

# HTML

## 1. Introducción a HTML

HTML es un lenguaje de **marcado de hipertexto**. Es el lenguaje con el que se estructuran las webs. Se basa en un sistema de “**cajas**” o contenedores que se sitúan uno debajo del otro. Con HTML no hay que tratar de dar estilo a esas cajas, de eso se encarga CSS.

HTML se basa en el uso de etiquetas. Las etiquetas se introducen entre los símbolos <>. Cuando se escribe una etiqueta, se crea un “objeto”, es decir un elemento o “caja”. Para terminar de crear la “caja” hay que cerrar la etiqueta con </>:

- `<etiqueta></etiqueta>`

Las etiquetas se componen de atributos o propiedades que tienen un valor que se pone entre comillas:

- `<etiqueta atributo="valor"></etiqueta>`  
`<button color="red"></button>`

Los atributos se secuencian dentro de la etiqueta y se ponen tantos como se consideren necesarios.

A pesar de no ser considerado un lenguaje de programación en sí, HTML tiene unas determinadas reglas que debe cumplir a nivel semántico. Por ejemplo, el título h1, debe ser único, ya que éste posiciona en google.

## 2. Directorio de archivos

[https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/Manejando\\_los\\_archivos](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/Manejando_los_archivos)

<https://www.htmlquick.com/es/tutorials/content-grouping-and-structure.html>

Cuando estés trabajando en un sitio web localmente en tu computadora, debes mantener todos los archivos relacionados en un solo directorio que **refleje la estructura de archivos del sitio web** publicado en el servidor.

### Recomendaciones:

- Usar nombres de directorios en minúscula y sin espacios.
- Mejor separar las palabras con guiones (-) que con barras bajas (\_)

### Estructura básica:

- index.html
- Directorios:
  - images
  - style
  - script

### Rutas:

- Mismo directorio: nombre archivo
- Subdirectorio: /
- Directorios anteriores: ../

### 3. HTML Semántico

A continuación, se va a ver la manera de definir HTML semántico, es decir, la manera de escribir código HTML de forma que sea más legible y óptima para el navegador.



Se han de respetar las secciones (su posición, función y características) de las webs. Estas secciones se corresponde con una serie de elementos/etiquetas que se introdujeron en HTML5: header, nav, section, aside, article, footer.

- **<header></header>**  
Encabezado del body o de una determinada sección o article. Se suele usar como **presentación de la página**.
- **<nav></nav>**  
Sección o barra de navegación. Es un contenedor para enlaces de navegación, que se supone proveen **formas de acceder a las secciones** del sitio web, así como a documentos externos. Se supone que los vínculos dentro de este elemento corresponden a un bloque de navegación principal únicamente. A veces se incluye en el header.
- **<main></main>**  
Representa el contenido principal del <body>. Este contenido debe ser único al documento, excluyendo cualquier contenido que se repita a través de un conjunto de documentos como barras laterales, enlaces de navegación, información de derechos de autor, logos del sitio y formularios de búsqueda. Además, el main no debe ser descendiente de un elemento <article>, <aside>, <footer>, <header>, o <nav>.
- **<article></article>**  
Contenido redistribuible. Contenedor de bloques de contenido que se consideran **bloques independientes** del sitio web y pueden, por lo tanto, ser vistos, reutilizados y distribuidos por separado, como por ejemplo, en la sindicación. Puedes encontrar habitualmente a este elemento encerrando artículos, entradas de blogs o mensajes de un foro.
- **<section></section>**  
Es un elemento de seccionamiento genérico, diseñado para contener una parte de un documento **temáticamente definida**. Los contenidos de un elemento section son usualmente precedidos por un encabezado.
- **<aside></aside>**  
Contenedor para información que se considera solo **levemente relacionada** al documento o la sección en la que es definida. En otras palabras, el contenido del elemento aside debe estar relacionado al contenido circundante, pero no ser completamente necesario para su comprensión.
- **<footer></footer>**  
Pie de una sección o documento, donde los autores habitualmente colocan firmas, información **acerca del autor**, información de licencias, documentos relacionados, etc. En muchos casos, los contenidos del pie de un documento son consistentes a lo largo de todo el sitio.
- **<address></address>**  
Provee información de contacto para el autor de la sección o documento, como una dirección postal, una dirección de correo electrónico, un número telefónico, una ciudad, un país, un código postal, etc.

