Multipage Websites Learning Guide

This guide covers all aspects of building multipage websites, including file paths, project structure, HTML boilerplate, portfolio projects, and hosting. Each section combines theory and practical code examples for easy teaching.

# 1. Computer File Paths

# Computer File Paths - Summary  
  
## Introduction  
File paths are essential for web development.  
They specify unique file or folder locations.  
Like giving directions to a specific place.  
  
## Types of File Paths  
  
### Absolute File Paths  
Start from the root of the computer.  
Specify full path to file or folder.  
Examples:  
- Windows: C:\Project\Images\cat.png  
- Mac: /Users/username/Project/Images/cat.png  
Always point to same location.  
Can be long and inflexible if moved.  
  
### Relative File Paths  
Specified relative to current file or folder.  
Shorter and more flexible.  
Remain valid if project is moved.  
Most used in web development.  
  
## Special Characters  
../ goes up one level to parent directory.  
Example: ../filename accesses parent directory file.  
./ refers to current directory.  
Can be omitted but ensures compatibility.  
Example: ./filename or just filename.  
  
## Practical Exercise  
1. Rabbit Image: ./rabbit.png  
2. Cat Image: ./Folder3/cat.png  
3. Dog Image: ../../dog.png  
4. Fish Image: ../Folder1/fish.png  
5. Bird Image: ../Folder1/Folder2/bird.png  
  
## Key Takeaways  
- File paths identify file/folder locations.  
- Absolute paths start from root directory.  
- Relative paths are preferred in web development.  
- ./ and ../ help navigate file structures.  
- Understanding paths is key for linking resources.  
- Use code editor autocomplete to avoid errors.  
  
## Next Steps  
Apply this knowledge to link HTML files and build websites.

Example: Folder0/index.html (before):

<h1>All the Animals</h1>  
<h2>Rabbit:</h2>  
  
<h2>Cat:</h2>  
  
<h2>Dog:</h2>  
  
<h2>Fish:</h2>  
  
<h2>Bird:</h2>

Solution: Folder0/solution.html (with correct file paths):

<h1>All the Animals</h1>  
  
<h2>Rabbit:</h2>  
<img src="./rabbit.png" alt="rabbit" />  
  
<h2>Cat:</h2>  
<img src="./Folder3/cat.png" alt="cat" />  
  
<h2>Dog:</h2>  
<img src="../dog.png" alt="dog" />  
  
<h2>Fish:</h2>  
<img src="../Folder1/fish.png" alt="fish" />  
  
<h2>Bird:</h2>  
<img src="../Folder1/Folder2/bird.png" alt="bird" />

Note: Images referenced in the code (e.g., cat.png, dog.png) would appear here if this were a web page.

# 2. Webpages and Multi-Page Websites

# Webpages and Multi-Page Websites  
  
## What are Webpages?  
Webpages are individual HTML files in a website.  
Multi-page websites have multiple HTML files.  
Users navigate between different pages.  
  
## Key Concepts  
  
### Website Structure  
- Project folder: contains all website files  
- index.html: main/home page  
- Additional HTML: about, contact, etc.  
- Assets folder: images, CSS, resources  
- Public folder: other HTML files  
  
### File Paths  
- Use ./ for current directory  
- Anchor tags (<a>) use href to link pages  
- Image tags (<img>) use src for images  
  
### Example Structure  
Project Folder/  
├── index.html  
├── assets/  
│ └── images/  
│ └── cat.png  
└── public/  
 ├── about.html  
 └── contact.html  
  
## Quiz Example  
How to display cat.png on index.html?  
<img src="./assets/images/cat.png" alt="cat">  
  
## Exercise  
1. Add image linking to About page  
2. Add link to Contact Me page  
  
## Solution Example  
<a href="./public/about.html">  
 <img src="./assets/images/cat.png" alt="cat" />  
</a>  
<a href="./public/contact.html">Contact Me</a>  
  
## Key Points  
- Multi-page sites need good file organization  
- Use correct relative file paths  
- Images can be clickable links  
- <a> uses href, <img> uses src  
- ./ means current directory  
  
## File Structure Used  
- index.html: Home page  
- public/about.html: About page  
- public/contact.html: Contact page  
- assets/images/cat.png: Sample image

index.html:

<h1>Welcome to My Website!</h1>  
<!-- Add an image of yourself that links to the about page -->  
  
<hr />  
<!-- Add a link to your contact me page here -->

Solution (solution.html):

<h1>Welcome to My Website!</h1>  
<!-- Add an image of yourself that links to the about page -->  
<a href="./public/about.html"><img src="./assets/images/cat.png" alt="cat" /></a>  
<hr />  
  
