



**EBook Gratis**

# APRENDIZAJE HTML

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#html**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con HTML.....</b>	<b>2</b>
Observaciones.....	2
Versiones.....	2
Examples.....	2
Hola Mundo.....	2
<b>Introducción.....</b>	<b>2</b>
<b>Perspectiva del elemento.....</b>	<b>3</b>
<b>Creando una página simple.....</b>	<b>3</b>
<b>Desglose de página simple.....</b>	<b>4</b>
<b>Capítulo 2: Anclas e hipervínculos.....</b>	<b>6</b>
Introducción.....	6
Sintaxis.....	6
Parámetros.....	6
Examples.....	7
Enlace a otro sitio.....	7
Abrir enlace en una nueva pestaña / ventana.....	7
Enlace a un ancla.....	8
Enlace que ejecuta JavaScript.....	8
Enlace a una página en el mismo sitio.....	9
Enlace que ejecuta el cliente de correo electrónico.....	10
Enlace que marca un número.....	10
<b>Capítulo 3: ARIA.....</b>	<b>11</b>
Sintaxis.....	11
Observaciones.....	11
Examples.....	12
role = "alerta".....	12
role = "alertdialog".....	12
role = "application".....	12
role = "artículo".....	12

role = "banner" .....	13
role = "button" .....	13
role = "cell" .....	13
role = "checkbox" .....	13
role = "columnheader" .....	14
role = "combobox" .....	14
role = "complementario" .....	14
role = "contentinfo" .....	14
role = "definición" .....	15
role = "dialog" .....	15
role = "directorio" .....	15
role = "documento" .....	15
role = "form" .....	15
role = "grid" .....	16
role = "gridcell" .....	16
role = "grupo" .....	17
role = "encabezado" .....	17
role = "img" .....	17
role = "link" .....	17
role = "list" .....	18
role = "listbox" .....	18
role = "listitem" .....	18
role = "log" .....	18
role = "main" .....	18
role = "marquee" .....	19
role = "math" .....	19
role = "menu" .....	19
role = "menubar" .....	19
role = "menuitem" .....	19
role = "menuitemcheckbox" .....	20
role = "menuitemradio" .....	20
role = "navegación" .....	20
role = "note" .....	20

role = "option" .....	20
role = "presentación" .....	21
role = "barra de progreso" .....	21
role = "radio" .....	21
role = "region" .....	21
role = "radiogroup" .....	21
role = "row" .....	22
role = "rowgroup" .....	22
role = "rowheader" .....	22
role = "scrollbar" .....	22
role = "search" .....	23
role = "searchbox" .....	23
role = "separator" .....	23
role = "control deslizante" .....	23
role = "spinbutton" .....	24
role = "status" .....	24
role = "switch" .....	24
role = "tab" .....	24
role = "table" .....	24
role = "tablist" .....	25
role = "tabpanel" .....	25
role = "cuadro de texto" .....	25
role = "timer" .....	25
role = "toolbar" .....	25
role = "tooltip" .....	26
role = "árbol" .....	26
role = "treegrid" .....	26
role = "treeitem" .....	27
<b>Capítulo 4: Atributos de datos</b> .....	<b>28</b>
Sintaxis .....	28
Parámetros .....	28
Examples .....	28
Uso de atributos de datos .....	28

Soporte de navegadores antiguos.....	28
<b>Capítulo 5: Atributos de evento HTML.....</b>	<b>30</b>
Examples.....	30
Eventos de formulario HTML.....	30
Eventos del teclado.....	30
<b>Capítulo 6: Atributos globales.....</b>	<b>32</b>
Parámetros.....	32
Observaciones.....	32
Examples.....	32
Atributo contenteditable.....	32
<b>Capítulo 7: Barras de navegación.....</b>	<b>34</b>
Examples.....	34
Barra de navegación básica.....	34
Barra de navegación HTML5.....	34
<b>Capítulo 8: Clases e identificaciones.....</b>	<b>35</b>
Introducción.....	35
Sintaxis.....	35
Parámetros.....	35
Observaciones.....	35
Examples.....	35
Dar un elemento a una clase.....	35
Usando clases en CSS.....	36
Dando a un elemento una identificación.....	37
Problemas relacionados con identificaciones duplicadas.....	37
Valores aceptables.....	38
Para una identificación.....	38
Para una clase.....	39
Nota importante: cómo los valores de ID y clase se tratan fuera de HTML.....	39
Referencias W3C.....	40
<b>Capítulo 9: Comentarios.....</b>	<b>41</b>
Introducción.....	41
Sintaxis.....	41

Observaciones.....	41
Examples.....	41
Creando comentarios.....	41
Comentarios condicionales para Internet Explorer.....	42
Bajo nivel oculto.....	42
Nivel bajo revelado.....	43
Comentando espacios en blanco entre elementos en línea.....	43
<b>Capítulo 10: Controles del menú de selección.....</b>	<b>45</b>
Sintaxis.....	45
Examples.....	45
Seleccione el menú.....	45
Cambiando el tamaño.....	45
Menús de selección multi-opciones.....	45
Grupos opcionales.....	46
Opciones.....	46
Seleccionando una opción por defecto.....	47
Lista de datos.....	47
<b>Soporte del navegador.....</b>	<b>47</b>
<b>Capítulo 11: Cotizaciones de mercado.....</b>	<b>49</b>
Observaciones.....	49
Examples.....	49
En línea con.....	49
Comillas.....	49
URL de origen (atributo de cite).....	49
Bloque con.....	49
URL de origen (atributo de cite).....	50
Citación / Atribución.....	50
<b>Capítulo 12: Doctypes.....</b>	<b>51</b>
Introducción.....	51
Sintaxis.....	51
Observaciones.....	51
Examples.....	51

Añadiendo el Doctype .....	51
HTML 4.01 Doctypes .....	52
HTML 4.01 estricto .....	52
HTML 4.01 Transicional .....	52
HTML 4.01 conjunto de marcos .....	52
HTML 5 Doctype .....	52
<b>Insensibilidad de casos .....</b>	<b>53</b>
Doctypes antiguos .....	53
HTML 3.2 .....	53
HTML 2.0 .....	53
<b>Capítulo 13: Elemento de etiqueta .....</b>	<b>54</b>
Sintaxis .....	54
Parámetros .....	54
Examples .....	54
Uso básico .....	54
Sobre la etiqueta .....	55
<b>Capítulo 14: Elemento de progreso .....</b>	<b>56</b>
Parámetros .....	56
Observaciones .....	56
Examples .....	56
Progreso .....	56
Cambiando el color de una barra de progreso .....	56
Chrome / Safari / Opera .....	57
Firefox .....	57
explorador de Internet .....	57
HTML Fallback .....	57
<b>Capítulo 15: Elemento de salida .....</b>	<b>59</b>
Parámetros .....	59
Examples .....	59
Elemento de salida usando para y atributos de formulario .....	59
Elemento de salida con atributos .....	60
<b>Capítulo 16: Elemento div .....</b>	<b>61</b>

Introducción.....	61
Sintaxis.....	61
Examples.....	61
Anidamiento.....	61
Uso básico.....	62
<b>Capítulo 17: Elementos de control de entrada.....</b>	<b>63</b>
Introducción.....	63
Sintaxis.....	63
Parámetros.....	63
Observaciones.....	64
Examples.....	65
Casilla de verificación y botones de radio.....	65
<b>Visión general.....</b>	<b>65</b>
<b>Atributos.....</b>	<b>66</b>
value.....	66
checked.....	66
<b>Accesibilidad.....</b>	<b>66</b>
Etiquetas.....	66
Grupos de botones.....	67
Oculto.....	68
Contraseña.....	68
Enviar.....	68
Expediente.....	68
Validación de entrada.....	69
Necesario.....	69
Longitud mínima / máxima.....	70
Especificando un rango.....	70
Emparejar un patrón.....	70
Aceptar tipo de archivo.....	71
Reiniciar.....	71
Número.....	71
Tel.....	72



Email.....	72
Botón.....	72
<b>Atributos.....</b>	<b>73</b>
[name].....	73
[type].....	73
[value].....	73
Atributos extra para botones de envío.....	73
Color.....	74
Url.....	74
Fecha.....	75
DateTime-Local.....	75
Imagen.....	75
Distancia.....	75
Mes.....	76
Hora.....	76
Semana.....	76
Texto.....	76
Buscar.....	77
DateTime (Global).....	77
<b>Capítulo 18: Elementos de los medios.....</b>	<b>79</b>
Parámetros.....	79
Observaciones.....	79
Soporte en los navegadores.....	79
Examples.....	80
Utilizando `y` Elemento ` para mostrar contenido de audio / video.....	80
Audio.....	81
Vídeo.....	81
Encabezado de video o fondo.....	81
<b>Capítulo 19: Elementos de seccionamiento.....</b>	<b>83</b>
Observaciones.....	83
Examples.....	83
Elemento del artículo.....	83

<b>Evitar el uso innecesario</b> .....	<b>83</b>
Elemento principal.....	84
Elemento de navegación.....	85
<b>Artículos en línea</b> .....	<b>85</b>
<b>Use los elementos de la lista cuando sea necesario</b> .....	<b>86</b>
<b>Evitar el uso innecesario</b> .....	<b>86</b>
Elemento de sección.....	87
Elemento de encabezado.....	88
Ejemplos:.....	88
Elemento de pie de página.....	88
<b>Capítulo 20: Elementos vacíos</b> .....	<b>89</b>
Introducción.....	89
Observaciones.....	89
Examples.....	89
Elementos vacíos.....	89
<b>Capítulo 21: Empotrar</b> .....	<b>91</b>
Parámetros.....	91
Examples.....	91
Uso básico.....	91
Definiendo el tipo MIME.....	91
<b>Capítulo 22: Encabezados</b> .....	<b>92</b>
Introducción.....	92
Sintaxis.....	92
Observaciones.....	92
Examples.....	92
Usando encabezados.....	92
<b>La estructura correcta importa</b> .....	<b>93</b>
<b>Capítulo 23: Entidades de carácter</b> .....	<b>94</b>
Examples.....	94
Caracteres especiales comunes.....	94
Entidades de personajes en HTML.....	95

<b>Capítulo 24: Formas</b>	<b>96</b>
Introducción	96
Sintaxis	96
Parámetros	96
Observaciones	96
Examples	97
Sumisión	97
El atributo de acción	97
El atributo del método	97
Mas atributos	97
Atributo de destino en la etiqueta de formulario	97
Cargando archivos	98
Agrupando algunos campos de entrada	98
<b>Capítulo 25: Formato de texto</b>	<b>100</b>
Introducción	100
Sintaxis	100
Examples	100
Negrita, cursiva y subrayado	100
Texto en negrita	100
Texto en cursiva	101
Texto subrayado	101
Destacando	101
Insertado, Borrado o Enojado	102
Superíndice y subíndice	102
Abreviatura	102
<b>Capítulo 26: HTML 5 Caché</b>	<b>103</b>
Observaciones	103
Examples	103
Ejemplo básico de caché de HTML 5	103
<b>Capítulo 27: Idiomas de contenido</b>	<b>104</b>
Sintaxis	104
Observaciones	104

Accesibilidad.....	104
Examples.....	104
Lenguaje de elementos.....	104
Elementos con múltiples idiomas.....	105
Manejo de atributos con diferentes idiomas.....	105
Idioma base del documento.....	105
URL regionales.....	105
<b>Capítulo 28: IFrames.....</b>	<b>106</b>
Parámetros.....	106
Observaciones.....	106
<b>Política del mismo origen.....</b>	<b>106</b>
atributo de sandbox.....	107
Examples.....	107
Fundamentos de un marco en línea.....	107
Configuración del tamaño del marco.....	108
Usando Anclas Con IFrames.....	108
Usando el atributo "srcdoc".....	108
Caja de arena.....	108
<b>Capítulo 29: Imágenes.....</b>	<b>110</b>
Sintaxis.....	110
Parámetros.....	110
Examples.....	110
Creando una imagen.....	110
Ancho y altura de la imagen.....	111
Elegir texto alternativo.....	112
Notas al pie.....	113
Imagen sensible utilizando el atributo srcset.....	113
<b>Usando srcset con tamaños.....</b>	<b>113</b>
<b>Usando srcset sin tallas.....</b>	<b>114</b>
Imagen sensible utilizando un elemento de imagen.....	114
<b>Capítulo 30: Incluir código JavaScript en HTML.....</b>	<b>115</b>
Sintaxis.....	115

Parámetros.....	115
Observaciones.....	115
Examples.....	115
Enlace a un archivo JavaScript externo.....	116
Incluyendo directamente el código JavaScript.....	116
Incluyendo un archivo JavaScript ejecutándose de forma asíncrona.....	116
Manejo de Javascript deshabilitado.....	116
<b>Capítulo 31: Liza.....</b>	<b>117</b>
Introducción.....	117
Sintaxis.....	117
Observaciones.....	117
Examples.....	117
Lista desordenada.....	117
Lista ordenada.....	118
Cambiando manualmente los números.....	118
Cambiando el tipo de numeral.....	120
Descripción de la lista.....	120
Listas anidadas.....	121
<b>Capítulo 32: Lona.....</b>	<b>122</b>
Parámetros.....	122
Observaciones.....	122
Examples.....	122
Ejemplo básico.....	122
Dibujando dos rectángulos en una.....	122
<b>Capítulo 33: Mapas de imagen.....</b>	<b>124</b>
Sintaxis.....	124
Parámetros.....	124
Observaciones.....	125
Examples.....	125
Introducción a los mapas de imagen.....	125
<b>Descripción.....</b>	<b>125</b>
<b>Ejemplo básico.....</b>	<b>125</b>

<b>Capítulo 34: Marcando código informático</b>	<b>127</b>
Sintaxis	127
Observaciones	127
Elementos relacionados	127
Examples	127
En línea con	127
Bloque con y	127
<b>Capítulo 35: Mesas</b>	<b>129</b>
Introducción	129
Sintaxis	129
Observaciones	129
Examples	130
Mesa simple	130
Abarcando columnas o filas	130
Tabla con thead, tbody, tfoot y subtítulo	131
Grupos de columnas	132
Ámbito de aplicación	133
<b>Capítulo 36: Meta informacion</b>	<b>136</b>
Introducción	136
Sintaxis	136
Observaciones	136
Examples	136
Codificación de caracteres	136
Actualización automática	137
Control de diseño móvil	137
Información de la página	138
application-name	138
author	138
description	138
generator	139
keywords	139
Los robots	139

Reconocimiento de números de teléfono.....	140
Medios de comunicación social.....	140
Facebook / Open Graph.....	141
Facebook / Artículos instantáneos.....	141
Gorjeo.....	141
Google+ / Schema.org.....	141
Redireccionamiento automatico.....	142
Aplicación Web.....	142
<b>Capítulo 37: Párrafos.....</b>	<b>144</b>
Introducción.....	144
Parámetros.....	144
Examples.....	144
Párrafos HTML.....	144
<b>Capítulo 38: Recursos de enlace.....</b>	<b>145</b>
Introducción.....	145
Sintaxis.....	145
Parámetros.....	145
Examples.....	145
Hoja de estilo CSS externa.....	146
JavaScript.....	146
<b>Sincrónico.....</b>	<b>146</b>
<b>Asincrónico.....</b>	<b>146</b>
<b>Diferido.....</b>	<b>147</b>
<b>&lt;noscript&gt;.....</b>	<b>147</b>
Favicon.....	147
CSS alternativo.....	147
Alimentación web.....	148
Enlace del atributo 'media'.....	148
Anterior y Siguiente.....	148
Sugerencia de recursos: dns-prefetch, prefetch, prerender.....	148
<b>Preconectarse.....</b>	<b>148</b>

<b>DNS-Prefetch</b> .....	<b>148</b>
<b>Prefetch</b> .....	<b>149</b>
<b>Prerender</b> .....	<b>149</b>
<b>Capítulo 39: SVG</b> .....	<b>150</b>
Introducción.....	150
Observaciones.....	150
Examples.....	150
Incrustar archivos SVG externos en HTML.....	150
Usando el elemento de imagen.....	150
Usando el elemento objeto.....	150
SVG en línea.....	150
Incrustar SVG usando CSS.....	151
<b>Capítulo 40: Tabindex</b> .....	<b>153</b>
Parámetros.....	153
Observaciones.....	153
Examples.....	153
Agregar un elemento al orden de tabulación.....	153
Eliminar un elemento del orden de tabulación.....	153
Definir un orden de tabulación personalizado (no recomendado).....	153
<b>Capítulo 41: Usando HTML con CSS</b> .....	<b>155</b>
Introducción.....	155
Sintaxis.....	155
Examples.....	155
Uso externo de la hoja de estilo.....	155
Hoja de estilo interna.....	155
Estilo en línea.....	156
Múltiples hojas de estilo.....	156
<b>Creditos</b> .....	<b>158</b>



---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [html](#)

It is an unofficial and free HTML ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official HTML.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capítulo 1: Empezando con HTML

## Observaciones

**HTML** ( **H**yper **t**ext **M**arkup **L**anguage) es un sistema compatible con **XML** para anotar documentos con 'etiquetas'. Se utiliza específicamente para crear contenido para páginas web y aplicaciones web, que luego se pueden compartir a través de una red.

Además del texto, la versión actual de HTML admite muchos **tipos** diferentes **de medios** , incluidas imágenes y videos.

## Versiones

Versión	Especificación	Fecha de lanzamiento
1.0	N / A	1994-01-01
2.0	<a href="#">RFC 1866</a>	1995-11-24
3.2	<a href="#">W3C: Especificación HTML 3.2</a>	1997-01-14
4.0	<a href="#">W3C: Especificación de HTML 4.0</a>	1998-04-24
4.01	<a href="#">W3C: HTML 4.01 Especificación</a>	1999-12-24
5	<a href="#">WHATWG: HTML Living Standard</a>	2014-10-28
5.1	<a href="#">W3C: Especificación de HTML 5.1</a>	2016-11-01

## Examples

Hola Mundo

---

---

## Introducción

**HTML** ( **H**yper **t**ext **M**arkup **L**anguage) utiliza un sistema de marcado compuesto por elementos que representan contenido específico. *El marcado* significa que con HTML usted declara *lo que* se presenta a un espectador, no *cómo* se presenta. Las representaciones visuales se definen mediante **hojas de estilo en cascada (CSS)** y se realizan mediante navegadores. **Los elementos aún existentes que permiten este tipo de `font`** , como por ejemplo la `font` , "están completamente obsoletos y no deben ser utilizados por los autores" <sup>[1]</sup> .

HTML a veces se denomina lenguaje de programación, pero no tiene lógica, por lo que es un

**lenguaje de marcado** . Las etiquetas HTML proporcionan significado semántico y legibilidad de la máquina para el contenido de la página.

Un elemento generalmente consiste en una etiqueta de apertura ( `<element_name>` ), una etiqueta de cierre ( `</element_name>` ), que contiene el nombre del elemento rodeado de corchetes angulares, y el contenido entre: `<element_name>...content...</element_name>`

Hay algunos elementos HTML que no tienen una etiqueta de cierre o ningún contenido. Estos se llaman **elementos vacíos** . Los elementos vacíos incluyen `<img>` , `<meta>` , `<link>` y `<input>` .

Los nombres de los elementos pueden considerarse palabras clave descriptivas para el contenido que contienen, como `video` , `audio` , `table` , `footer` .

Una página HTML puede consistir en cientos de elementos que luego se leen en un navegador web, se interpretan y se convierten en contenido legible o audible en la pantalla.

Para este documento es importante tener en cuenta la diferencia entre elementos y etiquetas:

**Elementos:** `video` , `audio` , `table` , `footer`

**Etiquetas:** `<video>` , `<audio>` , `<table>` , `<footer>` , `</html>` , `</body>`

---

## Perspectiva del elemento

Vamos a romper una etiqueta ...

La etiqueta `<p>` representa un párrafo común.

Los elementos comúnmente tienen una etiqueta de apertura y una etiqueta de cierre. La etiqueta de apertura contiene el nombre del elemento entre paréntesis angulares ( `<p>` ). La etiqueta de cierre es idéntica a la etiqueta de apertura con la adición de una barra diagonal ( `/` ) entre el soporte de apertura y el nombre del elemento ( `</p>` ).

El contenido puede ir entre estas dos etiquetas: `<p>This is a simple paragraph.</p>` .

---

## Creando una página simple

El siguiente ejemplo de HTML crea una página web simple "Hello World" .

Los archivos HTML se pueden crear utilizando cualquier **editor de texto** . Los archivos se deben guardar con una extensión `.html` o `.htm` <sup>[2]</sup> para que se reconozcan como archivos HTML.

Una vez creado, este archivo se puede abrir en cualquier navegador web.