## 4. Tipos de etiquetas

Atendiendo al modo en que se cierran las etiquetas existen dos tipos de etiquetas:

- **Etiquetas simples o Self-Closing Tags:** Se cierran solas
  - `<br>` o `<br/>` (en HTML5 da igual, se puede usar sin barra)
  - `<meta>`, `<link>`, `<input>`, `<img>`, `<iframe>`...
- **Etiquetas dobles o de Apertura y Cierre:** Se cierran con otra etiqueta que incluye "/" antes del nombre de la etiqueta. Estas etiquetas suelen tener algún tipo de contenido:  
`<h1>Titulo</h1>`

## 5. Estructura básica



```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi página</title>
  </head>
  <body>
  </body>
</html>
```

- `<!DOCTYPE html>`: indica que se está usando la última versión de HTML, HTML5
- `<html></html>`: Dentro de estas etiquetas se encuentra toda la página, es decir, todo el código HTML.
- `<head></head>`: Parte no visible de la página. Donde se debe incluir de manera obligatoria el título de la misma (`<title> </title>`). Se ponen metadatos, configuraciones, enlaces al estilo...
- `<body></body>`: Parte visible de la página. Será el contenedor o caja que incluya el resto de cajas que conformarán la estructura HTML.

## 6. Head



```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

En el head de una página web se incluye información de la página o de soporte. Se suelen incluir las siguientes etiquetas:

- **Título:** Esta etiqueta es obligatoria, ya que especificará el título de la página que aparecerá en la pestaña del navegador.
  - `<title>Desarrollo web en entorno cliente</title>`
- **Metainformación:** Aporta información al navegador:

Atributo **name**: (Tantas etiquetas meta como valores de name se quieran usar)

- **viewport** (área disponible en la pantalla del navegador)
  - `<meta name="viewport" content="width=device-width", initial-scale=1.0">`



- width=device-width: viewport igual a anchura real de pantalla del dispositivo
- initial-scale=1: No se hace zoom

- **author** (autor de la página)
  - `<meta name="author" content="Damian" />`
- **keywords** (palabras clave de la página)
  - `<meta name="keywords" lang="es" content="instituto, cadiz, DAW" />`
- **description** (descripción de la página)
  - `<meta name="description" content="Página de ejemplo" />`

Atributo **charset** → Definición el conjunto de caracteres de la página

- HTML5 → `<meta charset = "UTF-8">`

Atributo **http-equiv** → Representa directivas y tipos

- `<meta http-equiv="X-UA-Compatible" content="IE=edge">`
- `<meta http-equiv="refresh" content="60; URL=http://www.htmlquick.com/">`
- HTML4.01 → `<meta http-equiv = "Content-Type" content="text/ html; charset = UTF-8">`
- **Enlaces:** link → Sirve para relacionar un documento con los recursos externos que utiliza, como por ejemplo la hoja de estilos aplicable al documento
  - `<link rel ="stylesheet" type="text/css" href="estilo.css"/>`

También se usa para añadir un icono a la pestaña de la página:

- `<link rel="icon" href="imagen.ico">`
- **Estilos; style** → Contiene a un bloque de declaraciones de estilo o CSS
  - `body {background: black; color:white;}`

## 7. Etiquetas

### • Títulos

Los títulos van del h1 al h6 (de mayor a menor). h1 debe ser único.

`<h1>Título1</h1>...`

### • Etiquetas de texto

- **Párrafo:** Ideal para trabajar con textos  
`<p>Cualquier texto</p>`
- **Negrita:** Pone el texto en negrita  
`<b>Negrita</b>` o `<strong>Negrita</strong>`
- **Itálica:** Pone el texto en cursiva  
`<i>Cursiva</i>` o `<em>Cursiva</em>`
- **Strike:** Pone el texto tachado  
`<strike>Tachado</strike>`

### • Enlaces de hipertexto

- **Vínculos:** Se usa para incluir enlaces tanto externos, internos y a la propia página.

`<a href=" ... ">Enlace</a>`

- Enlace externo:
  - No están accesibles de manera local: Debe usar el protocolos HTTP o HTTPS
- Enlace interno:
  - Estarán en nuestro servidor, de manera local. Hay que poner la ruta y el nombre del archivo.