<!-- Add a link to your contact me page here -->  
<a href="./public/contact.html">Contact Me</a>

About page (about.html):

<h1>About Me</h1>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sem tellus, sagittis non odio nec, interdum  
 elementum lacus. Ut ac justo eget risus sollicitudin fringilla sed ut leo. Ut condimentum elit nec fermentum lobortis.  
 Quisque tincidunt quam nec tincidunt ullamcorper. Integer vitae pretium justo. In placerat volutpat pellentesque.  
 Etiam  
 gravida quam vitae odio pulvinar imperdiet. Vivamus venenatis gravida purus sit amet aliquet. Ut in nibh sed quam  
 laoreet lacinia. Cras cursus ut dui vitae posuere. Vivamus volutpat urna vitae odio fringilla placerat. Nulla vel  
 rhoncus sapien. Phasellus at interdum enim.</p>  
  
<p>Curabitur elementum sagittis mollis. In porta nec quam ut semper. Nulla volutpat neque quis lacus mattis eleifend.  
 Phasellus et congue odio. Mauris tristique, nisl quis porttitor porttitor, tortor ante mattis ex, at dictum tortor  
 dolor  
 ut sem. Nullam nec libero non ex porta vulputate a eu sapien. In nec bibendum mauris. Fusce iaculis lectus vel magna  
 laoreet gravida.</p>  
  
<p>Donec euismod vestibulum arcu, a blandit metus mattis ultrices. Integer quis hendrerit justo. Maecenas sed tempor mi.  
 Fusce egestas urna leo. Mauris viverra sem sed libero egestas, volutpat aliquet magna porta. Quisque id diam sed ipsum  
 interdum sollicitudin id quis augue. Donec congue nisl nec massa ornare imperdiet quis quis odio. Etiam ut volutpat  
 nisl. Vestibulum hendrerit justo nibh, eget consequat magna fermentum finibus. Pellentesque et urna fringilla, rutrum  
 neque in, varius leo. Vestibulum efficitur id massa eget pellentesque.</p>

Contact page (contact.html):

<h1>Contact Me</h1>  
<p>Tel: +123456789</p>  
<p>Email: me@gmail.com</p>  
<p>Address:</p>  
<p>  
 123 North Street<br />  
 Some City<br />  
 Some Country <br />  
</p>

# 3. HTML Boilerplate

# HTML Boilerplate Summary  
  
## What is HTML Boilerplate?  
Standard structure every HTML file should follow.  
Foundation or template for HTML documents.  
Like a letter's structure: address, greeting, body, closing.  
  
## Boilerplate Structure  
```html  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>My Website</title>  
</head>  
<body>  
 <!-- All visible content goes here -->  
</body>  
</html>  
```  
  
## Key Components  
1. DOCTYPE: <!DOCTYPE html> (first line, HTML5)  
2. HTML Root: <html lang="en"> (contains everything)  
3. Head: metadata, not visible, includes charset, title, viewport  
4. Body: all visible content (headings, images, links, text)  
  
## Nesting Structure  
HTML elements are nested like a hamburger.  
Proper indentation shows nesting clearly.  
  
## VS Code Shortcuts  
Type "!" and press Enter for boilerplate.  
Works with .html files, saves time.  
  
## Optional Meta Tags  
<meta http-equiv> for IE (can delete, IE deprecated)  
<meta name="viewport"> for mobile responsiveness.  
  
## Best Practices  
- Always start with boilerplate structure  
- Use proper indentation for readability  
- Include charset meta tag  
- Set language attribute  
- Remove deprecated tags  
  
## Summary  
HTML Boilerplate is the foundation for well-structured web pages.

# 4. Portfolio Website Project

# Portfolio Website Project Summary  
  
## Project Overview  
Create a Web Developer Portfolio using only HTML.  
Showcase projects and provide professional online presence.  
Employers or clients can view work and contact info.  
  
## Key Components  
- Main page (index.html) with custom title  
- H1, H2 headings for structure  
- Horizontal rules for separation  
- Project section: Movie Ranking, Birthday Invite  
- Each project: screenshot, link, H3 title  
- Footer links: About Me, Contact Me  
  
## Project Structure  
Portfolio Website/  
├── index.html  
├── public/  
│ ├── about.html  
│ ├── contact.html  
│ ├── movie-ranking.html  
│ └── birthday-invite.html  
└── assets/  
 └── images/  
 ├── movie-ranking.png  
 └── birthday-invite.png  
  
## Implementation Steps  
1. Create HTML boilerplate  
2. Customize title  
3. Add project code  
4. Take screenshots  
5. Rename images  
6. Add headings  
7. Create links  
  