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title>Hello!</title>
  </head>

  <body>
    <h1>Hello World!</h1>
    <p>This is a simple paragraph.</p>
  </body>

</html>
```

## Desglose de página simple

Estas son las etiquetas utilizadas en el ejemplo:

Etiqueta	Sentido
<!DOCTYPE>	Define la versión HTML utilizada en el documento. En este caso es HTML5. Consulte el <a href="#">tema doctypes</a> para más información.
<html>	Abre la página. No debe haber marcas después de la etiqueta de cierre ( </html> ). El atributo <code>lang</code> declara el idioma principal de la página utilizando los <a href="#">códigos de idioma ISO</a> ( <code>en</code> para inglés). Consulte el <a href="#">tema Idioma</a> del <a href="#">contenido</a> para obtener más información.
<head>	Abre la sección de encabezado, que no aparece en la ventana principal del navegador, pero contiene principalmente información <i>sobre</i> el documento HTML, llamado <i>metadatos</i> . También puede contener importaciones desde hojas de estilo externas y scripts. La etiqueta de cierre es </head> .
<meta>	Le da al navegador algunos metadatos sobre el documento. El atributo <code>charset</code> declara la <a href="#">codificación de caracteres</a> . Los documentos HTML modernos siempre deben usar <b>UTF-8</b> , aunque no sea un requisito. En HTML, la etiqueta <meta> no requiere una etiqueta de cierre. Vea el <a href="#">tema Meta</a> para más información.
<title>	El título de la página. El texto escrito entre esta apertura y la etiqueta de cierre ( </title> ) se mostrará en la pestaña de la página o en la barra de título del navegador.

Etiqueta	Sentido
<body>	Abre la parte del documento que se muestra a los usuarios, es decir, todo el contenido visible o audible de una página. No se debe agregar contenido después de la etiqueta de cierre </body> .
<h1>	Un título de nivel 1 para la página. Ver <a href="#">encabezados</a> para más información.
<p>	Representa un párrafo común de texto.

1. ↑ [HTML5, 11.2 características no conformes](#)
2. ↑ `.htm` se hereda del límite de extensión de archivo de tres caracteres del [DOS](#) heredado.

Lea [Empezando con HTML en línea](#): <https://riptutorial.com/es/html/topic/217/empezando-con-html>

# Capítulo 2: Anclas e hipervínculos

## Introducción

Las etiquetas de anclaje se usan comúnmente para vincular páginas web separadas, pero también se pueden usar para vincular diferentes lugares en un solo documento, a menudo dentro de la tabla de contenido o incluso para lanzar aplicaciones externas. Este tema explica la implementación y la aplicación de etiquetas de anclaje HTML en varias funciones.

## Sintaxis

- `<a href="URL or anchor">Link Text</a>`

## Parámetros

Parámetro	Detalles
href	Especifica la dirección de destino. Puede ser una URL absoluta o relativa, o el <code>name</code> de un ancla. Una URL absoluta es la URL completa de un sitio web como <a href="http://example.com/">http://example.com/</a> . Una URL relativa apunta a otro directorio y / o documento dentro del mismo sitio web, por ejemplo, <code>/about-us/</code> apunta al directorio "about-us" dentro del directorio raíz ( <code>/</code> ). Cuando se apunta a otro directorio sin especificar explícitamente el documento, los servidores web normalmente devuelven el documento "index.html" dentro de ese directorio.
hreflang	Especifica el idioma del recurso vinculado por el atributo <code>href</code> (que debe estar presente con este). Utilice los valores de idioma de <a href="#">BCP 47</a> para HTML5 y <a href="#">RFC 1766</a> para HTML 4.
rel	Especifica la relación entre el documento actual y el documento vinculado. Para HTML5, los valores deben <a href="#">definirse en la especificación</a> o <a href="#">registrarse en la wiki de Microformats</a> .
target	Especifica dónde abrir el enlace, por ejemplo, en una nueva pestaña o ventana. Los valores posibles son <code>_blank</code> , <code>_self</code> , <code>_parent</code> , <code>_top</code> y <code>framename</code> (en desuso). No se recomienda forzar dicho comportamiento, ya que viola el control del usuario sobre un sitio web.
title	Especifica información extra sobre un enlace. La información se muestra con mayor frecuencia como un texto de información sobre herramientas cuando el cursor se mueve sobre el enlace. Este atributo no está restringido a enlaces, se puede usar en casi todas las etiquetas HTML.
download	Especifica que el destino se descargará cuando un usuario haga clic en el hipervínculo. El valor del atributo será el nombre del archivo descargado. No

Parámetro	Detalles
	hay restricciones en los valores permitidos, y el navegador detectará automáticamente la extensión de archivo correcta y la agregará al archivo (.img, .pdf, etc.). Si se omite el valor, se usa el nombre de archivo original.

## Examples

### Enlace a otro sitio

Este es el uso básico del elemento `<a>` ( [un elemento nchor](#) ) :

```
<a href="http://example.com/">Link to example.com</a>
```

Crea un hipervínculo a la URL `http://example.com/` según lo especificado por el atributo `href` (referencia de hipertexto), con el texto de enlace "Vincular a example.com". Se vería algo como lo siguiente:

[Enlace a example.com](http://example.com/)

Para indicar que este enlace lleva a un sitio web externo, puede utilizar el tipo de enlace `external` :

```
<a href="http://example.com/" rel="external">example site</a>
```

Puede vincular a un sitio que utiliza un protocolo distinto de HTTP. Por ejemplo, para enlazar a un sitio FTP, puede hacer,

```
<a href="ftp://example.com/">This could be a link to a FTP site</a>
```

En este caso, la diferencia es que esta etiqueta de anclaje solicita que el navegador del usuario se conecte a `example.com` mediante el Protocolo de transferencia de archivos (FTP) en lugar del Protocolo de transferencia de hipertexto (HTTP).

[Esto podría ser un enlace a un sitio FTP](#)

### Abrir enlace en una nueva pestaña / ventana

```
<a href="example.com" target="_blank">Text Here</a>
```

El atributo de `target` especifica dónde abrir el enlace. Al configurarlo en `_blank` , le indica al navegador que lo abra en una nueva pestaña o ventana (según la preferencia del usuario).

### SEGURIDAD VULNERABILIDAD ADVERTENCIA!

El uso de `target="_blank"` le da al sitio de apertura acceso parcial al objeto `window.opener` través de JavaScript, lo que le permite a esa página acceder y cambiar

la `window.opener.location` de *su* página y potencialmente redirigir a los usuarios a sitios de malware o phishing.

Cuando utilice esto para las páginas que no controla, agregue `rel="noopener"` a su enlace para evitar que el objeto `window.opener` se envíe con la solicitud.

Actualmente, Firefox no admite `noopener`, por lo que deberá usar `rel="noopener noreferrer"` para obtener el máximo efecto.

## Enlace a un ancla

Los anclajes se pueden usar para saltar a etiquetas específicas en una página HTML. La etiqueta `<a>` puede apuntar a cualquier elemento que tenga un atributo `id`. Para obtener más información sobre las identificaciones, visite la [documentación sobre Clases e identificaciones](#). Los anclajes se usan principalmente para saltar a una subsección de una página y se usan junto con las etiquetas de encabezado.

Supongamos que ha creado una página ( `page1.html` ) sobre muchos temas:

```
<h2>First topic</h2>
<p>Content about the first topic</p>
<h2>Second topic</h2>
<p>Content about the second topic</p>
```

Una vez que tenga varias secciones, es posible que desee crear una Tabla de contenido en la parte superior de la página con enlaces rápidos (o marcadores) a secciones específicas.

Si asignó un atributo de `id` a sus temas, podría vincularlos

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

Ahora puedes usar el ancla en tu tabla de contenido:

```
<h1>Table of Contents</h1>
  <a href='#Topic1'>Click to jump to the First Topic</a>
  <a href='#Topic2'>Click to jump to the Second Topic</a>
```

Estos anclajes también se adjuntan a la página web en la que se encuentran ( `page1.html` ). Por lo tanto, puede enlazar el sitio de una página a otra haciendo referencia a la página y *al* nombre del ancla.

```
Remember, you can always <a href="page1.html#Topic1">look back in the First Topic</a> for supporting information.
```

## Enlace que ejecuta JavaScript



Simplemente use el protocolo `javascript:` para ejecutar el texto como JavaScript en lugar de abrirlo como un enlace normal:

```
<a href="javascript:myFunction();">Run Code</a>
```

También puedes lograr lo mismo usando el atributo `onclick` :

```
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

El `return false;` es necesario para evitar que su página se desplace hacia la parte superior cuando se hace clic en el enlace a `#` . Asegúrese de incluir todo el código que le gustaría ejecutar antes, ya que la devolución detendrá la ejecución de otro código.

También cabe destacar, puede incluir un signo de exclamación `!` después del hashtag para evitar que la página se desplace hacia arriba. Esto funciona porque cualquier barra no válida hará que el enlace no se desplace a *ninguna parte* de la página, porque no pudo ubicar el elemento al que hace referencia (un elemento con `id="!"` ). También puedes usar cualquier slug no válido (como `#scrollsNowhere` ) para lograr el mismo efecto. En este caso, `return false;` no es requerido:

```
<a href="#!" onclick="myFunction();">Run Code</a>
```

### ¿Deberías estar usando algo de esto?

La respuesta es casi seguro que **no** . Ejecutar JavaScript en línea con el elemento como este es una práctica bastante mala. Considere utilizar soluciones de JavaScript puras que busquen el elemento en la página y vincule una función a él en su lugar.

[Escuchando un evento](#)

También considere si este elemento es realmente un *botón* en lugar de un *enlace* . Si es así, debes usar `<button>` .

## Enlace a una página en el mismo sitio

Puede utilizar una [ruta relativa](#) para vincular a las páginas en el mismo sitio web.

```
<a href="/example">Text Here</a>
```

El ejemplo anterior iría al `example` archivo en el directorio raíz ( `/` ) del servidor.

---

Si este enlace estuviera en <http://example.com> , los siguientes dos enlaces llevarían al usuario a la misma ubicación

```
<a href="/page">Text Here</a>
<a href="http://example.com/page">Text Here</a>
```

Ambos de los anteriores irán al archivo de la `page` en el directorio raíz de `example.com` .

## Enlace que ejecuta el cliente de correo electrónico.

### Uso básico

Si el valor del atributo `href` comienza con `mailto:` intentará abrir un cliente de correo electrónico al hacer clic:

```
<a href="mailto:example@example.com">Send email</a>
```

Esto colocará la dirección de correo electrónico `example@example.com` como el destinatario del correo electrónico recién creado.

---

### Cc y Bcc

También puede agregar direcciones para los destinatarios de `cc` o `bcc` usando la siguiente sintaxis:

```
<a href="mailto:example@example.com?cc=john@example.com&bcc=jane@example.com">Send email</a>
```

### Texto del asunto y del cuerpo

También puede rellenar el asunto y el cuerpo del nuevo correo electrónico:

```
<a href="mailto:example@example.com?subject=Example+subject&body=Message+text">Send email</a>
```

Esos valores deben estar [codificados en URL](#) .

---

Al hacer clic en un enlace con `mailto:` intentará abrir el cliente de correo electrónico predeterminado especificado por su sistema operativo o le pedirá que elija qué cliente desea usar. No todas las opciones especificadas después de la dirección del destinatario son compatibles con todos los clientes de correo electrónico.

## Enlace que marca un número.

Si el valor del atributo `href` comienza con `tel:` su dispositivo marcará el número cuando haga clic en él. Esto funciona en dispositivos móviles o en computadoras / tabletas que ejecutan software, como Skype o FaceTime, que pueden hacer llamadas telefónicas.

```
<a href="tel:11234567890">Call us</a>
```

La mayoría de los dispositivos y programas le pedirán al usuario que confirme el número que está a punto de marcar.

Lea Anclas e hipervínculos en línea: <https://riptutorial.com/es/html/topic/254/anclas-e-hipervinculos>

---

# Capítulo 3: ARIA

## Sintaxis

- aria-live
- aria-relevante
- aria-autocompletar
- revisado por aria
- aria-desactivado
- aria expandida
- aria-haspopup
- aria escondido
- aria-invalido
- aria-label
- nivel de aria
- aria-multilínea
- aria-multiselectable
- orientación aria
- aria presionada
- aria-readonly
- aria requerido
- aria seleccionado
- aria-sort
- aria-valuemax
- aria-valuemin
- aria-valuenow
- aria-valuetext
- ariaatómico
- aria ocupado
- efecto aria-drop
- aria arrastrado
- aria-activedescendant
- aria-controles
- aria-descrita por
- aria-flowto
- aria-labelledby
- aria-posee
- aria-posinset
- aria-setsize

## Observaciones

ARIA es una especificación para describir semánticamente aplicaciones web enriquecidas. Seguir los estándares de ARIA puede aumentar la accesibilidad para aquellos que utilizan tecnologías de

asistencia (como un lector de pantalla) para acceder a su contenido.

## Examples

### role = "alerta"

Un mensaje con información importante, y generalmente sensible al tiempo.

```
<div role="alert" aria-live="assertive">Your session will expire in 60 seconds.</div>
```

Tenga en cuenta que he incluido tanto `role="alert"` como `aria-live="assertive"` al mismo tiempo. Estos son atributos sinónimos, pero algunos lectores de pantalla solo admiten uno u otro. Al usar ambos simultáneamente, por lo tanto, maximizamos las posibilidades de que la región en vivo funcione como se espera.

Fuente - Heydon Pickering '[Algunos ejemplos prácticos de ARIA](#)'

### role = "alertdialog"

Un tipo de diálogo que contiene un mensaje de alerta, donde el enfoque inicial va a un elemento dentro del diálogo.

```
<div role="alertdialog">
  <h1>Warning</h1>
  <div role="alert">Your session will expire in 60 seconds.</div>
</div>
```

### role = "application"

Una región declarada como una aplicación web, a diferencia de un documento web. En este ejemplo, la aplicación es una calculadora simple que podría sumar dos números juntos.

```
<div role="application">
  <h1>Calculator</h1>
  <input id="num1" type="text"> + <input id="num2" type="text"> =
  <span id="result"></span>
</div>
```

### role = "artículo"

Una sección de una página que consta de una composición que forma parte independiente de un documento, página o sitio.

---

Establecer una función de ARIA y / o un atributo `aria-*` que coincida con la semántica implícita predeterminada de ARIA no es necesario y no se recomienda ya que estas propiedades ya están establecidas por el navegador.

```
<article>
  <h1>My first article</h1>
  <p>Lorem ipsum...</p>
</article>
```

Usaría `role=article` en elementos no semánticos (no recomendado, no válido)

```
<div role="article">
  <h1>My first article</h1>
  <p>Lorem ipsum...</p>
</div>
```

W3C Entry para [role=article](#)

## role = "banner"

Una región que contiene principalmente contenido orientado al sitio, en lugar de contenido específico de la página.

```
<div role="banner">
  <h1>My Site</h1>

  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</div>
```

## role = "button"

Una entrada que permite acciones activadas por el usuario cuando se hace clic o se presiona.

```
<button role="button">Add</button>
```

## role = "cell"

Una celda en un contenedor tabular.

```
<table>
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <td role="cell">95</td>
    <td role="cell">14</td>
    <td role="cell">25</td>
  </tbody>
</table>
```

## role = "checkbox"

Una entrada verificable que tiene tres valores posibles: verdadero, falso o mixto.

```
<p>
  <input type="checkbox" role="checkbox" aria-checked="false">
  I agree to the terms
</p>
```

## role = "columnheader"

Una celda que contiene información de encabezado para una columna.

```
<table role="grid">
  <thead>
    <tr>
      <th role="columnheader">Day 1</th>
      <th role="columnheader">Day 2</th>
      <th role="columnheader">Day 3</th>
    </tr>
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

## role = "combobox"

Una presentación de un selecto; por lo general, es similar a un cuadro de texto donde los usuarios pueden escribir antes para seleccionar una opción o escribir para ingresar texto arbitrario como un nuevo elemento en la lista.

```
<input type="text" role="combobox" aria-expanded="false">
```

Normalmente, usaría JavaScript para construir el resto de la funcionalidad de selección de lista o encabezado de tipo.

## role = "complementario"

Una sección de soporte del documento, diseñada para ser complementaria al contenido principal en un nivel similar en la jerarquía DOM, pero sigue siendo significativa cuando se separa del contenido principal.

```
<div role="complementary">
  <h2>More Articles</h2>

  <ul>
    <!-- etc -->
  </ul>
</div>
```

## role = "contentinfo"

Una gran región perceptible que contiene información sobre el documento padre.

```
<p role="contentinfo">
  Author: Albert Einstein<br>
  Published: August 15, 1940
</p>
```

## role = "definición"

Una definición de un término o concepto.

```
<span role="term" aria-labelledby="def1">Love</span>
<span id="def1" role="definition">an intense feeling of deep affection.</span>
```

## role = "dialog"

Un diálogo es una ventana de aplicación que está diseñada para interrumpir el procesamiento actual de una aplicación con el fin de solicitar al usuario que ingrese información o que requiera una respuesta.

```
<div role="dialog">
  <p>Are you sure?</p>
  <button role="button">Yes</button>
  <button role="button">No</button>
</div>
```

## role = "directorio"

Una lista de referencias a miembros de un grupo, como una tabla de contenido estática.

```
<ul role="directory">
  <li><a href="/chapter-1">Chapter 1</a></li>
  <li><a href="/chapter-2">Chapter 2</a></li>
  <li><a href="/chapter-3">Chapter 3</a></li>
</ul>
```

## role = "documento"

Una región que contiene información relacionada que se declara como contenido del documento, a diferencia de una aplicación web.

```
<div role="document">
  <h1>The Life of Albert Einstein</h1>
  <p>Lorem ipsum...</p>
</div>
```

## role = "form"

Una región de hito que contiene una colección de elementos y objetos que, en conjunto, se

combinan para crear un formulario.

El uso del elemento HTML semánticamente correcto `<form>` implica la semántica ARIA predeterminada, lo que significa que `role=form` no es obligatorio, ya que no debe aplicar un rol de contraste a un elemento que ya es semántico, ya que agregar un rol invalida la semántica nativa de un elemento.

Establecer una función de ARIA y / o un atributo `aria-*` que coincida con la semántica implícita predeterminada de ARIA no es necesario y no se recomienda ya que estas propiedades ya están establecidas por el navegador.

```
<form action="">
  <fieldset>
    <legend>Login form</legend>
    <div>
      <label for="username">Your username</label>
      <input type="text" id="username" aria-describedby="username-tip" required />
      <div role="tooltip" id="username-tip">Your username is your email address</div>
    </div>
    <div>
      <label for="password">Your password</label>
      <input type="text" id="password" aria-describedby="password-tip" required />
      <div role="tooltip" id="password-tip">Was emailed to you when you signed up</div>
    </div>
  </fieldset>
</form>
```

Usaría `role=form` en elementos no semánticos (no recomendado, no válido)

```
<div role=form>
  <input type="email" placeholder="Your email address">
  <button>Sign up</button>
</div>
```

## role = "grid"

Una cuadrícula es un control interactivo que contiene celdas de datos tabulares organizados en filas y columnas, como una tabla.

```
<table role="grid">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

## role = "gridcell"

Una celda en una cuadrícula o treegrid.



```
<table role="grid">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <tr>
      <td role="gridcell">17</td>
      <td role="gridcell">64</td>
      <td role="gridcell">18</td>
    </tr>
  </tbody>
</table>
```

## role = "grupo"

Un conjunto de objetos de la interfaz de usuario que no están destinados a ser incluidos en un resumen de página o tabla de contenido por tecnologías de asistencia.

```
<div role="group">
  <button role="button">Previous</button>
  <button role="button">Next</button>
</div>
```

## role = "encabezado"

Un encabezado para una sección de la página.

```
<h1 role="heading">Introduction</h1>
<p>Lorem ipsum...</p>
```

## role = "img"

Un contenedor para una colección de elementos que forman una imagen.

```
<figure role="img">
  
  <figcaption>This is my cat, Albert.</figcaption>
</figure>
```

## role = "link"

Una referencia interactiva a un recurso interno o externo que, cuando se activa, hace que el agente de usuario navegue a ese recurso.

En la mayoría de los casos, no es necesario establecer una función de ARIA y / o un atributo aria- \* que coincida con la [semántica implícita predeterminada de ARIA](#), ya que estas propiedades ya están establecidas por el navegador.

Fuente: <https://www.w3.org/TR/html5/dom.html#aria-usage-note>

## role = "list"

Un grupo de elementos de lista no interactivos.

```
<ul role="list">
  <li role="listitem">One</li>
  <li role="listitem">Two</li>
  <li role="listitem">Three</li>
</ul>
```

## role = "listbox"

Un widget que permite al usuario seleccionar uno o más elementos de una lista de opciones.

```
<ul role="listbox">
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
</ul>
```

Normalmente, usaría JavaScript para construir la funcionalidad de selección múltiple.