```
<a href="https://google.com">Enlace</a>
```

```
<a href="pages/pagina1.html">Enlace</a>
```

(Recuerda: para volver una carpeta hacia arriba se usan dos puntos: ../index.html)

- Enlace a la propia página:
  - **Atributo name:**  
Con el name se crea un ancla en la página http://ejemplo.com/página1.html

```
<a name="primera_seccion"></a>
```

Para crear un enlace a esta sección se añade #:

```
<a href="http://ejemplo.com/página1.html#primera_seccion">Enlace</a>
```

- **Atributo id** (más recomendable):

También se pueden generar enlaces directos a secciones con el atributo id:

```
<h1 id="inicio">Titulo de la página</h1>
```

...

```
<a href="#inicio" target: >Volver al inicio de la página</a>
```

- Modos de abrir los enlaces:
  - **Atributo target:**
    - Para abrir en una nueva ventana: **target:"\_blank"**
    - Para abrir en el contexto padre inmediato: **target:"\_parent"**
    - Para abrir en el mismo contexto del vínculo: **target:"\_self"**
    - Para abrir en en el contexto más alto: **target:"\_top"**

- Enlaces a email, llamadas, sms y whatsapp

```
<a href="mailto:{email}?subject={subject}&body={content}">
  Send us an email
</a>

<a href="tel:{phone}">
  Call us
</a>

<a href="sms:{phone}?body={content}">
  Send us a message
</a>
```

fuelle:https://twitter.com/denicmarko/status/1429015087436664842

