

Bài Tập Thực Hành: Quản Lý File và Thư Mục

Bài Tập 1: Tạo Dự Án Web Cơ Bản

Mục tiêu:

- Tạo cấu trúc thư mục cho dự án web
- Thực hành tạo file và thư mục
- Hiểu về tổ chức file trong dự án thực tế

Các bước thực hiện:

1. Tạo thư mục gốc cho dự án:

```
mkdir ~/mywebsite  
cd ~/mywebsite
```

Giải thích:

- Dùng `~` để tham chiếu đến home directory
- Tạo thư mục riêng để dễ quản lý
- `cd` để di chuyển vào thư mục vừa tạo

2. Tạo cấu trúc thư mục con:

```
mkdir -p {css,js,images,pages}
```

Giải thích:

- `-p` cho phép tạo nhiều thư mục cùng lúc
- Dùng `{ }` để liệt kê các thư mục cần tạo
- Tổ chức theo chức năng của files

3. Tạo các file CSS:

```
touch css/{style.css,reset.css,responsive.css}
```

Giải thích:

- Tạo nhiều file CSS cùng lúc
- `style.css` cho style chính
- `reset.css` để reset CSS mặc định
- `responsive.css` cho thiết kế responsive

4. Tạo file HTML:

```
vi index.html
```

Thêm nội dung cơ bản:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Website</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <h1>Welcome to My Website</h1>
</body>
</html>
```

5. Tạo file JavaScript:

```
touch js/{main.js,utils.js}
```

Giải thích:

- Phân tách code JavaScript theo chức năng
- `main.js` cho code chính
- `utils.js` cho các hàm tiện ích

Bài Tập 2: Tạo Cấu Trúc Dự Án Blog

Mục tiêu:

- Tạo hệ thống quản lý bài viết
- Phân loại nội dung
- Quản lý media

Các bước thực hiện:

1. Tạo thư mục gốc và di chuyển vào:

```
mkdir ~/myblog
cd ~/myblog
```

2. Tạo cấu trúc thư mục cho bài viết:

```
mkdir -p content/{posts,drafts,pages}
mkdir -p content/posts/{2023,2024}/{01,02,03}
```

Giải thích:

- Tổ chức bài viết theo năm và tháng
- Tách riêng draft và bài đã xuất bản
- Dễ dàng quản lý và tìm kiếm

3. Tạo cấu trúc media:

```
mkdir -p media/{images,videos,documents}
mkdir -p media/images/{original,thumbnails}
```

Giải thích:

- Phân loại media theo loại
- Tách ảnh gốc và thumbnail
- Dễ quản lý dung lượng

4. Tạo file cấu hình:

```
touch config.yaml
vi config.yaml
```

Thêm nội dung:

```
site_name: My Blog
author: Your Name
theme: default
posts_per_page: 10
```

Bài Tập 3: Tạo Môi Trường Phát Triển

Mục tiêu:

- Tạo workspace cho developer
- Tổ chức code và tài liệu
- Quản lý nhiều dự án

Các bước thực hiện:

1. Tạo workspace:

```
mkdir -p ~/workspace  
cd ~/workspace
```

2. Tạo cấu trúc cho nhiều dự án:

```
mkdir -p {projects,documents,tools,backup}
```

Giải thích:

- **projects**: chứa các dự án
- **documents**: tài liệu kỹ thuật
- **tools**: công cụ phát triển
- **backup**: sao lưu dữ liệu

3. Tạo cấu trúc cho dự án mẫu:

```
mkdir -p projects/project1/{src,tests,docs,build}  
touch projects/project1/README.md
```

Giải thích:

- **src**: mã nguồn
- **tests**: file kiểm thử
- **docs**: tài liệu
- **build**: file biên dịch

4. Tạo file gitignore:

```
vi projects/project1/.gitignore
```

Thêm nội dung:

```
build/  
*.log  
.env  
node_modules/
```

Giải thích:

- Loại trừ file build
- Không theo dõi file log
- Bảo mật file môi trường

- Bỏ qua thư mục node_modules

Bài Tập 4: Quản Lý File Cấu Hình

Mục tiêu:

- Tạo và quản lý file cấu hình
- Hiểu về backup và version
- Thực hành với file dot (.)

Các bước thực hiện:

1. Tạo thư mục cho file cấu hình:

```
mkdir -p ~/.config/myapp  
cd ~/.config/myapp
```

Giải thích:

- Dùng `.config` theo chuẩn XDG
- Tách riêng cấu hình cho từng ứng dụng

2. Tạo file cấu hình:

```
touch {config.ini,settings.json}
```

3. Tạo backup tự động:

```
# Tạo script backup  
vi backup.sh
```

Thêm nội dung:

```
#!/bin/bash  
backup_dir="$HOME/.config/myapp/backup"  
mkdir -p "$backup_dir"  
cp config.ini "$backup_dir/config.ini.$(date +%Y%m%d)"  
cp settings.json "$backup_dir/settings.json.$(date +%Y%m%d)"
```

Giải thích:

- Tự động tạo backup hàng ngày
- Thêm timestamp vào tên file
- Dễ dàng khôi phục khi cần

Lời Khuyên Khi Thực Hành:

1. Lập kế hoạch trước:

- Vẽ sơ đồ cấu trúc thư mục
- Xác định rõ mục đích mỗi thư mục
- Dự tính khả năng mở rộng

2. Đặt tên file/thư mục:

- Dùng tên có ý nghĩa
- Tránh ký tự đặc biệt
- Nhất quán trong quy ước đặt tên

3. Kiểm tra thường xuyên:

- Dùng `ls -la` để xem chi tiết
- Kiểm tra quyền truy cập
- Xác nhận cấu trúc đúng

4. Lưu ý về backup:

- Backup trước khi thay đổi lớn
- Đặt tên backup có ý nghĩa
- Dọn dẹp backup cũ định kỳ