## role = "listitem"

Un solo elemento en una lista o directorio.

```
<ul role="list">
  <li role="listitem">One</li>
  <li role="listitem">Two</li>
  <li role="listitem">Three</li>
</ul>
```

## role = "log"

Un tipo de región en vivo donde la nueva información se agrega en un orden significativo y la información antigua puede desaparecer.

```
<ul role="log">
  <li>User 1 logged in.</li>
  <li>User 2 logged in.</li>
  <li>User 1 logged out.</li>
</ul>
```

## role = "main"

El contenido principal de un documento.

```
<!-- header & nav here -->
<div role="main">
  <p>Lorem ipsum...</p>
```

```
</div>
<!-- footer here -->
```

## role = "marquee"

Un tipo de región en vivo donde la información no esencial cambia con frecuencia.

```
<ul role="marquee">
  <li>Dow +0.26%</li>
  <li>Nasdaq +0.54%</li>
  <li>S&amp;P +0.44%</li>
</ul>
```

## role = "math"

Contenido que representa una expresión matemática.

```

```

## role = "menu"

Un tipo de widget que ofrece una lista de opciones para el usuario.

```
<ul role="menu">
  <li role="menuitem">New</li>
  <li role="menuitem">Open</li>
  <li role="menuitem">Save</li>
  <li role="menuitem">Close</li>
</ul>
```

## role = "menubar"

Una presentación de menú que generalmente permanece visible y generalmente se presenta horizontalmente.

```
<ul role="menubar">
  <li role="menuitem">File</li>
  <li role="menuitem">Edit</li>
  <li role="menuitem">View</li>
  <li role="menuitem">Help</li>
</ul>
```

## role = "menuitem"

Una opción en un grupo de opciones contenidas en un menú o barra de menú.

```
<ul role="menubar">
  <li role="menuitem">File</li>
  <li role="menuitem">Edit</li>
  <li role="menuitem">View</li>
</ul>
```

```
<li role="menuitem">Help</li>
</ul>
```

## role = "menuitemcheckbox"

Un menuitem comprobable que tiene tres valores posibles: verdadero, falso o mixto.

```
<ul role="menu">
  <li role="menuitem">Console</li>
  <li role="menuitem">Layout</li>
  <li role="menuitemcheckbox" aria-checked="true">Word wrap</li>
</ul>
```

## role = "menuitemradio"

Un menuitem comprobable en un grupo de roles de menuitemradio, solo uno de los cuales se puede verificar a la vez.

```
<ul role="menu">
  <li role="menuitemradio" aria-checked="true">Left</li>
  <li role="menuitemradio" aria-checked="false">Center</li>
  <li role="menuitemradio" aria-checked="false">Right</li>
</ul>
```

## role = "navegación"

Una colección de elementos de navegación (generalmente enlaces) para navegar por el documento o documentos relacionados.

```
<ul role="navigation">
  <li><a href="/">Home</a></li>
  <li><a href="/about">About</a></li>
  <li><a href="/contact">Contact</a></li>
</ul>
```

## role = "note"

Una sección cuyo contenido es parentético o auxiliar al contenido principal del recurso.

```
<p>Lorem ipsum...</p>
<p>Lorem ipsum...</p>
<p role="note">Lorem ipsum...</p>
```

## role = "option"

Un elemento seleccionable en una lista de selección.

```
<ul role="listbox">
  <li role="option">Option 1</li>
```

```
<li role="option">Option 2</li>
<li role="option">Option 3</li>
</ul>
```

## role = "presentación"

Un elemento cuya semántica implícita de rol nativo no se asignará a la API de accesibilidad.

```
<div style="float:left;">Some content on the left.</div>
<div style="float:right;">Some content on the right</div>
<div role="presentation" style="clear:both;"></div> <!-- Only used to clear floats -->
```

## role = "barra de progreso"

Un elemento que muestra el estado de progreso de las tareas que llevan mucho tiempo.

```
<progress role="progressbar" value="25" max="100">25%</progress>
```

## role = "radio"

Una entrada que se puede verificar en un grupo de roles de radio, solo uno de los cuales se puede verificar a la vez.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> One<br>
  <input role="radio" type="radio" aria-checked="false"> Two<br>
  <input role="radio" type="radio" aria-checked="false"> Three
</div>
```

## role = "region"

Una gran sección perceptible de una página web o documento, que el autor considera que es lo suficientemente importante como para incluirla en un resumen de la página o en una tabla de contenido, por ejemplo, un área de la página que contiene estadísticas de eventos deportivos en vivo.

```
<div role="region">
  Home team: 4<br>
  Away team: 2
</div>
```

## role = "radiogroup"

Un grupo de botones de radio.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> One<br>
  <input role="radio" type="radio" aria-checked="false"> Two<br>
  <input role="radio" type="radio" aria-checked="false"> Three
</div>
```

```
</div>
```

## **role = "row"**

Una fila de celdas en un contenedor tabular.

```
<table>
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <tr role="row">
      <!-- etc -->
    </tr>
  </tbody>
</table>
```

## **role = "rowgroup"**

Un grupo que contiene uno o más elementos de fila en una cuadrícula.

```
<table>
  <thead role="rowgroup">
    <!-- etc -->
  </thead>
  <tbody role="rowgroup">
    <!-- etc -->
  </tbody>
</table>
```

## **role = "rowheader"**

Una celda que contiene información de encabezado para una fila en una cuadrícula.

```
<table role="grid">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <tr>
      <th role="rowheader">Day 1</th>
      <td>65</td>
    </tr>
    <tr>
      <th role="rowheader">Day 2</th>
      <td>74</td>
    </tr>
  </tbody>
</table>
```

## **role = "scrollbar"**

Un objeto gráfico que controla el desplazamiento del contenido dentro de un área de

visualización, independientemente de si el contenido se muestra completamente dentro del área de visualización.

```
<div id="content1">Lorem ipsum...</div>
<div
  role="scrollbar"
  aria-controls="content1"
  aria-orientation="vertical"
  aria-valuemax="100"
  aria-valuemin="0"
  aria-valuenow="25">
  <div class="scrollhandle"></div>
</div>
```

## role = "search"

Una región emblemática que contiene una colección de elementos y objetos que, en conjunto, se combinan para crear un servicio de búsqueda.

```
<div role="search">
  <input role="searchbox" type="text">
  <button role="button">Search</button>
</div>
```

## role = "searchbox"

Un tipo de cuadro de texto destinado a especificar criterios de búsqueda.

```
<div role="search">
  <input role="searchbox" type="text">
  <button role="button">Search</button>
</div>
```

## role = "separator"

Un divisor que separa y distingue secciones de contenido o grupos de menútems.

```
<p>Lorem ipsum...</p>
<hr role="separator">
<p>Lorem ipsum...</p>
```

## role = "control deslizante"

Una entrada de usuario donde el usuario selecciona un valor dentro de un rango determinado.

```
<div
  role="slider"
  aria-valuemax="100"
  aria-valuemin="0"
  aria-valuenow="25">
  <div class="sliderhandle"></div>
```

```
</div>
```

## role = "spinbutton"

Una forma de rango que espera que el usuario seleccione entre opciones discretas.

```
<input
  role="spinbutton"
  aria-valuemax="100"
  aria-valuemin="0"
  aria-valuenow="25"
  type="number"
  value="25">
```

## role = "status"

Un contenedor cuyo contenido es información de asesoramiento para el usuario, pero no es lo suficientemente importante como para justificar una alerta, pero a menudo se presenta como una barra de estado.

```
<div role="status">Online</div>
```

## role = "switch"

Un tipo de casilla de verificación que representa los valores de activación / desactivación, a diferencia de los valores marcados / desactivados.

```
<select role="switch" aria-checked="false">
  <option>On</option>
  <option selected>Off</option>
</select>
```

## role = "tab"

Una etiqueta de agrupación que proporciona un mecanismo para seleccionar el contenido de la pestaña que se va a representar al usuario.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
```

## role = "table"

Una sección que contiene datos ordenados en filas y columnas. La función de tabla está destinada a contenedores tabulares que no son interactivos.



```
<table role="table">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

## role = "tablist"

Una lista de elementos de tabulación, que son referencias a elementos de tabpanel.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
```

## role = "tabpanel"

Un contenedor para los recursos asociados con una pestaña, donde cada pestaña está contenida en un cuadro.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
<div role="tabpanel">
  <!-- etc -->
</div>
```

## role = "cuadro de texto"

Entrada que permite texto de forma libre como su valor.

```
<textarea role="textbox"></textarea>
```

## role = "timer"

Un tipo de región activa que contiene un contador numérico que indica la cantidad de tiempo transcurrido desde un punto de inicio, o el tiempo restante hasta un punto final.

```
<p>
  <span role="timer">60</span> seconds remaining.
</p>
```

## role = "toolbar"

Una colección de botones de función de uso común representados en forma visual compacta.

```
<ul role="toolbar">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

## role = "tooltip"

Una ventana emergente contextual que muestra una descripción de un elemento.

```
<span aria-describedby="slopedesc">Slope</span>
<div role="tooltip" id="slopedesc">y=mx+b</div>
```

Normalmente, la información sobre herramientas estaría oculta. Usando JavaScript, la información sobre herramientas se mostrará después de un retraso cuando el usuario se desplace sobre el elemento que describe.

## role = "árbol"

Un tipo de lista que puede contener grupos anidados de subnivel que se pueden contraer y expandir.

```
<ul role="tree">
  <li role="treeitem">
    Part 1
    <ul>
      <li role="treeitem">Chapter 1</li>
      <li role="treeitem">Chapter 2</li>
      <li role="treeitem">Chapter 3</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 2
    <ul>
      <li role="treeitem">Chapter 4</li>
      <li role="treeitem">Chapter 5</li>
      <li role="treeitem">Chapter 6</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 3
    <ul>
      <li role="treeitem">Chapter 7</li>
      <li role="treeitem">Chapter 8</li>
      <li role="treeitem">Chapter 9</li>
    </ul>
  </li>
</ul>
```

## role = "treegrid"

Una cuadrícula cuyas filas se pueden expandir y contraer de la misma manera que para un árbol.

## **role = "treeitem"**

Un elemento de opción de un árbol. Este es un elemento dentro de un árbol que puede expandirse o contraerse si contiene un grupo de subnivel de treeitems.

```
<ul role="tree">
  <li role="treeitem">
    Part 1
    <ul>
      <li role="treeitem">Chapter 1</li>
      <li role="treeitem">Chapter 2</li>
      <li role="treeitem">Chapter 3</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 2
    <ul>
      <li role="treeitem">Chapter 4</li>
      <li role="treeitem">Chapter 5</li>
      <li role="treeitem">Chapter 6</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 3
    <ul>
      <li role="treeitem">Chapter 7</li>
      <li role="treeitem">Chapter 8</li>
      <li role="treeitem">Chapter 9</li>
    </ul>
  </li>
</ul>
```

Lea ARIA en línea: <https://riptutorial.com/es/html/topic/2734/aria>

# Capítulo 4: Atributos de datos

## Sintaxis

- `<element data-custom-name="somevalue">`

## Parámetros

Valor	Descripción
algún valor	Especifica el valor del atributo (como una cadena)

## Examples

### Uso de atributos de datos

Los atributos de `data-*` HTML5 proporcionan una manera conveniente de almacenar datos en elementos HTML. Los datos almacenados se pueden leer o modificar usando JavaScript

```
<div data-submitted="yes" class="user_profile">
  ... some content ...
</div>
```

- La estructura del atributo de `data-*` es `data-*`, es decir, el nombre del atributo de datos viene después de la parte de `data-`. Usando este nombre, se puede acceder al atributo.
- Los datos en formato de cadena (incluido `json`) se pueden almacenar utilizando el atributo `data-*`.

### Soporte de navegadores antiguos

Los atributos de datos se introdujeron en HTML5, que es compatible con todos los navegadores modernos, pero los navegadores más antiguos antes de HTML5 no reconocen los atributos de datos.

Sin embargo, en las especificaciones HTML, los atributos que no son reconocidos por el navegador deben dejarse solos y el navegador simplemente los ignorará al renderizar la página.

Los desarrolladores web han utilizado este hecho para crear atributos no estándar que son atributos que no forman parte de las especificaciones HTML. Por ejemplo, el atributo de `value` en la línea siguiente se considera un atributo no estándar porque las especificaciones para la etiqueta `<img>` no tienen un atributo de `value` y no es un atributo global:

```

```

Esto significa que aunque los atributos de datos no son compatibles con los navegadores más antiguos, siguen funcionando y puede configurarlos y recuperarlos utilizando los mismos métodos genéricos de `setAttribute` y `getAttribute` de JavaScript, pero no puede usar la nueva propiedad de `dataset` que solo se admite en los navegadores modernos.

Lea Atributos de datos en línea: <https://riptutorial.com/es/html/topic/1182/atributos-de-datos>

# Capítulo 5: Atributos de evento HTML

## Examples

### Eventos de formulario HTML

Eventos desencadenados por acciones dentro de un formulario HTML (se aplica a casi todos los elementos HTML, pero es más utilizado en elementos de formulario):

Atributo	Descripción
en la falta de definición	Dispara el momento en que el elemento pierde foco.
onchange	Dispara el momento en que se cambia el valor del elemento.
oncontextmenu	Script que se ejecutará cuando se active un menú contextual
enfocado	Dispara el momento en que el elemento se enfoca.
oninput	Script que se ejecutará cuando un elemento obtenga la entrada del usuario
oninvalido	Script a ejecutar cuando un elemento no es válido
onreset	Se dispara cuando se hace clic en el botón Restablecer en un formulario
búsqueda	Se activa cuando el usuario escribe algo en un campo de búsqueda (para <code>&lt;input = "search"&gt;</code> )
onselect	Se dispara después de que algún texto haya sido seleccionado en un elemento.
enviar	Incendios cuando se envía un formulario

### Eventos del teclado

Atributo	Descripción
onkeydown	Se dispara cuando un usuario está presionando una tecla
onkeypress	Se dispara cuando un usuario presiona una tecla.
onkeyup	Se dispara cuando un usuario libera una tecla.

Lea Atributos de evento HTML en línea: <https://riptutorial.com/es/html/topic/10924/atributos-de-evento-html>

# Capítulo 6: Atributos globales

## Parámetros

Atributo	Descripción
<code>class</code>	Define uno o más nombres de clase para un elemento. Ver <a href="#">Clases e IDs</a> .
<code>contenteditable</code>	Establece si el contenido de un elemento puede ser editado.
<code>contextmenu</code>	Define un menú contextual que se muestra cuando un usuario hace clic con el botón derecho en un elemento.
<code>dir</code>	Establece la dirección del texto para el texto dentro de un elemento.
<code>draggable</code>	Establece si un elemento puede ser arrastrado.
<code>hidden</code>	Oculto un elemento que no está actualmente en uso en la página.
<code>id</code>	Define un identificador único para un elemento. Ver <a href="#">Clases e IDs</a> .
<code>lang</code>	Define el idioma del contenido de un elemento y sus valores de atributos de texto. Ver <a href="#">lenguajes de contenido</a> .
<code>spellcheck</code>	Establece si la ortografía / gramática verifica el contenido de un elemento.
<code>style</code>	Define un conjunto de estilos CSS en línea para un elemento.
<code>tabindex</code>	Establece el orden en que los elementos de una página se navegan mediante el método abreviado de teclado de la pestaña.
<code>title</code>	Define información adicional sobre un elemento, generalmente en forma de texto de información sobre herramientas en el mouseover.
<code>translate</code>	Define si traducir el contenido de un elemento.

## Observaciones

Los atributos globales simplemente se atribuyen y se pueden aplicar a cualquier elemento en todo el documento.

## Examples

### Atributo `contenteditable`



```
<p contenteditable>This is an editable paragraph.</p>
```

Al hacer clic en el párrafo, su contenido se puede editar de manera similar a un campo de texto de entrada.

Cuando el atributo `contenteditable` no se establece en un elemento, el elemento lo heredará de su elemento primario. Por lo tanto, todo el texto secundario de un elemento editable del contenido también será editable, pero *puede* desactivarlo para un texto específico, así:

```
<p contenteditable>
  This is an editable paragraph.
  <span contenteditable="false">But not this.</span>
</p>
```

Tenga en cuenta que un elemento de texto no editable dentro de un elemento editable también tendrá un cursor de texto como heredado de su padre también.

Lea Atributos globales en línea: <https://riptutorial.com/es/html/topic/2811/atributos-globales>

---

# Capítulo 7: Barras de navegación

## Examples

### Barra de navegación básica

Las barras de navegación son esencialmente una lista de enlaces, por lo que los elementos `ul` y `li` se utilizan para encerrar los enlaces de navegación.

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Contact</a></li>
</ul>
```

### Barra de navegación HTML5

Para hacer una barra de navegación usando el elemento de navegación HTML5, encierre los enlaces dentro de la etiqueta de navegación.

```
<nav>
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Contact</a>
</nav>
```

Lea Barras de navegación en línea: <https://riptutorial.com/es/html/topic/10725/barras-de-navegacion>

# Capítulo 8: Clases e identificaciones

## Introducción

Las clases y las identificaciones facilitan la referenciación de elementos HTML desde scripts y hojas de estilo. El atributo de clase se puede usar en una o más etiquetas y es usado por CSS para el estilo. Sin embargo, las ID están destinadas a referirse a un solo elemento, lo que significa que la misma ID nunca debe usarse dos veces. Las identificaciones se usan generalmente con JavaScript y enlaces de documentos internos, y no se recomiendan en CSS. Este tema contiene explicaciones y ejemplos útiles sobre el uso adecuado de los atributos de clase e ID en HTML.

## Sintaxis

- `class = "class1 class2 class3"`
- `id = "uniqueid"`

## Parámetros

Parámetro	Detalles
clase	Indica la Clase del elemento (no único)
carné de identidad	Indica el ID del elemento (único en el mismo contexto)

## Observaciones

- Tanto `class` como `id` son atributos globales y, por lo tanto, pueden asignarse a cualquier elemento HTML.
- Los nombres de las clases deben comenzar con una letra (AZ o az) y pueden ir seguidos de letras, dígitos, guiones y guiones bajos.
- En `HTML5`, los atributos de clase e id pueden usarse en cualquier elemento. En `HTML 4.0.1`, estaban fuera de los límites de las etiquetas `<base>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` y `<title>`.
- Un elemento puede tener una o más clases. Las clases están separadas por espacios y no pueden contener espacios.
- Un elemento solo puede tener una ID y debe ser único dentro de su contexto (es decir, una página web). Las identificaciones tampoco pueden contener espacios en sí mismos.

## Examples

### Dar un elemento a una clase

Las clases son identificadores de los elementos a los que están asignados. Utilice el atributo de

`class` para asignar una clase a un elemento.

```
<div class="example-class"></div>
```

Para asignar varias clases a un elemento, separe los nombres de las clases con espacios.

```
<div class="class1 class2"></div>
```

## Usando clases en CSS

Las clases se pueden usar para estilizar ciertos elementos sin cambiar todos los elementos de ese tipo. Por ejemplo, estos dos elementos de `span` pueden tener estilos completamente diferentes:

```
<span></span>  
<span class="special"></span>
```

Se pueden dar clases del mismo nombre a cualquier número de elementos en una página y todos recibirán el estilo asociado con esa clase. Esto siempre será cierto a menos que especifique el elemento dentro del CSS.

Por ejemplo, tenemos dos elementos, ambos con el `highlight` clase:

```
<div class="highlight">Lorem ipsum</div>  
<span class="highlight">Lorem ipsum</span>
```

Si nuestro CSS es el siguiente, el color verde se aplicará al texto dentro de ambos elementos:

```
.highlight { color: green; }
```

Sin embargo, si solo queremos apuntar `div` con el `highlight` clase, podemos agregar especificidad como a continuación:

```
div.highlight { color: green; }
```

Sin embargo, cuando se diseña con CSS, generalmente se recomienda que solo se `.highlight` clases (por ejemplo, `.highlight`) en lugar de elementos con clases (por ejemplo, `div.highlight`).

Como con cualquier otro selector, las clases pueden ser anidadas:

```
.main .highlight { color: red; } /* Descendant combinator */  
.footer > .highlight { color: blue; } /* Child combinator */
```

También puede encadenar el selector de clase para seleccionar solo elementos que tengan una combinación de varias clases. Por ejemplo, si este es nuestro HTML:

```
<div class="special left menu">This text will be pink</div>
```

Y queremos colorear esta pieza específica de texto de color rosa, podemos hacer lo siguiente en nuestro CSS:

```
.special.left.menu { color: pink; }
```

## Dando a un elemento una identificación

El atributo ID de un elemento es un identificador que debe ser único en todo el documento. Su propósito es identificar de forma única el elemento cuando se vincula (mediante un ancla), las secuencias de comandos o el estilo (con CSS).

```
<div id="example-id"></div>
```

No debe tener dos elementos con la misma ID en el mismo documento, incluso si los atributos están unidos a dos tipos diferentes de elementos. Por ejemplo, el siguiente código es incorrecto:

```
<div id="example-id"></div>
<span id="example-id"></span>
```

Los navegadores harán todo lo posible para representar este código, pero puede producirse un comportamiento inesperado al estilizar el estilo con CSS o agregar funcionalidad con JavaScript.

Para hacer referencia a los elementos por su ID en CSS, prefije el ID con # .

```
#example-id { color: green; }
```

Para saltar a un elemento con una ID en una página determinada, agregue # con el nombre del elemento en la URL.

```
http://example.com/about#example-id
```

Esta función es compatible con la mayoría de los navegadores y no requiere JavaScript o CSS adicional para funcionar.

## Problemas relacionados con identificaciones duplicadas

Tener más de un elemento con la misma ID es un problema difícil de solucionar. El analizador de HTML generalmente intentará renderizar la página en cualquier caso. Por lo general no se produce ningún error. Pero el ritmo podría terminar en una página web de mal comportamiento.

En este ejemplo:

```
<div id="aDiv">a</div>
<div id="aDiv">b</div>
```

Los selectores de CSS todavía funcionan

```
#aDiv {
  color: red;
}
```

Pero JavaScript no puede manejar ambos elementos:

```
var html = document.getElementById("aDiv").innerHTML;
```

En este caso, la variable `html` solo lleva el primer contenido `div ("a")` .

