

# Cómo incluir *assets* en un documento HTML sin morir en el intento

En un contexto web, llamamos "assets" (o recursos) a los archivos que se utilizan para construir y dar estilo a una página web, así como para proporcionar funcionalidad interactiva. Estos archivos pueden incluir imágenes, hojas de estilo CSS, archivos JavaScript, fuentes tipográficas, archivos de audio o video, y cualquier otro elemento multimedia o recurso necesario para el funcionamiento y la apariencia de la página web.

Antes de nada, tenemos que entender qué diferencia existe entre **rutas relativas** y **rutas absolutas**.

## 1. Rutas relativas y absolutas

### • Rutas Relativas:

- Las rutas relativas especifican la ubicación de un archivo en relación con la ubicación actual del archivo de referencia.
  - Se utilizan rutas relativas cuando quieres hacer referencia a archivos que están dentro del mismo sitio web o proyecto.
  - Las rutas relativas pueden ser más cortas y más fáciles de mantener, ya que no necesitas especificar la ruta completa desde la raíz del sitio web.
  - Ejemplo: Si tienes un archivo HTML en el directorio **/blog** y quieres referenciar un archivo de estilo CSS llamado **styles.css** que está en el mismo directorio, la ruta relativa sería simplemente **styles.css**.
- 
- Las rutas relativas especifican la ubicación de un archivo en relación con la ubicación actual del archivo de referencia.
  - Se utilizan rutas relativas cuando quieres hacer referencia a archivos que están dentro del mismo sitio web o proyecto.
  - Las rutas relativas pueden ser más cortas y más fáciles de mantener, ya que no necesitas especificar la ruta completa desde la raíz del sitio web.
  - Son portátiles: funcionan independientemente del dominio o servidor donde esté alojado el sitio.
  - Ejemplo: Si tienes un archivo HTML en el directorio **/blog** y quieres referenciar un archivo de estilo CSS llamado **styles.css** que está en el mismo directorio, la ruta relativa sería simplemente **styles.css**.

- **Rutas Absolutas (en contexto web):**

- Las rutas absolutas en el contexto web especifican la ubicación de un archivo desde la raíz del sitio web (no del sistema de archivos del servidor).
- Se utilizan rutas absolutas cuando necesitas hacer referencia a archivos que están en ubicaciones específicas desde la raíz del sitio web, independientemente de dónde se encuentre el archivo HTML que las referencia.
- Las rutas absolutas pueden ser útiles en sitios web complejos con múltiples niveles de directorios.
- **Importante:** En desarrollo local, la "raíz" será la carpeta que sirva tu servidor local.
- Ejemplo: Si tienes un archivo HTML en cualquier directorio y quieres referenciar una imagen llamada **image.jpg** que está en el directorio **/images** en la raíz del sitio web, la ruta absoluta sería **/images/image.jpg**.

La elección entre rutas relativas y absolutas se basa en las necesidades y la estructura del proyecto. Algunos puntos a considerar son:

- **Tamaño del proyecto:**

- En proyectos más pequeños y simples, las rutas relativas suelen ser más convenientes y fáciles de manejar.
- En proyectos más grandes y complejos, donde hay una estructura de directorios más profunda y múltiples páginas HTML, las rutas absolutas pueden ser más robustas y menos propensas a errores.

- **Portabilidad:**

- Las rutas relativas suelen ser más portátiles ya que no dependen de la ubicación del sitio web en el servidor.
- Las rutas absolutas pueden ser menos portátiles si el sitio web se mueve a un nuevo dominio o servidor, ya que las rutas pueden necesitar modificarse.

- **Mantenimiento:**

- Las rutas relativas pueden ser más fáciles de mantener, especialmente si los archivos y directorios se reorganizan en el proyecto.
- Las rutas absolutas pueden requerir menos ajustes si la estructura del proyecto cambia significativamente, ya que no dependen de la ubicación relativa de los archivos.