## Technical Requirements  
- Screenshots in assets/images, height="200"  
- Replace placeholder code with personal versions  
- Customize About and Contact pages  
- Add unique touches  
  
## File Organization  
- Main portfolio: index.html  
- Project pages: public/  
- Images: assets/images/  
- Keep structure clean and organized  
  
## Key Features  
- Internal links between pages  
- Project previews with screenshots  
- Custom project content  
- Personal info in About Me  
- Contact details in Contact Me  
  
## Learning Outcomes  
- HTML structure and organization  
- File path management  
- Image integration and sizing  
- Anchor tag use  
- Multi-page website creation  
- Professional portfolio development  
  
## Next Steps  
- Deploy portfolio online  
- Share with employers/clients  
- Add new projects  
- Enhance with CSS (future)  
  
## Best Practices  
- Use semantic HTML  
- Organize files logically  
- Descriptive image names  
- Test all links  
- Keep content current  
  
## Summary  
Portfolio is foundation for showcasing web skills and growth.

index.html (starter with TODOs):

<!-- TODO 1: Create the HTML Boilerplate -->  
  
<!-- TODO 2: Add Your previous projects' HTML into the public folder -->  
  
<!-- TODO 3: Take screenshots of your project previews and add the images to the images folder -->  
  
<!-- TODO 4: Add titles/subtitles etc. -->  
  
<!-- TODO 5: Add a link to the project pages -->  
  
<!-- TODO 6: Add images to show the project previews  
HINT for TODO 6: You can use the height attribute set to 200 to make the image smaller:  
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img#attr-height -->  
  
<!-- TODO 7: Add the Contact Me and About Me page links -->

Solution (solution.html):

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <title>Angela's Portfolio</title>  
</head>  
  
<body>  
 <h1>Angela Yu's Portfolio</h1>  
 <h2>I'm a Web Developer</h2>  
 <hr />  
 <h3><a href="./public/movie-ranking.html">Movie Ranking Project</a></h3>  
 <img src="./assets/images/movie-ranking.png" height="200" alt="movie ranking project preview"/>  
 <h3><a href="./public/birthday-invite.html">Birthday Invite Project</a></h3>  
 <img src="./assets/images/birthday-invite.png" height="200" alt="birthday invite project preview"/>  
 <hr />  
 <a href="./public/about.html">About Me</a>  
 <a href="./public/contact.html">Contact Me</a>  
</body>  
  
</html>

About page (about.html):

<h1>About Me</h1>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sem tellus, sagittis non odio nec, interdum  
 elementum lacus. Ut ac justo eget risus sollicitudin fringilla sed ut leo. Ut condimentum elit nec fermentum lobortis.  
 Quisque tincidunt quam nec tincidunt ullamcorper. Integer vitae pretium justo. In placerat volutpat pellentesque.  
 Etiam  
 gravida quam vitae odio pulvinar imperdiet. Vivamus venenatis gravida purus sit amet aliquet. Ut in nibh sed quam  
 laoreet lacinia. Cras cursus ut dui vitae posuere. Vivamus volutpat urna vitae odio fringilla placerat. Nulla vel  
 rhoncus sapien. Phasellus at interdum enim.</p>  
  
<p>Curabitur elementum sagittis mollis. In porta nec quam ut semper. Nulla volutpat neque quis lacus mattis eleifend.  
 Phasellus et congue odio. Mauris tristique, nisl quis porttitor porttitor, tortor ante mattis ex, at dictum tortor  
 dolor  
 ut sem. Nullam nec libero non ex porta vulputate a eu sapien. In nec bibendum mauris. Fusce iaculis lectus vel magna  
 laoreet gravida.</p>  
  
<p>Donec euismod vestibulum arcu, a blandit metus mattis ultrices. Integer quis hendrerit justo. Maecenas sed tempor mi.  
 Fusce egestas urna leo. Mauris viverra sem sed libero egestas, volutpat aliquet magna porta. Quisque id diam sed ipsum  
 interdum sollicitudin id quis augue. Donec congue nisl nec massa ornare imperdiet quis quis odio. Etiam ut volutpat  
 nisl. Vestibulum hendrerit justo nibh, eget consequat magna fermentum finibus. Pellentesque et urna fringilla, rutrum  
 neque in, varius leo. Vestibulum efficitur id massa eget pellentesque.</p>

Contact page (contact.html):

<h1>Contact Me</h1>  
<p>Tel: +123456789</p>  
<p>Email: me@gmail.com</p>  
<p>Address:</p>  
<p>  
 123 North Street<br />  
 Some City<br />  
 Some Country <br />  
</p>

Movie Ranking Project (movie-ranking.html):