## Valores aceptables

### Para una identificación

5

Las únicas restricciones en el valor de una `id` son:

1. debe ser único en el documento
2. no debe contener caracteres de espacio
3. debe contener al menos un carácter

Por lo tanto, el valor puede ser todos los dígitos, solo un dígito, solo caracteres de puntuación, incluir caracteres especiales, lo que sea. Simplemente no hay espacios en blanco.

Así que estos son válidos:

```
<div id="container"> ... </div>
<div id="999"> ... </div>
<div id="#%LV-||"> ... </div>
<div id="____V"> ... </div>
<div id="[]"> ... </div>
<div id="♥">... </div>
<div id="{ ">... </div>
<div id="@"> ... </div>
<div id="☺☻☹☹~¥"> ... </div>
```

Esto no es válido:

```
<div id=" "> ... </div>
```

Esto también es inválido, cuando se incluye en el mismo documento:

```
<div id="results"> ... </div>
<div id="results"> ... </div>
```

4.01

Un valor de `id` debe comenzar con una letra, que solo puede ser seguida por:

- letras (AZ / az)

- dígitos (0-9)
- guiones ("-")
- guiones bajos ("\_")
- dos puntos (":")
- puntos (".")

Refiriéndose al primer grupo de ejemplos en la sección de HTML5 anterior, solo uno es válido:

```
<div id="container"> ... </div>
```

Estos también son válidos:

```
<div id="sampletext"> ... </div>  
<div id="sample-text"> ... </div>  
<div id="sample_text"> ... </div>  
<div id="sample:text"> ... </div>  
<div id="sample.text"> ... </div>
```

Nuevamente, si no comienza con una letra (mayúscula o minúscula), no es válido.

---

## Para una clase

Las reglas para las clases son esencialmente las mismas que para una `id`. La diferencia es que `class` valores de `class` *no* tienen que ser únicos en el documento.

Refiriéndose a los ejemplos anteriores, aunque esto no es válido en el mismo documento:

```
<div id="results"> ... </div>  
<div id="results"> ... </div>
```

Esto está perfectamente bien:

```
<div class="results"> ... </div>  
<div class="results"> ... </div>
```

---

## Nota importante: cómo los valores de ID y clase se tratan fuera de HTML

Tenga en cuenta que las reglas y los ejemplos anteriores se aplican dentro del contexto de HTML.

El uso de números, puntuación o caracteres especiales en el valor de una `id` o una `class` puede causar problemas en otros contextos, como CSS, JavaScript y expresiones regulares.

Por ejemplo, aunque la siguiente `id` es válida en HTML5:

```
<div id="9lions"> ... </div>
```

... no es válido en CSS:

### 4.1.3 Caracteres y estuche

En CSS, los *identificadores* (incluidos los nombres de elementos, las clases y las ID en los selectores) pueden contener solo los caracteres [a-zA-Z0-9] y los caracteres ISO 10646 U + 00A0 y superior, más el guión (-) y el guión bajo ( \_); **no pueden comenzar con un dígito, dos guiones o un guión seguido de un dígito** . (énfasis añadido)

En la mayoría de los casos, es posible que pueda escapar de los personajes en contextos donde tienen restricciones o un significado especial.

---

## Referencias W3C

- [3.2.5.1 El atributo `id`](#)
- [3.2.5.7 El atributo de `class`](#)
- [6.2 tipos básicos de SGML](#)

Lea Clases e identificaciones en línea: <https://riptutorial.com/es/html/topic/586/clases-e-identificaciones>



---

# Capítulo 9: Comentarios

## Introducción

Al igual que en otros lenguajes de programación, marcado y markdown, los comentarios en HTML brindan a otros desarrolladores información específica para el desarrollo sin afectar la interfaz del usuario. Sin embargo, a diferencia de otros idiomas, los comentarios HTML se pueden usar para especificar elementos HTML solo para Internet Explorer. Este tema explica cómo escribir comentarios HTML y sus aplicaciones funcionales.

## Sintaxis

- `<!-- Comment text -->`

## Observaciones

Cualquier cosa que comience con `<!--` y que termine con `-->` es un comentario. Los comentarios no pueden contener dos guiones adyacentes ( `--` ) y deben terminar con exactamente dos guiones (es decir, `---->` no es correcto).

Los comentarios no se pueden ver en una página web y no se pueden diseñar con CSS. El desarrollador de la página puede utilizarlos para tomar notas dentro del HTML o para ocultar cierto contenido durante el desarrollo.

Para páginas dinámicas o interactivas, ocultar y mostrar contenido se realiza con JavaScript y CSS en lugar de con comentarios HTML.

Se puede usar JavaScript para obtener el contenido de los nodos de comentarios HTML y estos nodos pueden crearse, agregarse y eliminarse dinámicamente del documento, pero esto no afectará la forma en que se muestra la página.

Dado que los comentarios HTML forman parte del código fuente de la página, se descargan en el navegador junto con el resto de la página. El código fuente normalmente se puede ver usando la opción de menú del navegador web para "Ver código fuente" o "Ver código fuente de página".

## Examples

### Creando comentarios

Los comentarios HTML se pueden usar para dejar notas a usted u otros desarrolladores sobre un punto específico en el código. Se pueden iniciar con `<!--` y concluir con `-->` , así:

```
<!-- I'm an HTML comment! -->
```

Se pueden incorporar en línea dentro de otros contenidos:

```
<h1>This part will be displayed <!-- while this will not be displayed -->.</h1>
```

También pueden abarcar varias líneas para proporcionar más información:

```
<!-- This is a multiline HTML comment.  
  Whatever is in here will not be rendered by the browser.  
  You can "comment out" entire sections of HTML code.  
-->
```

Sin embargo, no **pueden** aparecer dentro de otra etiqueta HTML, como esta:

```
<h1 <!-- testAttribute="something" -->>This will not work</h1>
```

Esto produce HTML no válido ya que todo el bloque `<h1 <!-- testAttribute="something" --> se` consideraría una sola etiqueta de inicio `h1` con alguna otra información no válida contenida en él, seguida de un solo `>` corchete de cierre que no hace nada.

Para la compatibilidad con herramientas que tratan de analizar HTML como XML o SGML, el cuerpo de su comentario no debe contener dos guiones `--`.

## Comentarios condicionales para Internet Explorer

Los comentarios condicionales se pueden usar para personalizar el código para diferentes versiones de Microsoft Internet Explorer. Por ejemplo, se pueden proporcionar diferentes clases de HTML, etiquetas de script o hojas de estilo. Los comentarios condicionales son compatibles con las versiones 5 a 9 de Internet Explorer. Las versiones anteriores y nuevas de Internet Explorer, y todos los navegadores que no son de IE, se consideran "de bajo nivel" y tratan los comentarios condicionales como comentarios HTML ordinarios.

### Bajo nivel oculto

Los comentarios ocultos de nivel inferior funcionan al encapsular todo el contenido dentro de lo que parece ser un comentario HTML normal. Solo IE 5 a 9 todavía lo leerá como un comentario condicional, y ocultarán o mostrarán el contenido en consecuencia. En otros navegadores se ocultará el contenido.

```
<!--[if IE]>  
  Revealed in IE 5 through 9. Commented out and hidden in all other browsers.  
<![endif]-->  
  
<!--[if lt IE 8]>  
  Revealed only in specified versions of IE 5-9 (here, IE less than 8).  
<![endif]-->  
  
<!--[if !IE]>  
  Revealed in no browsers. Equivalent to a regular HTML comment.  
<![endif]-->
```

```
<!--  
  For purposes of comparison, this is a regular HTML comment.  
-->
```

## Nivel bajo revelado

Estos son ligeramente diferentes a los comentarios ocultos en el nivel inferior: solo el comentario condicional está contenido dentro de la sintaxis de comentarios normal. Los navegadores que no admiten comentarios condicionales simplemente los ignorarán y mostrarán el resto del contenido entre ellos.

```
<!--[if IE]>-->  
  The HTML inside this comment is revealed in IE 5-9, and in all other browsers.  
<!--<![endif]-->  
  
<!--[if IE 9]>-->  
  This is revealed in specified versions of IE 5-9, and in all other browsers.  
<!--<![endif]-->  
  
<!--[if !IE]>-->  
  This is not revealed in IE 5-9. It's still revealed in other browsers.  
<!--<![endif]-->
```

## Comentando espacios en blanco entre elementos en línea

Los elementos de visualización en línea, normalmente como `span` o `a`, incluirán hasta un carácter de espacio en blanco antes y después de ellos en el documento. Para evitar líneas muy largas en el marcado (que son difíciles de leer) y espacios en blanco no intencionales (que afectan el formato), se puede comentar el espacio en blanco.

```
<!-- Use an HTML comment to nullify the newline character below: -->  
<a href="#">I hope there will be no extra whitespace after this!</a><!--  
--><button>Foo</button>
```

Pruébalo sin comentarios entre los elementos en línea, y habrá un espacio entre ellos. A veces se desea recoger el carácter de espacio.

Código de ejemplo:

```
<!-- Use an HTML comment to nullify the newline character below: -->  
<a href="#">I hope there will be no extra whitespace after this!</a><!--  
--><button>Foo</button>  
<hr>  
<!-- Without it, you can notice a small formatting difference: -->  
<a href="#">I hope there will be no extra whitespace after this!</a>  
<button>Foo</button>
```

Salida:

[I hope there will be no extra whitespace after this!](#)

---

[I hope there will be no extra whitespace after this!](#)

Lea Comentarios en línea: <https://riptutorial.com/es/html/topic/468/comentarios>

# Capítulo 10: Controles del menú de selección

## Sintaxis

- `<select name=""></select>`
- `<datalist id=""></datalist>`
- `<optgroup label="Option Group"></optgroup>`
- `<option value="">Option</option>`

## Examples

### Seleccione el menú

El elemento `<select>` genera un menú desplegable desde el cual el usuario puede elegir una opción.

```
<select name="">
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
  <option value="4">Four</option>
</select>
```

### Cambiando el tamaño

Puede cambiar el tamaño del menú de selección con el atributo de `size`. Un tamaño de 0 o 1 muestra el menú de estilo desplegable estándar. Un tamaño mayor que 1 convertirá el menú desplegable en un cuadro que muestra tantas líneas, con una opción por línea y una barra de desplazamiento para desplazarse por las opciones disponibles.

```
<select name="" size="4"></select>
```

### Menús de selección multi-opciones

Por defecto, los usuarios solo pueden seleccionar una sola opción. Agregar el atributo `multiple` permite a los usuarios seleccionar múltiples opciones a la vez y enviar todas las opciones seleccionadas con el formulario. El uso del atributo `multiple` convierte automáticamente el menú desplegable en un cuadro como si tuviera un tamaño definido. El tamaño predeterminado cuando esto ocurre está determinado por el navegador específico que está utilizando, y no es posible cambiarlo de nuevo a un menú de estilo desplegable mientras se permiten varias selecciones.

```
<select name="" multiple></select>
```

Cuando se usa el atributo `multiple`, hay una diferencia entre usar 0 y 1 para el tamaño, mientras que no existe diferencia cuando no se usa el atributo. El uso de 0 hará que el navegador se

comporte de la manera predeterminada para la que está programado. El uso de 1 establecerá explícitamente el tamaño del cuadro resultante en solo una fila.

## Grupos opcionales

Puede agrupar cuidadosamente sus opciones dentro de un menú de selección para proporcionar un diseño más estructurado en una larga lista de opciones utilizando el elemento `<optgroup>` .

La sintaxis es muy básica, simplemente usando el elemento con un atributo de `label` para identificar el título para el grupo, y que contiene cero o más opciones que deberían estar dentro de ese grupo.

```
<select name="">
  <option value="milk">Milk</option>
  <optgroup label="Fruits">
    <option value="banana">Bananas</option>
    <option value="strawberry">Strawberries</option>
  </optgroup>
  <optgroup label="Vegetables" disabled>
    <option value="carrot">Carrots</option>
    <option value="zucchini">Zucchini</option>
  </optgroup>
</select>
```

Cuando se usan grupos de opciones, no todas las opciones deben estar contenidas dentro de un grupo. Además, deshabilitar un grupo de opciones deshabilitará todas las opciones dentro del grupo, y no es posible volver a habilitar manualmente una sola opción dentro de un grupo deshabilitado.

## Opciones

Las opciones dentro de un menú de selección son las que el usuario seleccionará. La sintaxis normal para una opción es la siguiente:

```
<option>Some Option</option>
```

Sin embargo, es importante tener en cuenta que el texto dentro del elemento `<option>` sí no siempre se usa, y esencialmente se convierte en el valor predeterminado para los atributos que no se especifican.

Los atributos que controlan el aspecto real y la función de la opción son `value` y `label` . La etiqueta representa el texto que se mostrará en el menú desplegable (lo que está viendo y hará clic para seleccionarlo). El valor representa el texto que se enviará junto con el envío del formulario. Si se omite alguno de estos valores, en su lugar utiliza el texto dentro del elemento como el valor. Así que el ejemplo que dimos arriba podría ser "expandido" a esto:

```
<option label="Some Option" value="Some Option">
```

Tenga en cuenta la omisión del texto interno y la etiqueta final, que no se requieren para construir

una opción dentro del menú. Si se incluyeran, el texto interno se ignoraría porque ambos atributos ya están especificados y el texto no es necesario. Sin embargo, probablemente no verá a mucha gente escribirlos de esta manera. La forma más común en que se escribe es con un valor que se enviará al servidor, junto con el texto interno que eventualmente se convierte en el atributo de la etiqueta, de esta manera:

```
<option value="option1">Some Option</option>
```

## Seleccionando una opción por defecto

También puede especificar una determinada opción para ser seleccionada en el menú por defecto adjuntando el atributo `selected`. De forma predeterminada, si no se especifica ninguna opción como seleccionada en el menú, la primera opción en el menú se seleccionará cuando se renderice. Si más de una opción tiene el atributo `selected` adjunto, entonces la última opción presente en el menú con el atributo será la seleccionada de forma predeterminada.

```
<option value="option1" selected>Some option</option>
```

Si está utilizando el atributo en un menú de selección de múltiples opciones, entonces todas las opciones con el atributo se seleccionarán de forma predeterminada, y ninguna se seleccionará si ninguna opción tiene el atributo.

```
<select multiple>
  <option value="option1" selected>Some option</option>
  <option value="option2" selected>Some option</option>
</select>
```

## Lista de datos

La etiqueta `<datalist>` especifica una lista de opciones predefinidas para un elemento `<input>`. Proporciona una función "autocompletar" en `<input>` elementos `<input>`. Los usuarios verán una lista desplegable de opciones a medida que escriben.

```
<input list="Languages">

<datalist id="Languages">
  <option value="PHP">
  <option value="Perl">
  <option value="Python">
  <option value="Ruby">
  <option value="C+">
</datalist>
```

---

## Soporte del navegador

Cromo	Borde	Mozilla	Safari	Ópera
20.0	10.0	4.0	No soportado	9.0

Lea Controles del menú de selección en línea: <https://riptutorial.com/es/html/topic/722/controles-del-menu-de-seleccion>



---

# Capítulo 11: Cotizaciones de mercado

## Observaciones

`cite` elementos de `cite` y `blockquote` no deben utilizarse con el propósito de representar una conversación, transcripciones de conversaciones, diálogos en guiones, registros de mensajes instantáneos y otras situaciones en las que diferentes jugadores se turnan para hablar.

## Examples

### En línea con

El **elemento** `q` se puede usar para una cita que forma parte de una oración:

```
<p>She wrote <q>The answer is 42.</q> and everyone agreed.</p>
```

### Comillas

#### 4.01

Las comillas no deben añadirse. Los agentes de usuario deben (en HTML 4.01) resp. debe (en HTML 4.0) renderizarlos automáticamente.

#### 5

Las comillas no deben ser agregadas. Los agentes de usuario los renderizarán automáticamente.

### URL de origen (atributo de `cite` )

El **atributo** `cite` se puede usar para hacer referencia a la URL de la fuente citada:

```
<p>She wrote <q cite="http://example.com/blog/hello-world">The answer is 42.</q> and everyone agreed.</p>
```

Tenga en cuenta que los navegadores normalmente no muestran esta URL, por lo que si la fuente es relevante, debe agregar un hipervínculo ( `a` elemento) además.

### Bloque con

El **elemento** `blockquote` se puede usar para una cotización (a nivel de bloque):

```
<blockquote>
  <p>The answer is 42.</p>
</blockquote>
```

## URL de origen (atributo de `cite`)

El **atributo** `cite` se puede usar para hacer referencia a la URL de la fuente citada:

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
</blockquote>
```

Tenga en cuenta que los navegadores normalmente no muestran esta URL, por lo que si la fuente es relevante, debe agregar un hipervínculo ( a elemento) además (consulte la sección *Cita / Atribución* sobre dónde colocar este enlace).

## Citación / Atribución

### 4.01

La cita / atribución no debe formar parte del elemento `blockquote` :

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
</blockquote>
<p>Source: <cite><a href="http://example.com/blog/hello-world" rel="external">Hello
World</a></cite></p>
```

Puede agregar un elemento `div` para agrupar la cita y la cita, pero no existe manera de asociarlos semánticamente.

El **elemento** `cite` se puede usar para la referencia de la fuente citada (pero no para el nombre del autor).

### 5

La cita / atribución (p. Ej., El hipervínculo que proporciona la URL de origen) puede estar dentro de la cita en `blockquote` , pero en ese caso debe estar dentro de un elemento de `cite` (para atribuciones en el texto) o un elemento de `footer` :

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
  <footer>
    <p>Source: <cite><a href="http://example.com/blog/hello-world" rel="external">Hello
World</a></cite></p>
  </footer>
</blockquote>
```

El **elemento** `cite` se puede usar para la referencia de la fuente citada, o para el nombre del autor de la cita.

Lea Cotizaciones de marcado en línea: <https://riptutorial.com/es/html/topic/2943/cotizaciones-de-marcado>

---

# Capítulo 12: Doctypes

## Introducción

Tipos de documentos: abreviatura de "tipo de documento": ayuda a los navegadores a comprender la versión de HTML en la que está escrito el documento para una mejor interpretación. Las declaraciones de doctype no son etiquetas HTML y pertenecen a la parte superior de un documento. Este tema explica la estructura y declaración de varios tipos de documentos en HTML.

## Sintaxis

- `<!DOCTYPE [cadena específica de la versión]>`

## Observaciones

La declaración `<!DOCTYPE>` no es una etiqueta HTML. Se utiliza para especificar qué versión de HTML está utilizando el documento. Esto se conoce como la declaración de tipo de documento (DTD).

La declaración `<!DOCTYPE>` NO distingue entre mayúsculas y minúsculas. Para verificar si el HTML de sus páginas web es válido, vaya al [servicio de validación de W3C](#) .

- Algunas versiones anteriores de IE no admiten algunas etiquetas HTML a menos que haya disponible un tipo de documento adecuado.
- Es *vital* que un doctype se declare para asegurarse de que el navegador no use el modo de peculiaridades. [Más información en MDN](#).

## Examples

### Añadiendo el Doctype

La declaración `<!DOCTYPE>` siempre debe incluirse en la parte superior del documento HTML, antes de la etiqueta `<html>` .

5

Consulte [HTML 5 Doctype](#) para obtener detalles sobre el HTML 5 Doctype.

---

```
<!DOCTYPE html>
```

4.01

Consulte [HTML 4.01 Doctypes](#) para obtener detalles sobre cómo estos tipos difieren entre sí.

## Estricto

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

## Transicional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

## Conjunto de marcos

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

## HTML 4.01 Doctypes

La especificación HTML 4.01 proporciona varios tipos diferentes de tipos de documentos que permiten especificar diferentes tipos de elementos dentro del documento.

### HTML 4.01 estricto

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Incluye todos los elementos y atributos HTML, pero **no incluye elementos de presentación o desaprobados** y **no se permiten los conjuntos de marcos** .

### HTML 4.01 Transicional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Incluye todos los elementos y atributos HTML y los elementos de presentación y en desuso, pero **no se permiten los conjuntos de marcos** .

### HTML 4.01 conjunto de marcos

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Incluye todos los elementos y atributos HTML, elementos de presentación y desaprobados. Se permiten marcos de cuadros.

## HTML 5 Doctype

HTML5 no se basa en SGML y, por lo tanto, no requiere una referencia a una DTD.

Declaración de HTML 5 Doctype:

```
<!DOCTYPE html>
```

## Insensibilidad de casos

Según las [especificaciones de W3.org HTML 5 DOCTYPE](#) :

Un DOCTYPE debe constar de los siguientes componentes, en este orden:

1. Una cadena que es una coincidencia ASCII **que no distingue entre mayúsculas y minúsculas** para la cadena "`<!DOCTYPE`".

por lo tanto los siguientes DOCTYPE s también son válidos:

```
<!doctype html>  
<!dOCTyPe html>  
<!DocTYpe html>
```

Este artículo de SO analiza el tema ampliamente: ¿ [Mayúsculas o minúsculas?](#)

## Doctypes antiguos

### HTML 3.2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

HTML 3.2 es compatible con la mayoría de los navegadores en uso. Sin embargo, HTML 3.2 tiene soporte limitado para hojas de estilo y no admite funciones de HTML 4 como marcos e internacionalización.

### HTML 2.0

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

HTML 2.0 es ampliamente compatible con los navegadores, pero carece de soporte para tablas, marcos e internacionalización, así como muchos elementos y atributos de presentación de uso común.

Lea Doctypes en línea: <https://riptutorial.com/es/html/topic/806/doctypes>

# Capítulo 13: Elemento de etiqueta

## Sintaxis

- `<label>Example <input type="radio" name="r"></label>` // Envolviendo un elemento de control
- `<label for="rad1">Example</label> <input id="rad1" type="radio" name="r">` // Using for attribute

## Parámetros

Atributos	Descripción
para	Referencia al elemento de ID de destino. Es decir: <code>for="surname"</code>
formar	<b>HTML5 , [Obsoleto]</b> Referencia al formulario que contiene el elemento de destino. Se esperan elementos de etiqueta dentro de un elemento <code>&lt;form&gt;</code> . Si se proporciona el <code>form="someFormId"</code> esto le permite colocar la etiqueta en cualquier parte del documento.

## Examples

### Uso básico

Forma simple con etiquetas ...

