



RUTAS DE ARCHIVOS HTML

Una ruta de archivo describe la ubicación de un archivo en la estructura de carpetas de un sitio web

EJEMPLOS DE RUTAS DE ARCHIVOS

Path	Descripción
<code></code>	El archivo "picture.jpg" se encuentra en la misma carpeta que la página actual
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta images de la carpeta actual
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta images de la raíz de la web actual
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta situada un nivel por encima de la carpeta actual



RUTAS DE ARCHIVOS HTML

- Una ruta de archivo describe la ubicación de un archivo en la estructura de carpetas de un sitio web.
- Las rutas de archivos se utilizan cuando se enlaza a archivos externos, como:
 - Páginas web
 - Imágenes
 - Hojas de estilo
 - JavaScripts

RUTAS ABSOLUTAS DE ARCHIVOS

- Una ruta de archivo absoluta es la URL completa de un archivo:

Ejemplo

```

```

RUTAS RELATIVAS DE ARCHIVOS

- Una ruta de archivo relativa apunta a un archivo relativo a la página actual.
- En el siguiente ejemplo, la ruta del archivo apunta a un archivo de la carpeta images situada en la raíz de la web actual:

Ejemplo

```

```

- En el siguiente ejemplo, la ruta del archivo apunta a un archivo de la carpeta images situada en la carpeta actual:

Ejemplo

```

```

- En el siguiente ejemplo, la ruta del archivo apunta a un archivo de la carpeta images que se encuentra en la carpeta situada un nivel por encima de la carpeta actual:

Ejemplo

```

```

PRÁCTICAS RECOMENDADAS

- La mejor práctica es utilizar rutas de archivo relativas (si es posible).
- Al utilizar rutas de archivo relativas, sus páginas web no estarán vinculadas a su URL base actual. Todos los enlaces funcionarán en su propio ordenador (localhost), así como en su dominio público actual y en sus futuros dominios públicos.



HTML - EL ELEMENTO HEAD

El elemento HTML <head> es un contenedor para los siguientes elementos: <title>, <style>, <meta>, <link>, <script> y <base>.

EL ELEMENTO <HEAD> DE HTML

- El elemento <head> es un contenedor de metadatos (datos sobre datos) y se coloca entre la etiqueta <html> y la etiqueta <body>.
- Los metadatos HTML son datos sobre el documento HTML. Los metadatos no se muestran.
- Los metadatos suelen definir el título del documento, el juego de caracteres, los estilos, las secuencias de comandos y otra metainformación.

EL ELEMENTO HTML <TITLE>

- El elemento <title> define el título del documento. El título debe ser sólo texto y se muestra en la barra de título del navegador o en la pestaña de la página.
- El elemento <title> es obligatorio en los documentos HTML.
- El contenido del título de una página es muy importante para la optimización de los motores de búsqueda (SEO). El título de la página es utilizado por los algoritmos de los motores de búsqueda para decidir el orden al listar las páginas en los resultados de búsqueda.
- El elemento <title>:
 1. define un título en la barra de herramientas del navegador
 2. proporciona un título para la página cuando se añade a favoritos
 3. muestra un título para la página en los resultados del motor de búsqueda
- Por lo tanto, ¡intente que el título sea lo más preciso y significativo posible!

EL ELEMENTO HTML <STYLE>

- El elemento <style> se utiliza para definir información de estilo para una sola página HTML:

Ejemplo

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

EL ELEMENTO HTML <LINK>

- El elemento <link> define la relación entre el documento actual y un recurso externo.