## 2. Navegación por directorios

Antes de ver ejemplos específicos, es importante entender cómo navegar por la estructura de directorios:

- `'./'` o sin prefijo: Directorio actual.
- `'../'`: Directorio padre (subir un nivel).
- `'../../'`: Subir dos niveles de directorio.
- `'/'`: Raíz del sitio web.

Un ejemplo de estructura de directorios podría ser:

```
mi-proyecto/  
├─ index.html  
├─ css/  
│   ├─ styles.css  
│   └─ components.css  
├─ js/  
│   ├─ main.js  
│   └─ utils.js  
├─ images/  
│   └─ logo.png  
└─ pages/  
    ├─ about.html  
    └─ contact/  
        └─ index.html
```

## 3. ¿Cómo incluir assets?

Para este ejemplo usaremos CSS, pero valdría igual para cualquier otro asset.

- Desde el archivo raíz (index.html):

```
<!-- Ruta relativa sin prefijo (equivale a ./css/styles.css) -->  
<link rel="stylesheet" href="css/styles.css">  
  
<!-- Ruta relativa explícita -->  
<link rel="stylesheet" href="./css/styles.css">  
  
<!-- Ruta absoluta (desde la raíz del sitio) -->  
<link rel="stylesheet" href="/css/styles.css">
```

- Desde pages/about.html:

```
<!-- Subimos un nivel para acceder a la carpeta css -->  
<link rel="stylesheet" href="../../css/styles.css">  
  
<!-- Ruta absoluta (siempre funciona igual) -->  
<link rel="stylesheet" href="/css/styles.css">
```

- Desde `pages/contact/index.html`:

```
<!-- Subimos dos niveles para acceder a la carpeta css -->
<link rel="stylesheet" href="../../css/styles.css">

<!-- Ruta absoluta (siempre funciona igual) -->
<link rel="stylesheet" href="/css/styles.css">
```

## 4. Diferencias entre las formas de especificar rutas

Imagina que tienes un HTML en el que quieres incluir un archivo CSS externo. Básicamente, tienes 3 formas de especificar la ruta del archivo:

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
<link rel="stylesheet" type="text/css" href="/css/styles.css">
<link rel="stylesheet" type="text/css" href="./css/styles.css">
```

Nota: Ten en cuenta que `type="text/css"` es opcional en HTML5, puedes omitirlo si quieres.

Las 3 formas son válidas, pero difieren en cómo el navegador interpreta la ruta:

```
<link rel="stylesheet" href="css/styles.css">
```

Esta forma significa que el navegador buscará el archivo `styles.css` dentro de un directorio llamado `css` ubicado en el mismo nivel que la página HTML que contiene este link. Esta es una **ruta relativa**.

```
<link rel="stylesheet" href="/css/styles.css">
```

El prefijo `"/` indica que la ruta del archivo comienza desde el directorio raíz del sitio web. Esto significa que el navegador buscará el archivo `styles.css` dentro de un directorio llamado `css` ubicado en la raíz del sitio web, independientemente de la ubicación de la página HTML que contiene este link. Esta es una **ruta absoluta**.

```
<link rel="stylesheet" href="./css/styles.css">
```

El prefijo `"/` indica que la ruta del archivo comienza desde el directorio actual. Esto significa que el navegador buscará el archivo `styles.css` dentro de un directorio llamado `css` ubicado en el mismo nivel que la página HTML que contiene este script.

Es similar al primer ejemplo, también una **ruta relativa**, pero más explícito al indicar que la ruta comienza desde el directorio actual.

## 5. Buenas prácticas

### 5.1. Recomendaciones

- **Usa rutas relativas** para proyectos pequeños y medianos, ya que son más portátiles.
- Organiza los assets en carpetas específicas (css/, js/, images/, etc.).
- Mantén una estructura consistente en todo el proyecto.
- Usa nombres descriptivos para archivos y carpetas.
- Evita espacios en nombres de archivos y carpetas (usa guiones o guiones bajos).

### 5.2. Errores comunes:

- Mezclar rutas relativas y absolutas.
- No considerar la estructura de directorios al mover archivos.
- Usar rutas del sistema de archivos en lugar de rutas web.
- No probar las rutas en diferentes niveles de la estructura del sitio.

## 6. Servidor local vs. abrir archivos en el navegador

Una de las confusiones más comunes al desarrollar sitios web es entender la diferencia entre visualizar nuestro proyecto usando un servidor local (como Live Server de Visual Studio Code) versus simplemente abrir el archivo HTML directamente en el navegador.

### 6.1. Con servidor local (recomendado)

Cuando utilizamos un servidor local como Live Server, nuestro proyecto se sirve a través del **protocolo HTTP**:

```
http://localhost:5500/index.html  
http://127.0.0.1:5500/pages/about.html
```

Características:

- Las rutas absolutas funcionan correctamente.
- Podemos usar **/css/styles.css** sin problemas.
- Simula el comportamiento real de un servidor web.
- Permite usar tecnologías que requieren HTTP (AJAX, fetch, etc.).

## 6.2. Abriendo archivos directamente (no recomendado)

Cuando abrimos el archivo HTML directamente desde el explorador de archivos se usa el **protocolo FILE**:

```
file:///C:/Users/usuario/mi-proyecto/index.html
file:///C:/Users/usuario/mi-proyecto/pages/about.html
```

Características:

- Las rutas absolutas con `'/'` NO funcionan.
- `'/css/styles.css'` dará error 404.
- No simula el comportamiento real de un servidor.
- Limitaciones de seguridad del navegador, como no poder hacer peticiones AJAX/fetch a otros archivos locales por restricciones de CORS.
- Solo funcionan las rutas relativas.

## 6.3. Ejemplo práctico

Imagina esta estructura de proyecto:

```
mi-proyecto/
├─ index.html
├─ css/
│   └─ styles.css
└─ pages/
    └─ about.html
```

- En index.html:

```
<head>
  <!-- Funciona en ambos casos -->
  <link rel="stylesheet" href="css/styles.css">

  <!-- Solo funciona con servidor local -->
  <link rel="stylesheet" href="/css/styles.css">
</head>
```

- En pages/about.html:

```
<head>
  <!-- Funciona en ambos casos -->
  <link rel="stylesheet" href="../../css/styles.css">

  <!-- Solo funciona con servidor local -->
  <link rel="stylesheet" href="/css/styles.css">
</head>
```

## 6.4. ¿Por qué las rutas absolutas no funcionan sin servidor?

Cuando abrimos un archivo directamente en el navegador, este interpreta `/css/styles.css` como que debe buscar el archivo en la ruta:

```
file:///C:/css/styles.css
```

En lugar de:

```
file:///C:/Users/usuario/mi-proyecto/css/styles.css
```

Por tanto,

**usa siempre un servidor local durante el desarrollo**

## 6.5. Detección de problemas comunes

- Síntoma: Los estilos CSS no se aplican en algunas páginas.
- Posible causa: Estás usando rutas absolutas, pero abriendo los archivos directamente.
- Síntoma: Todo funciona localmente, pero falla al subir al servidor.
- Posible causa: Rutas absolutas que apuntan a ubicaciones incorrectas en el servidor de producción.
- Síntoma: Las imágenes no se cargan en páginas internas.
- Posible causa: Rutas relativas mal configuradas o no considerar la profundidad de directorios.

## 7. Consejos para depuración

Si cualquiera de nuestros assets no se carga:

1. Abre las herramientas de desarrollador del navegador.
2. Revisa la pestaña "Network" para ver si hay errores 404.
3. Verifica las rutas copiando y pegando la URL del archivo en una nueva pestaña.
4. Comprueba que el archivo existe en la ubicación especificada.
5. Revisa las mayúsculas y minúsculas de los nombres de assets.