```
<a href="https://api.whatsapp.com/send?phone=34XXXXXXXXX&text=Hola">
  Enviar Whatsapp
</a>
```



- **Ruptura de línea**

- **br:** El elemento br inserta una ruptura de línea, causando que el texto continúe en una línea nueva. Este elemento debería ser utilizado únicamente para rupturas de línea que realmente forman parte del **flujo del texto**. Su utilización para separaciones temáticas o de presentación es inadecuada.

<br>

- **Listas**

- **Lista desordenada:** Son listas sin indicador de orden. Los elementos de la lista (li) no aparecen numerados.

<ul>

<li>Elemento1</li>

<li>Elemento2</li>

</ul>

- Elemento1
- Elemento2

- **Lista ordenada:** Son listas con indicador de orden. Los elementos de la lista (li) aparecen numerados.

<ol>

<li>Elemento1</li>

<li>Elemento2</li>

</ol>

1. Elemento1
2. Elemento2

- **Tablas**

Permiten a los desarrolladores representar información tabulada en un documento. En lo que respecta al código, una tabla consiste básicamente de un grupo de filas que contiene, cada una, a un grupo de celdas.

Un elemento table actúa únicamente como el contenedor donde otros elementos pueden ser declarados. Estos elementos son:

- **caption:** Provee un título para la tabla.
- **colgroup:** agrupa columnas temáticamente.
- **col:** aplica atributos comunes y estilos a una o más columnas. Sólo puede ser declarado como hijo de **colgroup**.
- **tbody:** representa a un cuerpo de la tabla.
- **thead:** representa la sección de encabezado de la tabla.
- **tfoot:** representa la sección de pie de la tabla.
- **tr:** representa una **fila**. Puede ser declarado como hijo de **tbody**, **thead**, **tfoot** o **table**.
- **td:** representa una **celda regular**. Sólo puede ser declarado como hijo de **tr**.
- **th:** representa una **celda de encabezado**. Debe ser declarado como hijo de **tr**.

## 8. Multimedia

- **Imágenes**

`` (etiqueta simple)

¿Qué problemas se pueden encontrar si cargamos una página con enlace externo?  
Que no cargue la imagen por algún motivo

Es importante tener en cuenta que las imágenes tienen un comportamiento **en línea** y que es "obligatorio" incluir los atributos src y alt (importante en SEO) :

- **Atributo alt:**  
`alt="texto descriptivo"`, texto que aparece si no se carga la imagen.
- **Atributo title:**  
`title="título"`, título de la imagen
- **Cambiar el tamaño:**
  - **Atributo width:** Cambia el ancho de la imagen
  - **Atributo height:** Cambia el alto de la imagen  
``

Ojo, no es recomendable utilizar ambas si no conocemos la **relación de aspecto**. Mejor modificar solo una.  
Además, si se especifica el tamaño en HTML, luego en CSS tendremos problemas a la hora de modificarlo, ya que por **jerarquía (especificidad)** solo **!important** estaría por encima de los estilos en línea como estos.
- Carga lazy: Retrasa la descarga de la imagen hasta que el usuario no hace scroll en ella:  
``
- **Imagen de fondo:**  
`<body background="URI">`

- **Vídeo**

`<video src="ruta_del_archivo_interna_o_externa" controls></video>`

- **Atributo src:** debe ser un archivo formato vídeo no un enlace a youtube, facebook...
- **Atributo controls:** es obligatorio, aunque esté vacío, para que el navegador muestre los controles necesarios para reproducir el vídeo

- **Audio**

`<audio src="ruta_del_archivo_interna_o_externa" controls></audio>`

- **Atributo src:** debe ser un archivo formato audio no un enlace a youtube, facebook...
- **Atributo controls:** es obligatorio, aunque esté vacío, para que el navegador muestre los controles necesarios para reproducir el vídeo



## 9. DIV vs SPAN

El elemento **DIV** es un contenedor genérico sin un significado semántico en particular.

Para escribir un HTML correcto a nivel semántico, se debe evitar trabajar con divs, ya que las etiquetas div se usan para separar y agrupar contenido pero no aporta información de su funcionalidad. Aunque aparentemente con HTML no parezca demasiado útil, será con CSS cuando se alcance la máxima funcionalidad de estas etiquetas.

```
<div id="rojo" style="background-color: red;">
  <h1> Titulo en contenedor rojo</h1>
  <div id="azul" style="background-color: blue;">
    <p> Este es un párrafo dentro de un contenedor div con fondo rojo y que a su vez esta
    dentro de otro contenedor div con fondo azul</p>
  </div>
  <br>
</div>
<br>
<div id="verde" style="background-color: green;">
  <h2> Subtitulo en contenedor verde</h2>
  <p> Este es otro párrafo que esta dentro de un contenedor div con fondo verde</p>
</div>
```

### Titulo en contenedor rojo

Este es un párrafo dentro de un contenedor div con fondo rojo y que a su vez esta dentro de otro contenedor div con fondo azul

### Subtitulo en contenedor verde

Este es otro párrafo que esta dentro de un contenedor div con fondo verde

**SPAN**, es un contenedor genérico para líneas de texto. El elemento span está diseñado para contener líneas simples de texto únicamente. Para bloques más grandes de contenido usa el elemento div en su lugar.

La principal diferencia entre ellos es que **div es un elemento en bloque y span es en línea**.