La etiqueta <link> se suele utilizar para enlazar a hojas de estilo externas:

Ejemplo

```
<link rel="stylesheet" href="mystyle.css">
```

EL ELEMENTO <META> DE HTML

- El elemento <meta> se utiliza normalmente para especificar el juego de caracteres, la descripción de la página, las palabras clave, el autor del documento y la configuración de la ventana gráfica.
- Los metadatos no se muestran en la página, pero los utilizan los navegadores (para mostrar el contenido o recargar la página), los motores de búsqueda (palabras clave) y otros servicios web

EL ELEMENTO <META> DE HTML

- **Definir el juego de caracteres utilizado:**

```
<meta charset="UTF-8">
```

- **Definir palabras clave para los motores de búsqueda:**

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

- **Defina una descripción de su página web:**

```
<meta name="description" content="Free Web tutorials">
```

- **Definir el autor de una página:**

```
<meta name="author" content="John Doe">
```

- **Definir el autor de una página:**

```
<meta http-equiv="refresh" content="30">
```

- **Configurar la ventana gráfica para que su sitio web se vea bien en todos los dispositivos:**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Ejemplo

- ```
<meta charset="UTF-8">
<meta name="description" content="Tutoriales web gratuitos">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

# CONFIGURAR LA VENTANA GRÁFICA

- La ventana gráfica es el área visible de una página web para el usuario. Varía según el dispositivo: será más pequeña en un teléfono móvil que en la pantalla de un ordenador.
- Debe incluir el siguiente elemento <meta> en todas sus páginas web:

**<meta name="viewport" content="width=device-width, initial-scale=1.0">.**

- Esto da instrucciones al navegador sobre cómo controlar las dimensiones y el escalado de la página.
- La parte width=device-width establece el ancho de la página para seguir el ancho de pantalla del dispositivo (que variará dependiendo del dispositivo).
- La parte initial-scale=1.0 establece el nivel de zoom inicial cuando el navegador carga la página por primera vez.



# EL ELEMENTO <SCRIPT> DE HTML

- El elemento <script> se utiliza para definir JavaScripts del lado del cliente.
- El siguiente JavaScript escribe "¡Hola JavaScript!" en un elemento HTML con id="demo":

## Ejemplo

```
<script>función miFunción() { document.getElementById("demo").innerHTML = "¡Hola JavaScript!";}</script>
```

**Consejo:** Para aprenderlo todo sobre JavaScript, visite nuestro [Tutorial de JavaScript](#).

# EL ELEMENTO HTML <BASE>

- El elemento <base> especifica la URL base y/o el destino de todas las URL relativas de una página.
- La etiqueta <base> debe contener un atributo href o un atributo target, o ambos.
- Sólo puede haber un único elemento <base> en un documento.

## Ejemplo

- Especificar una URL por defecto y un destino por defecto para todos los enlaces de una página:

```
<head>
<base href="https://www.mipagina.com/" target="_blank">
</head>

<body>

Etiquetas base deHTML
</body>
```

Tag	Description
<u>&lt;head&gt;</u>	Define información sobre el documento
<u>&lt;title&gt;</u>	Define el título de un documento
<u>&lt;base&gt;</u>	Define una dirección por defecto o un destino por defecto para todos los enlaces de una página.
<u>&lt;link&gt;</u>	Define la relación entre un documento y un recurso externo
<u>&lt;meta&gt;</u>	Define metadatos sobre un documento HTML
<u>&lt;script&gt;</u>	Define un script del lado del cliente
<u>&lt;style&gt;</u>	Define la información de estilo de un documento



# GUÍA DE ESTILO HTML

Un código HTML coherente, limpio y ordenado facilita a los demás la lectura y comprensión de tu código.

# DECLARAR SIEMPRE EL TIPO DE DOCUMENTO

- Declare siempre el tipo de documento en la primera línea de su documento.
- El tipo de documento correcto para HTML es:

```
<!DOCTYPE html>
```

# UTILICE MINÚSCULAS EN LOS NOMBRES DE LOS ELEMENTOS

- HTML permite mezclar mayúsculas y minúsculas en los nombres de los elementos.
- Sin embargo, recomendamos utilizar minúsculas en los nombres de los elementos, porque:
  1. La mezcla de mayúsculas y minúsculas da mala imagen.
  2. Los desarrolladores suelen utilizar minúsculas
  3. Las minúsculas tienen un aspecto más limpio
  4. Las minúsculas son más fáciles de escribir

Bien:

```
<body><p>Esto es un párrafo.</p></body>
```

Malo:

```
<BODY><P>Esto es un párrafo.</P></BODY>
```



# CERRAR TODOS LOS ELEMENTOS HTML

- En HTML, no es necesario cerrar todos los elementos (por ejemplo, el elemento <p>).
- Sin embargo, recomendamos encarecidamente cerrar todos los elementos HTML, así:

Bien:

```
<section> <p>Esto es un párrafo.</p> <p>Esto es un párrafo.</p></section>
```

Malo:

```
<sección> <p>Esto es un párrafo.</sección>
```

# UTILICE MINÚSCULAS EN LOS NOMBRES DE ATRIBUTOS

- HTML permite mezclar mayúsculas y minúsculas en los nombres de atributos.
- Sin embargo, recomendamos utilizar minúsculas en los nombres de atributos, porque:
  1. La mezcla de mayúsculas y minúsculas da mala imagen.
  2. Los desarrolladores suelen utilizar minúsculas
  3. Las minúsculas tienen un aspecto más limpio
  4. Las minúsculas son más fáciles de escribir

Bien:

```
Visita nuestro tutorial HTML
```

Mal:

```
Visita nuestro tutorial de HTML
```

# VALORES DE ATRIBUTO SIEMPRE ENTRECOMILLADOS

- HTML permite valores de atributo sin comillas.
- Sin embargo, recomendamos entrecomillar los valores de los atributos porque:
  1. Los desarrolladores suelen entrecomillar los valores de los atributos.
  2. Los valores entrecomillados son más fáciles de leer
  3. DEBE utilizar comillas si el valor contiene espacios

Bien:

```
<table class="striped">
```

Malo:

```
<table class=striped>
```

Muy mal:

Esto no funcionará, porque el valor contiene espacios:

```
<table class=table striped>
```

# ESPECIFIQUE SIEMPRE ALT, ANCHURA Y ALTURA PARA LAS IMÁGENES

- Especifique siempre el atributo alt para las imágenes. Este atributo es importante si la imagen, por alguna razón, no puede mostrarse.
- Además, defina siempre la anchura y la altura de las imágenes. Esto reduce el parpadeo, porque el navegador puede reservar espacio para la imagen antes de cargarla.

Bien:

```

```

Malo:

```

```

# ESPACIOS Y SIGNOS DE IGUALDAD

- HTML permite espacios alrededor de signos iguales. Pero sin espacio es más fácil de leer y agrupa mejor las entidades juntas.

Bien:

```
<link rel="hoja de estilo" href="styles.css">
```

Malo:

```
<link rel = "hoja de estilo" href = "styles.css">
```

# EVITE LAS LÍNEAS DE CÓDIGO LARGAS

- Cuando se utiliza un editor HTML, NO es conveniente desplazarse hacia la derecha y hacia la izquierda para leer el código HTML.
- Trate de evitar líneas de código demasiado largas.



# LÍNEAS EN BLANCO Y SANGRÍA

No agregue líneas en blanco, espacios o sangrías sin una razón.

Para facilitar la lectura, agregue líneas en blanco para separar bloques de código grandes o lógicos.

Para facilitar la lectura, agregue dos espacios de sangría. No utilice la tecla de tabulación.

## Bien:

```
<body>
```

```
<h1>Ciudades famosas</h1>
```

```
<h2>Tokio</h2><p>Tokio es la capital de
Japón, el centro del área metropolitana de
Tokio y el área metropolitana más poblada
del mundo.</p>
```

```
<h2>Londres</h2><p>Londres es la
capital de Inglaterra. Es la ciudad más
poblada del Reino Unido.</p>
```

```
<h2>Paris</h2><p>Paris es la capital de
Francia. El área de París es uno de los
centros de población más grandes de
Europa.</p>
```

## Malo

```
<body>
<h1>Ciudades famosas</h1>
```

```
<h2>Tokio</h2>
```

```
<p>Tokio es la capital de Japón, el centro del área
metropolitana de Tokio y el área metropolitana más
poblada del mundo.</p>
```

