

# Introducción a HTML

---

## ¿Qué es HTML?

HTML es el lenguaje de marcado estándar para crear páginas Web.

- HTML significa Hyper Text Markup Language
- HTML describe la estructura de las páginas Web mediante marcas
- Los elementos HTML son los bloques de construcción de las páginas HTML
- Los elementos HTML están representados por etiquetas
- Las etiquetas HTML etiquetan partes de contenido como "encabezado", "párrafo", "tabla", etc.
- Los navegadores no muestran las etiquetas HTML, pero las utilizan para procesar el contenido de la página

---

## Un documento HTML simple

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Ejemplo explicado:

- La declaración `<!DOCTYPE html>` define este documento como HTML5
- El elemento `<html>` es el elemento raíz de una página HTML
- El elemento `<head>` contiene información meta sobre el documento
- El elemento `<title>` especifica un título para el documento
- El elemento `<body>` contiene el contenido visible de la página
- El elemento `<h1>` define un encabezado grande

- El elemento <p> define un párrafo

---

## Etiquetas HTML

Las etiquetas HTML son nombres de elementos rodeados por corchetes angulares:

<tagname> el contenido va aquí ... </ tagname>

- Las etiquetas HTML normalmente vienen en parejas como <p> y </p>
- La primera etiqueta de un par es la etiqueta de inicio, la segunda etiqueta es la etiqueta de fin
- La etiqueta de fin se escribe como la etiqueta de inicio, pero con una barra diagonal insertada antes del nombre de la etiqueta

Sugerencia: La etiqueta de inicio también se denomina etiqueta de apertura y la etiqueta de fin la etiqueta de cierre.

---

## Navegadores web

El propósito de un navegador web (Chrome, IE, Firefox, Safari) es leer documentos HTML y mostrarlos.

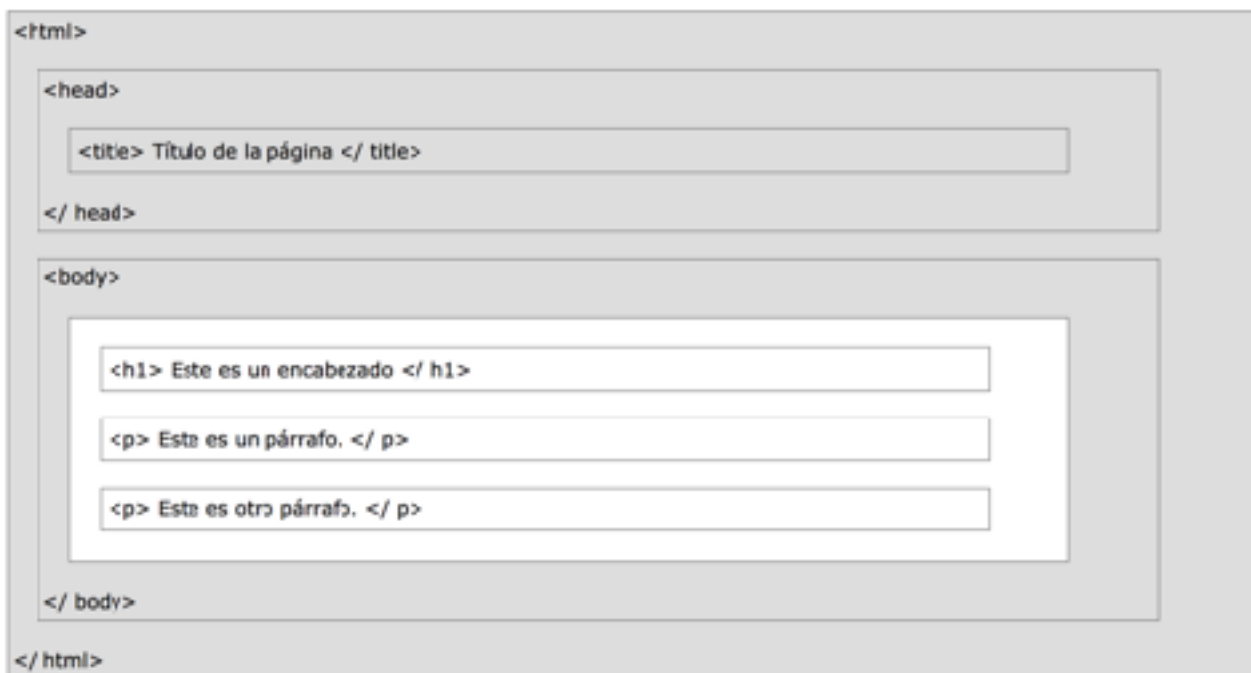
El navegador no muestra las etiquetas HTML, pero las utiliza para determinar cómo mostrar el documento:



---

## Estructura de la página HTML

A continuación se muestra una visualización de una estructura de página HTML:



**Nota:** Sólo el contenido dentro de la sección `<body>` (el área blanca arriba) se muestra en un navegador.

---

## La declaración `<!DOCTYPE>`

La declaración `<!DOCTYPE>` representa el tipo de documento y ayuda a los navegadores a mostrar correctamente las páginas web.

Solo debe aparecer una vez, en la parte superior de la página (antes de las etiquetas HTML).

La declaración `<!DOCTYPE>` no distingue entre mayúsculas y minúsculas.

La declaración `<!DOCTYPE>` para HTML5 es:

```
<!DOCTYPE html>
```

---

## Versiones HTML

Desde los primeros días de la web, ha habido muchas versiones de HTML

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

## Ejemplos básicos de HTML

---

### Documentos HTML

Todos los documentos HTML deben comenzar con una declaración de tipo de documento:  
<!DOCTYPE html> .

El documento HTML comienza con <html> y termina con </html> .

La parte visible del documento HTML está entre <body> y </body> .

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>

<h1>Mi primer encabezado</h1>
<p>Mi primer párrafo.</p>

</body>
</html>
```

---

## Encabezados HTML

Los encabezados HTML se definen con las etiquetas <h1> a <h6> .

<h1> define el encabezado más importante. <h6> define el encabezado menos importante:

Ejemplo:

```
<h1>Esto es un encabezado 1</h1>  
<h2>Esto es un encabezado 2</h2>  
<h3>Esto es un encabezado 3</h3>
```

---

## Párrafos HTML

Los párrafos HTML se definen con la etiqueta <p> :

Ejemplo:

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

---

## Enlaces HTML

Los enlaces HTML se definen con la etiqueta <a> :

Ejemplo:

```
<a href="https://www.w3schools.com">Esto es un enlace</a>
```

El destino del enlace se especifica en el atributo href .

Los atributos se utilizan para proporcionar información adicional sobre elementos HTML.

---

## Imágenes HTML

Las imágenes HTML se definen con la etiqueta `<img>` .

El archivo fuente (`src`), el texto alternativo (`alt`), el ancho y la altura se proporcionan como atributos:

Ejemplo:

```

```

## Elementos HTML

Un elemento HTML normalmente consta de una etiqueta de inicio y una etiqueta de fin , con el contenido insertado entre:

```
< tagname > El contenido va aquí ... < / tagname >
```

El elemento HTML es todo, desde la etiqueta de inicio hasta la etiqueta de fin:

```
< p > Mi primer párrafo. < / p >
```

Los elementos HTML sin contenido se denominan elementos vacíos. Elementos vacíos no tienen una etiqueta de finalización, como el elemento `<br>` (que indica un salto de línea).

---

## Elementos HTML anidados

Los elementos HTML pueden anidarse (los elementos pueden contener elementos).

Todos los documentos HTML constan de elementos HTML anidados.

Este ejemplo contiene cuatro elementos HTML:

Ejemplo:

```
<!DOCTYPE html>
```

```
<html>
<body>

<h1>Mi primer encabezado</h1>
<p>Mi primer párrafo.</p>

</body>
</html>
```

Ejemplo explicado:

El elemento <html> define todo el documento .

Tiene una etiqueta de inicio <html> y una etiqueta de fin </ html>.

El contenido del elemento es otro elemento HTML (el elemento <body>).

```
<html>
<body>

<h1>Mi primer encabezado</h1>
<p>Mi primer párrafo.</p>

</body>
</html>
```

El elemento <body> define el cuerpo del documento .

Tiene una etiqueta de inicio <body> y una etiqueta de fin </ body>.

El contenido del elemento es dos otros elementos HTML (<h1> y <p>).

```
<body>

<h1>Mi primer encabezado</h1>
<p>Mi primer párrafo.</p>

</body>
```

El elemento <h1> define un encabezado .

Tiene una etiqueta de inicio <h1> y una etiqueta de fin </ h1>.

El contenido del elemento es: Mi primer encabezado.

```
<h1>Mi primer encabezado</h1>
```

El elemento `<p>` define un párrafo .

Tiene una etiqueta de inicio `<p>` y una etiqueta de fin `</p>`.

El contenido del elemento es: Mi primer párrafo.

```
<p>Mi primer párrafo.</p>
```

No te olvides de la etiqueta final

Algunos elementos HTML se mostrarán correctamente, incluso si se olvida la etiqueta de fin:

Ejemplo:

```
<html>
<body>

<p>Esto es un párrafo
<p>Esto es un párrafo

</body>
</html>
```

El ejemplo anterior funciona en todos los navegadores, ya que la etiqueta de cierre se considera opcional.

Nunca confíes en esto. Puede producir resultados inesperados y / o errores si olvidas la etiqueta final.

---

## Elementos HTML vacíos

Los elementos HTML sin contenido se denominan elementos vacíos.

`<br>` es un elemento vacío sin una etiqueta de cierre (la etiqueta `<br>` define un salto de línea).

Los elementos vacíos pueden ser "cerrados" en la etiqueta de apertura como esto: `<br />`.



HTML5 no requiere que los elementos vacíos se cierren. Pero si deseas una validación más estricta, o si necesitas hacer tu documento legible por analizadores XML, debes cerrar todos los elementos HTML correctamente.

Utilizar etiquetas en minúsculas

Las etiquetas HTML no distinguen entre mayúsculas y minúsculas: `<P>` significa lo mismo que `<p>`.

El estándar HTML5 no requiere etiquetas en minúsculas, pero W3C recomienda minúsculas en HTML y exige minúsculas para tipos de documentos más estrictos como XHTML.

## Atributos HTML

Los atributos proporcionan información adicional sobre elementos HTML.

Atributos HTML:

- Todos los elementos HTML pueden tener atributos
- Los atributos proporcionan información adicional acerca de un elemento
- Los atributos siempre se especifican en la etiqueta de inicio
- Los atributos normalmente vienen en pares nombre / valor como: `name = "value"`

---

### El atributo lang

El idioma del documento se puede declarar en la etiqueta `<html>` .

El lenguaje se declara con el atributo `lang` .

Declarar un idioma es importante para las aplicaciones de accesibilidad (lectores de pantalla) y los motores de búsqueda:

```
<!DOCTYPE html>
<html lang="es-ES">
<body>

...

