

# Phần 02: Xử lý chuỗi



## Vấn đề 01: Khái niệm string được hiểu như thế nào?

- Khái niệm string được hiểu như là chuỗi, văn bản.
- Kiểu string được sử dụng để lưu trữ các giá trị có chứa ký tự. Các giá trị này luôn nằm trong cặp dấu nháy đôi hoặc dấu nháy đơn



## Vấn đề 02: Hiển thị ký tự nháy đơn và nháy đôi trong chuỗi ?

- Vì dấu nháy đơn và nháy đôi là các ký tự đặc biệt do đó để hiển thị các ký tự này trong chuỗi chúng ta sử dụng thêm ký tự \



## Vấn đề 03: Nối 2 hay nhiều chuỗi lại với nhau ?

- Sử dụng dấu chấm (kí hiệu .) để nối các giá trị kiểu chuỗi lại với nhau thành một giá trị duy nhất
- Lưu ý phân biệt giữa dấu cộng (+) và dấu chấm (.)
- Viết hàm nối 2 chuỗi bất kỳ bởi một ký tự nối nào đó



## Vấn đề 04: Đếm tổng số ký tự có trong chuỗi ?

- Tổng số ký tự có trong chuỗi, chúng ta thường gọi ngắn gọn là chiều dài của chuỗi.
- Trong PHP để lấy chiều dài của chuỗi, chúng ta sử dụng hàm `strlen()`
- Lưu ý trường hợp chuỗi có chứa các ký tự UTF-8 chúng ta sử dụng hàm `mb_strlen()`



## Vấn đề 05: Đếm số từ có trong chuỗi ?

- Chúng ta sử dụng hàm `str_word_count` để đếm số từ xuất hiện trong chuỗi



## Vấn đề 06: Chuyển đổi chữ thường thành chữ hoa và ngược lại ?

- Để chuyển đổi chữ thường thành chữ HOA, chúng ta dùng hàm `strtoupper($str)`. Ngược lại, ta dùng hàm `strtolower($str)`
- Chuyển đổi ký tự đầu tiên trong chuỗi thành chữ HOA `ucfirst($str)`
- Chuyển đổi ký tự đầu tiên trong chuỗi thành chữ thường `lcfirst($str)`
- Chuyển đổi tất cả các ký tự đầu tiên của các từ trong một chuỗi thành chữ in HOA `ucwords($str)`



## Vấn đề 07: Tìm kiếm vị trí xuất hiện của một từ nào đó trong chuỗi ?

- Sử dụng hàm strpos() để tìm kiếm chỉ số xuất hiện đầu tiên của một từ nào đó trong chuỗi
- Sử dụng hàm strrpos() để tìm kiếm chỉ số xuất hiện cuối cùng của một từ nào đó trong chuỗi
- Lưu ý strpos() và strrpos() dành cho PHP version 4



## Vấn đề 08: Đảo ngược một chuỗi

- Khi cần đảo ngược một chuỗi nào đó chúng ta sử dụng hàm `strrev()`



## Vấn đề 09: Trích xuất nội dung nào đó trong chuỗi

- Ví dụ lấy các ký tự từ vị trí thứ 2 đến vị trí thứ 6 trong một chuỗi nào đó. Khi gặp các yêu cầu dạng này chúng ta sử dụng hàm `substr()`



## Vấn đề 10: Xóa bỏ ký tự nằm bên trái chuỗi

- ltrim(\$str, \$params) sẽ xóa các ký nằm bên trái của một chuỗi nào đó.
- Tham số \$params khi bằng rỗng sẽ xóa bỏ các ký tự sau:
  - "\0" - NULL
  - "\t" - tab
  - "\n" - new line
  - "\x0B" - vertical tab
  - "\r" - carriage return
  - " " - ordinary white space



