); // $sound is "NGHE"
```

Hàm **substr_count**(*Mẹ*, *Con*); đếm số lần xuất hiện chuỗi con trong chuỗi mẹ.

Hàm **substr_replace**(*original*, *new*, *start* [, *length*]); chèn (thay thế nếu *length* khác không) chuỗi con *new* vào trong chuỗi mẹ *original* tại vị trí *start*.

Hàm **strrev**(*string*); trả về chuỗi đảo ngược với chuỗi ban đầu.

Hàm **strpos**(*Mẹ*, *Con*); trả về vị trí đầu tiên mà chuỗi con xuất hiện trong chuỗi mẹ.

Hàm **strrpos**(*Mẹ*, *Con*); trả về vị trí cuối cùng mà chuỗi con xuất hiện trong chuỗi mẹ.

Hàm **strpos**(*Mẹ*, *Con*); tìm vị trí đầu tiên mà chuỗi con xuất hiện trong chuỗi mẹ. Sau đó trả về chuỗi bắt đầu từ vị trí tìm thấy.

Hàm **stristr**(*Mẹ*, *Con*); cũng tương tự như trên nhưng không phân biệt in hay thường.