```
<form action="/login" method="POST">

  <label for="username">Username:</label>
  <input id="username" type="text" name="username" />

  <label for="pass">Password:</label>
  <input id="pass" type="password" name="pass" />

  <input type="submit" name="submit" />

</form>
```

5

```
<form id="my-form" action="/login" method="POST">

  <input id="username" type="text" name="username" />

  <label for="pass">Password:</label>
  <input id="pass" type="password" name="pass" />

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

```
</form>

<label for="username" form="my-form">Username:</label>
```

## Sobre la etiqueta

El elemento `<label>` se utiliza para hacer referencia a un elemento de acción de formulario. En el ámbito de **la Interfaz de usuario**, se utiliza para facilitar el destino / la selección de elementos como la `radio` Tipo o la `checkbox` .

### `<label>` como envoltorio

Puede encerrar el elemento de acción deseado.

```
<label>
  <input type="checkbox" name="Cats">
  I like Cats!
</label>
```

(Al hacer clic en el texto, la `input` destino cambiará su estado / valor)

### `<label>` como referencia

Al usar el atributo `for` , no tiene que colocar el elemento de control como descendiente de la `label` , pero el valor `for` debe coincidir con su ID

```
<input id="cats" type="checkbox" name="Cats">
<label for="cats" >I like Cats!</label>
```

## Nota

No utilice más de un elemento de control dentro de un elemento `<label>`

Lea Elemento de etiqueta en línea: <https://riptutorial.com/es/html/topic/1704/elemento-de-etiqueta>

# Capítulo 14: Elemento de progreso

## Parámetros

Parámetro	Valor
max	¿Cuánto trabajo requiere la tarea en total?
valor	¿Cuánto del trabajo ya se ha realizado?
posición	Este atributo devuelve la posición actual del elemento <code>&lt;progress&gt;</code>
etiquetas	Este atributo devuelve una lista de etiquetas de elemento <code>&lt;progress&gt;</code> (si las hay)

## Observaciones

El elemento `<progress>` no es compatible con versiones de Internet Explorer de menos de 10

El elemento `<progress>` es el elemento incorrecto que se utiliza para algo que es solo un indicador, en lugar del progreso de la tarea. Por ejemplo, mostrar el uso del espacio en disco utilizando el elemento `<progress>` es inapropiado. En su lugar, el elemento `<meter>` está disponible para este tipo de casos de uso.

## Examples

### Progreso

El elemento `<progress>` es nuevo en HTML5 y se utiliza para representar el progreso de una tarea

```
<progress value="22" max="100"></progress>
```

Esto crea una barra llena de 22%.

### Cambiando el color de una barra de progreso.

Las barras de progreso se pueden diseñar con el selector de `progress[value]` .

Este ejemplo da una barra de progreso una anchura de 250px y una altura de 20px

```
progress[value] {  
  width: 250px;  
  height: 20px;  
}
```

Las barras de progreso pueden ser especialmente difíciles de diseñar.



## Chrome / Safari / Opera

Estos navegadores utilizan el `-webkit-appearance` para dar estilo a la etiqueta de progreso. Para anular esto, podemos restablecer la apariencia.

```
progress[value] {
  -webkit-appearance: none;
  appearance: none;
}
```

Ahora, podemos estilizar el propio contenedor.

```
progress[value]::-webkit-progress-bar {
  background-color: "green";
}
```

## Firefox

Firefox le da un estilo diferente a la barra de progreso. Tenemos que usar estos estilos.

```
progress[value] {
  -moz-appearance: none;
  appearance: none;
  border: none; /* Firefox also renders a border */
}
```

## explorador de Internet

Internet Explorer 10+ admite el elemento de `progress`. Sin embargo, no admite la propiedad de `background-color`. Tendrá que usar la propiedad de color en su lugar.

```
progress[value] {
  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none;

  border: none; /* Remove border from Firefox */

  width: 250px;
  height: 20px;

  color: blue;
}
```

## HTML Fallback

Para los navegadores que no admiten el elemento de `progress`, puede usar esto como solución alternativa.

```
<progress max="100" value="20">
```

```
<div class="progress-bar">
  <span style="width: 20%;">Progress: 20%</span>
</div>
</progress>
```

Los navegadores que admiten la etiqueta de progreso ignorarán el div anidado en el interior. Los navegadores heredados que no pueden identificar la etiqueta de progreso mostrarán el div en su lugar.

Lea Elemento de progreso en línea: <https://riptutorial.com/es/html/topic/5055/elemento-de-progreso>

# Capítulo 15: Elemento de salida

## Parámetros

Atributo	Descripción
Global	Atributos que están disponibles para cualquier elemento HTML5. Para obtener una documentación completa de estos atributos, consulte: <a href="#">Atributos globales de MDN</a>
nombre	Una cadena que representa el nombre de una salida. Como elemento de formulario, se puede hacer referencia a la salida por su nombre usando la propiedad <code>document.forms</code> . Este atributo también se utiliza para recopilar valores en un formulario enviado.
para	Una lista separada por espacios de los identificadores de elementos de formulario (por ejemplo, <code>&lt;inputs id="inp1"&gt; for value is "inp1" )</code> para la cual la salida debe mostrar los cálculos.
formar	Una cadena que representa la <code>&lt;form&gt;</code> que está asociada a la salida. Si la salida está realmente fuera de la <code>&lt;form&gt;</code> , este atributo asegurará que la salida aún pertenezca a la <code>&lt;form&gt;</code> y esté sujeta a las colecciones y envíos de dicha <code>&lt;form&gt;</code> .

## Examples

### Elemento de salida usando para y atributos de formulario

La siguiente demostración muestra el uso del elemento `<output>` de los atributos `[for]` y `[form]`. Tenga en cuenta que `<output>` **necesita JavaScript** para funcionar. El JavaScript en línea se usa comúnmente en formularios como lo demuestra este ejemplo. Aunque los elementos `<input>` son `type="number"`, sus `value`s no son números, son texto. Por lo tanto, si necesita que se calcule el `value`s, debe convertir cada `value` en un número utilizando métodos como: `parseInt()`, `parseFloat()`, `Number()`, etc.

### Demo en vivo

```
<!--form1 will collect the values of in1 and in2 on 'input' event.-->
<!--out1 value will be the sum of in1 and in2 values.-->

<form id="form1" name="form1" oninput="out1.value = parseInt(in1.value, 10) +
parseInt(in2.value, 10)">

  <fieldset>

    <legend>Output Example</legend>

    <input type="number" id="in1" name="in1" value="0">
```

```
<br/>
+
<input type="number" id="in2" name="in2" value="0">

</fieldset>

</form>

<!--[for] attribute enables out1 to display calculations for in1 and in2.-->
<!--[form] attribute designates form1 as the form owner of out1 even if it isn't a
descendant.-->

<output name="out1" for="in1 in2" form="form1">0</output>
```

## Elemento de salida con atributos

```
<output name="out1" form="form1" for="inp1 inp2"></output>
```

Lea Elemento de salida en línea: <https://riptutorial.com/es/html/topic/723/elemento-de-salida>

# Capítulo 16: Elemento div

## Introducción

El elemento div en HTML es un elemento contenedor que encapsula otros elementos y se puede utilizar para agrupar y separar partes de una página web. Un div en sí mismo no representa nada inherentemente, pero es una herramienta poderosa en el diseño web. Este tema cubre el propósito y las aplicaciones del elemento div.

## Sintaxis

- `<div>example div</div>`

## Examples

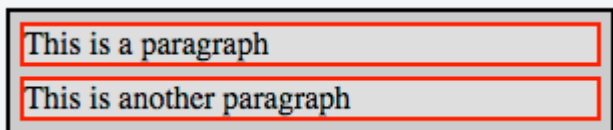
### Anidamiento

Es una práctica común colocar varios `<div>` dentro de otro `<div>` . Esto generalmente se conoce como elementos "anidados" y permite dividir los elementos en subsecciones o ayudar a los desarrolladores con el estilo CSS.

`<div class="outer-div">` se usa para agrupar dos elementos `<div class="inner-div">` ; cada uno contiene un elemento `<p>` .

```
<div class="outer-div">
  <div class="inner-div">
    <p>This is a paragraph</p>
  </div>
  <div class="inner-div">
    <p>This is another paragraph</p>
  </div>
</div>
```

Esto producirá el siguiente resultado (estilos CSS aplicados para mayor claridad):



**Anidado de elementos en línea y de bloque** Al anidar elementos, debe tener en cuenta que hay elementos en línea y de bloque. mientras que los elementos de bloque "agregan un salto de línea en el fondo", lo que significa que otros elementos anidados se muestran en la siguiente línea automáticamente, los elementos en línea se pueden colocar uno al lado del otro de forma predeterminada

**Evitar la anidación profunda** `<div>`

Un diseño de contenedor anidado profundo y de uso frecuente muestra un estilo de codificación incorrecto.

Las esquinas redondeadas o algunas funciones similares a menudo crean un código HTML de este tipo. Para la mayoría de los navegadores de última generación hay contrapartes de CSS3. Trate de usar lo menos posible los elementos HTML para aumentar el contenido a la relación de etiqueta y reducir la carga de la página, lo que resulta en una mejor clasificación en los motores de búsqueda.

`div` elemento `div` section no debe estar anidado a más de 6 capas.

## Uso básico

El elemento `<div>` generalmente no tiene un significado semántico específico por sí mismo, simplemente representa una división, y se usa normalmente para agrupar y encapsular otros elementos dentro de un documento HTML y separar aquellos de otros grupos de contenido. Como tal, cada `<div>` se describe mejor por su contenido.

```
<div>
  <p>Hello! This is a paragraph.</p>
</div>
```

El elemento `div` suele ser un [elemento de nivel de bloque](#), lo que significa que separa un bloque de un documento HTML y ocupa el ancho máximo de la página. Los navegadores suelen tener la siguiente regla de CSS predeterminada:

```
div {
  display: block;
}
```

---

**El World Wide Web Consortium (W3C)** lo recomienda encarecidamente a ver el elemento `div` como un elemento de último recurso, ya que cuando ningún otro elemento es adecuado. El uso de elementos más apropiados en lugar del elemento `div` conduce a una mejor accesibilidad para los lectores y una facilidad de mantenimiento más fácil para los autores.

Por ejemplo, una publicación de blog se marcaría con `<article>`, un capítulo con `<section>`, las ayudas de navegación de una página con `<nav>` y un grupo de controles de formulario con `<fieldset>`.

Los elementos `div` pueden ser útiles para propósitos estilísticos o para encerrar varios párrafos dentro de una sección que deben ser anotados de manera similar.

Lea Elemento `div` en línea: <https://riptutorial.com/es/html/topic/1468/elemento-div>

# Capítulo 17: Elementos de control de entrada

## Introducción

Un componente clave de los sistemas web interactivos, las etiquetas de entrada son elementos HTML diseñados para tomar una forma específica de entrada de los usuarios. Diferentes tipos de elementos de entrada pueden regular los datos ingresados para que se ajusten a un formato específico y brinden seguridad a la entrada de contraseña.

## Sintaxis

- `<input type="" name="" value="">`

## Parámetros

Parámetro	Detalles
clase	Indica la clase de la entrada
carné de identidad	Indica el ID de la entrada
tipo	Identifica el tipo de control de entrada a visualizar. Los valores aceptables están <code>hidden</code> , <code>text</code> , <code>tel</code> , <code>url</code> , <code>email</code> , <code>password</code> , <code>date</code> , <code>time</code> , <code>number</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> , <code>submit</code> , <code>image</code> , <code>reset</code> y <code>button</code> . El valor predeterminado es <code>text</code> si no se especifica, si el valor no es válido o si el navegador no admite el tipo especificado.
nombre	Indica el nombre de la entrada.
discapacitado	Valor booleano que indica que la entrada debe estar deshabilitada. Los controles deshabilitados no pueden editarse, no se envían al enviar el formulario y no pueden recibir el foco.
comprobado	Cuando el valor del atributo de tipo es <code>radio</code> o casilla de verificación, la presencia de este atributo booleano indica que el control está seleccionado de forma predeterminada; De lo contrario se ignora.
múltiple	<b>HTML5</b> indica que se pueden pasar varios archivos o valores (se aplica solo a las entradas de tipo de <code>file</code> y <code>email</code> )
marcador de posición	<b>HTML5</b> Una sugerencia para el usuario de lo que se puede ingresar en el control. El texto del marcador de posición no debe contener retornos de carro o saltos de línea
autocompletar	<b>HTML5</b> Indica si el navegador puede completar automáticamente el valor

Parámetro	Detalles
	del control.
solo lectura	Valor booleano que indica que la entrada no es editable. Los controles de solo lectura aún se envían en el envío del formulario, pero no recibirán el foco. <b>HTML5:</b> este atributo se ignora cuando el valor del atributo de <code>type</code> se establece en <code>hidden</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> o <code>button</code> .
necesario	<b>HTML5</b> Indica que un valor debe estar presente o que el elemento debe verificarse para que se envíe el formulario
alt	Un texto alternativo para las imágenes, en caso de que no se muestren.
enfoque automático	El elemento <code>&lt;input&gt;</code> debe obtener el foco cuando se carga la página.
valor	Especifica el valor del elemento <code>&lt;input&gt;</code> .
paso	El atributo de <code>step</code> especifica los intervalos de números legales. Funciona con los siguientes tipos de entrada: <code>number</code> , <code>range</code> , <code>date</code> , <code>datetime-local</code> , <code>month</code> , <code>time</code> y <code>week</code> .

## Observaciones

Al igual que con otros elementos de HTML5 void, `<input>` se cierra automáticamente y puede escribirse `<input />`. HTML5 no requiere esta barra.

Los siguientes son tipos de entrada válidos en HTML:

- `button`
- `checkbox`
- `file`
- `hidden`
- `image`
- `password`
- `radio`
- `reset`
- `submit`
- `text` (valor por defecto)

5

Los siguientes son tipos de entrada recientemente introducidos como parte del estándar HTML 5. Algunos de estos tipos no son compatibles con todos los navegadores web. En el caso de que un tipo no sea compatible, el elemento de entrada se establecerá de forma predeterminada en el tipo de `text`.

- `color`
- `date`
- `datetime`



(obsoleto y obsoleto)

- `datetime-local`
- `email`
- `month`
- `number`
- `range`
- `search`
- `tel`
- `time`
- `url`
- `week`

Para comprobar qué navegadores admiten qué tipos, puede ir a [caniuse.com](https://caniuse.com) .

## Examples

### Casilla de verificación y botones de radio

## Visión general

Las casillas de verificación y los botones de opción se escriben con la etiqueta HTML `<input>` , y su comportamiento se define en la [especificación HTML](#) .

La casilla de verificación o el botón de opción más simple es un elemento `<input>` con un atributo de `type` de `checkbox` o `radio` , respectivamente:

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

Se utiliza un único elemento de casilla de verificación independiente para una única opción binaria, como una pregunta de sí o no. Las casillas de verificación son independientes, lo que significa que el usuario puede seleccionar tantas opciones como desee en un grupo de casillas de verificación. En otras palabras, marcar una casilla de verificación *no* desactiva las otras casillas de verificación en el grupo de casillas de verificación.

Los botones de radio generalmente vienen en grupos (si no están agrupados con otro botón de radio, probablemente querías usar una casilla de verificación) identificados usando el mismo atributo de `name` en todos los botones dentro de ese grupo. La selección de los botones de opción *se excluyen mutuamente* , lo que significa que el usuario solo puede seleccionar una opción de un grupo de botones de opción. Cuando se marca un botón de opción, cualquier otro botón de opción con el mismo `name` que se verificó previamente se desmarca.

Ejemplo:

```
<input type="radio" name="color" id="red" value="#F00">
<input type="radio" name="color" id="green" value="#0F0">
<input type="radio" name="color" id="blue" value="#00F">
```

Cuando se ve, los botones de radio aparecen como un círculo (sin marcar) o un círculo relleno (marcado). Las casillas de verificación aparecen como un cuadrado (sin marcar) o como un cuadrado relleno (marcado). Dependiendo del navegador y del sistema operativo, el cuadrado a veces tiene esquinas redondeadas.

---

## Atributos

Las casillas de verificación y los botones de radio tienen una serie de atributos para controlar su comportamiento:

### `value`

Al igual que cualquier otro elemento de entrada, el atributo de `value` especifica el valor de cadena para asociar con el botón en caso de envío de formulario. Sin embargo, casillas de verificación y botones de radio son especiales ya que cuando se omite el valor, el valor predeterminado es `on` cuando se presentó, en lugar de enviar un valor en blanco. El atributo de `value` no se refleja en la apariencia del botón.

### `checked`

El atributo `checked` especifica el estado inicial de una casilla de verificación o botón de radio. Este es un atributo booleano y se puede omitir.

Cada una de estas son formas válidas y equivalentes de definir un botón de opción marcado:

```
<input checked>
<input checked="">
<input checked="checked">
<input checked="ChEcKeD">
```

La ausencia del atributo `checked` es la única sintaxis válida para un botón sin marcar:

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

Al restablecer un `<form>`, las casillas de verificación y los botones de radio vuelven al estado de su atributo `checked`.

---

## Accesibilidad

### Etiquetas

Para dar contexto a los botones y mostrar a los usuarios para qué sirve cada botón, cada uno de ellos debe tener una etiqueta. Esto se puede hacer usando un elemento `<label>` para envolver el botón. Además, esto hace que la etiqueta sea clickeable, por lo que selecciona el botón correspondiente.

## Ejemplo:

```
<label>
  <input type="radio" name="color" value="#F00">
  Red
</label>
```

o con un elemento `<label>` con un atributo `for` establecido en el atributo `id` del botón:

```
<input type="checkbox" name="color" value="#F00" id="red">
<label for="red">Red</label>
```

## Grupos de botones

Como cada botón de opción afecta a los otros del grupo, es común proporcionar una etiqueta o contexto para todo el grupo de botones de opción.

Para proporcionar una etiqueta para todo el grupo, los botones de `<fieldset>` deben incluirse en un elemento `<fieldset>` con un elemento `<legend>` dentro de él.

## Ejemplo:

```
<fieldset>
  <legend>Theme color:</legend>
  <p>
    <input type="radio" name="color" id="red" value="#F00">
    <label for="red">Red</label>
  </p>
  <p>
    <input type="radio" name="color" id="green" value="#0F0">
    <label for="green">Green</label>
  </p>
  <p>
    <input type="radio" name="color" id="blue" value="#00F">
    <label for="blue">Blue</label>
  </p>
</fieldset>
```

Las casillas de verificación también se pueden agrupar de manera similar, con un conjunto de campos y una leyenda que identifican el grupo de casillas de verificación relacionadas. Sin embargo, tenga en cuenta que las casillas de verificación *no* deben compartir el mismo nombre porque no se excluyen mutuamente. Al hacer esto, el formulario enviará múltiples valores para la misma clave y no todos los idiomas del lado del servidor lo manejarán de la misma manera (comportamiento indefinido). Cada casilla de verificación debe tener un nombre único o usar un conjunto de corchetes ( `[]` ) para indicar que el formulario debe enviar una matriz de valores para esa clave. El método que elija dependerá de cómo planea manejar los datos del formulario del lado del cliente o del lado del servidor. También debe mantener la leyenda corta, ya que algunas combinaciones de navegadores y lectores de pantalla leen la leyenda antes de cada campo de entrada en el campo.

## Oculto

```
<input type="hidden" name="inputName" value="inputValue">
```

Una entrada oculta no será visible para el usuario, pero su valor se enviará al servidor cuando se envíe el formulario.

## Contraseña

```
<input type="password" name="password">
```

El elemento de entrada con un atributo de tipo cuyo valor es `password` crea un campo de texto de una sola línea similar al `type=text` [entrada type=text](#), excepto que el texto no se muestra cuando el usuario lo ingresa.

```
<input type="password" name="password" placeholder="Password">
```

El texto del marcador de posición se muestra en texto sin formato y se sobrescribe automáticamente cuando un usuario comienza a escribir.