</body>
</html>
```

Las dos primeras letras especifican el idioma (es). Si hay un dialecto, use dos letras más (ES).

---

## El atributo title

Aquí, un atributo de título se añade al elemento `<p>` . El valor del atributo de título se mostrará como una información de herramientas al pasar el mouse sobre el párrafo:

Ejemplo:

```
<p title="I'm a tooltip">  
Esto es un párrafo.  
</p>
```

---

## El atributo href

Los enlaces HTML se definen con la etiqueta `<a>` . La dirección de enlace se especifica en el atributo href :

Ejemplo

```
<a href="https://www.w3schools.com">Esto es un enlace</a>
```

Más adelante aprenderás más sobre los vínculos y la etiqueta `<a>`.

---

## Atributos de tamaño

Las imágenes HTML se definen con la etiqueta `<img>` .

El nombre de archivo del origen ( `src` ) y el tamaño de la imagen ( ancho y alto ) se proporcionan como atributos :

Ejemplo:

```

```

El tamaño de la imagen se especifica en píxeles: width = "104" significa 104 píxeles de pantalla de ancho.

Más adelante aprenderás más sobre las imágenes y la etiqueta <img>.

---

## El atributo alt

El atributo alt especifica un texto alternativo que se utilizará cuando no se pueda mostrar una imagen.

El lector de pantalla puede leer el valor del atributo. De esta manera, alguien que "escucha" la página web, por ejemplo, una persona ciega, puede "escuchar" el elemento.

Ejemplo:

```

```

---

## Sugerencia: Utilizar atributos en minúsculas

El estándar HTML5 no requiere nombres de atributos en minúsculas.

El atributo title puede escribirse con mayúsculas o minúsculas como title o TITLE .

W3C recomienda minúsculas en HTML y exige minúsculas para tipos de documentos más estrictos como XHTML.

---

## Sugerencia: Valores de los atributos en comillas

El estándar HTML5 no requiere comillas alrededor de valores de atributo.

El atributo href , demostrado anteriormente, puede escribirse como:

Ejemplo:

```
<a href=https://www.w3schools.com>
```

W3C recomienda cotizaciones en HTML y exige comillas para tipos de documentos más estrictos como XHTML.

A veces es necesario usar comillas. Este ejemplo no mostrará correctamente el atributo title, ya que contiene un espacio:

Ejemplo:

```
<p title=About W3Schools>
```

El uso de comillas es más común. Omitir comillas puede producir errores.

---

## Comillas simples o dobles?

Las comillas dobles alrededor de los valores de atributo son las más comunes en HTML, pero también se pueden usar comillas simples.

En algunas situaciones, cuando el valor del atributo en sí contiene comillas dobles, es necesario utilizar comillas simples:

```
<p title='John "ShotGun" Nelson'>
```

O viceversa:

```
<p title="John 'ShotGun' Nelson">
```

Resumen:

- Todos los elementos HTML pueden tener atributos
- El atributo title proporciona información adicional de "tool-tip"
- El atributo href proporciona información de direcciones para enlaces
- Los atributos width y height proporcionan información sobre el tamaño de las

imágenes

- El atributo alt proporciona texto para los lectores de pantalla
- Siempre nombres de atributos en minúsculas
- Siempre comillas dobles en los atributos.

## Entidades HTML

Los caracteres reservados en HTML deben reemplazarse por entidades de caracteres.

Los caracteres que no están presentes en el teclado también pueden ser reemplazados por entidades.

---

## Entidades HTML

Algunos caracteres están reservados en HTML.

Si usa los signos menos de (<) o mayores que (>) en su texto, el navegador puede mezclarlos con etiquetas.

Las entidades de caracteres se utilizan para mostrar caracteres reservados en HTML.

Una entidad de caracteres se ve así:

&entity\_name;  
OR  
&#entity\_number;

Para mostrar un signo menor que (<) debemos escribir: &lt; o & #60;

Ventaja de usar un nombre de entidad: Un nombre de entidad es fácil de recordar.

Desventaja de usar un nombre de entidad: Los navegadores pueden no admitir todos los nombres de entidad, pero el soporte para números es bueno.

---

## Espacio irrompible

Una entidad de caracteres común utilizada en HTML es el espacio que no se rompe: &nbsp;

Un espacio que no se rompe es un espacio que no entrará en una nueva línea.

Dos palabras separadas por un espacio que no se rompe se pegarán juntas (no se rompen en una nueva línea). Esto es útil cuando romper las palabras puede ser perjudicial.

Ejemplos:

- § 10
- 10 km / h
- 10 PM

Otro uso común del espacio sin interrupciones es evitar que los navegadores trunquen espacios en páginas HTML.

Si escribe 10 espacios en el texto, el navegador eliminará 9 de ellos. Para añadir espacios reales a su texto, puede utilizar los &nbsp; entidad de caracteres.

El guión no rompiente ( &#8209; ) le permite usar un carácter de guión (-) que no se rompa.

---

## Algunas otras entidades de caracteres HTML útiles

Result	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	double quotation mark	&quot;	&#34;
'	single quotation mark (apostrophe)	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;

## Símbolos HTML

---

### Entidades de símbolos HTML

Las entidades HTML se describieron en el capítulo anterior.

Muchos símbolos matemáticos, técnicos y de moneda no están presentes en un teclado normal.

Para agregar estos símbolos a una página HTML, puede utilizar un nombre de entidad HTML.

Si no existe ningún nombre de entidad, puede utilizar un número de entidad, un decimal o una referencia hexadecimal.

Ejemplo:

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

Importante: las vocales españolas con tilde se escriben como &acute; &eacute; &iacute; &oacute; y &uacute;

# Codificación HTML (Conjuntos de caracteres)

Para mostrar una página HTML correctamente, un navegador web debe saber qué juego de caracteres (codificación de caracteres) debe utilizar.

---

## ¿Qué es la codificación de caracteres?

ASCII fue el primer estándar de codificación de caracteres (también llamado conjunto de caracteres). ASCII define 128 caracteres alfanuméricos diferentes que pueden utilizarse en Internet: números (0-9), letras en inglés (AZ) y algunos caracteres especiales como! \$ + - () @ <>.

ANSI (Windows-1252) fue el juego de caracteres original de Windows, con soporte para 256 códigos de caracteres diferentes.

ISO-8859-1 fue el conjunto de caracteres predeterminado para HTML 4. Este conjunto de caracteres también soporta 256 códigos de caracteres diferentes.

Debido a que ANSI y ISO-8859-1 eran tan limitados, HTML 4 también soporta UTF-8.

UTF-8 (Unicode) cubre casi todos los personajes y símbolos del mundo.

La codificación de caracteres predeterminada para HTML5 es UTF-8.

---

## El atributo de conjunto de caracteres HTML

Para mostrar una página HTML correctamente, un navegador web debe conocer el conjunto de caracteres utilizado en la página.

Esto se especifica en la etiqueta <meta>:

Para HTML4:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Para HTML5:

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

Si un navegador detecta ISO-8859-1 en una página web, el valor predeterminado es ANSI, porque ANSI es idéntico a ISO-8859-1, excepto que ANSI tiene 32 caracteres adicionales.

# Encabezados HTML

Los encabezados son importantes en los documentos HTML.

Los encabezados se definen con las etiquetas `<h1>` a `<h6>`.

`<h1>` define el encabezado más importante. `<h6>` define el encabezado menos importante.

Ejemplo:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

Nota: Los navegadores añaden automáticamente un espacio en blanco (un margen) antes y después de un encabezado.

Los encabezados son importantes

Los motores de búsqueda utilizan los encabezados para indexar la estructura y el contenido de sus páginas web.

Los usuarios navegan por sus páginas por sus encabezados. Es importante utilizar encabezados para mostrar la estructura del documento.

Los encabezados `<h1>` deben usarse para los encabezamientos principales, seguidos por los encabezados `<h2>`, luego los menos importantes `<h3>`, etc.

Nota: Utiliza encabezados HTML sólo para encabezados. No utilices encabezados para hacer texto GRANDE o en negrita .

---

## Reglas horizontales HTML

La etiqueta `<hr>` define una ruptura temática en una página HTML y se muestra con mayor frecuencia como una regla horizontal.

El elemento `<hr>` se utiliza para separar el contenido (o definir un cambio) en una página HTML:

Ejemplo:



```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

---

## El elemento HTML <head>

El elemento HTML <head> no tiene nada que ver con los encabezados HTML.

El elemento <head> es un contenedor para metadatos. Los metadatos HTML son datos sobre el documento HTML. Los metadatos no se muestran.

El elemento <head> se coloca entre la etiqueta <html> y la etiqueta <body>:

Ejemplo:

```
<!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>

<body>
  .
  .
  .
```

Nota: Los metadatos suelen definir el título del documento, el conjunto de caracteres, los estilos, los vínculos, los scripts y otra meta información.

---

## ¿Cómo ver la fuente del HTML?

¿Alguna vez has visto una página web y te has preguntado "¿Cómo lo hicieron?"

### **Ver código fuente HTML:**

Para averiguarlo, haz clic con el botón derecho en la página y selecciona "Ver origen de página" (en Chrome) o "Ver código fuente" (en IE), o similar en otros navegadores. Esto abrirá una ventana que contiene el código fuente HTML de la página.

### **Inspeccionar un elemento HTML:**

Haz clic con el botón derecho en un elemento (o un área en blanco) y seleccione "Inspeccionar" o "Inspeccionar elemento" para ver qué elementos están compuestos (verá tanto el HTML como el CSS). También puede editar el HTML o CSS al vuelo en el panel Elementos o Estilos que se abre.