## Vấn đề 11: Xóa bỏ ký tự nằm bên phải chuỗi

- `rtrim($str, $params)` sẽ xóa các ký nằm bên phải của một chuỗi nào đó.
- Tham số `$params` khi bằng rỗng sẽ xóa bỏ các ký tự sau:
  - `"\0"` - NULL
  - `"\t"` - tab
  - `"\n"` - new line
  - `"\x0B"` - vertical tab
  - `"\r"` - carriage return
  - `" "` - ordinary white space



## Vấn đề 12: Xóa bỏ ký tự nằm bên trái và bên phải chuỗi

- trim(\$str, \$params) sẽ xóa các ký tự nằm bên trái và bên phải của một chuỗi nào đó.
- Tham số \$params khi bằng rỗng sẽ xóa bỏ các ký tự sau:
  - "\0" - NULL
  - "\t" - tab
  - "\n" - new line
  - "\x0B" - vertical tab
  - "\r" - carriage return
  - " " - ordinary white space



## Vấn đề 13: Kiểm tra chuỗi khác rỗng

- Sử dụng các hàm `isset()`, `trim()` để kiểm tra một chuỗi nào đó có khác rỗng hay không



## Vấn đề 14: Chuyển đổi qua lại giữa mảng và chuỗi ?

- implode (\$str, \$array) chuyển các giá trị của mảng \$array thành một chuỗi bao gồm các phần tử cách nhau bởi ký tự \$str
- explode (\$delimiter, \$str) chuyển một chuỗi thành một mảng. Tách chuỗi dựa vào \$delimiter, mỗi đoàn tách ra sẽ thành một phần tử của mảng mới



## Vấn đề 15: Trích xuất nội dung trong chuỗi

- Sử dụng hàm `substr($str, $start, $length)` để truy xuất các đoạn nội dung trong chuỗi
- Xây dựng hàm rút gọn chuỗi. Chúng ta thường gặp ở phần giới thiệu 1 bài viết, trang web chỉ xuất hiện 1 số từ được quy định sẵn và theo sau có thể là dấu ...



## Vấn đề 16: Lặp chuỗi

- Với yêu cầu lặp lại chuỗi \$str với số lần lặp là n, chúng ta sẽ gọi hàm str\_repeat(\$str, \$n)



## Vấn đề 17: Ký tự và mã ASCII

- `chr()` Trả về ký tự tương ứng với mã ASCII được truyền vào
- `ord()` Trả về giá trị ASCII của ký tự đầu tiên trong chuỗi



## Vấn đề 18: Phân tích chuỗi truy vấn

- Chúng ta sẽ sử dụng hàm `parse_str()` để chuyển các nội dung truy vấn vào các biến hoặc mảng



## Vấn đề 19: Phân tích URL

- Sử dụng hàm `parse_url` để truy xuất các thành phần protocol, domain name, path, .. của một URL nào đó



## Các thành phần của URL

**href**

**protocol**

**://**

**host**

**/**

**pathname**

**hash**

**hostname**

**port**



## Xác định các thành phần của URL

<http://www.zend.vn:8080/public/tin-cong-nghe/cd-o-viet-nam.html#title>

- Href: *http://www.zend.vn:8080/public/tin-cong-nghe/cd-o-viet-nam.html#title*
- Protocol: *http*
- Host: *www.zend.vn:8080*
- Hostname: *www.zend.vn*
- Port: *8080*
- Pathname: *public/tin-cong-nghe/cd-o-viet-nam.html#title*
- Hash: *#title*



# Bài tập 01: Truy xuất nội dung

Cho URL như sau:

[http://210.245.126.171/Music/NhacTre/TinhYeu\\_LyMaiTrang/wma32/06\\_BienTham\\_TinhYeu\\_LyMaiTrang.wma](http://210.245.126.171/Music/NhacTre/TinhYeu_LyMaiTrang/wma32/06_BienTham_TinhYeu_LyMaiTrang.wma)

Lấy các giá trị:

- No: 06
- Name: Bien Tham
- Album: Tinh Yeu
- Singer: Ly Mai Trang
- Type: wma