

# Estructura y Etiquetas dentro de <HEAD> y atributos

## HyperText Markup Language

# Comentarios en HTML

Los comentarios sirven para anotar o documentar el código sin que se muestre en la página web.

El navegador los ignora al renderizar el contenido.

Son muy útiles para:

- Explicar partes del código.
- Recordar cambios o pendientes.
- Desactivar temporalmente líneas de código.

<!-- Esto es un comentario en HTML -->

# Comentarios en HTML

```
<!-- Título principal -->
<h1>Bienvenidos a mi página</h1>
```

```
<!-- Lista de hobbies -->
<ul>
  <li>Leer</li>
  <li>Programar</li>
  <li>Deporte</li>
</ul>
```

# Comentarios en HTML - Mala práctica

Dejar grandes bloques de código comentado que ya no se usan → ensucia y hace más difícil mantener el archivo.

Usar comentarios como “basurero” de código viejo en lugar de borrarlo.

Poner comentarios obvios (ej. `<!-- aquí empieza el body -->`).

# Comentarios en HTML - Buena práctica

Documentar solo lo necesario y de forma clara.

Eliminar código que ya no se usará (para eso está el control de versiones como Git).

Usar comentarios solo como ayuda, no como sustituto de un código limpio y ordenado.

# Atributos en HTML

Los atributos proporcionan información extra a las etiquetas HTML.

Van dentro de la etiqueta de apertura.

Su sintaxis es:

```
<tagname atributo="valor">contenido</tagname>
```

# Ejemplos clásicos de atributos

type → Define el tipo de input.

```
<input type="checkbox">
```

disabled → Desactiva un campo.

```
<input type="checkbox" disabled>
```

checked → Marca un checkbox por defecto.

```
<input type="checkbox" checked>
```

# Ejemplos clásicos de atributos

value → Valor inicial de un input.

```
<input type="text" value="Hola">
```

placeholder → Texto guía dentro de un input.

```
<input type="text" placeholder="Escribe tu nombre">
```

src y alt → Para imágenes.

```

```

href y target → Para enlaces.

```
<a href="https://google.com" target="_blank">Google</a>
```

# Actividad

Crea un archivo atributos.html que contenga lo siguiente:

Un título principal (**<h1>**) con el texto “Practicando atributos en HTML”.

Un checkbox normal, uno marcado por defecto y otro deshabilitado.

Un campo de texto con:

un value inicial que diga “Texto por defecto”.

un placeholder que diga “Escribe tu nombre aquí...”.

Una imagen que use los atributos src y alt.  
(puedes usar: <https://via.placeholder.com/150>)

Un enlace a Google que se abra en una nueva pestaña usando target="\_blank"

The background of the slide features a photograph of a person's hands typing on a laptop keyboard. A large green plant is positioned in the lower-left foreground, partially obscuring the bottom of the laptop.

**HTML**



**ETIQUETA  
`<head>`**

# <head>

La etiqueta <head> en HTML5 es una sección especial del documento HTML donde se colocan metadatos y recursos que el navegador necesita antes de mostrar el contenido visual al usuario.



# ¿Qué contiene <head>?

Etiqueta	Función principal
<title>	Título de la pestaña del navegador
<meta>	Información sobre el documento (idioma, codificación, autor, etc.)
<link>	Vincula recursos externos (por ejemplo, hojas de estilo CSS)
<style>	Estilos CSS internos
<script>	Código JavaScript (puede ir en <head> o antes de cerrar <body>)
<base>	Define una URL base para los enlaces relativos

# <title>

La etiqueta <title> es un elemento que se coloca dentro de la sección <head> de un documento HTML y define el título de la página web.



## ¿Para qué sirve?

- Aparece en la pestaña del navegador o en la barra del título de la ventana.
- Se usa como el nombre de la página cuando el usuario la guarda en favoritos o marcadores.
- Los motores de búsqueda (como Google) lo usan como título principal en los resultados.

# <meta>

La etiqueta <meta> se usa dentro de la sección <head> de un documento HTML para proporcionar metadatos (información sobre la página) al navegador, motores de búsqueda y otros servicios.

Es una etiqueta auto-cerrada, es decir, no necesita etiqueta de cierre.



