



Phần 02: PHP File

Vấn đề 01: Kiểm tra sự tồn tại của tập tin, thư mục

Sử dụng hàm `file_exist($fileName)` để kiểm tra sự tồn tại của tập tin, thư mục

- `$filename`: tên (đường dẫn) tập tin, thư mục cần kiểm tra
- Kết quả trả về `true` → Tồn tại
- Kết quả trả về `false` → Không tồn tại

Vấn đề 02: Xem một số thông tin cơ bản của tập tin, thư mục

- `filetype($fileName)` trả về kiểu của \$filename (tập tin hoặc thư mục)
- `filesize($fileName)` trả về dung lượng của \$filename (đơn vị bytes)
- `is_readable($fileName)` kiểm tra \$fileName có được quyền đọc hay không ?
- `is_writable($fileName)` kiểm tra \$fileName có được quyền ghi hay không ?
- `is_executable($fileName)` kiểm tra \$fileName có được quyền thực thi hay không ?

Vấn đề 03: Các hàm lấy thông tin từ đường dẫn

- `basename($path)` trả về kết quả là tên của tập tin từ đường dẫn `$path`
- `dirname($path)` trả về tên thư mục tại đường dẫn `$path`
- `pathinfo ($path, $options)` trả về một mảng các thông tin từ đường dẫn `$path`
 - `dirname`
 - `basename`
 - `extension`

Vấn đề 04: Thống kê số dòng, số từ và số ký tự trong file

- `file ($fileName)` đọc tập tin `$fileName` thành một mảng, mỗi dòng trong tập tin `$fileName` tương ứng với một phần tử của mảng
- `file_get_contents ($fileName)` đọc tập tin `$fileName` thành một chuỗi

Vấn đề 05: Ghi nội dung vào tập tin với `file_put_contents`

- Sử dụng hàm `file_put_contents` (`$fileName`, `$data`, `$mod`) để ghi nội dung `$data` vào tập tin `$fileName`
- Quá trình thực thi của hàm `file_put_contents()`: Tạo ra tập tin nếu tập tin đó chưa tồn tại → Mở tập tin → Ghi nội dung → Đóng tập tin
- Nếu `$mod = FILE_APPEND` nội dung cũ ở tập tin được giữ, nội dung mới được ghi vào cuối tập tin
- Nếu ghi thành công, trả về tổng số ký tự đã ghi ngược lại trả về `FALSE`

Vấn đề 06: Đổi tên tập tin, thư mục

- Sử dụng hàm `rename ($oldName, $newName)` để đổi tên tập tin, thư mục từ giá trị `$oldName` thành giá trị `$newName` (lưu ý thay đổi cả phần đường dẫn)
- Hàm trả về kết quả `TRUE` nếu rename thành công, trả về kết quả `FALSE` nếu rename không thành công

Vấn đề 07: Sao chép tập tin

- Sử dụng hàm `copy ($sourceFile, $destinationFile)` để copy nội dung từ tập tin `$sourceFile` sang tập tin `$destinationFile`
- Hàm trả về kết quả `TRUE` nếu copy thành công, trả về kết quả `FALSE` nếu copy không thành công
- Làm sao copy một thư mục nào đó ?

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P1)

Xét mối quan hệ giữa người dùng và một tập tin, thư mục. Chúng ta có 3 nhóm sau đây:

- Owner/User: Người sở hữu, chủ tài khoản.
- Group: Các tài khoản cùng hoạt động trong một nhóm.
- Other/Guest: Ngoài 2 đối tượng trên.

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P2)

Xét về quyền thao tác giữa người dùng và **một tập tin**. Chúng ta có 3 loại quyền sau:

- Read: có thể đọc nội dung trong tập tin
- Write: có thể thay đổi nội dung của tập tin
- Execute: có thể mở tập tin (khởi chạy chương trình nếu là một tập tin ứng dụng)

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P3)

Xét về quyền thao tác giữa người dùng và **một thư mục**. Chúng ta có 3 loại quyền sau:

- **Read**: Có thể duyệt nội dung trong thư mục (Xem trong thư mục có gì, phải đi kèm với quyền Execute).
- **Write**: Có thể thay đổi nội dung trong thư mục này (Thêm, xóa, sửa, đổi tên thư mục/tập tin bên trong thư mục này).
- **Execute**: Có thể mở thư mục (Mở thư mục ra, nhưng không thể thấy gì trong thư mục này nếu không đi kèm với quyền Read).