## **Párrafos HTML**

El elemento HTML `<p>` define un párrafo :

Ejemplo:

```
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>
```

Nota: Los navegadores añaden automáticamente un espacio en blanco (un margen) antes y después de un párrafo.

---

## **Pantallas y HTML**

No puedes estar seguro de cómo se mostrará HTML.

Las pantallas grandes o pequeñas y las ventanas redimensionadas crearán diferentes resultados. Con HTML, no puede cambiar la salida agregando espacios adicionales o líneas adicionales en su código HTML.

El navegador eliminará los espacios adicionales y las líneas adicionales cuando se muestre la página:

Ejemplo:

```
<p>
```

This paragraph  
contains a lot of lines  
in the source code,  
but the browser  
ignores it.  
</p>

<p>  
This paragraph  
contains        a lot of spaces  
in the source        code,  
but the        browser  
ignores it.  
</p>

No te olvides de la etiqueta final.

La mayoría de los navegadores mostrarán HTML correctamente incluso si se te olvida la etiqueta de fin.

Ejemplo:

<p>This is a paragraph.  
<p>This is another paragraph.

El ejemplo anterior funcionará en la mayoría de los navegadores, pero no confíe en él.

Nota: Eliminar la etiqueta de fin puede producir resultados o errores inesperados.

---

## Saltos de línea HTML

El elemento HTML <br> define un salto de línea .

Use <br> si desea un salto de línea (una nueva línea) sin iniciar un nuevo párrafo:

Ejemplo:

<p>This is<br>a paragraph<br>with line breaks.</p>

La etiqueta <br> es una etiqueta vacía, lo que significa que no tiene etiqueta de final.

---

## El problema del poema

Este poema se mostrará en una sola línea:

Ejemplo:

```
<p>
```

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

```
</p>
```

---

## El elemento HTML <pre>

El elemento HTML <pre> define el texto preformateado.

El texto dentro de un elemento <pre> se muestra en una fuente de anchura fija (normalmente Courier) y conserva espacios y saltos de línea:

Ejemplo:

```
<pre>
```

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

```
</pre>
```

# Estilos HTML

---

## El atributo de estilo HTML

Establecer el estilo de un elemento HTML, se puede hacer con el atributo de estilo .

El atributo de estilo HTML tiene la siguiente sintaxis :

```
<tagname style="property:value;">
```

La propiedad es una propiedad CSS. El valor es un valor CSS.

---

## Color de fondo HTML

La propiedad background-color define el color de fondo para un elemento HTML.

Este ejemplo establece el color de fondo de una página en powderblue:

Ejemplo:

```
<body style="background-color:powderblue;">
```

```
<h1>This is a heading</h1>
```

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

```
</body>
```

---

## Color del texto HTML

La propiedad de color define el color de texto de un elemento HTML:

Ejemplo:

```
<h1 style="color:blue;">This is a heading</h1>
```

```
<p style="color:red;">This is a paragraph.</p>
```

---

## Fuentes HTML

La propiedad font-family define la fuente a utilizar para un elemento HTML:

Ejemplo:

```
<h1 style="font-family:verdana;">This is a heading</h1>  
<p style="font-family:courier;">This is a paragraph.</p>
```

---

## Tamaño del texto HTML

La propiedad font-size define el tamaño de texto para un elemento HTML:

Ejemplo:

```
<h1 style="font-size:300%;">This is a heading</h1>  
<p style="font-size:160%;">This is a paragraph.</p>
```

---

## Alineación de texto HTML

La propiedad text-align define la alineación horizontal del texto para un elemento HTML:

Ejemplo:

```
<h1 style="text-align:center;">Centered Heading</h1>  
<p style="text-align:center;">Centered paragraph.</p>
```

## Formato de texto HTML

HTML también define elementos especiales para definir texto con un significado especial .

HTML utiliza elementos como <b> y <i> para dar formato a la salida, como texto en negrita o en cursiva .

Los elementos de formato fueron diseñados para mostrar tipos especiales de texto:

- <b> - Texto en negrita
- <strong> - Texto importante

- <i> - Texto en cursiva
- <em> - Texto enfatizado
- <mark> - Texto marcado
- <small> - Texto pequeño
- <del> - Texto eliminado
- <ins> - Texto insertado
- <sub> - Texto de subíndice
- <sup> - Texto superíndice

---

## Elementos HTML <b> y <strong>

El elemento HTML <b> define el texto en negrita , sin ninguna importancia extra.

Ejemplo:

```
<b>This text is bold</b>
```

El elemento HTML <strong> define el texto fuerte , con una mayor importancia semántica.

Ejemplo:

```
<strong>This text is strong</strong>
```

---

## Elementos HTML <i> y <em>

El elemento HTML <i> define el texto en cursiva , sin ninguna importancia extra.

Ejemplo:

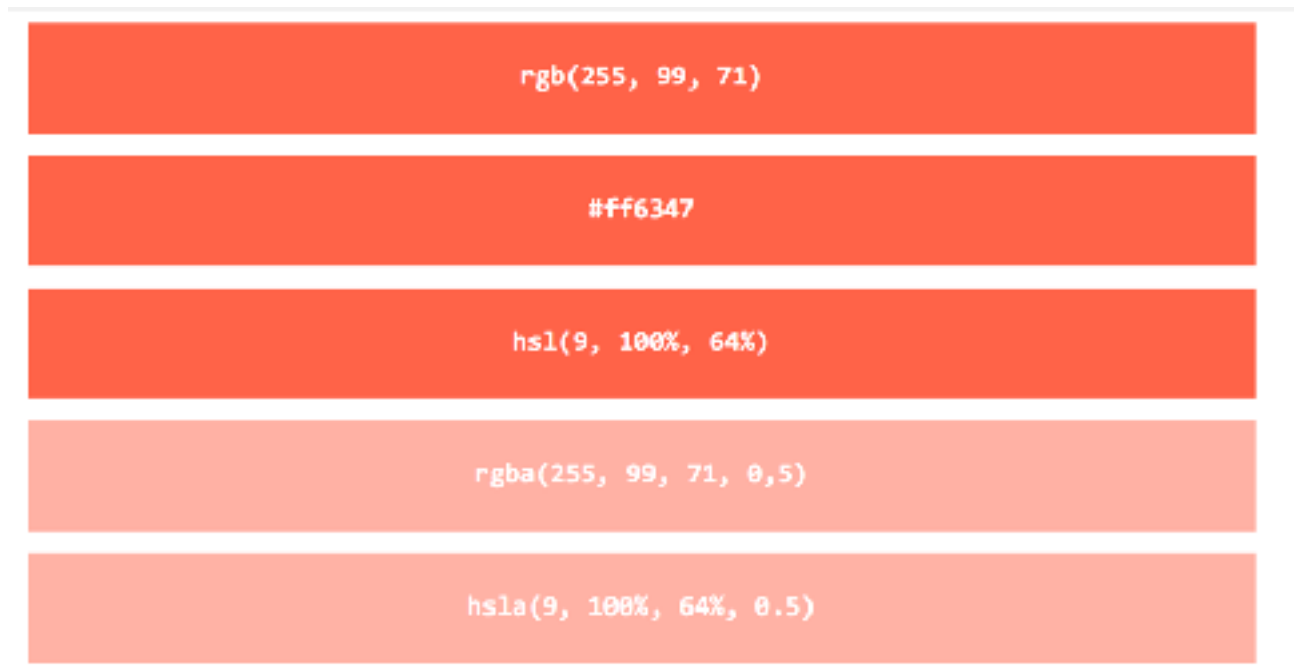
```
<i>This text is italic</i>
```

El elemento HTML <em> define el texto enfatizado , con una importancia semántica añadida.

Ejemplo:

```
<em>This text is emphasized</em>
```

Nota: Los navegadores muestran `<strong>` como `<b>` y `<em>` como `<i>`. Sin embargo, hay una diferencia en el significado de estas etiquetas: `<b>` y `<i>` definen el texto en negrita y en cursiva, pero `<strong>` y `<em>` significan que el texto es "importante".



---

## Elemento HTML `<small>`

El elemento HTML `<small>` define un texto más pequeño :

Ejemplo:

```
<h2>HTML <small>Small</small> Formatting</h2>
```

I

---

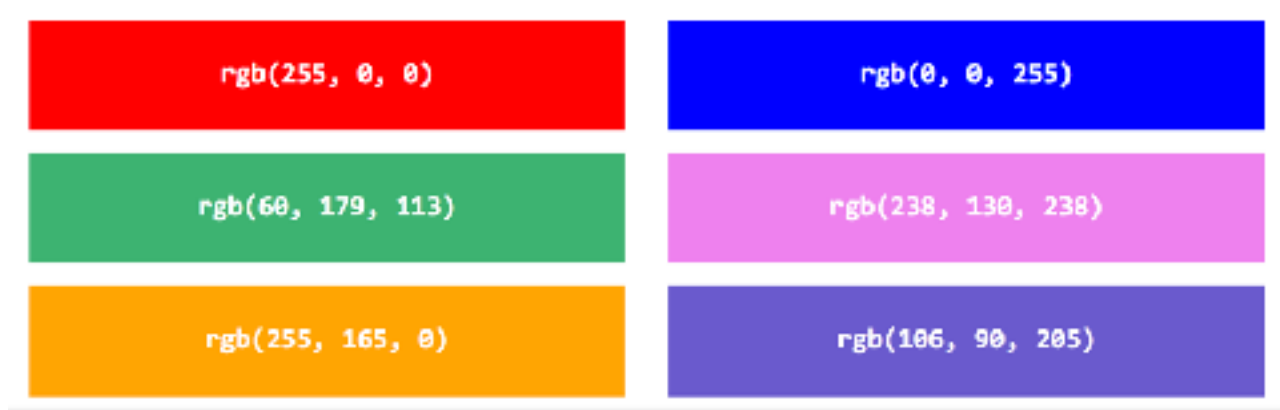
## Elemento HTML `<mark>`

El elemento HTML `<mark>` define texto marcado o destacado :

Ejemplo:

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```





---

## Elemento HTML <del>

El elemento HTML <del> define suprimido (eliminado).