```
<h1>Pintando el arcoiris</h1>
<p>Chic@s, esta vez usaremos los siguientes colores:
  <span style="color: #ed1b24">rojo</span>, <span style="color: #d9cc00">amarillo</span>,
  <span style="color: #00a3e8">azul</span>, <span style="color: #a349a3">violeta</span>,
  <span style="color: #ff7f26">naranja</span>, <span style="color: #a1cc1b">verde</span> e
  <span style="color: #5701ae">indigo</span>.
</p>
```

## Pintando el arcoíris

Chicos, esta vez usaremos los siguientes colores: rojo, amarillo, azul, violeta, naranja, verde e indigo.

## 10. FORMULARIOS

**<form>...</form>**

HTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

Los usuarios completan los distintos campos del formulario antes de enviarlos a un agente para que lo procese, por ejemplo, a un servidor web.

De todos sus atributos, dos tienen mayor importancia y son habitualmente utilizados: **action** y **method**.



- **Atributo action:** indica la ubicación del programa que procesará la información enviada por el navegador, cuando el formulario sea despachado. Este programa está escrito en cualquier lenguaje del lado servidor y se supone que está preparado para recibir y procesar los datos del formulario.
- **Atributo method:** indica cómo debe el navegador adjuntar los datos del formulario a la petición. Es recomendable usar el método "POST" debido a que éste oculta la información enviada y permite la transmisión de datos binarios. Sin embargo, algunas situaciones específicas pueden requerir el uso del método GET (visible en la url - ?).

`<form action="../../../form-result.php" method="post" target="_blank">`

Dentro de la etiqueta form, se ponen los **controles**, cuyas etiquetas se nombran a continuación:

- **<input>**

Control de entrada. Se usará para introducir la entrada de datos.

Atributos:

- **type=""**; con los siguientes posibles valores (hay más: [aquí](#)):

- [text](#)
- [password](#)
- [checkbox](#)
- [radio](#)
- [submit](#)
- [reset](#)
- [file](#)
- [hidden](#)
- [image](#)
- [button](#)
- [search](#)
- [range](#)
- [color](#)

Dependiendo del valor se podrán utilizar otros atributos.

- **name="texto"**, servirá para identificar los inputs en el backend
- **value="texto"**, **valor del control. Es lo que la info que se usa**
- **size="número"**, ancho del control en número de caracteres
- **maxlength="número"**, longitud máxima de caracteres que se puede teclear (text y password)
- **checked="checked"**, indica si el control está marcado (radio y checkbox)
- **disable="disable"**, deshabilita el control, no se envía
- **readonly="readonly"**, solo lectura (text o password)
- **src="url"**, ruta de la imagen que actuará como botón de envío)
- **placeholder="texto"**, introduce leyenda dentro del input hasta que se escribe en él.
- **autofocus="autofocus"**
- **required="required"**

### ■ <select>

El elemento [select](#) presenta un control que permite al usuario elegir entre un número de opciones. Las opciones en una lista select son provistas con el [elemento option](#) y pueden ser agrupadas con el [elemento optgroup](#).

### ■ <textarea>

Representa un campo para la entrada de texto multilínea. El control asociado a este campo es una caja de texto que permite a los usuarios editar múltiples líneas de texto regular. Los controles textarea son útiles para recolectar o editar líneas largas de texto como mensajes, contenido de archivos, listas, reseñas, artículos, etc.

```
<form action="../../../form-result.php" target="_blank">
  <p>Edita tu comentario:</p>

  <p>
    <textarea name="comment">Hola! Creo que estás completamente equivocado!</textarea>
  </p>

  <p>
    <input type="submit" value="Editar">
    <input type="reset" value="Restaurar">
  </p>
</form>
```

Edita tu comentario:

Hola! Creo que  
estás completamente

EditarRestaurar

### ■ <label>

Etiqueta de un control. El elemento [label](#) representa una etiqueta que puede ser asociada a un control de formulario, y que se supone provee una descripción corta para éste. Los navegadores pueden enlazar ambos elementos permitiendo que los usuarios **establezcan el enfoque sobre el control al hacer clic en su etiqueta**.