```
<h2>Londres</h2>
```

```
<p>Londres es la capital de Inglaterra. Es la ciudad más
poblada del Reino Unido.</p>
```

```
<h2>Paris</h2>
```

```
<p>Paris es la capital de Francia. El área de París es uno de
los centros de población más grandes de Europa.</p>
</body>
```

# NUNCA OMITAS EL ELEMENTO <TITLE>

- El elemento <title> es obligatorio en HTML.
- ¡El contenido del título de una página es muy importante para la optimización de motores de búsqueda (SEO)! Los algoritmos de los motores de búsqueda utilizan el título de la página para decidir el orden de las páginas en los resultados de búsqueda.
- El elemento <title>:
  1. Define un título en la barra de herramientas del navegador
  2. Proporciona un título para la página cuando se agrega a favoritos
  3. Muestra un título para la página en los resultados de los motores de búsqueda
- Por lo tanto, intenta que el título sea lo más preciso y significativo posible:
- <title>Guía de estilo HTML y convenciones de codificación</title>

# ¿OMITIR <HTML> Y <BODY>?

- Una página HTML validará sin las etiquetas <html> y <body>:

## Ejemplo

```
<!DOCTYPE html><head> <title>Título de la página</title></head>
```

```
<h1>This is a heading</h1><p>This is a paragraph.</p>
```

- Sin embargo, recomendamos encarecidamente añadir siempre las etiquetas <html> y <body>.
- Omitir <body> puede producir errores en los navegadores más antiguos.
- Omitir <html> y <body> también puede bloquear el software DOM y XML.

# ¿OMITIR <HEAD>?

- También se puede omitir la etiqueta HTML <head>.
- Los navegadores agregarán todos los elementos antes de <body>, a un elemento predeterminado <head>.

## Ejemplo

```
<!DOCTYPE html><html><title>Título de la página</title><body>
```

```
<h1>Este es un encabezado</h1><p>Este es un párrafo.</p>
```

```
</body></html>
```

- Sin embargo, recomendamos usar la etiqueta <head>.

# ¿CERRAR ELEMENTOS HTML VACÍOS?

- En HTML, es opcional cerrar los elementos vacíos.
- Permitido:

```
<meta charset="utf-8">
```

- También se permite:

```
<meta charset="utf-8" />
```

- Si espera que el software XML/XHTML acceda a su página, mantenga la barra de cierre (/), ya que es necesaria en XML y XHTML.

# AGREGUE EL ATRIBUTO LANG

- Siempre debe incluir el atributo lang dentro de la etiqueta <html> para declarar el idioma de la página web. Esto está destinado a ayudar a los motores de búsqueda y navegadores.

## Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
 <title>Título de la página</title>
</head>
<body>

<h1>Esto es un encabezado</h1>

<p>Esto es un párrafo.</p>

</body>
</html>
```