**Nota:** Algunos navegadores y sistemas modifican el comportamiento predeterminado del campo de contraseña para mostrar también el carácter escrito más recientemente por un breve período de tiempo, como por ejemplo:

## Enviar

```
<input type="submit" value="Submit">
```

Una entrada de envío crea un botón que envía el formulario que contiene cuando se hace clic.

También puede usar el elemento `<button>` si necesita un botón de envío que se pueda diseñar más fácilmente o que contenga otros elementos:

```
<button type="submit">  
   Submit  
</button>
```

## Expediente

```
<input type="file" name="fileSubmission">
```

Las entradas de archivo permiten a los usuarios seleccionar un archivo de su sistema de archivos local para usar con la página actual. Si se usan junto con un elemento de `form`, se pueden usar para permitir a los usuarios cargar archivos en un servidor (para obtener más información, consulte [Carga de archivos](#)).

El siguiente ejemplo les permite a los usuarios usar la entrada del `file` para seleccionar un archivo de su sistema de archivos y cargar ese archivo en un script en el servidor llamado `upload_file.php`.

```
<form action="upload_file.php" method="post" enctype="multipart/form-data">
  Select file to upload:
  <input type="file" name="fileSubmission" id="fileSubmission">
  <input type="submit" value="Upload your file" name="submit">
</form>
```

## Múltiples archivos

Agregando el atributo `multiple` el usuario podrá seleccionar **más de un** archivo:

```
<input type="file" name="fileSubmission" id="fileSubmission" multiple>
```

## Aceptar archivos

El atributo `Accept` especifica los tipos de archivos que el usuario puede seleccionar. Por ejemplo, `.png`, `.gif`, `.jpeg`.

```
<input type="file" name="fileSubmission" accept="image/x-png,image/gif,image/jpeg" />
```

## Validación de entrada

La validación de entrada HTML se realiza automáticamente por el navegador en función de atributos especiales en el elemento de entrada. Podría reemplazar parcial o completamente la validación de entrada de JavaScript. Este tipo de validación puede ser evitada por el usuario a través de solicitudes HTTP especialmente diseñadas, por lo que no reemplaza la validación de entrada del lado del servidor. La validación solo se produce cuando se intenta enviar el formulario, por lo que todas las entradas restringidas deben estar dentro de un formulario para que se produzca la validación (a menos que esté utilizando JavaScript). Tenga en cuenta que las entradas que están deshabilitadas o de solo lectura no activarán la validación.

Algunos tipos de entrada más nuevos (como `email`, `url`, `tel`, `date` y muchos más) se validan automáticamente y no requieren sus propias restricciones de validación.

5

## Necesario

Use el atributo `required` para indicar que se debe completar un campo para pasar la validación.

```
<input required>
```

## Longitud mínima / máxima

Utilice los `minlength` y `maxlength` para indicar los requisitos de longitud. La mayoría de los navegadores evitarán que el usuario escriba más de un *máximo de caracteres* en el cuadro, lo que evitará que su entrada sea inválida incluso antes de intentar el envío.

```
<input minlength="3">  
<input maxlength="15">  
<input minlength="3" maxlength="15">
```

## Especificando un rango

Use los atributos `min` y `max` para restringir el rango de números que un usuario puede ingresar en una entrada de `number` de tipo o `range`

```
Marks: <input type="number" size="6" name="marks" min="0" max="100" />  
Subject Feedback: <input type="range" size="2" name="feedback" min="1" max="5" />
```

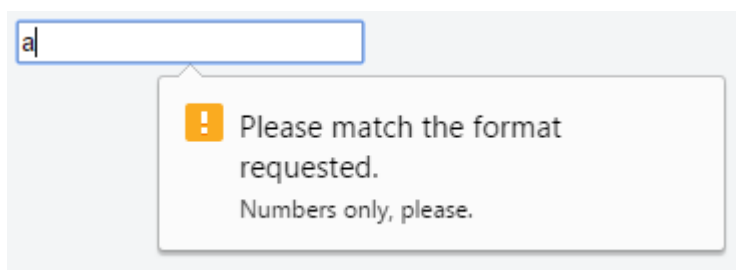
5

## Emparejar un patrón

Para obtener más control, use el atributo de `pattern` para especificar cualquier expresión regular que deba coincidir para pasar la validación. También puede especificar un `title`, que se incluye en el mensaje de validación si el campo no pasa.

```
<input pattern="\d*" title="Numbers only, please.">
```

Este es el mensaje que se muestra en Google Chrome versión 51 cuando intenta enviar el formulario con un valor no válido dentro de este campo:



No todos los navegadores muestran un mensaje para patrones no válidos, aunque hay soporte completo entre los navegadores modernos más usados.

Verifique el soporte más reciente en [CanIUse](#) e implemente en consecuencia.

5

## Aceptar tipo de archivo

Para los campos de entrada de tipo de `file` , es posible aceptar solo ciertos tipos de archivos, como videos, imágenes, audios, extensiones de archivo específicas o ciertos [tipos de medios](#) . Por ejemplo:

```
<input type="file" accept="image/*" title="Only images are allowed">
```

Se pueden especificar varios valores con una coma, por ejemplo:

```
<input type="file" accept="image/*,.rar,application/zip">
```

**Nota:** al agregar el atributo `novalidate` al elemento de `form` o `formnovalidate` atributo `formnovalidate` en el botón de envío, se evita la validación de los elementos de formulario. Por ejemplo:

```
<form>
  <input type="text" name="name" required>
  <input type="email" name="email" required>
  <input pattern="\d*" name="number" required>

  <input type="submit" value="Publish"> <!-- form will be validated -->
  <input type="submit" value="Save" formnovalidate> <!-- form will NOT be validated -->
</form>
```

El formulario tiene campos que son necesarios para "publicar" el borrador pero no son necesarios para "guardar" el borrador.

## Reiniciar

```
<input type="reset" value="Reset">
```

Una entrada de tipo `reset` crea un botón que, al hacer clic, restablece todas las entradas en la forma en que se encuentra en su estado predeterminado.

- El texto en un campo de entrada se restablecerá en blanco o su valor predeterminado (especificado mediante el atributo de `value` ).
- Cualquier opción (s) en un menú de selección se deseleccionará a menos que tengan el atributo `selected` .
- Todas las casillas de verificación y casillas de radio se deseleccionarán a menos que tengan el atributo `checked` .

**Nota:** Un botón de reinicio debe estar dentro o adjuntado a (a través del atributo de `form` ) un elemento `<form>` para que tenga algún efecto. El botón solo restablecerá los elementos dentro de este formulario.

## Número

5

```
<input type="number" value="0" name="quantity">
```

El elemento de entrada con un atributo de tipo cuyo valor es `number` representa un control preciso para establecer el valor del elemento en una cadena que representa un número.

Tenga en cuenta que este campo no garantiza tener un número correcto. Simplemente permite todos los símbolos que podrían usarse en cualquier número real, por ejemplo, el usuario podrá ingresar valores como `ele-,0`.

## Tel

```
<input type="tel" value="+8400000000">
```

El elemento de entrada con un atributo de tipo cuyo valor es `tel` representa un control de edición de texto sin formato de una línea para ingresar un número de teléfono.

## Email

### 5

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

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

La dirección de correo electrónico se puede validar automáticamente cuando se envía, según la compatibilidad del navegador.

## Botón

```
<input type="button" value="Button Text">
```

Los botones se pueden utilizar para desencadenar acciones en la página, sin enviar el formulario. También puede usar el elemento `<button>` si necesita un botón que se pueda diseñar más fácilmente o que contenga otros elementos:

```
<button type="button">Button Text</button>
```

Los botones se utilizan normalmente con un evento "onclick":

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

### 0



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

## Atributos

[name]

El `name` del botón, que se envía con los datos del formulario.

[type]

El `type` de botón.

### Los valores posibles son:

`submit` : el botón envía los datos del formulario al servidor. Este es el valor predeterminado si no se especifica el atributo, o si el atributo se cambia dinámicamente a un valor vacío o no válido.

`reset` : el botón restablece todos los controles a sus valores iniciales.

`button` : El botón no tiene un comportamiento por defecto. Puede tener scripts del lado del cliente asociados con los eventos del elemento, que se activan cuando ocurren los eventos.

`menu` : el botón abre un menú emergente definido a través de su elemento designado.

[value]

El valor inicial del botón.

5

## Atributos extra para botones de envío

Atributo	Descripción
<code>form</code>	Especifica el ID del formulario al que pertenece el botón. Si no se especifica ninguno, pertenecerá a su elemento de formulario ancestro (si existe).
<code>formaction</code>	Especifica dónde enviar los datos del formulario. cuando se envía el formulario utilizando este botón.
<code>formenctype</code>	Especifica cómo se deben codificar los datos de formulario. al enviarlo al servidor mediante este botón. Solo se puede utilizar con <code>formmethod="post"</code> .
<code>formmethod</code>	Especifica el método HTTP a utilizar (POST o GET)

Atributo	Descripción
	al enviar datos de formulario utilizando este botón.
formnovalidate	Especifica que los datos de formulario no deben validarse en el envío.
formtarget	Especifica dónde mostrar la respuesta que se recibe. Después de enviar el formulario utilizando este botón.

## Color

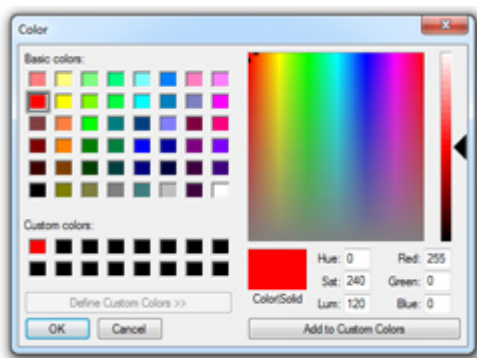
5

```
<input type="color" name="favcolor" value="#ff0000">
```

En los navegadores compatibles, el elemento de entrada con un atributo de tipo cuyo valor es `color` crea un control similar a un botón, con un color igual al valor del atributo de `color` (el valor predeterminado es negro si el valor no se especifica o es un formato hexadecimal no válido).



Al hacer clic en este botón, se abre el widget de color del sistema operativo, que permite al usuario seleccionar un color.



El respaldo para los navegadores que no admiten este tipo de entrada es un `type=text` [entrada](#) normal `type=text`.

## Url

5

```
<input type="url" name="Homepage">
```

Esto se utiliza para los campos de entrada que deben contener una dirección URL.

Dependiendo de la compatibilidad del navegador, el campo `url` puede validarse 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.

## Fecha

5

```
<input type="date" />
```

Aparecerá un selector de fecha en la pantalla para que usted elija una fecha. Esto no es compatible con Firefox o Internet Explorer.

## DateTime-Local

5

```
<input type="datetime-local" />
```

Dependiendo de la compatibilidad con el navegador, aparecerá un selector de fecha y hora en la pantalla para que elijas una fecha y una hora.

## Imagen

```
<input type="image" src="img.png" alt="image_name" height="50px" width="50px"/>
```

Una imagen. Debe usar el atributo `src` para definir el origen de la imagen y el atributo `alt` para definir un texto alternativo. Puede utilizar los atributos de altura y anchura para definir el tamaño de la imagen en píxeles.

## Distancia

5

```
<input type="range" min="" max="" step="" />
```

Un control para ingresar un número cuyo valor exacto no es importante.

Atributo	Descripción	Valor por defecto
min	Valor mínimo por rango	0
max	Valor máximo por rango	100
paso	Cantidad a aumentar en cada incremento.	1

## Mes

5

```
<input type="month" />
```

Dependiendo de la compatibilidad con el navegador, se mostrará un control para elegir el mes.

## Hora

5

```
<input type="time" />
```

La entrada de `time` marca este elemento como aceptando una cadena que representa un tiempo. El formato está definido en [RFC 3339](#) y debe ser un tiempo parcial como

```
19:04:39  
08:20:39.04
```

Actualmente, todas las versiones de Edge, Chrome, Opera y Chrome para Android admiten el tipo = "tiempo". Las versiones más recientes de Android Browser, específicamente 4.4 y versiones superiores lo admiten. Safari para iOS ofrece soporte parcial, no admite los atributos mínimo, máximo y paso.

## Semana

5

```
<input type="week" />
```

Dependiendo de la compatibilidad con el navegador, se mostrará un control para ingresar un número de año-semana y un número de semana sin zona horaria.

## Texto

El tipo de entrada más básico y la entrada predeterminada si no se especifica ningún `type`. Este tipo de entrada define un campo de texto de una sola línea con saltos de línea eliminados automáticamente del valor de entrada. Todos los demás personajes pueden ser ingresados en esto. `<input>` elementos `<input>` se utilizan dentro de un elemento `<form>` para declarar controles de entrada que permiten a los usuarios ingresar datos.

## Sintaxis

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

o (sin especificar un `type`, usando el atributo predeterminado):

```
<input>
```

El ancho predeterminado de una entrada de campo de texto es de 20 caracteres. Esto se puede cambiar especificando un valor para el atributo de `size` como este:

```
<input type="text" size="50">
```

El atributo de `size` es claramente diferente a establecer un ancho con CSS. El uso de un ancho define un valor específico (en número de píxeles, porcentaje del elemento principal, etc.) que la entrada siempre debe ser amplia. El uso del `size` calcula la cantidad de ancho para asignar en función de la fuente que se utiliza y el ancho de los caracteres normalmente.

**Nota:** El uso del atributo de `size` no limita intrínsecamente el número de caracteres que se pueden ingresar en el cuadro, solo la anchura del cuadro. Para limitar la longitud, ver [Validación de entrada](#).

Un campo de entrada solo permite una línea de texto. Si necesita una entrada de texto de varias líneas para una cantidad sustancial de texto, use [un elemento](#) `<textarea>` lugar.

## Buscar

5

La búsqueda de tipo de entrada se utiliza para la búsqueda textual. Agregará un símbolo de lupa junto al espacio para texto en la mayoría de los navegadores

```
<input type="search" name="googlesearch">
```

## DateTime (Global)

El elemento de entrada con un atributo de tipo cuyo valor es "**datetime**" representa un control para establecer el valor del elemento en una cadena que representa una **fecha y hora global (con información de zona horaria)**.

```
<fieldset>
  <p><label>Meeting time: <input type=datetime name="meeting.start"></label>
</fieldset>
```

*Atributos permitidos:*

- atributos globales
- nombre
- discapacitado
- formar
- tipo
- autocompletar
- enfoque automático
- lista

- mínimo máximo
- paso (flotar)
- solo lectura
- valor requerido

Lea Elementos de control de entrada en línea: <https://riptutorial.com/es/html/topic/277/elementos-de-control-de-entrada>

# Capítulo 18: Elementos de los medios

## Parámetros

Atributo	Detalles
anchura	Establece el ancho del elemento en píxeles.
altura	Establece la altura del elemento en píxeles.
<source>	Define los recursos de los archivos de audio o video.
pista	Define la pista de texto para elementos multimedia.
controles	Controles de pantallas
auto-reproducción	Empieza a reproducir automáticamente los medios de comunicación.
lazo	Reproduce los medios en un ciclo repetido.
apagado	Reproduce los medios sin sonido.
póster	Asigna una imagen para mostrar hasta que se carga un video

## Observaciones

### Soporte en los navegadores.

Característica	Cromo	Firefox (Gecko)	explorador de Internet	Ópera	Safari
Soporte basico	3.0	3.5 (1.9.1)	9.0	10.50	3.1
<audio> : PCM en WAVE	(Sí)	3.5 (1.9.1)	Sin soporte	10.50	3.1
<audio> : Vorbis en WebM	(Sí)	4.0 (2.0)	Sin soporte	10.60	3.1
<audio> : Streaming Vorbis / Opus en WebM a través de MSE	?	36.0	?	?	?
<audio> : Vorbis en Ogg	(Sí)	3.5 (1.9.1)	Sin soporte	10.50	Sin soporte
<audio> : MP3	(Sí)	(Sí)	9.0	(Sí)	3.1

Característica	Cromo	Firefox (Gecko)	explorador de Internet	Ópera	Safari
<audio> : MP3 en MP4	?	?	?	?	(Sí)
<audio> : AAC en MP4	(Sí)	(Sí)	9.0	(Sí)	3.1
<audio> : Opus en Ogg	27.0	15.0 (15.0)	?	?	?
<video> : VP8 y Vorbis en WebM	6.0	4.0 (2.0)	9.0	10.60	3.1
<video> : VP9 y Opus en WebM	29.0	28.0 (28.0)	?	(Sí)	?
<video> : Streaming WebM a través de MSE	?	42.0 (42.0)	?	?	?
<video> : Theora y Vorbis en Ogg	(Sí)	3.5 (1.9.1)	Sin soporte	10.50	Sin soporte
<video> : H.264 y MP3 en MP4	(Sí)	(Sí)	9.0	(Sí)	(Sí)
<video> : H.264 y AAC en MP4	(Sí)	(Sí)	9.0	(Sí)	3.1
cualquier otro formato	Sin soporte	Sin soporte	Sin soporte	Sin soporte	3.1

## Examples

### Utilizando `<video>` Elemento `<video>` para mostrar contenido de audio / video

Use el elemento HTML o `<audio>` para incrustar contenido de video / audio en un documento. El elemento de video / audio contiene una o más fuentes de video / audio. Para especificar una fuente, use el atributo `src` o el elemento `<source>` ; El navegador elegirá el más adecuado.

Ejemplo de etiqueta de audio:

```
<!-- Simple video example -->
<video src="videofile.webm" autoplay poster="posterimage.jpg">
  Sorry, your browser doesn't support embedded videos,
  but don't worry, you can <a href="videofile.webm">download it</a>
  and watch it with your favorite video player!
</video>

<!-- Video with subtitles -->
<video src="foo.webm">
```



```
<track kind="subtitles" src="foo.en.vtt" srclang="en" label="English">
<track kind="subtitles" src="foo.sv.vtt" srclang="sv" label="Svenska">
</video>
<!-- Simple video example -->
<video width="480" controls poster="https://archive.org/download/WebmVp8Vorbis/webmvp8.gif" >
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.webm" type="video/webm">
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8_512kb.mp4" type="video/mp4">
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.ogv" type="video/ogg">
  Your browser doesn't support HTML5 video tag.
</video>
```

## Ejemplo de etiqueta de audio:

```
<!-- Simple audio playback -->
<audio src="http://developer.mozilla.org/@api/deki/files/2926/=AudioTest_(1).ogg" autoplay>
  Your browser does not support the <code>audio</code> element.
</audio>

<!-- Audio playback with captions -->
<audio src="foo.ogg">
  <track kind="captions" src="foo.en.vtt" srclang="en" label="English">
  <track kind="captions" src="foo.sv.vtt" srclang="sv" label="Svenska">
</audio>
```

## Audio

HTML5 proporciona un nuevo estándar para incrustar un archivo de audio en una página web.

Puede incrustar un archivo de audio en una página utilizando el elemento `<audio>` :

```
<audio controls>
  <source src="file.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

## Vídeo

También puede insertar un video en una página web utilizando el elemento `<video>` :

```
<video width="500" height="700" controls>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

## Encabezado de video o fondo

Agregar un video que se reproducirá automáticamente en un bucle y no tiene controles ni sonido. Perfecto para un encabezado de video o fondo.

```
<video width="1280" height="720" autoplay muted loop poster="video.jpg" id="videobg">
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  <source src="video.ogv" type="video/ogg">
```

```
</video>
```

Este CSS proporciona un respaldo si el video no se puede cargar. Tenga en cuenta que se recomienda utilizar el primer fotograma del video como el video poster.jpg.

```
#videobg {  
  background: url(video.jpg) no-repeat;  
  background-size: cover;  
}
```

Lea Elementos de los medios en línea: <https://riptutorial.com/es/html/topic/2111/elementos-de-los-medios>

---

# Capítulo 19: Elementos de seccionamiento

## Observaciones

Los estándares HTML5 no enumeran el elemento principal como un elemento de sección.

## Examples

### Elemento del artículo

El elemento `<article>` contiene **contenido autónomo** como artículos, publicaciones de blog, comentarios de usuarios o un widget interactivo que podría distribuirse fuera del contexto de la página, por ejemplo, por RSS.

- Cuando los elementos del artículo están anidados, el contenido del nodo del artículo interno debe relacionarse con el elemento del artículo externo.

Un blog ( `section` ) con varias publicaciones ( `article` ), y comentarios ( `article` ) podría tener este aspecto.

```
<section>
  <!-- Each individual blog post is an <article> -->
  <article>
    <header>
      <h1>Blog Post</h1>
      <time datetime="2016-03-13">13th March 2016</time>
    </header>

    <p>The article element represents a self contained article or document.</p>
    <p>The section element represents a grouping of content.</p>

    <section>
      <h2>Comments <small>relating to "Blog Post"</small></h2>

      <!-- Related comment is also a self-contained article -->
      <article id="user-comment-1">
        <p>Excellent!</p>
        <footer><p>...</p><time>...</time></footer>
      </article>
    </section>
  </article>

  <!-- ./repeat: <article> -->

</section>

<!-- Content unrelated to the blog or posts should be outside the section. -->
<footer>
  <p>This content should be unrelated to the blog.</p>
</footer>
```

# Evitar el uso innecesario

Cuando el contenido principal de la página (excluyendo encabezados, pies de página, barras de navegación, etc.) es simplemente un grupo de elementos. Puede omitir el `<article>` a favor del elemento `<main>` .

```
<article>
  <p>This doesn't make sense, this article has no real `context`.</p>
</article>
```

En su lugar, reemplace el artículo con un elemento `<main>` para indicar que este es el contenido `main` de esta página.

```
<main>
  <p>I'm the main content, I don't need to belong to an article.</p>
</main>
```

Si usa otro elemento, asegúrese de especificar la función `<main>` [ARIA](#) para una interpretación y representación correctas en múltiples dispositivos y navegadores que no sean HTML5.

```
<section role="main">
  <p>This section is the main content of this page.</p>
</section>
```

## Notas:

- `<main>` descendientes del [elemento](#) `<main>` no se permiten dentro de un `<article>`

[Haga clic aquí para leer la especificación oficial de HTML5 para el elemento](#) `<article>`

## Elemento principal

El elemento `<main>` contiene el **contenido principal** de su página web. Este contenido es exclusivo de la página individual y no debe aparecer en ninguna otra parte del sitio. El contenido repetido, como encabezados, pies de página, navegación, logotipos, etc., se coloca fuera del elemento.

