Web Development - Day 1

1. Course Overview

Welcome to Web Development! This course will introduce you to the fundamental concepts and tools used in modern web development.

Course Objectives:

- Understand the basics of web development
- Learn HTML fundamentals
- Master basic Git and GitHub workflows
- Create and deploy simple web pages

2. Setting Up Your Development Environment

Required Tools:

- Text Editor (VS Code recommended)
- Web Browser (Chrome or Firefox recommended)
- Git
- GitHub Account

VS Code Extensions:

- Live Server
- HTML CSS Support
- Auto Rename Tag

- Prettier - Code formatter

> **Good Practice:**
>
> - Use a dedicated code editor with syntax highlighting
> - Install relevant extensions for productivity
> - Keep your development tools updated

> **Bad Practice:**
>
> - Using basic text editors like Notepad
> - Ignoring tool updates and security patches
> - Working without version control

3. HTML Basics

Basic HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First Webpage</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

Essential HTML Tags:

- <h1> to <h6> - Headings

- `<p>` - Paragraphs
- `<a>` - Links
- `<img>` - Images
- `<ul>`, `<ol>`, `<li>` - Lists

> **Good HTML Practices:**
>
> - Use semantic HTML elements
> - Include proper meta tags
> - Set proper document language
> - Use descriptive alt text for images
> - Properly indent your code

> **Bad HTML Practices:**
>
> - Using tables for layout
> - Using deprecated tags
> - Missing closing tags
> - Using inline styles extensively
> - Skipping alt attributes for images

4. GitHub Basics

Getting Started:

1. Create a GitHub account at github.com
2. Install Git on your local machine
3. Configure Git with your credentials
4. Create your first repository

Essential Git Commands:

```
### Initial Git Setup:

# Configure global username and email
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"

# Configure default branch name
git config --global init.defaultBranch main

# Configure default editor
git config --global core.editor "code --wait"

# View all configurations
git config --list


### Authentication Setup:

# Generate SSH key
ssh-keygen -t ed25519 -C "your.email@example.com"

# Start SSH agent
eval "$(ssh-agent -s)"

# Add SSH key to agent
ssh-add ~/.ssh/id_ed25519

# Copy public key to clipboard (Mac)
pbcopy < ~/.ssh/id_ed25519.pub
# For Windows, use:
clip < ~/.ssh/id_ed25519.pub


### Repository Operations:

# Initialize new repository
git init

# Clone existing repository
git clone

# Add remote repository
git remote add origin

# Change remote URL
git remote set-url origin
```

```
# View remote repositories
git remote -v


### Basic Git Workflow:

# Check status
git status

# Stage changes
git add       # Stage specific file
git add .           # Stage all changes
git add -p         # Stage changes interactively

# Commit changes
git commit -m "Descriptive message"
git commit -am "Message"  # Stage and commit tracked files

# Push changes
git push origin

# Pull changes
git pull origin


### Branch Management:

# Create and switch to new branch
git checkout -b feature-branch

# Switch branches
git checkout main

# List branches
git branch          # Local branches
git branch -r       # Remote branches
git branch -a       # All branches

# Delete branch
git branch -d feature-branch    # Safe delete
git branch -D feature-branch    # Force delete
```

> **Good Git Practices:**

- Write clear commit messages
- Commit frequently with logical changes
- Use meaningful branch names
- Keep repositories organized

Bad Git Practices:

- Committing directly to main branch
- Writing vague commit messages
- Pushing broken code
- Ignoring .gitignore file

5. Today's Assignment

Task: Create a Personal Introduction Page

1. Create a new GitHub repository named "web-dev-intro"
2. Clone the repository to your local machine
3. Create index.html with your personal introduction
4. Include:
   - A heading with your name
   - A paragraph about yourself
   - A list of your interests
   - A link to your GitHub profile
   - An image (optional)
5. Commit and push your changes to GitHub

Submission Instructions:

Submit your GitHub repository URL by the end of the day. Ensure your repository is public and the page is properly formatted.

6. Additional Resources

- MDN Web Docs: HTML Learning Area
- GitHub Guides: guides.github.com
- W3Schools HTML Tutorial: w3schools.com/html