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P4)

Cấp quyền truy cập cho tập tin, thư mục:

- Quyền cho một tập tin, thư mục thường được biểu diễn tương ứng với một số nguyên có 3 chữ số xyz
 - Số thứ nhất x - đại diện cho quyền hạn của Owner/User.
 - Số thứ hai y - đại diện cho quyền hạn của Group.
 - Số thứ ba z - đại diện cho quyền hạn của Other/Guest.

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P5)

Giá trị của các chữ số xyz, quy định như sau:

- Read = 4
- Write = 2
- Execute = 1

→ $\text{Read} + \text{Write} + \text{Execute} = 4 + 2 + 1 = 7$ (Toàn quyền)

→ $\text{Read} + \text{Execute} = 4 + 1 = 5$ (Chỉ có thể đọc và thực thi tập tin)

Vấn đề 08: Khái quát về vấn đề phân quyền tập tin, thư mục (P6)

Một tập tin được cấu hình với quyền 754, như vậy chúng ta có các thông tin sau:

- Chữ số đầu tiên $7 = 4 + 2 + 1 \rightarrow$ quyền: toàn quyền \rightarrow Owner/User có toàn quyền
- Chữ số thứ hai $5 = 4 + 1 \rightarrow$ quyền: đọc và thực thi \rightarrow Group có quyền đọc và thực thi
- Chữ số thứ ba $4 \rightarrow$ quyền: đọc \rightarrow Other/Guest chỉ có quyền đọc tập tin.

Vấn đề 09: Tạo – Xóa – Cấp quyền cho thư mục

- Sử dụng hàm `mkdir ($path, $mode)` để tạo ra thư mục ở đường dẫn `$path`, với quyền truy cập vào thư mục là `$mode`
- Sử dụng hàm `rmdir ($path)` để xóa thư mục ở đường dẫn `$path`
- Sử dụng hàm `fileperms ($dirName)` để xem quyền truy cập đối với thư mục `$dirName`
- Sử dụng hàm `chmod ($dirName, $mod)` để cấp quyền truy cập cho thư mục `$dirName`

Vấn đề 10: Tạo – Xóa – Cấp quyền cho tập tin

- Sử dụng hàm `file_put_contents ($path, null)` để tạo ra tập tin ở đường dẫn `$path`
- Sử dụng hàm `unlink ($path)` để xóa tập tin ở đường dẫn `$path`
- Sử dụng hàm `fileperms ($fileName)` để xem quyền truy cập đối với tập tin `$fileName`
- Sử dụng hàm `chmod ($fileName, $mod)` để cấp quyền truy cập cho tập tin `$fileName`

Vấn đề 11: Liệt kê danh sách tập tin và thư mục

- Sử dụng hàm **glob** (\$pattern) để lấy danh sách các tập tin và thư mục với tên thỏa \$pattern
- Sử dụng tham số thứ hai **GLOB_ONLYDIR** để kết quả trả về chỉ bao gồm các thư mục.

Vấn đề 12: Tìm hiểu hàm parse_ini_file (1)