## ¿Para qué sirve?

- Definir la codificación de caracteres (muy importante para evitar errores con acentos o eñes).
- Indicar el autor del sitio.
- Optimizar el SEO (títulos, descripciones).
- Hacer que la página sea responsive en móviles.
- Instrucciones para robots de búsqueda.



## Ejemplos comunes:

### ✓ 1. charset: Codificación de caracteres

<meta charset="UTF-8">

- ✓ Define el conjunto de caracteres usado en el documento (ej. UTF-8).
- ✓ UTF-8 es el estándar moderno que soporta caracteres latinos, acentos, ñ, símbolos, emojis, etc.
- ✓ Evita errores como:
- ✓ Siempre debe ser la primera etiqueta <meta> dentro del <head>.

¡Hola! se ve como Á¡Hola!



## 2. viewport: Diseño adaptable (responsive)

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

- ✓ Permite que la página se adapte al tamaño de pantalla del dispositivo (móvil, tableta, PC).
- ✓ Sin esta etiqueta, los sitios pueden aparecer demasiado pequeños o desajustados en móviles.
- ✓ El valor width=device-width ajusta el contenido al ancho del dispositivo.
- ✓ initial-scale=1.0 establece el nivel de zoom inicial.

Fundamental para sitios modernos y responsivos.



### 3. description: Descripción para buscadores (SEO)

```
<meta name="description"  
content="Sitio web con tutoriales de programación.">
```

- ✓ Proporciona una breve descripción del contenido del sitio.
- ✓ Aparece como el texto debajo del título en los resultados de búsqueda de Google.
- ✓ Ideal para atraer clics y mejorar el SEO (Search Engine Optimization).



Importante: no garantiza que Google lo use, pero ayuda bastante.



## 4. author: Autor del sitio

```
<meta name="author"  
content="Luis Francisco Contreras">
```

- ✓ Define quién creó o mantiene el contenido de la página.
- ✓ Útil para fines de documentación, crédito o licencias.
- ✓ Algunas herramientas SEO o navegadores pueden mostrar esta info.



Es más útil a nivel organizacional o académico.



## 5. robots: Instrucciones para motores de búsqueda

```
<meta name="robots"  
content="index, follow">
```

Indica a los motores de búsqueda (Google, Bing, etc.) qué hacer con la página:

Valor	Significado
index	Puede indexar (incluir en búsquedas)
noindex	No debe indexarse
follow	Puede seguir los enlaces
nofollow	No debe seguir los enlaces

Ejemplo para bloquear una página de prueba:

```
<meta name="robots" content="noindex,nofollow">
```



Útil para evitar indexar páginas en construcción, paneles de administración, o contenido duplicado.

# <link>

La etiqueta <link> se utiliza en el elemento <head> de un documento HTML para vincular recursos externos al archivo HTML, como hojas de estilo CSS, íconos, fuentes, archivos RSS, entre otros.

Es una etiqueta auto-cerrada, lo que significa que no necesita etiqueta de cierre



## Función principal

La función más común de <link> es conectar una hoja de estilos CSS al archivo HTML, permitiendo que el contenido tenga diseño visual (colores, márgenes, tamaños, etc.).



## Ejemplo básico

- rel="stylesheet" indica que el recurso es una hoja de estilo.
- href="estilos.css" especifica la ruta del archivo externo que se va a aplicar.

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



## Atributos principales de <link>

Atributo	Descripción
rel	Define la relación entre el documento HTML y el recurso externo. Ej: "stylesheet", "icon", "manifest"
href	Ruta al recurso externo (local o URL)
type	Tipo MIME del recurso (ej. "text/css" para hojas de estilo, opcional en HTML5)
media	Especifica el tipo de dispositivo o condiciones para aplicar el recurso (ej. screen, print)



## Ejemplos útiles

### ✓ 1. Vincular hoja de estilos

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

### ✓ 2. Ícono del sitio (favicon)

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

### ✓ 3. Fuente externa (Bootstrap)

```
<!-- Bootstrap 5 CSS -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0  
/dist/css/bootstrap.min.css"  
rel="stylesheet">
```

### ✓ 4. Estilo específico para impresión

```
<link rel="stylesheet" href="print.css" media="print">
```

# <style>

La etiqueta <style> se utiliza para escribir código CSS directamente dentro del archivo HTML, con el fin de aplicar estilos (colores, fuentes, márgenes, tamaños, etc.) a los elementos de la página.