- El elemento `<main>` solo debe usarse **una vez como** máximo en una sola página.
- El elemento `<main>` no debe incluirse como descendiente de un `article` , `aside` , `footer` , `header` o elemento de `nav` .

En el siguiente ejemplo, estamos mostrando una **única publicación de blog** (e información relacionada, como referencias y comentarios).

```
<body>
  <header>
    <nav>...</nav>
  </header>
```

```
<main>
  <h1>Individual Blog Post</h1>
  <p>An introduction for the post.</p>

  <article>
    <h2>References</h2>
    <p>...</p>
  </article>

  <article>
    <h2>Comments</h2> ...
  </article>
</main>

<footer>...</footer>
</body>
```

- La publicación del blog se encuentra dentro del elemento `<main>` para indicar que este es el contenido principal de esta página (y, por lo tanto, único en todo el sitio web).
- Las etiquetas `<header>` y `<footer>` son *hermanos* del elemento `<main>` .

---

### Notas:

La especificación HTML5 reconoce el elemento `<main>` como un elemento de **agrupación** y no como un elemento de *sección* .

- **Atributos del rol ARIA** : `main` (*predeterminado*) , `presentation`

Se recomienda agregar el atributo de **función ARIA** `role="main"` a **otros elementos** destinados a ser utilizados como contenido principal para ayudar a los agentes de usuario que no admiten HTML5 y también para proporcionar más contexto para aquellos que lo hacen.

El elemento `<main>` por defecto tiene la función principal, por lo que no es necesario proporcionarlo.

[Haga clic aquí para leer la especificación oficial de HTML5 para el elemento `<main>`](#)

## Elemento de navegación

El elemento `<nav>` está destinado principalmente a ser utilizado para las secciones que contienen los **principales bloques de navegación** para el sitio web, esto puede incluir enlaces a otras partes de la página web (*por ejemplo, anclas para una tabla de contenido*) u otras páginas por completo.

---

## Artículos en línea

Lo siguiente mostrará un conjunto en línea de hipervínculos.

```
<nav>
  <a href="https://google.com">Google</a>
  <a href="https://www.yahoo.com">Yahoo!</a>
  <a href="https://www.bing.com">Bing</a>
</nav>
```

## Use los elementos de la lista cuando sea necesario

Si el contenido representa una lista de elementos, utilice un elemento de lista para mostrar esto y mejorar la experiencia del usuario.

Tenga en cuenta el `role="navigation"` , *más sobre esto a continuación*.

```
<nav role="navigation">
  <ul>
    <li><a href="https://google.com">Google</a></li>
    <li><a href="https://www.yahoo.com">Yahoo!</a></li>
    <li><a href="https://www.bing.com">Bing</a></li>
  </ul>
</nav>
```

## Evitar el uso innecesario

`<footer>` elementos `<footer>` pueden tener una lista de enlaces a otras partes del sitio (preguntas frecuentes, T&C, etc.). El elemento de pie de página solo es suficiente en este caso, no es *necesario* que envuelva más sus enlaces con un elemento `<nav>` en el `<footer>` .

```
<!-- the <nav> is not required in the <footer> -->
<footer>
  <nav>
    <a href="#">...</a>
  </nav>
</footer>

<!-- The footer alone is sufficient -->
<footer>
  <a href="#">...</a>
</footer>
```

### Notas:

- `<main>` descendientes del elemento `<main>` no están permitidos dentro de un `<nav>`

Se recomienda agregar el `role="navigation"` [ARIA](#) `role="navigation"` al elemento `<nav>` para ayudar a los agentes de usuario que no admiten HTML5 y también para proporcionar más contexto para aquellos que lo hacen.

```
<nav role="navigation"><!-- ... --></nav>
```

**Lectores de pantalla:** (*software que permite a los usuarios ciegos o discapacitados visuales navegar por el sitio*)

Los agentes de usuario, como los lectores de pantalla, interpretarán el elemento `<nav>` diferente en función de sus requisitos.

- Podría dar al elemento `<nav>` una prioridad más alta al renderizar la página
- Podría retrasar la prestación del elemento.
- Podría adaptar la página de una manera específica para adaptarse a las necesidades del usuario.

*Ejemplo:* haga que los enlaces de texto dentro de los elementos `<nav>` más grandes para una persona con discapacidad visual.

[Haga clic aquí para leer la especificación oficial de HTML5 para el elemento `<nav>`](#)

## Elemento de sección

El elemento `<section>` representa una sección genérica para agrupar temáticamente el contenido. Normalmente, cada sección debe poder identificarse con un elemento de encabezado como elemento secundario de la `section`.

- Puede utilizar el elemento `<section>` dentro de un `<article>` y viceversa.
- Cada sección debe tener un *tema* (un elemento de encabezado que identifique esta región)
- No utilice el elemento `<section>` como un 'contenedor' de estilo general.  
Si necesita un contenedor para aplicar el estilo, use un `<div>` lugar.

En el siguiente ejemplo, mostramos una **única publicación de blog** con varios capítulos, cada capítulo es una sección (*un conjunto de contenido agrupado temáticamente, que se puede identificar por los elementos del encabezado en cada sección*).

```
<article>
  <header>
    <h2>Blog Post</h2>
  </header>
  <p>An introduction for the post.</p>
  <section>
    <h3>Chapter 1</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Chapter 2</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Comments</h3> ...
  </section>
</article>
```

---

### Notas:

Los desarrolladores deben usar el elemento de **artículo** cuando tenga sentido syndicar los contenidos del elemento.

[Haga clic aquí para leer la especificación oficial de HTML5 para el elemento <main>](#)

## Elemento de encabezado

El elemento <header> representa el contenido introductorio para su antepasado más cercano que secciona el contenido o la sección del elemento raíz. Un <header> normalmente contiene un grupo de ayudas introductorias o de navegación.

**Nota:** El elemento de encabezado no es el contenido de sección; No introduce una nueva sección.

---

## Ejemplos:

```
<header>
  <p>Welcome to...</p>
  <h1>Voidwars!</h1>
</header>
```

En este ejemplo, el <article> tiene un <header> .

```
<article>
  <header>
    <h1>Flexbox: The definitive guide</h1>
  </header>
  <p>The guide about Flexbox was supposed to be here, but it turned out Wes wasn't a Flexbox expert either.</p>
</article>
```

---

## Recomendación propuesta del W3C

### Elemento de pie de página

El elemento <footer> contiene la parte del pie de página de la página.

Aquí hay un ejemplo para el elemento <footer> que contiene p etiqueta de párrafo.

```
<footer>
  <p>All rights reserved</p>
</footer>
```

Lea Elementos de seccionamiento en línea: <https://riptutorial.com/es/html/topic/311/elementos-de-seccionamiento>



---

# Capítulo 20: Elementos vacíos

## Introducción

No todas las etiquetas HTML son de la misma estructura. Si bien la mayoría de los elementos requieren una etiqueta de apertura, una etiqueta de cierre y contenido, algunos elementos, conocidos como elementos vacíos, solo requieren una etiqueta de apertura ya que ellos mismos no contienen ningún elemento. Este tema explica y demuestra el uso correcto de los elementos nulos en HTML

## Observaciones

Un elemento vacío no puede tener ningún contenido, pero puede tener atributos. Los elementos vacíos se cierran automáticamente, por lo que no deben tener una etiqueta de cierre.

En [HTML5](#), los siguientes elementos son nulos:

- `area`
- `base`
- `br`
- `col`
- `embed`
- `hr`
- `img`
- `input`
- `keygen`
- `link`
- `meta`
- `param`
- `source`
- `track`
- `wbr`

## Examples

### Elementos vacíos

HTML 4.01 / XHTML 1.0 Strict incluye los siguientes elementos nulos:

- `area` : `area` definida en la que se puede hacer clic en una imagen
- `base` : especifica una URL base desde la cual se basan todos los enlaces
- `br` - salto de línea
- `col` - columna en una tabla [obsoleta]
- `hr` - regla horizontal (línea)
- `img` - imagen
- `input` - campo donde los usuarios ingresan datos
- `link` - vincula un recurso externo al documento

- `meta` - proporciona información sobre el documento
- `param` - define parámetros para plugins

Los estándares HTML 5 incluyen todas las etiquetas no en desuso de la lista anterior y

- `command` : representa un comando que los usuarios pueden invocar [obsoleto]
- `keygen` : facilita la generación de claves públicas para certificados web [obsoleto]
- `source` : especifica fuentes de medios para elementos de `picture` , `audio` y `video`

El siguiente ejemplo **no** incluye elementos vacíos:

```
<div>
  <a href="http://stackoverflow.com/">
    <h3>Click here to visit <i>Stack Overflow!</i></h3>
  </a>
  <button onclick="alert('Hello!');">Say Hello!</button>
  <p>My favorite language is <b>HTML</b>. Here are my others:</p>
  <ol>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>PHP</li>
  </ol>
</div>
```

Observe cómo cada elemento tiene una etiqueta de apertura, una etiqueta de cierre y un texto u otros elementos dentro de las etiquetas de apertura y cierre. Las etiquetas nulas, sin embargo, se muestran en el siguiente ejemplo:

```

<br>
<hr>
<input type="number" placeholder="Enter your favorite number">
```

Con la excepción de la etiqueta `img`, todos estos elementos nulos tienen solo una etiqueta de apertura. La etiqueta `img`, a diferencia de cualquier otra etiqueta, tiene un cierre automático / antes del signo mayor que la etiqueta de apertura. Es una buena práctica tener un espacio antes de la barra.

Lea Elementos vacíos en línea: <https://riptutorial.com/es/html/topic/1449/elementos-vacios>

# Capítulo 21: Empotrar

## Parámetros

Parámetros	Detalles
src	Dirección del recurso.
type	Tipo de recurso incorporado
width	Dimensión horizontal
height	Dimensión vertical

## Examples

### Uso básico

La etiqueta de `embed` es nueva en HTML5. Este elemento proporciona un punto de integración para una aplicación externa (normalmente no HTML) o contenido interactivo.

```
<embed src="myflash.swf">
```

### Definiendo el tipo MIME

El tipo **MIME** debe definirse utilizando el atributo `type`.

```
<embed type="video/mp4" src="video.mp4" width="640" height="480">
```

Lea Empotrar en línea: <https://riptutorial.com/es/html/topic/8123/empotrar>

---

# Capítulo 22: Encabezados

## Introducción

HTML proporciona no solo etiquetas de párrafos simples, sino seis etiquetas de encabezado separadas para indicar encabezados de varios tamaños y grosores. Enumerado desde el encabezado 1 hasta el encabezado 6, el encabezado 1 tiene el texto más grande y grueso, mientras que el encabezado 6 es el más pequeño y delgado, hasta el nivel de párrafo. Este tema detalla el uso adecuado de estas etiquetas.

## Sintaxis

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

## Observaciones

- Un elemento `h1` - `h6` debe tener una etiqueta de inicio y una etiqueta de finalización. <sup>1</sup>
- `h1` - los elementos `h6` son elementos de nivel de bloque por defecto (estilo CSS: `display: block`). <sup>2</sup>
- `h1` elementos `h1` - `h6` no deben confundirse con el [elemento de sección](#)
- Las etiquetas de encabezado ( `h1` - `h6` ) no están relacionadas con la etiqueta de `head` .
- Contenido permitido: [contenido de fraseo](#)
- Los diferentes estilos CSS para los encabezados suelen diferir en `font-size` y `margin` . Las siguientes configuraciones de CSS para los elementos `h1` - `h6` pueden servir como orientación (caracterizada como 'informativa' por el [W3C](#) )
- Las arañas de los motores de búsqueda (el código que agrega una página a un motor de búsqueda) automáticamente prestan más atención a una mayor importancia (`h1` tiene más, `h2` tiene menos, `h3` tiene aún menos, ...) encabezados para discernir de qué se trata una página.

## Examples

### Usando encabezados

Los encabezados se pueden usar para describir el tema que preceden y se definen con las

etiquetas `<h1>` a `<h6>` . Los encabezados soportan todos los [atributos globales](#) .

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

### Definiendo un encabezado:

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

---

## La estructura correcta importa

**Los motores de búsqueda** y otros **agentes de usuario** generalmente indexan el contenido de la página basándose en elementos de encabezado, por ejemplo, para crear una tabla de contenido, por lo que es importante usar la estructura correcta para los encabezados.

En general, un artículo debe tener un elemento `h1` para el título principal seguido de los subtítulos `h2` : bajar una capa si es necesario. Si hay elementos `h1` en un nivel superior, no deben usarse para describir ningún contenido de nivel inferior.

### Documento de ejemplo (intención adicional para ilustrar la jerarquía):

```
<h1>Main title</h1>
<p>Introduction</p>

  <h2>Reasons</h2>

    <h3>Reason 1</h3>
    <p>Paragraph</p>

    <h3>Reason 2</h3>
    <p>Paragraph</p>

  <h2>In conclusion</h2>
  <p>Paragraph</p>
```

Lea Encabezados en línea: <https://riptutorial.com/es/html/topic/226/encabezados>

# Capítulo 23: Entidades de carácter

## Examples

### Caracteres especiales comunes

Algunos caracteres pueden estar reservados para HTML y no pueden usarse directamente, ya que pueden obstruir los códigos HTML reales. Por ejemplo, intentar mostrar los corchetes angulares izquierdo y derecho ( < > ) en el código fuente puede causar resultados inesperados en la salida. De manera similar, los espacios en blanco tal como están escritos en el código fuente pueden no mostrarse como se espera en el HTML de salida. Algunos, como ☎, no están disponibles en el conjunto de caracteres ASCII.

Para ello, se crean entidades de carácter. Estos son de la forma `&entity_name;` o `&#entity_number;`. Las siguientes son algunas de las entidades HTML disponibles.

Personaje	Descripción	Nombre de la entidad	Número de entidad
" "	espacio irrompible	<code>&amp;nbsp;</code>	<code>&amp;#160;</code>
"<"	menos que	<code>&amp;lt;</code>	<code>&amp;#60;</code>
">"	mas grande que	<code>&amp;gt;</code>	<code>&amp;#62;</code>
"&"	ampersand	<code>&amp;amp;</code>	<code>&amp;#38;</code>
"-"	guión largo	<code>&amp;mdash;</code>	<code>&amp;#8212;</code>
"-"	en el tablero	<code>&amp;ndash;</code>	<code>&amp;#8211;</code>
"©"	derechos de autor	<code>&amp;copy;</code>	<code>&amp;#169;</code>
"®"	marca registrada	<code>&amp;reg;</code>	<code>&amp;#174;</code>
"™"	marca	<code>&amp;trade;</code>	<code>&amp;#8482;</code>
"☎"	teléfono	<code>&amp;phone;</code>	<code>&amp;#9742;</code>

Así, para escribir

© 2016 Stack Exchange Inc.

Se utiliza el siguiente código HTML:

```
<b>&copy; 2016 Stack Exchange Inc.</b>
```

## Entidades de personajes en HTML

Se requieren muchos símbolos y caracteres especiales al desarrollar una página web en html, pero como sabemos, a veces el uso de caracteres directamente puede interferir con el código html real que tiene ciertos caracteres reservados y también que algunos caracteres no están disponibles en el teclado. Por lo tanto, para evitar el conflicto y, al mismo tiempo, poder utilizar diferentes símbolos en nuestro código w3 org, nos proporciona "Entidades de caracteres".

Las entidades de caracteres están predefinidas con 'Nombre de entidad' - & entity\_name; y 'Número de entidad' - & entity\_number; por lo tanto, necesitamos usar cualquiera de los dos para que el símbolo requerido se represente en nuestra página.

La lista de algunas entidades de caracteres se puede encontrar en <https://dev.w3.org/html5/html-author/charref>

Un ejemplo simple con el uso de la entidad de carácter para 'lupa':

```
<input type="text" placeholder=" 🔍 #128269; Search" />
```

que rinde como



Lea Entidades de carácter en línea: <https://riptutorial.com/es/html/topic/5229/entidades-de-caracter>

# Capítulo 24: Formas

## Introducción

Para agrupar elementos de entrada y enviar datos, HTML utiliza un elemento de formulario para encapsular elementos de entrada y envío. Estos formularios manejan el envío de los datos en el método especificado a una página manejada por un servidor o controlador. Este tema explica y demuestra el uso de formularios HTML para recopilar y enviar datos de entrada.

## Sintaxis

- `<form method="post|get" action="somePage.php" target="_blank|_self|_parent|_top|frameName">`

## Parámetros

Atributo	Descripción
<code>accept-charset</code>	Especifica las codificaciones de caracteres que se utilizarán para el envío del formulario.
<code>action</code>	Especifica dónde enviar los datos del formulario cuando se envía un formulario.
<code>autocomplete</code>	Especifica si un formulario debe tener autocompletado activado o desactivado.
<code>enctype</code>	Especifica cómo se deben codificar los datos de formulario al enviarlos al servidor (solo para <code>method = "post"</code> ).
<code>method</code>	Especifica el método HTTP que se debe utilizar al enviar datos de formulario (POST o GET).
<code>name</code>	Especifica el nombre de un formulario.
<code>novalidate</code>	Especifica que el formulario no debe validarse cuando se envía.
<code>target</code>	Especifica dónde mostrar la respuesta que se recibe después de enviar el formulario.

## Observaciones

El elemento `<form>` representa una sección que contiene elementos asociados al formulario (por ejemplo, `<button>` `<fieldset>` `<input>` `<label>` `<output>` `<select>` `<textarea>` ) que envía información a un servidor. Se requieren las etiquetas de inicio ( `<form>` ) y finalización ( `</form>` ).



# Examples

## Sumisión

### El atributo de acción

El atributo de acción define la acción que se realizará cuando se envía el formulario, lo que generalmente conduce a un script que recopila la información presentada y trabaja con ella. Si lo deja en blanco, lo enviará al mismo archivo.

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

### El atributo del método

El atributo de método se utiliza para definir el método HTTP de la forma que es GET o POST.

```
<form action="action.php" method="get">  
<form action="action.php" method="post">
```

El método GET se utiliza principalmente para *obtener* datos, por ejemplo, para recibir una publicación por su ID o nombre, o para enviar una consulta de búsqueda. El método GET adjuntará los datos del formulario a la URL especificada en el atributo de acción.

```
www.example.com/action.php?firstname=Mickey&lastname=Mouse
```

El método POST se utiliza al enviar datos a un script. El método POST no anexa los datos del formulario a la URL de acción, pero los envía utilizando el cuerpo de la solicitud.

Para enviar los datos del formulario correctamente, se debe especificar un nombre de atributo de nombre.

Como ejemplo, enviemos el valor del campo y establezcamos su nombre como *apellido* :

```
<input type="text" name="lastname" value="Mouse">
```

### Mas atributos

```
<form action="action.php" method="post" target="_blank" accept-charset="UTF-8"  
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>  
  
<!-- form elements -->  
  
</form>
```

### Atributo de destino en la etiqueta de formulario

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

El atributo de destino define un nombre de, o palabra clave para, un contexto de navegación (por ejemplo, pestaña, ventana o marco en línea).

Desde la etiqueta con un atributo de destino:

```
<form target="_blank">
```

## Valores de atributo

Valor	Descripción
_blanco	La respuesta se muestra en una nueva ventana o pestaña.
_yo	La respuesta se muestra en el mismo marco (esto es predeterminado)
_padre	La respuesta se muestra en el marco padre.
_parte superior	La respuesta se muestra en el cuerpo completo de la ventana.
<i>nombre de cuadro</i>	La respuesta se muestra en un iframe con nombre

Nota: el atributo de destino está en **desuso** en **HTML 4.01** . El atributo de destino es **compatible** con **HTML5** .

Los marcos y los conjuntos de marcos no son compatibles con **HTML5** , por lo que los **valores de \_parent, \_top y nombre de marco ahora se usan principalmente con iframes** .

## Cargando archivos

Las imágenes y los archivos se pueden cargar / enviar al servidor configurando el atributo `enctype` de la etiqueta de `form` en `multipart/form-data` . `enctype` especifica cómo se codificarían los datos del formulario al enviarlos al servidor.