Ejemplo:

```
<p>My favorite color is <del>blue</del> red.</p>
```

---

## Elemento HTML <ins>

El elemento HTML <ins> define insertado (añadido).

Ejemplo:

```
<p>My favorite <ins>color</ins> is red.</p>
```

---

## Elemento HTML <sub>

El elemento HTML <sub> define el texto con subíndices .

Ejemplo:

```
<p>This is <sub>subscripted</sub> text.</p>
```

---

## Elemento HTML <sup>

El elemento HTML <sup> define el texto superíndice .

Ejemplo:

```
<p>This is <sup>superscripted</sup> text.</p>
```

## Elementos de cita HTML

---

### Elemento HTML <q> para citas cortas

El elemento HTML <q> define una cita corta.

Normalmente, los navegadores insertan comillas alrededor del elemento <q>.

Ejemplo:

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

---

### Elemento HTML <blockquote> para citas

El elemento HTML <blockquote> define una sección que se cita de otra fuente.

Los navegadores suelen sangrar elementos <blockquote>.

Ejemplo:

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

---

### Elemento HTML <abbr> para abreviaturas

El elemento HTML <abbr> define una abreviatura o un acrónimo.

Marcando abreviaturas puedes dar información útil a los navegadores, sistemas de traducción y motores de búsqueda.

Ejemplo:

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

---

## Elemento HTML <address> para información de contacto

El elemento HTML <address> define la información de contacto (autor / propietario) de un documento o un artículo.

El elemento <address> suele aparecer en cursiva. La mayoría de los navegadores añadirán un salto de línea antes y después del elemento.

Ejemplo:

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

---

## Elemento HTML <cite> para el título de trabajo

El elemento HTML <cite> define el título de una obra.

Los navegadores suelen mostrar elementos <cite> en cursiva.

Ejemplo

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

---

## Elemento HTML <bdo> para la sustitución bidireccional

El elemento HTML <bdo> define la anulación bi-direccional.

El elemento <bdo> se utiliza para anular la dirección actual del texto:

Ejemplo:

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```

## Comentarios HTML

Puedes agregar comentarios a tu HTML utilizando la siguiente sintaxis:

```
<!-- Write your comments here -->
```

Observa que hay un signo de exclamación (!) En la etiqueta de apertura, pero no en la etiqueta de cierre.

Nota: los comentarios no se muestran en el navegador, pero pueden ayudar a documentar el código fuente HTML.

Con comentarios puedes colocar notificaciones y recordatorios en tu HTML:

Ejemplo:

```
<!-- This is a comment -->
```

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

```
<!-- Remember to add more information here -->
```

Los comentarios también son buenos para depurar HTML, ya que puedes comentar las líneas HTML de código, una a la vez, para buscar errores:

Ejemplo:

```
<!-- Do not display this at the moment
```

```

```

```
-->
```

---

## Comentarios condicionales

Puedes tropezar con comentarios condicionales en HTML:

```
<!--[if IE 9]>
```

```
.... some HTML here ....
```

```
<![endif]-->
```

Los comentarios condicionales definen algunas etiquetas HTML que deben ejecutar únicamente Internet Explorer.

## Colores HTML

Los colores HTML se pueden especificar utilizando nombres de color predefinidos, valores RGB, HEX, HSL, RGBA y HSLA.

---

### Nombres de color

En HTML, se puede especificar un color utilizando un nombre de color:



---

### Valores de color

En HTML, los colores también se pueden especificar mediante valores RGB, valores HEX, valores HSL, valores RGBA y valores HSLA:

---

### Valor RGB

En HTML, se puede especificar un color como un valor RGB, usando esta fórmula:

```
rgb ( rojo, verde , azul )
```

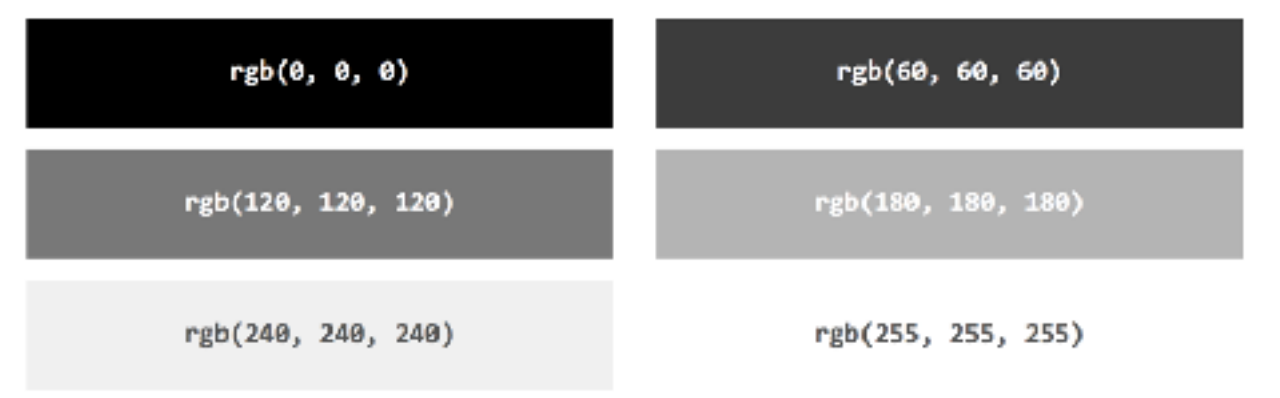
Cada parámetro (rojo, verde y azul) define la intensidad del color entre 0 y 255.

Por ejemplo, `rgb(255, 0, 0)` se muestra en rojo, porque el rojo se establece en su valor más alto (255) y los demás se ponen a 0.

Para mostrar el color negro, todos los parámetros de color se deben establecer en 0, como esto: `rgb(0, 0, 0)`.

Para mostrar el color blanco, todos los parámetros de color deben ajustarse a 255, como esto: `rgb(255, 255, 255)`.

Los tonos de gris se definen a menudo con valores iguales para todas las 3 fuentes de luz:



---

## Valor HEX

En HTML, se puede especificar un color utilizando un valor hexadecimal en la forma:

# rrggbb

Donde rr (rojo), gg (verde) y bb (azul) son valores hexadecimales entre 00 y ff (igual que decimal 0-255).

Por ejemplo, `#ff0000` se muestra en rojo, porque el rojo se establece en su valor más alto (ff) y los demás se establecen en el valor más bajo (00).

Los tonos de gris se definen a menudo con valores iguales para todas las 3 fuentes de luz:

#000000	#3c3c3c
#787878	#b4b4b4
#f0f0f0	#ffffff

## Valor HSL

En HTML, se puede especificar un color usando matiz, saturación y ligereza (HSL) en la forma:

```
hsl ( tonalidad , saturación , ligereza )
```

Tono es un grado en la rueda de color de 0 a 360. 0 es rojo, 120 es verde y 240 es azul.

Saturación es un valor porcentual, 0% significa un tono de gris y 100% es el color completo.

La ligereza es también un porcentaje, el 0% es negro, el 50% no es ni claro ni oscuro, el 100% es blanco.

hsl(0, 100%, 50%)	hsl(240, 100%, 50%)
hsl(147, 50%, 47%)	hsl(300, 76%, 72%)
hsl(39, 100%, 50%)	hsl(248, 53%, 58%)

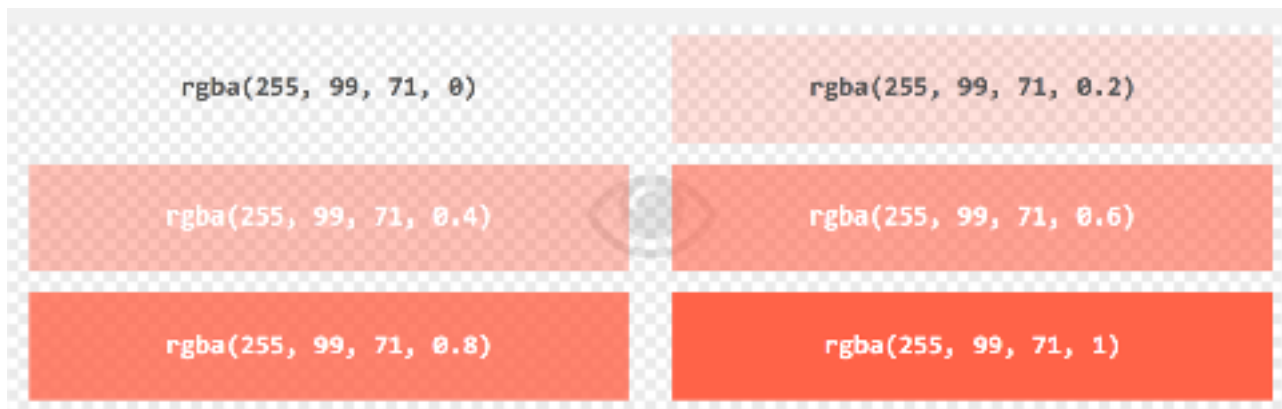
## Valor RGBA

Los valores de color RGBA son una extensión de valores de color RGB con un canal alfa, que especifica la opacidad de un color.

Se especifica un valor de color RGBA con:

rgba ( rojo, verde , azul, alfa )

El parámetro alfa es un número entre 0,0 (totalmente transparente) y 1,0 (no transparente en absoluto):



## Enlaces HTML

Los enlaces se encuentran en casi todas las páginas web. Los enlaces permiten a los usuarios hacer clic en su camino de una página a otra.

---

### Enlaces HTML - Hipervínculos

Los enlaces HTML son hipervínculos.

Puedes hacer clic en un enlace y saltar a otro documento.

Cuando mueves el ratón sobre un enlace, la flecha del ratón se convertirá en una pequeña mano.

Nota: Un enlace no tiene que ser texto. Puede ser una imagen o cualquier otro elemento HTML.

---

### Enlaces HTML - Sintaxis

En HTML, los vínculos se definen con la etiqueta `<a>` :

```
<a href="url">link text</a>
```

Ejemplo:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```



El atributo href especifica la dirección de destino (https://www.w3schools.com/html/) del enlace.

El texto del enlace es la parte visible (visite nuestro tutorial HTML).

Al hacer clic en el texto del enlace, se le enviará a la dirección especificada.