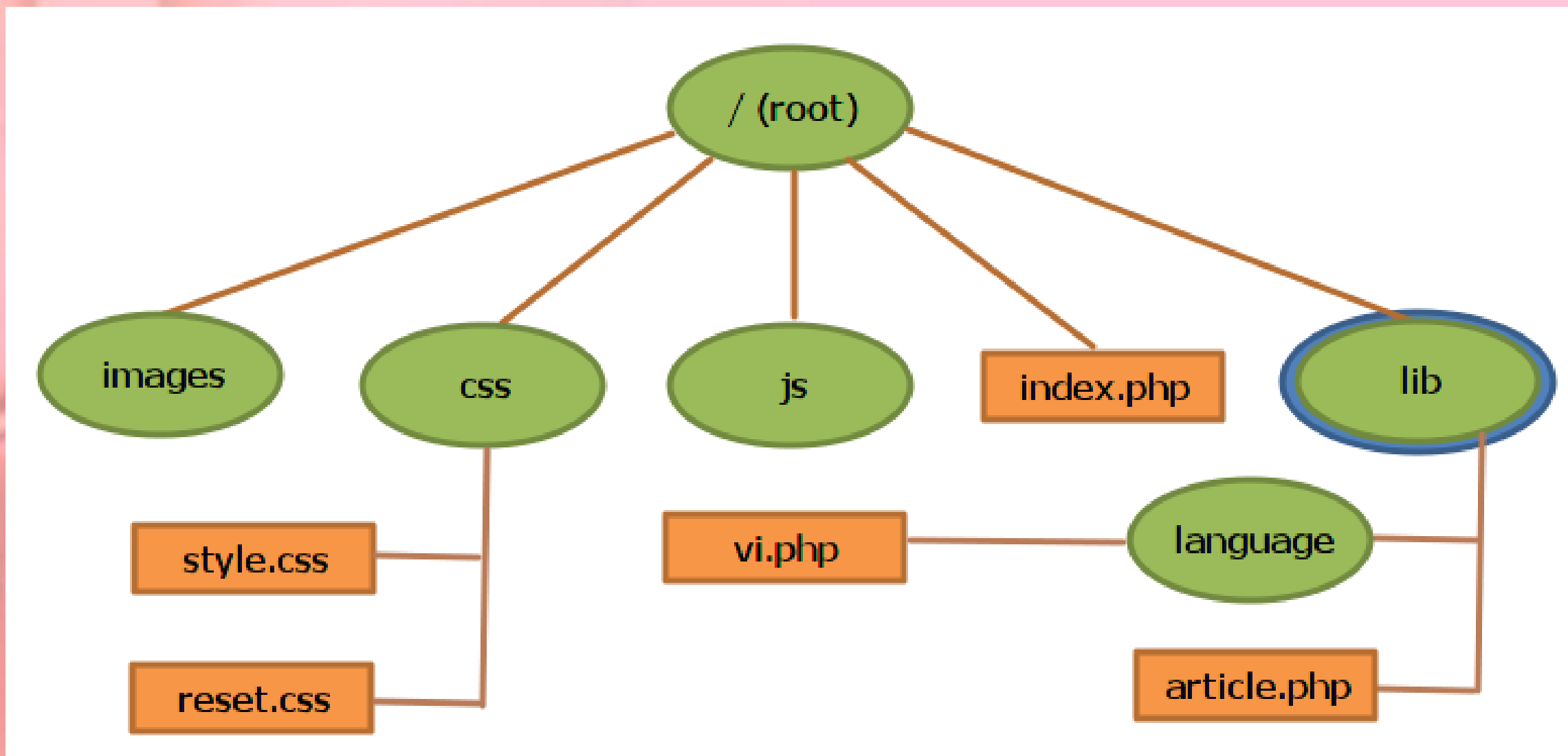
- Cấu trúc tập tin *.ini

```
1
2 ; comment
3
4 [info]
5 name=Nguyễn Văn A
6 age=23
7
8 [study]
9 school=Nguyễn Đình Chiểu
10 programming=PHP
11
```

Vấn đề 12: Tìm hiểu hàm `parse_ini_file` (2)

- Sử dụng hàm `parse_ini_file ($fileName)` để đọc nội dung `$fileName` thành một mảng.
- Lưu ý nếu muốn đọc luôn các phần section cần truyền giá trị `TRUE` vào tham số thứ hai

Vấn đề 13: Một số định nghĩa về hệ thống cây thư mục (1)



Vấn đề 13: Một số định nghĩa về hệ thống cây thư mục (2)