Se coloca normalmente dentro del <head>, aunque también puede usarse de forma muy puntual dentro del <body> (aunque no es buena práctica).

```
<head>
  <style>
    /* Aquí va el código css */
  </style>
</head>
```



## ¿Para qué sirve?

- Establece estilos internos para la página sin necesidad de un archivo CSS externo.
- Es útil para prototipos rápidos o pequeños proyectos.
- Permite aplicar diseño visual sin depender aún de Bootstrap o frameworks.



## Ejemplo de uso:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejemplo con Style</title>
    <style>
        body {
            background-color: #f0f0f0;
            font-family: Arial, sans-serif;
        }

        h1 {
            color: navy;
            text-align: center;
        }

        p {
            color: #333;
            font-size: 18px;
        }
    </style>
</head>
<body>
    <h1>Hola mundo</h1>
    <p>Este párrafo tiene estilo aplicado con CSS interno.</p>
</body>
</html>
```

## ⚠ Cuándo NO usar <style>

- En proyectos grandes o profesionales, se recomienda usar un archivo externo (.css) enlazado con <link>.
- Mantener estilos separados del contenido HTML mejora el mantenimiento, limpieza y reutilización del código.

# <script>

La etiqueta <script> se utiliza en HTML para incluir o escribir código JavaScript dentro de una página web. JavaScript es el lenguaje que le da interactividad y comportamiento dinámico a los sitios web.



## ¿Para qué sirve?

- Para manipular el contenido HTML dinámicamente (mostrar/ocultar elementos, cambiar estilos, etc.).
- Para validar formularios antes de enviarlos.
- Para manejar eventos como clics, teclas, desplazamientos, etc.
- Para cargar scripts externos desde archivos .js.



## Ejemplos

- ✓ 1. Script interno (escrito directamente en el HTML)

```
<script>
  alert("¡Hola, mundo!");
</script>
```

- ✓ 2. Script externo (vinculado desde un archivo .js)

```
<script src="scripts/miscript.js"></script>
```

- ✓ 3. Colocación del <script> en el documento

Ubicación recomendada	¿Por qué?
Al final del <body>	Así se carga primero el contenido HTML, y después el JavaScript, evitando errores.
En el <head> con defer	Ejecuta el script después de que el HTML esté cargado.

```
<script src="main.js" defer></script>
```

## Buenas prácticas

- Usar defer o colocar el <script> justo antes de </body> para mejorar el rendimiento.
- Mantener el JavaScript en archivos externos para separar estructura (HTML), estilo (CSS) y comportamiento (JS).
- No abusar de scripts en línea.

# <base>

La etiqueta <base> se usa dentro del <head> de un documento HTML para definir una URL base para todos los enlaces relativos (href o src) de la página



## ¿Para qué sirve?

Cuando usas rutas relativas, por ejemplo:

- images/logo.png
- pages/contacto.html)

la etiqueta <base> permite establecer desde qué dirección deben resolverse esas rutas.



## Características clave

- Solo puede haber una etiqueta <base> por documento HTML.
- Se debe colocar antes de cualquier uso de enlaces relativos.
- Se puede usar con los atributos:
  - href → Ruta base
  - target → Valor por defecto para todos los enlaces (\_blank, \_self, etc.)



## Ejemplo de uso:

```
<head>
  <base href="https://www.ejemplo.com/">
  <title>Mi sitio</title>
</head>
<body>
  <a href="contacto.html">Contacto</a>
</body>
```



En este caso, el enlace lleva a:

<https://www.ejemplo.com/contacto.html>

Aunque el href diga solo contacto.html, se resuelve desde la base especificada.



Ejemplo con target:

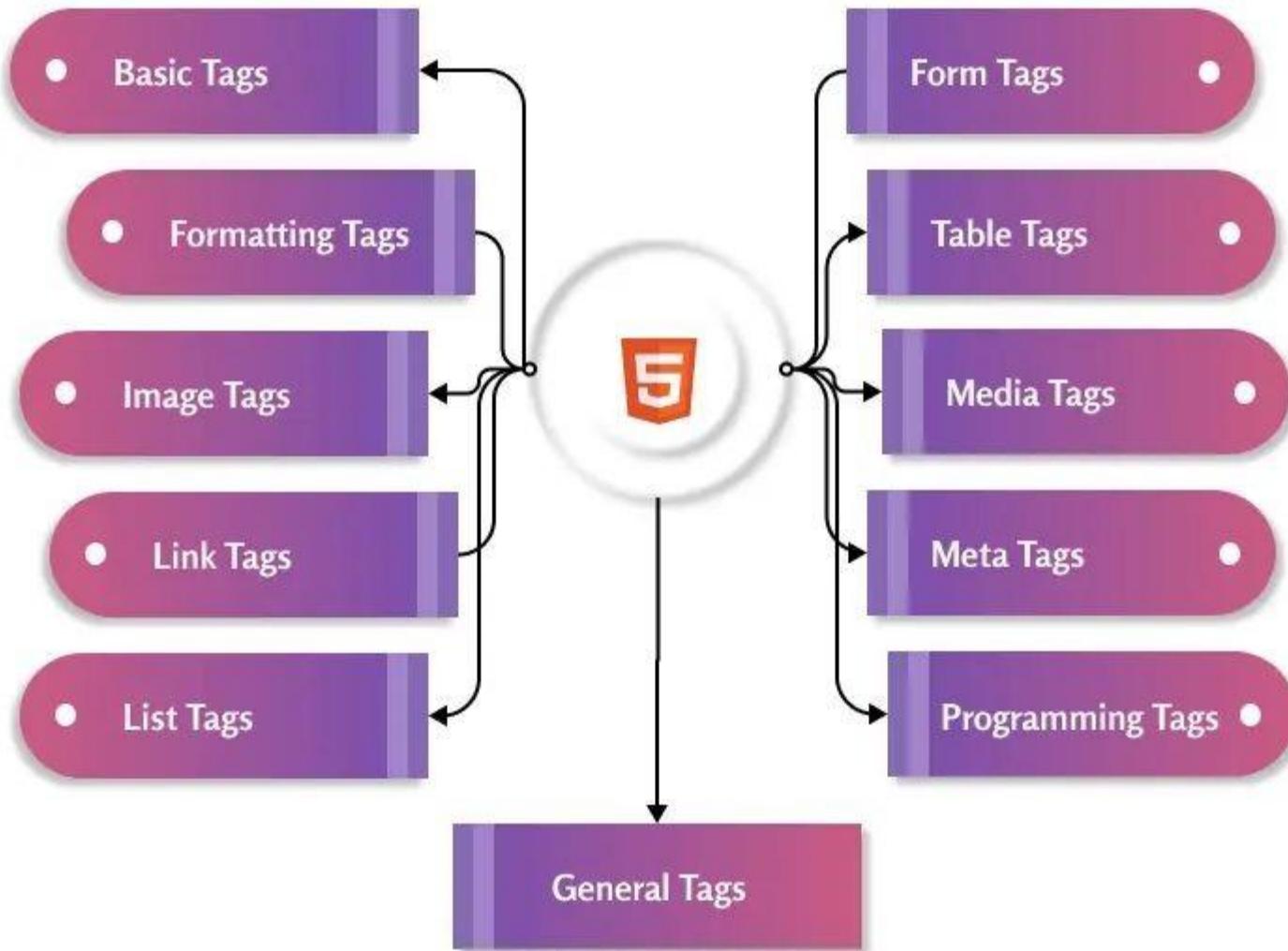
<base href="https://www.ejemplo.com/" target="\_blank">

Esto hace que **todos los enlaces** se abran en una nueva pestaña.

## ⚠ Cuándo usar <base>

- ✓ Cuando trabajas con rutas relativas en proyectos donde la estructura del sitio depende de una URL principal (como al cargar contenido desde un subdominio o CDN).
- ✗ Evítala si trabajas con muchas rutas relativas distintas y no quieres que todas apunten al mismo origen.

# HTML Tags List



## Actividad 4

Hacer una página web personal donde te presentes a tus amigos considerando los conceptos aprendidos de UI y UX (no uses CSS aun) agrega los elementos de head vistos en esta presentación.