Nota: sin una barra inclinada en las direcciones de subcarpeta, puede generar dos peticiones al servidor. Muchos servidores agregarán automáticamente una barra diagonal a la dirección y, a continuación, crearán una nueva solicitud.

---

## Enlaces Locales

El ejemplo anterior utiliza una URL absoluta (una dirección web completa).

Un enlace local (enlace al mismo sitio web) se especifica con una URL relativa (sin http: // www ....).

Ejemplo:

```
<a href="html_images.asp">HTML Images</a>
```

---

## Enlaces HTML: el atributo de destino

El atributo de destino especifica dónde abrir el documento vinculado.

El atributo target puede tener uno de los siguientes valores:

- `_blank` - Abre el documento vinculado en una nueva ventana o pestaña
- `_self` - Abre el documento vinculado en la misma ventana / pestaña que se hizo clic

(esto es el predeterminado)

- `_parent` - Abre el documento vinculado en el marco padre
- `_top` - Abre el documento vinculado en todo el cuerpo de la ventana
- `framename` - Abre el documento vinculado en un marco con nombre

En este ejemplo se abrirá el documento vinculado en una nueva ventana / pestaña de explorador:

Ejemplo:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

---

## Enlaces HTML - Imagen como enlace

Es común utilizar imágenes como enlaces:

Ejemplo:

```
<a href="default.asp">  
    
</a>
```

Nota: border: 0; se agrega para evitar IE9 (y anterior) de mostrar un borde alrededor de la imagen (cuando la imagen es un enlace).

---

## Enlaces HTML - Crear un marcador

Marcadores HTML se utilizan para permitir a los lectores saltar a partes específicas de una página Web.

Los marcadores pueden ser útiles si su página web es muy larga.

Para crear un marcador, primero debe crear el marcador y, a continuación, agregar un vínculo al mismo.

Cuando se hace clic en el enlace, la página se desplazará a la ubicación con el marcador.

Ejemplo:

Primero, cree un marcador con el atributo id:

```
<h2 id="C4">Chapter 4</h2>
```

A continuación, agregue un enlace al marcador ("Saltar al Capítulo 4"), desde la misma página:

```
<a href="#C4">Jump to Chapter 4</a>
```

O bien, agregue un enlace al marcador ("Saltar al Capítulo 4"), desde otra página:

Ejemplo:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

---

## Rutas externas

Las páginas externas pueden referenciarse con una URL completa o con una ruta relativa a la página web actual.

En este ejemplo se utiliza una URL completa para enlazar a una página web:

Ejemplo:

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

Este ejemplo enlaza a una página ubicada en la carpeta html del sitio web actual:

Ejemplo:

```
<a href="/html/default.asp">HTML tutorial</a>
```

Este ejemplo enlaza a una página ubicada en la misma carpeta que la página actual:

Ejemplo:

```
<a href="default.asp">HTML tutorial</a>
```

## Imágenes HTML

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>

<h2>Spectacular Mountain</h2>


</body>
</html>
```

---

## Sintaxis de imágenes HTML

En HTML, las imágenes se definen con la etiqueta `<img>` .

La etiqueta <img> está vacía, sólo contiene atributos y no tiene una etiqueta de cierre.  
El atributo src especifica la URL (dirección web) de la imagen:

```

```

---

## El atributo alt

El atributo alt proporciona un texto alternativo para una imagen, si por alguna razón el usuario no puede verla (debido a una conexión lenta, un error en el atributo src o si el usuario usa un lector de pantalla).

Si un navegador no puede encontrar una imagen, mostrará el valor del atributo alt:

Ejemplo:

```

```

Se requiere el atributo alt. Una página web no se validará correctamente sin ella.

---

## Tamaño de la imagen - Anchura y altura

Puedes utilizar el atributo style para especificar el ancho y la altura de una imagen.

Los valores se especifican en píxeles (utilice px después del valor):

Ejemplo:

```

```

Como alternativa, puede utilizar los atributos width y height . En este caso, los valores se especifican en píxeles de forma predeterminada:

Ejemplo:

```

```

Nota: especifica siempre el ancho y la altura de una imagen. Si el ancho y la altura no se especifican, la página parpadeará mientras se carga la imagen.

---

## ¿Anchura y altura, o estilo?

Los atributos de anchura, altura y estilo son válidos en HTML5.

Sin embargo, sugerimos utilizar el atributo style. Evita que las hojas de estilos internas o externas cambien el tamaño original de las imágenes:

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  width:100%;
}
</style>
</head>
<body>




</body>
</html>
```

---

## Imágenes en otra carpeta

Si no se especifica, el navegador espera encontrar la imagen en la misma carpeta que la página web.

Sin embargo, es común almacenar imágenes en una subcarpeta. A continuación, debe incluir el nombre de la carpeta en el atributo src:

Ejemplo:

```

```

---

## Imágenes en otro servidor

Algunos sitios web almacenan sus imágenes en servidores de imágenes.

En realidad, puede acceder a imágenes de cualquier dirección web del mundo:

Ejemplo:

```

```

---

## Imágenes animadas

El estándar GIF permite imágenes animadas:

Ejemplo:

```

```

Ten en cuenta que la sintaxis de insertar imágenes animadas no es diferente de las imágenes no animadas.

---

## Uso de una imagen como vínculo

Para utilizar una imagen como enlace, simplemente anida la etiqueta <img> dentro de la etiqueta <a>:

Ejemplo:

```
<a href="default.asp">  
    
</a>
```

Nota: border: 0; se agrega para evitar IE9 (y anterior) de mostrar un borde alrededor de la imagen (cuando la imagen es un enlace).

---

## Imagen flotante

Utilice la propiedad flotante CSS para dejar flotar la imagen a la derecha o la izquierda de un texto:

Ejemplo:

```
<p>
The image will float to the right of the text.</p>
```

```
<p>
The image will float to the left of the text.</p>
```

---

## Mapas de imágenes

Utiliza la etiqueta `<map>` para definir un mapa de imágenes. Un mapa de imágenes es una imagen con áreas que se pueden hacer clic.

El atributo `name` de la etiqueta `<map>` está asociado con el atributo `usemap` del `<img>` y crea una relación entre la imagen y el mapa.

La etiqueta `<map>` contiene una serie de etiquetas `<area>`, que definen las áreas que se pueden hacer clic en el mapa de imágenes:

Ejemplo:

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
```

---

## Imagen de fondo

Para agregar una imagen de fondo en un elemento HTML, utilice la propiedad CSS `background-image`:

Ejemplo:

Para agregar una imagen de fondo en una página web, especifique la propiedad background-image en el elemento BODY:

```
<body style="background-image:url('skies.jpg')">  
  
<h2>Background Image</h2>  
  
</body>
```

Ejemplo:

Para agregar una imagen de fondo en un párrafo, especifique la propiedad background-image en el elemento P:

```
<body>  
  
<p style="background-image:url('skies.jpg')">  
...  
</p>  
  
</body>
```

---

## El elemento <picture>

HTML5 introdujo el elemento <picture> para agregar más flexibilidad al especificar recursos de imagen.

El elemento <picture> contiene una serie de elementos <source>, cada uno de los cuales se refiere a diferentes fuentes de imagen. De esta manera el navegador puede elegir la imagen que mejor se ajuste a la vista y / o dispositivo actual.

Cada elemento <source> tiene atributos que describen cuándo su imagen es la más adecuada.

El navegador utilizará el primer elemento <source> con valores de atributo coincidentes e ignorará los siguientes elementos <source>.

Ejemplo:

Mostrar una imagen si la ventana del navegador (vista) es un mínimo de 650 píxeles, y otra imagen si no, pero más de 465 píxeles.

```
<picture>
```



```
<source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
<source media="(min-width: 465px)" srcset="img_white_flower.jpg">

</picture>
```

Nota: Siempre especifique un elemento `<img>` como último elemento secundario del elemento `<picture>`. El elemento `<img>` es utilizado por los navegadores que no admiten el elemento `<picture>`, o si ninguna de las etiquetas `<source>` coincide.

---

## Lectores de pantalla HTML

Un lector de pantalla es un programa de software que lee el código HTML, convierte el texto y permite al usuario "escuchar" el contenido. Los lectores de pantalla son útiles para las personas que son ciegas, con discapacidad visual o discapacitados de aprendizaje.

## Tablas HTML

---

### Definición de una tabla HTML

Una tabla HTML se define con la etiqueta `<table>` .

Cada fila de la tabla se define con la etiqueta `<tr>` . Un encabezado de tabla se define con la etiqueta `<th>` . De forma predeterminada, los encabezados de las tablas están en negrita y centrados. Una tabla de datos / celda se define con la etiqueta `<td>` .

Ejemplo:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
```

```
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

Nota: Los elementos <td> son los contenedores de datos de la tabla.

Pueden contener todo tipo de elementos HTML; texto, imágenes, listas, otras tablas, etc.

---

## Tabla HTML - Agregar un borde

Si no especifica un borde para la tabla, se mostrará sin bordes.

Un borde se establece mediante la propiedad de borde CSS :

Ejemplo:

```
table, th, td {
  border: 1px solid black;
}
```

Recuerda definir bordes para la tabla y las celdas de la tabla.

---

## Tabla HTML - Fronteras reducidas

Si deseas que los bordes se contraigan en un borde, agregue la propiedad CSS border-collapse :

Ejemplo:

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

---

## Tabla HTML - Añadir relleno a las celdas

Cell padding especifica el espacio entre el contenido de la celda y sus bordes.

Si no especifica un relleno, las celdas de la tabla se mostrarán sin relleno.

Para configurar el relleno, utilice la propiedad de relleno CSS :

Ejemplo:

```
th, td {  
    padding: 15px;  
}
```

---

## Tabla HTML: alinear a la izquierda los encabezados

De forma predeterminada, los encabezados de las tablas están en negrita y centrados.

Para alinear a la izquierda los encabezados de tabla, utilice la propiedad CSS text-align :

Ejemplo:

```
th {  
    text-align: left;  
}
```

---

## Tabla HTML: agregando el espaciado de los bordes

El espaciado entre bordes especifica el espacio entre las celdas.

Para establecer el espaciado de borde para una tabla, utilice la propiedad de espaciado de borde CSS :

Ejemplo:

```
table {  
    border-spacing: 5px;  
}
```

Nota: Si la tabla tiene bordes contraídos, el espaciado entre bordes no tiene efecto.

---

## Tabla HTML: celdas que abarcan muchas columnas

Para hacer que una celda cubra más de una columna, utilice el atributo colspan :

Ejemplo:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

---

## Tabla HTML: celdas que abarcan muchas filas

Para hacer un intervalo de celda más de una fila, utiliza el atributo rowspan :

Ejemplo:

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

---

## Tabla HTML: agregar una leyenda

Para agregar un título a una tabla, utilice la etiqueta <caption> :

Ejemplo:

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Nota: La etiqueta <caption> debe insertarse inmediatamente después de la etiqueta <table>.

---

## Un estilo especial para una tabla

Para definir un estilo especial para una tabla especial, agregue un atributo id a la tabla:

Ejemplo:

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

```
</tr>
</table>
```

Ahora puedes definir un estilo especial para esta tabla:

```
table#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

Y añade más estilos:

```
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
table#t01 tr:nth-child(odd) {
  background-color: #fff;
}
table#t01 th {
  color: white;
  background-color: black;
}
```

## Listas HTML

---

### Lista HTML no ordenada

Una lista desordenada comienza con la etiqueta `<ul>` . Cada elemento de lista comienza con la etiqueta `<li>` .