- **Root-directory**: thư mục cao nhất trong cây thư mục và trên nó không còn có một thư mục nào khác. Root-directory được đánh dấu với ký hiệu /
- **Working-directory** (current directory): thư mục mà người sử dụng đang làm việc
- **Parent-directory**: thư mục cha - thư mục nằm ngay phía trên một thư mục khác trong cây thư mục
- **Path** (đường dẫn): mỗi tệp tin hay thư mục trong hệ thống linux được chỉ định bởi một đường dẫn (để truy cập vào tệp tin hay thư mục đó)

Vấn đề 13: Phân biệt đường dẫn tuyệt đối – Đường dẫn tương đối

- Đường dẫn tuyệt đối:
 - Là những đường dẫn được tính từ root (/), đi qua các thư mục khác cho đến tập tin hoặc thư mục đích
 - Khi chúng ta đang ở `article.php`, đường dẫn tuyệt đối đến các tập tin
 - `style.css` sẽ là `/style/style.css`
 - `vi.php` sẽ là `/lib/language/vi.phpstyle.css`
 - Thường được dùng với các hàm `include`, `require` để nhúng các tập tin `.php` vào một tập tin php nào đó hoặc dùng để khai báo thư mục chứa hình ảnh trên host

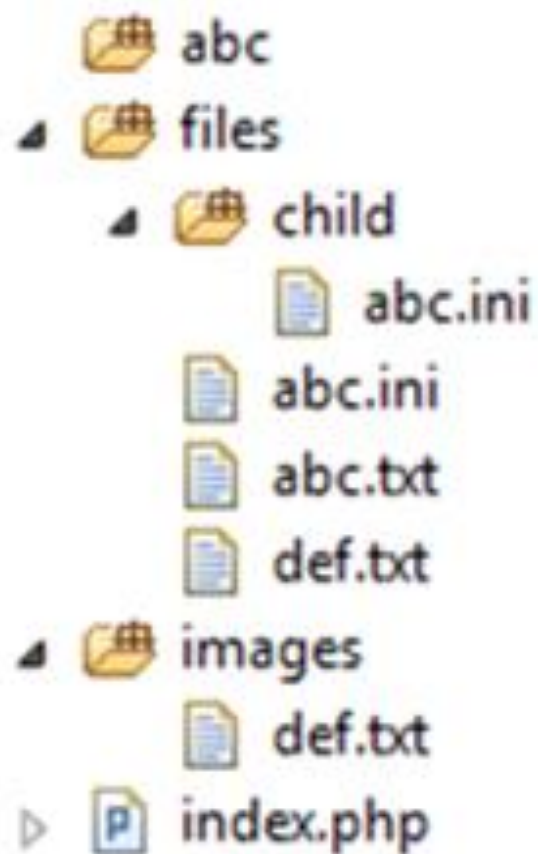
Vấn đề 13: Phân biệt đường dẫn tuyệt đối – Đường dẫn tương đối

- Đường dẫn tương đối:
 - Là những đường dẫn có điểm xuất phát là thư mục hiện tại (current directory).
 - Khi chúng ta đang ở `article.php`, đường dẫn tương đối đến các tập tin
 - `style.css` sẽ là `../css/style.css`
 - `vi.php` sẽ là `/lib/language/vi.php`
 - Thường được sử dụng để load các tập tin javascript, css và hình ảnh lên và hiển thị trong trang web

Vấn đề 14: Thao tác với thư mục

- `getcwd()` trả về thư mục hiện tại
- `realpath()` trả về đường dẫn tuyệt đối của đường dẫn `$path`
- `chdir()` thay đổi thư mục hiện tại
- `dir()` mở một thư mục lên và thao tác với thư mục đó (3 thao tác `read`, `rewind`, `close`)
- `opendir()` mở một thư mục lên và thao tác với thư mục đó
- `closedir()` đóng thư mục được mở bởi hàm `opendir()`
- `scandir()` lấy danh sách các tập tin và thư mục của đường dẫn `$path`

Vấn đề 15: Liệt kê cấu trúc tập tin thư mục đến cấp 2



- D: abc
- D: files
 - F: abc.ini
 - F: abc.txt
 - D: child
 - F: def.txt
- D: images
 - F: def.txt
- F: index.php