<h1>The Best Movies According to Angela</h1>  
<h2>My top 3 movies of all-time.</h2>  
<hr />  
<h3>Spirited Away</h3>  
<p>This is my favourite anime. I love the beautiful images.</p>  
<h3>Ex Machina</h3>  
<p>Really cool sci-fi movie.</p>  
<h3>Drive</h3>  
<p>Super beautiful film. Really artistic.</p>

Birthday Invite Project (birthday-invite.html):

<!-- This is one possible solution -->  
<h1>It's My Birthday!</h1>  
<h2>On the 12th May</h2>  
  
<img src="https://raw.githubusercontent.com/appbrewery/webdev/main/birthday-cake3.4.jpeg"  
 alt="purple birthday cake with candles" />  
  
<h3>What to bring:</h3>  
<ul>  
 <li>Baloons (I love baloons)</li>  
 <li>Cake (I'm really good at eating)</li>  
 <li>An appetite (There will be lots of food)</li>  
</ul>  
  
<h3>This is where you need to go:</h3>  
<a  
 href="https://www.google.com/maps/@35.7040744,139.5577317,3a,75y,289.6h,87.01t,0.72r/data=!3m6!1e1!3m4!1sgT28ssf0BB2LxZ63JNcL1w!2e0!7i13312!8i6656">Google  
 map link</a>

Note: Images referenced in the code (e.g., movie-ranking.png, birthday-invite.png) would appear here if this were a web page.

# 5. Hosting on Github

# How to Host Your Website for Free with GitHub  
  
## Introduction  
This lesson teaches how to make your HTML portfolio website live on the Internet using GitHub's free hosting service, allowing you to share your work with anyone worldwide.  
  
## What is Web Hosting?  
- \*\*Definition\*\*: The process of making websites available on the Internet  
- \*\*Purpose\*\*: Enables global access to your website from anywhere in the world  
- \*\*Process\*\*: Uploading website files (HTML, images, etc.) to a web server connected to the Internet  
- \*\*Availability\*\*: Web servers run 24/7, making your site accessible anytime  
  
## Local Development vs Web Hosting  
  
### Local Development  
- Files stored on your computer only  
- Website accessible only on your machine  
- Others need file copies to view your site  
- Not ideal for sharing with others  
  
### Web Hosting  
- Files uploaded to Internet-connected web server  
- Website accessible to anyone worldwide  
- Professional way to share your work  
- Enables real-world website deployment  
  
## Hosting Your HTML Portfolio on GitHub  
  
### Step 1: Create GitHub Repository  
1. Go to github.com  
2. Sign up/sign in to your account  
3. Click "+" icon → "New repository"  
4. Name: `html-portfolio`  
5. Add optional description  
6. Set repository to \*\*Public\*\* (required for hosting)  
7. Check "Add a README file"  
8. Click "Create Repository"  
  
### Step 2: Upload Website Files  
1. Click "Add file" → "Upload files"  
2. Navigate to your HTML Portfolio Project folder  
3. \*\*Important\*\*: Upload folder contents, not the folder itself  
4. Ensure `index.html` is named exactly (lowercase "i")  
5. Include all images and subfolders (public/, assets/, etc.)  
6. Click "Commit changes"  
  
### Step 3: Enable GitHub Pages  
1. Go to repository \*\*Settings\*\* tab  
2. Navigate to \*\*Pages\*\* section  
3. Under "Branch" dropdown, select \*\*main\*\* branch  
4. Click "Save"  
5. Wait 1-10 minutes for processing  
6. Refresh until you see "Your site is live" popup  
  
### Step 4: Access Your Live Website  
1. Click "Visit site" to view your website online  
2. Verify all pages work correctly  
3. Share the URL with others  
4. Post in course Q&A for feedback  
  
## Key Requirements  
- Repository must be \*\*public\*\*  
- Homepage must be named `index.html` (exact spelling)  
- All files and folders must be uploaded  
- GitHub Pages may take several minutes to activate  
  
## Benefits of GitHub Hosting  
- \*\*Free\*\*: No cost for hosting  
- \*\*Global\*\*: Accessible worldwide  
- \*\*Professional\*\*: Real web address  
- \*\*Portfolio\*\*: Showcase your work  
- \*\*Sharing\*\*: Easy to share with others  
  
## Next Steps  
This foundation prepares you for learning CSS to transform your basic HTML site into a beautiful, modern website.  
  
## Key Takeaways  
- Web hosting makes websites globally accessible via web servers  
- Local development limits access to your computer only  
- GitHub provides free hosting through public repositories  
- Proper file naming (`index.html`) is crucial for GitHub Pages  
- The hosting process enables professional portfolio sharing

# Conclusion

You have now learned how to build, structure, and host multipage websites. Practice by building your own projects and sharing them online!