Los elementos de la lista se marcarán con viñetas (pequeños círculos negros) de forma predeterminada:

Ejemplo:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
```

```
</ul>
```

---

## Lista HTML no ordenada: elige marcador de elemento de lista

La propiedad CSS `list-style-type` se utiliza para definir el estilo del marcador de elemento de lista:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Ejemplo - Disco:

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ejemplo - Círculo:

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ejemplo - Cuadrado:

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ejemplo - Ninguno:

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

---

## Lista HTML ordenada

Una lista ordenada comienza con la etiqueta `<ol>` . Cada elemento de lista comienza con la etiqueta `<li>` .

Los elementos de la lista se marcarán con números de forma predeterminada:

Ejemplo:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

---

## Lista HTML ordenada: el atributo de tipo

El atributo de tipo de la etiqueta `<ol>`, define el tipo de marcador de elemento de lista:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers



Números:

```
<ol type="1">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Letras mayúsculas:

```
<ol type="A">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Letras minúsculas:

```
<ol type="a">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Números romanos en mayúsculas:

```
<ol type="I">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Números romanos en minúsculas:

```
<ol type="i">  
  <li>Coffee</li>  
  <li>Tea</li>
```

```
<li>Milk</li>
</ol>
```

---

## HTML Listas de Descripción

HTML también admite listas de descripción.

Una lista de descripción es una lista de términos, con una descripción de cada término.

La etiqueta <dl> define la lista de descripción, la etiqueta <dt> define el término (nombre) y la etiqueta <dd> describe cada término:

Ejemplo:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

---

## Listas HTML anidadas

La lista puede anidarse (listas dentro de las listas):

Ejemplo:

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Nota: Los elementos de lista pueden contener nueva lista y otros elementos HTML, como imágenes y vínculos, etc.

## Listas horizontales

Las listas de HTML se pueden diseñar de muchas maneras diferentes con CSS. Una forma popular es horizontalmente, para crear un menú:



pueden diseñar de muchas CSS. diseñar una lista crear un menú:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>
```

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

## Bloque HTML y elementos en línea

Cada elemento HTML tiene un valor de visualización predeterminado dependiendo del tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es bloque o en línea.

---

### Elementos a nivel de bloque

Un elemento de nivel de bloque siempre comienza en una nueva línea y ocupa toda la anchura disponible (se extiende hacia la izquierda y la derecha hasta donde puede).

El elemento `<div>` es un elemento a nivel de bloque.

Ejemplos de elementos a nivel de bloque:

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`

---

### Elementos en línea

Un elemento en línea no se inicia en una nueva línea y sólo ocupa tanto ancho como sea necesario.

Este es un elemento `<span>` dentro de un párrafo.

Ejemplos de elementos en línea:

- `<span>`
- `<a>`
- `<img>`

---

## El elemento <div>

El elemento <div> se utiliza a menudo como un contenedor para otros elementos HTML.

El elemento <div> no tiene atributos obligatorios, pero el estilo y la clase son comunes.

Cuando se utiliza junto con CSS, el elemento <div> se puede utilizar para estilos de bloques de contenido:

Ejemplo:

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the United Kingdom,
  with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

---

## El elemento <span>

El elemento <span> se utiliza a menudo como un contenedor para algún texto.

El elemento <span> no tiene atributos obligatorios, pero el estilo y la clase son comunes.

Cuando se utiliza junto con CSS, el elemento <span> se puede utilizar para estilos de partes del texto:

Ejemplo:

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

## El atributo clase en HTML

---

### Uso del atributo clase

El atributo de clase HTML permite definir estilos iguales para elementos con el mismo nombre de clase.

Aquí tenemos tres elementos <div> que apuntan al mismo nombre de clase:

Ejemplo:

```
<!DOCTYPE html>
<html>
```

```
<head>
<style>
div.cities {
  background-color: black;
  color: white;
  margin: 20px 0 20px 0;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
<h2>London</h2>
<p>London is the capital of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="cities">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>

<div class="cities">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
</div>

</body>
</html>
```

---

## Uso del atributo clase en elementos en línea

El atributo de clase HTML también se puede utilizar para elementos en línea:

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
span.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

## Diseños HTML

Los sitios web suelen mostrar contenido en varias columnas (como una revista o un periódico). HTML5 ofrece nuevos elementos semánticos que definen las diferentes partes de una página web:

- <header> - Define un encabezado para un documento o una sección
- <nav> - Define un contenedor para los enlaces de navegación
- <section> - Define una sección en un documento
- <article> - Define un artículo autónomo independiente
- <aside> - Define el contenido aparte del contenido (como una barra lateral)
- <footer> - Define un pie de página para un documento o una sección
- <details> - Define detalles adicionales
- <summary> - Define un encabezado para el elemento

# Formularios HTML

Ejemplo de formulario HTML

First name:

Last name:

---

## El elemento <form>

El elemento HTML <form> define un formulario que se utiliza para recopilar la entrada del usuario:

```
<form>
.
form elements
.
</form>
```

Un formulario HTML contiene elementos de formulario .

Los elementos de formulario son diferentes tipos de elementos de entrada, como campos de texto, casillas de verificación, botones de opción, botones de envío y más.

---

## El elemento <input>

El elemento <input> es el elemento de formulario más importante.

El elemento <input> se puede mostrar de varias maneras, dependiendo del atributo type .

Aquí hay unos ejemplos:

Tipo	Descripción
<input type = "text">	Define un campo de entrada de texto de una línea
<input type = "radio">	Define un botón de radio (para seleccionar una de muchas opciones)
<input type = "submit">	Define un botón de envío (para enviar el formulario)



---

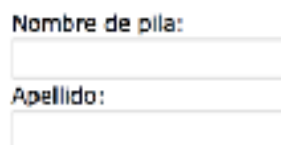
## Entrada de texto

`<input type = "text">` define un campo de entrada de una línea para la entrada de texto :

Ejemplo:

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Así es como se verá en un navegador:



Nombre de pila:

Apellido:

Nota: El formulario en sí no es visible. También ten en cuenta que el ancho predeterminado de un campo de texto es de 20 caracteres.

---

## Entrada del botón de radio

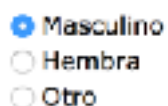
`<input type = "radio">` define un botón de radio .

Los botones de radio permiten a un usuario seleccionar UNO de un número limitado de opciones:

Ejemplo:

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



☒ Masculino  
☐ Hembra  
☐ Otro

---

## El botón Enviar

`<input type = "submit">` define un botón para enviar los datos del formulario a un form-handler .

El manejador de formularios suele ser una página de servidor con un script para procesar datos de entrada.

El form-handler se especifica en el atributo de acción del formulario :

Ejemplo:

```
<form action="/action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



---

## El atributo de acción

El atributo action define la acción a realizar cuando se envía el formulario.

Normalmente, los datos del formulario se envían a una página web en el servidor cuando el usuario hace clic en el botón Enviar.

En el ejemplo anterior, los datos del formulario se envían a una página del servidor denominada `" / action_page.php "`. Esta página contiene un script del lado del servidor que maneja los datos del formulario:

```
<form action="/action_page.php">
```

Si se omite el atributo de acción, la acción se establece en la página actual.

---

## El atributo de método

El atributo method especifica el método HTTP ( GET o POST ) que se utilizará al enviar los datos del formulario:

```
<form action="/action_page.php" method="get">
```

o:

```
<form action="/action_page.php" method="post">
```

---

## ¿Cuándo utilizar GET?

El método predeterminado al enviar datos de formulario es GET.

Sin embargo, cuando se utiliza GET, los datos del formulario presentados serán visibles en el campo de dirección de página :

```
/action_page.php?firstname=Mickey&lastname=Mouse
```

Nota: GET no se debe utilizar cuando se envía información confidencial. GET es más adecuado para cantidades cortas, no sensibles, de datos, ya que tiene limitaciones de tamaño también.

---

## ¿Cuándo utilizar POST?

Siempre usa POST si los datos del formulario contienen información sensible o personal. El método POST no muestra los datos de formulario enviados en el campo de dirección de página. POST no tiene limitaciones de tamaño, y se puede utilizar para enviar grandes cantidades de datos.

---

## El atributo de nombre

Cada campo de entrada debe tener un atributo de nombre para ser enviado.

Si se omite el atributo de nombre, los datos de ese campo de entrada no se enviarán en absoluto. En este ejemplo sólo se enviará el campo de entrada "Apellido":

Ejemplo:

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

---

## Agrupación de datos de formulario con <fieldset>

El elemento <fieldset> se utiliza para agrupar datos relacionados en un formulario.

El elemento <legend> define una leyenda para el elemento <fieldset>.

Ejemplo:

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



Información personal:

Nombre de pila:  
Mickey

Apellido:  
Mouse

Enviar

# Elementos del formulario HTML

A continuación describimos todos los elementos del formulario HTML.

---

## El elemento <input>

El elemento de formulario más importante es el elemento <input> .

El elemento <input> se puede mostrar de varias maneras, dependiendo del atributo type .

Todos los tipos de entrada diferentes se tratan más adelante.

---

## El elemento <select>

El elemento <select> define una lista desplegable :

Ejemplo:

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Los elementos <option> definen una opción que se puede seleccionar.

De forma predeterminada, se selecciona el primer elemento de la lista desplegable.

Para definir una opción preseleccionada, agregue el atributo seleccionado a la opción:

Ejemplo:

```
<option value="fiat" selected>Fiat</option>
```

---

## El elemento <textarea>

El elemento <textarea> define un campo de entrada de varias líneas ( un área de texto ):

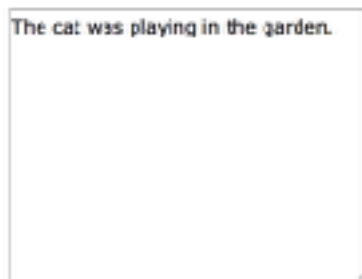
Ejemplo:

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

El atributo rows especifica el número visible de líneas en un área de texto.

El atributo cols especifica el ancho visible de un área de texto.

Así es como se mostrará el código HTML anterior en un navegador:



---

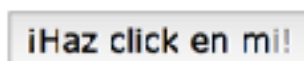
## El elemento <button>

El elemento <button> define un botón que se puede pulsar :

Ejemplo:

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

Así es como se mostrará el código HTML anterior en un navegador:



---

## Elementos del formulario HTML5

HTML5 agregó los siguientes elementos del formulario:

- <datalist>
- <keygen>

- `<output>`

Nota: Los navegadores no muestran elementos desconocidos. Los nuevos elementos que no son compatibles con los navegadores más antiguos no destruirán su página web.

---

## Elemento HTML5 `<datalist>`

El elemento `<datalist>` especifica una lista de opciones predefinidas para un elemento `<input>`. Los usuarios verán una lista desplegable de las opciones predefinidas a medida que introducen datos.

El atributo de lista del elemento `<input>`, debe referirse al atributo `id` del elemento `<datalist>`.

Ejemplo:

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

---

## Elemento HTML5 `<keygen>`

El objetivo del elemento `<keygen>` es proporcionar una forma segura de autenticar a los usuarios.

El elemento `<keygen>` especifica un campo generador de pares de claves en un formulario.

Cuando se envía el formulario, se generan dos claves, una privada y otra pública.

La clave privada se almacena localmente y la clave pública se envía al servidor.

La clave pública podría utilizarse para generar un certificado de cliente para autenticar al usuario en el futuro.

Ejemplo:

Un formulario con un campo keygen:

```
<form action="/action_page.php">
  Username: <input type="text" name="user">
  Encryption: <keygen name="security">
  <input type="submit">
</form>
```

---

## Elemento HTML5 <output>

El elemento <output> representa el resultado de un cálculo (como el realizado por un script).

Ejemplo:

Realiza un cálculo y muestra el resultado en un elemento <output>:

```
<form action="/action_page.php"
oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

## Tipos de entrada HTML

---

### Tipo de entrada Texto

<input type = "text"> define un campo de entrada de texto de una sola línea :

Ejemplo:



```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



---

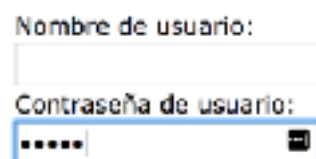
## Tipo de entrada Contraseña

`<input type = "password">` define un campo de contraseña :

Ejemplo:

```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



Los caracteres de un campo de contraseña están enmascarados (mostrados como asteriscos o círculos).

---

## Tipo de entrada Enviar

`<input type = "submit">` define un botón para enviar datos de formulario a un manejador de formularios .

El manejador de formularios suele ser una página de servidor con un script para procesar datos de entrada.

El form-handler se especifica en el atributo de acción del formulario :

Ejemplo:

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

I

Así es como se mostrará el código HTML anterior en un navegador:



Si omites el atributo de valor del botón Enviar, el botón tendrá un texto predeterminado:

Ejemplo:

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit">
</form>
```

---

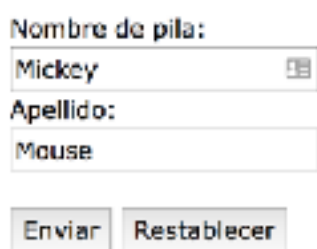
## Tipo de entrada Restablecer

`<input type = "reset">` define un botón de reinicio que restablecerá todos los valores de formulario a sus valores predeterminados:

Ejemplo:

```
<form action="/action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
  <input type="reset">  
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:



Nombre de pila:  
Mickey

Apellido:  
Mouse

Enviar Restablecer

Si cambias los valores de entrada y, a continuación, haga clic en el botón "Restablecer", los datos de formulario se restablecerán a los valores predeterminados.

---

## Tipo de entrada Radio

`<input type = "radio">` define un botón de radio .

Los botones de radio permiten que el usuario seleccione SÓLO UNA de un número limitado de opciones:

Ejemplo:

```
<form>  
  <input type="radio" name="gender" value="male" checked> Male<br>  
  <input type="radio" name="gender" value="female"> Female<br>  
  <input type="radio" name="gender" value="other"> Other  
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:

☒ Masculino  
☐ Hembra  
☐ Otro

---

## Tipo de entrada casilla de verificación

`<input type = "checkbox">` define una casilla de verificación .

Las casillas de verificación permiten a un usuario seleccionar opciones CERO o MÁS de un número limitado de opciones.

Ejemplo:

```
<form>  
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>  
  <input type="checkbox" name="vehicle2" value="Car"> I have a car  
</form>
```

Así es como se mostrará el código HTML anterior en un navegador:

☐ Tengo una bicicleta  
☐ tengo un carro

---

## Tipo de entrada botón

`<input type = "button">` define un botón :

Ejemplo:

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

Así es como se mostrará el código HTML anterior en un navegador:

¡Haz click en mí!

---

## Tipos de entrada HTML5

HTML5 agregó varios tipos de entrada nuevos:

- color
- fecha
- datetime local
- correo electrónico
- mes
- número
- distancia
- buscar
- tel
- hora
- url
- semana

Los nuevos tipos de entrada que no son compatibles con los navegadores web anteriores, se comportarán como `<input type = "text">`.

---

### Tipo de entrada Color

El `<input type = "color">` se utiliza para campos de entrada que deben contener un color.

Dependiendo del soporte del navegador, puede aparecer un selector de color en el campo de entrada.

Ejemplo:

```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```

---

### Tipo de entrada Fecha

El `<input type = "fecha">` se utiliza para campos de entrada que deben contener una fecha.

Dependiendo del soporte del navegador, un selector de fechas puede aparecer en el campo de entrada.

Ejemplo:

```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

También puedes agregar restricciones a las fechas:

Ejemplo:

```
<form>
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
</form>
```

---

## Tipo de entrada Datetime-local

El `<input type = "datetime-local">` especifica un campo de entrada de fecha y hora, con ninguna zona de tiempo.

Dependiendo del soporte del navegador, un selector de fechas puede aparecer en el campo de entrada.

Ejemplo:

```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

---

## Tipo de entrada Correo electrónico

El `<input type = "email">` se utiliza para campos de entrada que deben contener una dirección de correo electrónico.

Dependiendo del soporte del navegador, la dirección de correo electrónico puede ser validada automáticamente cuando se envía.

Algunos smartphones reconocen el tipo de correo electrónico y añaden ".com" al teclado para que coincida con la entrada de correo electrónico.

Ejemplo:

```
<form>  
  E-mail:  
  <input type="email" name="email">  
</form>
```

---

## Tipo de entrada Mes

El `<input type = "month">` permite al usuario seleccionar un mes y un año.

Dependiendo del soporte del navegador, un selector de fechas puede aparecer en el campo de entrada.

Ejemplo:

```
<form>  
  Birthday (month and year):  
  <input type="month" name="bdaymonth">  
</form>
```

---

## Entrada de número

El `<input type = "number">` define un campo numérico de entrada.

También puede establecer restricciones sobre qué números se aceptan.

El siguiente ejemplo muestra un campo de entrada numérico, donde puede introducir un valor de 1 a 5:






Ejemplo:

```
<form>  
  Quantity (between 1 and 5):  
  <input type="number" name="quantity" min="1" max="5">  
</form>
```

---

## Restricciones de entrada

Aquí hay una lista de algunas restricciones de entrada comunes (algunas son nuevas en HTML5):

Attribute	Description
disabled	Specifies that an input field should be disabled
max	 Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	 Specifies the minimum value for an input field
pattern	 Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	 Specifies that an input field is required (must be filled out)
size	Specifies the width (n characters) of an input field
step	 Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

El siguiente ejemplo muestra un campo numérico de entrada, donde puede introducir un valor de 0 a 100, en pasos de 10. El valor predeterminado es 30:

Ejemplo:

```
<form>
  Quantity:
  <input type="number" name="points" min="0" max="100" step="10" value="30">
</form>
```

---

## Tipo de entrada Rango

El `<input type = "range">` define un control para introducir un número cuyo valor exacto no es importante (como un control deslizante). El rango predeterminado es 0 a 100. Sin embargo, puede establecer restricciones sobre qué números se aceptan con los atributos `min`, `max` y `step`:

Ejemplo:

```
<form>
  <input type="range" name="points" min="0" max="10">
</form>
```



---

## Tipo de entrada búsqueda

El `<input type = "search">` se utiliza para los campos de búsqueda (un campo de búsqueda se comporta como un campo de texto normal).

Ejemplo:

```
<form>
  Search Google:
  <input type="search" name="googlesearch">
</form>
```

---

## Tipo de entrada Tel

El `<input type = "tel">` se utiliza para campos de entrada que deben contener un número de teléfono.

El tipo de teléfono sólo se admite actualmente en Safari 8.

Ejemplo:

```
<form>
  Telephone:
  <input type="tel" name="usrtel">
</form>
```

---

## Tipo de entrada Tiempo

El `<input type = "time">` permite al usuario seleccionar una hora (sin zona horaria).

Dependiendo del soporte del navegador, puede aparecer un selector de tiempo en el campo de entrada.

Ejemplo:

```
<form>
  Select a time:
  <input type="time" name="usr_time">
</form>
```

---

## Tipo de entrada URL

El `<input type = "URL">` se utiliza para campos de entrada que deben contener una dirección URL.