# METADATOS

- Para garantizar una interpretación adecuada y una correcta indexación en los motores de búsqueda, tanto el idioma como la codificación de caracteres `<meta charset="charset">` deben definirse lo antes posible en un documento HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
 <meta charset="UTF-8">
 <title>Título de la página</title>
</head>
```

# CONFIGURACIÓN DE LA VENTANA

- La ventana es el área visible del usuario de una página web. Varía según el dispositivo: será más pequeño en un teléfono móvil que en la pantalla de una computadora.
- Debes incluir el siguiente elemento `<meta>` en todas tus páginas web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Esto le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.
- La parte `width=device-width` establece el ancho de la página para seguir el ancho de pantalla del dispositivo (que variará según el dispositivo).
- La parte `initial-scale=1.0` establece el nivel de zoom inicial cuando el navegador carga la página por primera vez.
- A continuación, se muestra un ejemplo de una página web *sin* la metaetiqueta `viewport` y la misma página web *con* la metaetiqueta `viewport`:

**Consejo:** Si está navegando por esta página con un teléfono o una tableta, puede hacer clic en los dos enlaces a continuación para ver la diferencia

# COMENTARIOS HTML

- Los comentarios cortos deben escribirse en una línea, así:
- `<!-- Esto es un comentario -->`
- Los comentarios que abarquen más de una línea, deben escribirse así:
- `<!-- Este es un ejemplo de comentario largo. Este es un ejemplo de comentario largo. Este es un ejemplo de comentario largo. Este es un ejemplo de comentario largo.-->`
- Los comentarios largos son más fáciles de observar si tienen sangría con dos espacios.

# USO DE HOJAS DE ESTILO

- Utilice una sintaxis sencilla para enlazar a hojas de estilo (el atributo type no es necesario):
  - `<link rel="hoja de estilo" href="styles.css">`
  - Las reglas CSS cortas se pueden escribir comprimidas, así:
  - `p.intro {font-family:Verdana; tamaño de fuente:16em;}`
  - Las reglas CSS largas deben escribirse en varias líneas:
  - ```
body {  
  background-color: lightgrey;  
  font-family: "Arial Black", Helvetica, sans-serif;  
  font-size: 16em;  
  color: black;  
}
```
1. Coloque el soporte de apertura en la misma línea que el selector
 2. Utilice un espacio antes del soporte de apertura
 3. Utilice dos espacios de sangría
 4. Utilice punto y coma después de cada par propiedad-valor, incluido el último
 5. Utilice solo comillas alrededor de los valores si el valor contiene espacios
 6. Coloque el soporte de cierre en una nueva línea, sin espacios iniciales

CARGA DE JAVASCRIPT EN HTML

- Utilice una sintaxis sencilla para cargar scripts externos (el atributo type no es necesario):

```
<script src="myscript.js">
```

ACCESO A ELEMENTOS HTML CON JAVASCRIPT

- El uso de código HTML "desordenado" puede dar lugar a errores de JavaScript.
- Estas dos instrucciones JavaScript producirán resultados diferentes:

Ejemplo

```
<!DOCTYPE HTML>
<HTML>
<BODY>

<P ID="DEMO">ESTE ES EL PÁRRAFO 1.</P>
<P ID="DEMO">ESTE ES EL PÁRRAFO 2.</P>

<SCRIPT>
// SOLO SE SOBRESCRIBIRÁ EL PÁRRAFO 2
DOCUMENT.GETELEMENTBYID("DEMO").INNERHTML = "HELLO.";
</SCRIPT>

</BODY>
</html>
```

- Algunos servidores web (Apache, Unix) distinguen entre mayúsculas y minúsculas con respecto a los nombres de archivo: no se puede acceder a "london.jpg" como "London.jpg".
- Otros servidores web (Microsoft, IIS) no distinguen entre mayúsculas y minúsculas: se puede acceder a "london.jpg" como "London.jpg".
- Si utilizas una mezcla de mayúsculas y minúsculas, tienes que ser consciente de ello.
- Si pasas de un servidor que no distingue entre mayúsculas y minúsculas a uno que distingue entre mayúsculas y minúsculas, ¡incluso los errores más pequeños romperán tu web!
- Para evitar estos problemas, utilice siempre nombres de archivo en minúsculas.



EXTENSIONES DE ARCHIVO

- Los archivos HTML deben tener una **extensión .html** (.htm está permitida).
- Los archivos CSS deben tener una **extensión .css**.
- Los archivos JavaScript deben tener una **extensión .js**.

¿DIFERENCIAS ENTRE .HTM Y .HTML?

- ¡No hay diferencia entre las extensiones de archivo .htm y .html!
- Ambos serán tratados como HTML por cualquier navegador web y servidor web.

NOMBRES DE ARCHIVO PREDETERMINADOS

- Cuando una URL no especifica un nombre de archivo al final (como "https://www.mipagina.com/"), el servidor simplemente agrega un nombre de archivo predeterminado, como "index.html", "index.htm", "default.html" o "default.htm".
- Si su servidor está configurado solo con "index.html" como nombre de archivo predeterminado, su archivo debe llamarse "index.html" y no "default.html".
- Sin embargo, los servidores se pueden configurar con más de un nombre de archivo predeterminado; Por lo general, puede configurar tantos nombres de archivo predeterminados como desee.