Dependiendo del soporte del navegador, el campo url puede ser validado automáticamente cuando se envía.

Algunos teléfonos inteligentes reconocen el tipo de url y agregan ".com" al teclado para que coincida con la entrada de url.

Ejemplo:

```
<form>
  Add your homepage:
  <input type="url" name="homepage">
</form>
```

---

## Tipo de entrada Semana

El `<input type = "semana">` permite al usuario seleccionar una semana y el año.

Dependiendo del soporte del navegador, un selector de fechas puede aparecer en el campo de entrada.

Ejemplo:

```
<form>
  Select a week:
  <input type="week" name="week_year">
</form>
```

## Atributos de entrada HTML

---

### El atributo valor

El atributo value especifica el valor inicial de un campo de entrada:

Ejemplo:

```
<form action="">
First name:<br>
```

```
<input type="text" name="firstname" value="John">
</form>
```

---

## El atributo de sólo lectura

El atributo readonly especifica que el campo de entrada es de sólo lectura (no se puede cambiar):  
Ejemplo:

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" readonly>
</form>
```

---

## El atributo deshabilitado

El atributo disabled especifica que el campo de entrada está deshabilitado.  
Un campo de entrada inhabilitado es inutilizable y no se puede hacer clic, y su valor no se enviará al enviar el formulario:  
Ejemplo:

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" disabled>
</form>
```

---

## El atributo tamaño

El atributo size especifica el tamaño (en caracteres) del campo de entrada:  
Ejemplo:

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" size="40">
</form>
```

---

## El atributo maxlength

El atributo maxlength especifica la longitud máxima permitida para el campo de entrada:

Ejemplo:

```
<form action="">
First name:<br>
<input type="text" name="firstname" maxlength="10">
</form>
```

Con un atributo maxlength, el campo de entrada no aceptará más que el número permitido de caracteres.

El atributo maxlength no proporciona ninguna retroalimentación. Si desea alertar al usuario, debe escribir código JavaScript.

Nota: Las restricciones de entrada no son infalibles, y JavaScript proporciona muchas maneras de agregar entradas ilegales. Para restringir la entrada de forma segura, debe ser verificada por el receptor (el servidor) también!

---

## Atributos HTML5

HTML5 agregó los siguientes atributos para <input>:

- autocompletar
- autofocus
- formar
- forma de actuar
- tipo de formulario
- método de formulario
- formidable
- formatear
- alto y ancho
- lista
- min y max
- múltiple
- patrón (regexp)
- marcador de posición
- necesario
- paso

y los siguientes atributos para <form>:

- autocompletar

- novalidar

---

## El atributo de autocompletar

El atributo autocomplete especifica si un campo de formulario o de entrada debe tener el autocompletar activado o desactivado.

Cuando el autocompletar está activado, el navegador completa automáticamente los valores de entrada basados en valores que el usuario ha ingresado anteriormente.

Sugerencia: Es posible tener el autocompletar "on" para el formulario y "off" para campos de entrada específicos, o viceversa.

El atributo de autocompletar funciona con <form> y los siguientes <input> tipos: texto, búsqueda, url, tel, correo electrónico, contraseña, datepickers, rango y color.

Ejemplo:

Un formulario HTML con autocompletar activado (y desactivado para un campo de entrada):

```
<form action="/action_page.php" autocomplete="on">  
  First name:<input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>
```

Sugerencia: En algunos navegadores es posible que deba activar la función de autocompletar para que esto funcione.

---

## El atributo novalidado

El atributo novalidate es un atributo <form>.

Cuando está presente, novalidate especifica que los datos del formulario no se deben validar cuando se envían.

Ejemplo:

Indica que el formulario no se debe validar al enviar:

```
<form action="/action_page.php" novalidate>  
  E-mail: <input type="email" name="user_email">
```

```
<input type="submit">
</form>
```

---

## El atributo de enfoque automático

El atributo autofocus especifica que el campo de entrada debe obtener automáticamente el foco cuando se carga la página.

Ejemplo:

Deja que el campo de entrada "Nombre" obtenga automáticamente el foco cuando se cargue la página:

```
First name:<input type="text" name="fname" autofocus>
```

---

## El atributo de formulario

El atributo form especifica uno o más formularios a los que pertenece un elemento <input>.

Sugerencia: Para referirse a más de un formulario, utilice una lista separada por espacios de ID de formulario.

Ejemplo:

Un campo de entrada situado fuera del formulario HTML (pero sigue siendo parte del formulario):

```
<form action="/action_page.php" id="form1">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
</form>

Last name: <input type="text" name="lname" form="form1">
```

---

## El atributo formaction

El atributo formaction especifica la dirección URL de un archivo que procesará el control de entrada cuando se envíe el formulario.

El atributo formaction reemplaza el atributo action del elemento <form>.

El atributo formaction se utiliza con type = "submit" y type = "image".

Ejemplo:

Un formulario HTML con dos botones de envío, con diferentes acciones:

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formaction="/action_page2.php"  
    value="Submit as admin">  
</form>
```

---

## El atributo formenctype

El atributo formenctype especifica cómo se deben codificar los datos del formulario cuando se envían (sólo para formularios con method = "post").

El atributo formenctype anula el atributo enctype del elemento <form>.

El atributo formenctype se utiliza con type = "submit" y type = "image".

Ejemplo:

Enviar datos de formulario que están codificados por defecto (el primer botón de envío) y codificados como "multipart / form-data" (el segundo botón de envío):

```
<form action="/action_page_binary.asp" method="post">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formenctype="multipart/form-data"  
    value="Submit as Multipart/form-data">  
</form>
```

---

## El atributo formmethod

El atributo formmethod define el método HTTP para enviar datos de formulario a la URL de acción.

El atributo formmethod anula el atributo de método del elemento <form>.

El atributo formmethod se puede utilizar con type = "submit" y type = "image".

Ejemplo:

El segundo botón de envío reemplaza al método HTTP del formulario:

```
<form action="/action_page.php" method="get">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formmethod="post" formaction="action_page_post.asp"  
    value="Submit using POST">  
</form>
```

---

## El atributo formnovalidate

El atributo formnovalidate anula el atributo novalidate del elemento <form>.

El atributo formnovalidate se puede utilizar con type = "submit".

Ejemplo:

Un formulario con dos botones de envío (con y sin validación):

```
<form action="/action_page.php">  
  E-mail: <input type="email" name="userid"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formnovalidate value="Submit without validation">  
</form>
```

---

## El atributo formtarget

El atributo formtarget especifica un nombre o una palabra clave que indica dónde mostrar la respuesta que se recibe después de enviar el formulario.

El atributo formtarget reemplaza el atributo de destino del elemento <form>.

El atributo formtarget se puede utilizar con type = "submit" y type = "image".

Ejemplo:

Un formulario con dos botones de envío, con diferentes ventanas de destino:



```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit as normal">  
  <input type="submit" formtarget="_blank"  
    value="Submit to a new window">  
</form>
```

---

## Atributos altura y anchura

Los atributos height y width especifican la altura y el ancho de un elemento <input type = "image">.

Siempre especifique el tamaño de las imágenes. Si el navegador no conoce el tamaño, la página parpadeará mientras las imágenes se cargan.

Ejemplo:

Defina una imagen como el botón Enviar, con los atributos altura y anchura:

```
<input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
```

---

## El atributo lista

El atributo list se refiere a un elemento <datalist> que contiene opciones predefinidas para un elemento <input>.

Ejemplo:

Un elemento <input> con valores predefinidos en un <datalist>:

```
<input list="browsers">  
<datalist id="browsers">  
  <option value="Internet Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

---

## Los atributos min y max

Los atributos min y max especifican los valores mínimo y máximo de un elemento `<input>`.

Los atributos min y max funcionan con los siguientes tipos de entrada: número, intervalo, fecha, fecha-hora local, mes, hora y semana.

Ejemplo:

`<input>` elementos con valores min y max:

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```

---

## El atributo múltiple

El atributo multiple especifica que el usuario puede introducir más de un valor en el elemento `<input>`.

El atributo múltiple funciona con los siguientes tipos de entrada: correo electrónico y archivo.

Ejemplo:

Un campo de carga de archivos que acepta varios valores:

```
Select images: <input type="file" name="img" multiple>
```

---

## El atributo patrón

El atributo pattern especifica una expresión regular en la que se comprueba el valor del elemento `<input>`.

El atributo patrón funciona con los siguientes tipos de entrada: texto, búsqueda, url, tel, correo electrónico y contraseña.

Consejo: utiliza el atributo de título global para describir el patrón para ayudar al usuario.

Ejemplo:

Campo de entrada que sólo puede contener tres letras (sin números ni caracteres especiales):

```
Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">
```

---

## El atributo del marcador de posición

El atributo de marcador de posición especifica una sugerencia que describe el valor esperado de un campo de entrada (un valor de muestra o una breve descripción del formato).

La sugerencia se muestra en el campo de entrada antes de que el usuario introduzca un valor.

El atributo del marcador de posición funciona con los siguientes tipos de entrada: texto, búsqueda, url, tel, correo electrónico y contraseña.

Ejemplo:

Un campo de entrada con un texto de marcador de posición:

```
<input type="text" name="fname" placeholder="First name">
```

---

## El atributo requerido

El atributo required especifica que un campo de entrada debe ser llenado antes de enviar el formulario.

El atributo requerido funciona con los siguientes tipos de entrada: texto, búsqueda, url, tel, correo electrónico, contraseña, selectores de fecha, número, casilla de verificación, radio y archivo.

Ejemplo:

Un campo de entrada requerido:

```
Username: <input type="text" name="username" required>
```

---

## El atributo paso

El atributo step especifica los intervalos de número legales para un elemento <input>.

Ejemplo: si `step = "3"`, los números legales podrían ser -3, 0, 3, 6, etc.

Sugerencia: El atributo `step` se puede usar junto con los atributos `max` y `min` para crear un rango de valores legales.

El atributo `step` funciona con los siguientes tipos de entrada: número, intervalo, fecha, fecha-hora local, mes, hora y semana.

Ejemplo:

Campo de entrada con intervalos de número legal especificados:

```
<input type="number" name="points" step="3">
```