## Ejemplo

```
<form method="post" enctype="multipart/form-data" action="upload.php">
  <input type="file" name="pic" />
  <input type="submit" value="Upload" />
</form>
```

## Agrupando algunos campos de entrada

Al diseñar un formulario, es posible que desee agrupar algunos campos de entrada en un grupo para ayudar a organizar el diseño del formulario. Esto se puede hacer usando la etiqueta. Aquí

hay un ejemplo para usarlo.

Para cada conjunto de campo, puede establecer una leyenda para el conjunto utilizando la etiqueta LEGEND TEXT

## Ejemplo

```
<form>
  <fieldset>
    <legend>1st field set:</legend>
    Field one:<br>
    <input type="text"><br>
    Field two:<br>
    <input type="text"><br>
  </fieldset><br>
  <fieldset>
    <legend>2nd field set:</legend>
    Field three:<br>
    <input type="text"><br>
    Field four:<br>
    <input type="text"><br>
  </fieldset><br>
  <input type="submit" value="Submit">
</form>
```

## Resultado

1st field set:

Field one:

Field two:

2nd field set:

Field three:

Field four:

## Soporte del navegador

Las últimas versiones de Chrome, IE, Edge, FireFox, Safari y Opera también son compatibles con la etiqueta

Lea Formas en línea: <https://riptutorial.com/es/html/topic/1160/formas>

---

# Capítulo 25: Formato de texto

## Introducción

Si bien la mayoría de las etiquetas HTML se utilizan para crear elementos, HTML también proporciona etiquetas de formato en el texto para aplicar estilos específicos relacionados con el texto a partes del texto. Este tema incluye ejemplos de formato de texto HTML como resaltado, negrita, subrayado, subíndice y texto marcado.

## Sintaxis

- `<abbr>Abbreviation</abbr>`
- `<b>Bold Text</b>`
- `<del>Deleted Text</del>`
- `<em>Emphasized Text</em>`
- `<i>Italic Text</i>`
- `<ins>Inserted Text</ins>`
- `<mark>Marked (or Highlighted) Text</mark>`
- `<s>Stricken Text</s>`
- `<strong>Strong Text</strong>`
- `<sub>Subscript Text</sub>`
- `<sup>Superscript Text</sup>`
- `<u>Underlined Text</u>`

## Examples

### Negrita, cursiva y subrayado

#### Texto en negrita

Para el texto en negrita, use las etiquetas `<strong>` o `<b>` :

```
<strong>Bold Text Here</strong>
```

o

```
<b>Bold Text Here</b>
```

¿Cual es la diferencia? Semántica. `<strong>` se utiliza para indicar que el texto es fundamental o semánticamente *importante* para el texto circundante, mientras que `<b>` no indica tal importancia y simplemente representa texto que debe estar en negrita.

Si utilizara `<b>` un programa de conversión de texto a voz no diría la (s) palabra (s) de manera diferente a cualquiera de las otras palabras que lo rodean, simplemente está llamando la atención sobre ellos sin agregar ninguna importancia adicional. Sin embargo, al utilizar `<strong>` , el mismo programa querría decir esas palabras con un tono de voz diferente para transmitir que el texto es

importante de alguna manera.

---

## Texto en cursiva

Para poner el texto en cursiva, use las etiquetas `<em>` o `<i>` :

```
<em>Italicized Text Here</em>
```

O

```
<i>Italicized Text Here</i>
```

¿Cual es la diferencia? Semántica. `<em>` se utiliza para indicar que el texto debe tener un énfasis adicional que se debe resaltar, mientras que `<i>` simplemente representa el texto que se debe diferenciar del texto normal que lo rodea.

Por ejemplo, si desea enfatizar la acción dentro de una oración, puede hacerlo al enfatizarla en cursiva a través de `<em>` : "¿ *Enviaría* la edición ya?"

Pero si estuviera identificando un libro o un periódico que normalmente usaría en cursiva estilística, simplemente usaría `<i>` : "Me vi obligado a leer a *Romeo y Julieta* en la escuela secundaria."

---

## Texto subrayado

Si bien el elemento `<u>` sí estaba en desuso en HTML 4, se reintrodujo con un significado semántico alternativo en HTML 5, para representar una anotación no textual no articulada. Puede usar dicha representación para indicar el texto mal escrito en la página, o para una marca con el nombre propio chino.

```
<p>This paragraph contains some <u>mispelled</u> text.</p>
```

## Destacando

El elemento `<mark>` es nuevo en HTML5 y se usa para marcar o resaltar texto en un documento "debido a su relevancia en otro contexto". <sup>1</sup>

El ejemplo más común sería en los resultados de una búsqueda si el usuario ingresó una consulta de búsqueda y los resultados se muestran resaltando la consulta deseada.

```
<p>Here is some content from an article that contains the <mark>searched query</mark> that we are looking for. Highlighting the text will make it easier for the user to find what they are looking for.</p>
```

Salida:

Here is some content from an article that contains the **searched query** that we are looking for. Highlighting the text will make it easier for the user to find what they are looking for.

Un formato estándar común es el texto negro sobre un fondo amarillo, pero esto se puede cambiar con CSS.

## Insertado, Borrado o Enojado

Para marcar el texto como insertado, use la etiqueta `<ins>` :

```
<ins>New Text</ins>
```

Para marcar el texto como eliminado, use la etiqueta `<del>` :

```
<del>Deleted Text</del>
```

Para tachar el texto, use la etiqueta `<s>` :

```
<s>Struck-through text here</s>
```

## Superíndice y subíndice

Para desplazar el texto hacia arriba o hacia abajo, puede usar las etiquetas `<sup>` y `<sub>` .

Para crear superíndice:

```
<sup>superscript here</sup>
```

Para crear un subíndice:

```
<sub>subscript here</sub>
```

## Abreviatura

Para marcar alguna expresión como una abreviatura, use la etiqueta `<abbr>` :

```
<p>I like to write <abbr title="Hypertext Markup Language">HTML</abbr>!</p>
```

Si está presente, el atributo de `title` se utiliza para presentar la descripción completa de dicha abreviatura.

Lea Formato de texto en línea: <https://riptutorial.com/es/html/topic/526/formato-de-texto>

---

# Capítulo 26: HTML 5 Caché

## Observaciones

El archivo de manifiesto es un archivo de texto simple, que le dice al navegador qué almacenar en caché (y qué no almacenar nunca en caché). La extensión de archivo recomendada para los archivos de manifiesto es: ".appcache" El archivo de manifiesto tiene tres secciones:

CACHE MANIFEST - Los archivos enumerados bajo este encabezado se almacenarán en caché después de que se descarguen por primera vez

RED - Los archivos enumerados bajo este encabezado requieren una conexión al servidor y nunca se almacenarán en caché

FALLBACK: los archivos enumerados bajo este encabezado especifican las páginas de respaldo si una página no es accesible

## Examples

### Ejemplo básico de caché de HTML 5

este es nuestro archivo index.html

```
<!DOCTYPE html>
<html manifest="index.appcache">
<body>
  <p>Content</p>
</body>
</html>
```

luego crearemos el archivo index.appcache con los códigos a continuación

```
CACHE MANIFEST
index.html
```

escriba los archivos que desea que se almacenen en la memoria caché index.html, vaya al modo sin conexión y vuelva a cargar la pestaña

**Nota:** los dos archivos deben estar en la misma carpeta en este ejemplo

Lea HTML 5 Caché en línea: <https://riptutorial.com/es/html/topic/8024/html-5-cache>

---

# Capítulo 27: Idiomas de contenido

## Sintaxis

- `<element lang="language_code">` `<!--` El código de idioma debe estar en el formato [\[ISO 639-1\]](#) ([https://en.wikipedia.org/wiki/ISO\\_639-1](https://en.wikipedia.org/wiki/ISO_639-1)) `-->`

## Observaciones

El valor del atributo `lang` debe ser una **etiqueta de idioma BCP 47** válida o la **cadena vacía** (si se desconoce el idioma).

Las etiquetas de idioma de [BCP 47](#) se enumeran en el [Registro de subetiquetas de idioma de la IANA](#).

## Accesibilidad

Los criterios de éxito relevantes de WCAG 2.0 son:

- [3.1.1 Idioma de la página](#)
- [3.1.2 Idioma de las partes](#)

Las técnicas relacionadas de WCAG 2.0 son:

- [H57: Usando atributos de lenguaje en el elemento html](#)
- [H58: uso de atributos de lenguaje para identificar cambios en el lenguaje humano](#)

## Examples

### Lenguaje de elementos

El atributo `lang` se utiliza para especificar el idioma del contenido del elemento y los valores de texto del atributo:

```
<p lang="en">The content of this element is in English.</p>
```

```
<p lang="en" title="The value of this attribute is also in English.">The content of this element is in English.</p>
```

La declaración de idioma se hereda:

```
<div lang="en">
  <p>This element contains English content.</p>
  <p title="This attribute, too.">Same with this element.</p>
</div>
```



## Elementos con múltiples idiomas

Puede "sobrescribir" una declaración de idioma:

```
<p lang="en">This English sentence contains the German word <span lang="de">Hallo</span>.</p>
```

## Manejo de atributos con diferentes idiomas

Puede "sobrescribir" la declaración de lenguaje de un elemento principal introduciendo cualquier elemento aparte de `applet`, `base`, `basefont`, `br`, `frame`, `frameset`, `hr`, `iframe`, `meta`, `param`, `script` (de HTML 4.0) con un atributo `lang` propio:

```
<p lang="en" title="An English paragraph">
  <span lang="de" title="A German sentence">Hallo Welt!</span>
</p>
```

## Idioma base del documento

Es una buena práctica declarar el idioma principal del documento en el elemento `html` :

```
<html lang="en">
```

Si no se especifica ningún otro atributo de `lang` en el documento, significa que *todo* (es decir, el contenido del elemento y los valores de texto del atributo) está en ese idioma.

Si el documento contiene partes en otros idiomas, estas partes deben obtener sus propios atributos de `lang` para "sobrescribir" la declaración de idioma.

## URL regionales

Es posible agregar el atributo `hreflang` a los elementos `<a>` y `<area>` que crean hipervínculos. Así se especifica el idioma del recurso vinculado. El idioma definido debe ser una etiqueta de idioma [BCP 47](#) <sup>[1]</sup> válida.

```
<p>
  <a href="example.org" hreflang="en">example.org</a> is one of IANA's example domains.
</p>
```

1. ↑ Grupo de trabajo de la red IETF: [Etiquetas RFC 5646 para la identificación de idiomas](#), IETF, septiembre de 2009

Lea Idiomas de contenido en línea: <https://riptutorial.com/es/html/topic/737/idiomas-de-contenido>

---

# Capítulo 28: IFrames

## Parámetros

Atributo	Detalles
<code>name</code>	Establece el nombre del elemento, que se utilizará con la etiqueta para cambiar el <code>src</code> de <code>iframe</code> .
<code>width</code>	Establece el ancho del elemento en píxeles.
<code>height</code>	Establece la altura del elemento en píxeles.
<code>src</code>	Especifica la página que se mostrará en el marco.
<code>srcdoc</code>	Especifica el contenido que se mostrará en el marco, suponiendo que el navegador lo admita. El contenido debe ser HTML válido.
<code>sandbox</code>	Cuando se configura, el contenido del <code>iframe</code> se trata desde un origen único y se desactivarán las funciones que incluyen scripts, complementos, formularios y ventanas emergentes. Las restricciones se pueden relajar selectivamente agregando una lista de valores separados por espacios. Vea la tabla en Observaciones para los posibles valores.
<code>allowfullscreen</code>	Ya sea para permitir que los contenidos del <code>iframe</code> utilicen <code>requestFullscreen()</code>

## Observaciones

Un `iframe` se utiliza para incrustar otro documento en el documento HTML actual.

Puedes usar `iframes` para mostrar:

- otras páginas HTML en el mismo dominio;
- otras páginas HTML en otro dominio (ver más abajo - Política del mismo origen);
- Documentos PDF (aunque IE podría tener algunos problemas, [esta pregunta SO](#) podría ayudar);

DEBE usar un `iframe` como último recurso, ya que tiene problemas con los marcadores y la navegación, y siempre hay mejores opciones aparte de un `iframe`. [Esta pregunta de SO](#) debería ayudarlo a comprender más sobre los altibajos de los `iframes`.

---

## Política del mismo origen

Algunos sitios no pueden mostrarse usando un `iframe`, porque aplican una política llamada política [de origen idéntico](#) . Esto significa que el sitio en el que se encuentra el `iframe` debe estar en el mismo dominio que el que se muestra.

Esta política también se aplica a la manipulación de contenido que vive dentro de un `iFrame`. Si el `iFrame` accede al contenido de un dominio diferente, no podrá acceder o manipular el contenido dentro de un `iFrame`.

El elemento `iframe` en [W3C](#)

## atributo de `sandbox`

El atributo `sandbox` , cuando se establece, agrega restricciones adicionales al `iframe`. Se puede usar una lista de fichas separadas por espacios para relajar estas restricciones.

Valor	Detalles
<code>allow-forms</code>	Permite la presentación de formularios.
<code>allow-pointer-lock</code>	Habilita la API de puntero de JavaScript.
<code>allow-popups</code>	Las ventanas emergentes se pueden crear usando <code>window.open 0 &lt;a target="_blank"</code>
<code>allow-same-origin</code>	El documento <code>iframe</code> utiliza su origen real en lugar de recibir uno único. Si se usa con <code>allow-scripts</code> el documento <code>iframe</code> puede eliminar todo el espacio aislado si es del mismo origen que el documento principal.
<code>allow-scripts</code>	Habilita scripts. El documento <code>iframe</code> y el documento principal pueden comunicarse entre sí mediante la API <code>postMessage()</code> . Si se utiliza con <code>allow-same-origin</code> el documento <code>iframe</code> puede eliminar todo el espacio aislado si es del mismo origen que el documento principal.
<code>allow-top-navigation</code>	Permite que el contenido del <code>iframe</code> cambie la ubicación del documento de nivel superior.

## Examples

### Fundamentos de un marco en línea

El término "IFrame" significa marco en línea. Se puede usar para incluir otra página en tu página. Esto producirá un pequeño marco que muestra el contenido exacto del `base.html` .

```
<iframe src="base.html"></iframe>
```

## Configuración del tamaño del marco

El IFrame puede redimensionarse utilizando los atributos de `width` y `height` , donde los valores están representados en píxeles (HTML 4.01 permite valores de porcentaje, pero HTML 5 solo permite valores en píxeles de CSS).

```
<iframe src="base.html" width="800" height="600"></iframe>
```

## Usando Anclas Con IFrames

Normalmente, un cambio de página web dentro de un IFrame se inicia con el IFrame, por ejemplo, al hacer clic en un enlace dentro de IFrame. Sin embargo, es posible cambiar el contenido de un IFrame desde fuera del IFrame. Puede usar una etiqueta de anclaje cuyo atributo `href` se establezca en la URL deseada y cuyo atributo de `target` se establezca en el atributo de `name` de IFrame.

```
<iframe src="webpage.html" name="myIframe"></iframe>
<a href="different_webpage.html" target="myIframe">Change the Iframe content to
different_webpage.html</a>
```

## Usando el atributo "srcdoc"

El atributo `srcdoc` se puede usar (en lugar del atributo `src` ) para especificar el contenido exacto del IFrame como un documento HTML completo. Esto producirá un IFrame con el texto "¡Los IFrames son geniales!"

```
<iframe srcdoc="<p>IFrames are cool!</p>"></iframe>
```

Si el `srcdoc` atributo no está soportado por el navegador, el IFrame en lugar de recurrir al uso de la `src` atributo, pero si tanto el `src` y `srcdoc` atributos están presentes y apoyado por el navegador, `srcdoc` tiene prioridad.

```
<iframe srcdoc="<p>IFrames are cool!</p>" src="base.html"></iframe>
```

En el ejemplo anterior, si el navegador no admite el atributo `srcdoc` , mostrará los contenidos de la página `base.html` .

## Caja de arena

Lo siguiente incrusta una página web no confiable con todas las restricciones habilitadas

```
<iframe sandbox src="http://example.com/"></iframe>
```

Para permitir que la página ejecute scripts y envíe formularios, agregue `allow-scripts` y `allow-forms` al `sandbox` attribute .

```
<iframe sandbox="allow-scripts allow-forms" src="http://example.com/"></iframe>
```

Si hay contenido no confiable (como comentarios del usuario) en el mismo dominio que la página web principal, se puede usar un iframe para deshabilitar los scripts y al mismo tiempo permitir que el documento principal interactúe con su contenido mediante JavaScript.

```
<iframe sandbox="allow-same-origin allow-top-navigation"
src="http://example.com/untrusted/comments/page2">
```

El documento principal puede agregar detectores de eventos y cambiar el tamaño de IFrame para que se ajuste a su contenido. Esto, junto con el `allow-top-navigation`, puede hacer que el iframe de espacio aislado aparezca como parte del documento principal.

Esta caja de arena no es un reemplazo para la entrada de desinfección, pero se puede usar como parte de una estrategia de [defensa en profundidad](#).

También tenga en cuenta que un atacante puede subvertir esta zona de pruebas convenciendo a un usuario de que visite directamente la fuente del iframe. El encabezado HTTP de la [Política de seguridad del contenido](#) se puede usar para mitigar este ataque.

Lea IFrames en línea: <https://riptutorial.com/es/html/topic/499/iframes>

# Capítulo 29: Imágenes

## Sintaxis

- `<img src="" alt="">`

## Parámetros

Parámetros	Detalles
<code>src</code>	Especifica la URL de la imagen.
<code>srcset</code>	Imágenes para usar en diferentes situaciones (por ejemplo, pantallas de alta resolución, monitores pequeños, etc.)
<code>sizes</code>	Tamaños de imagen entre puntos de ruptura
<code>crossorigin</code>	Cómo el elemento maneja las solicitudes de origen cruzado.
<code>usemap</code>	Nombre del mapa de imagen a utilizar.
<code>ismap</code>	Si la imagen es un mapa de imagen del lado del servidor
<code>alt</code>	Texto alternativo que se debe mostrar si, por algún motivo, no se pudo mostrar la imagen
<code>width</code>	Especifica el ancho de la imagen (opcional)
<code>height</code>	Especifica la altura de la imagen (opcional)

## Examples

### Creando una imagen

Para agregar una imagen a una página, use la etiqueta de imagen.

Las etiquetas de imagen ( `img` ) no tienen etiquetas de cierre. Los dos atributos principales que le da a la etiqueta `img` son `src` , la fuente de la imagen y `alt` , que es un texto alternativo que describe la imagen.

```

```

También puede obtener imágenes de una URL web:

```

```

```
Kleveter">
```

Nota: las imágenes no se insertan técnicamente en una página HTML, las imágenes están vinculadas a páginas HTML. La etiqueta `<img>` crea un espacio de retención para la imagen a la que se hace referencia.

También es posible incrustar imágenes directamente dentro de la página usando base64:

```

```

Sugerencia: para vincular una imagen a otro documento, simplemente anide la etiqueta `<img>` dentro de las etiquetas `<a>` .

## Ancho y altura de la imagen

**Nota:** Los atributos de ancho y alto *no* están en desuso en las imágenes y nunca lo han sido. Su uso se ha hecho mucho más estricto.

Las dimensiones de una imagen se pueden especificar utilizando los atributos de `width` y `height` de la etiqueta de imagen:

```

```

Al especificar el `width` y el `height` de una imagen, su estructura le da al navegador una pista sobre cómo se debe diseñar la página, incluso si solo está especificando el tamaño real de la imagen. Si no se especifican las dimensiones de la imagen, el navegador tendrá que volver a calcular el diseño de la página después de que se cargue la imagen, lo que puede hacer que la página "salte" mientras se carga.

### 4.1

Puede dar a la imagen un ancho y alto, ya sea en la cantidad de píxeles CSS o en un porcentaje de las dimensiones reales de la imagen.

Estos ejemplos son todos válidos:

```




```

## 5

El ancho y el alto de la imagen deben especificarse en píxeles CSS; un valor de porcentaje ya no es un valor válido. Además, si se especifican ambos atributos, deben encajar en [una de las tres fórmulas](#) que mantienen la relación de aspecto. Aunque es válido, no debe usar los atributos de ancho y alto para estirar una imagen a un tamaño más grande.

Estos ejemplos son válidos:

```



```

Este ejemplo no es recomendable:

```

```

Estos ejemplos son inválidos:

```


```

## Elegir texto alternativo

Alt-text es utilizado por lectores de pantalla para usuarios con discapacidades visuales y por motores de búsqueda. Por lo tanto, es importante escribir un buen texto alternativo para sus imágenes.

El texto debe verse correcto incluso si reemplaza la imagen con su atributo alt. Por ejemplo:

```
<!-- Incorrect -->
 An anonymous user wrote:
<blockquote>Lorem ipsum dolor sed.</blockquote>
<a href="https://google.com/"></a> /
<a href="https://google.com/"></a>
```

Sin las imágenes, esto se vería así:

Usuario anónimo avatar Un usuario anónimo escribió:

Lorem ipsum dolor sed.

[Editar icono](#) / [Eliminar icono](#)

Para corregir esto:

- Eliminar el texto alternativo para el avatar. Esta imagen agrega información para los usuarios videntes (un icono fácilmente identificable para mostrar que el usuario es anónimo) pero esta información ya está disponible en el texto. <sup>1</sup>
- Elimine el "icono" del texto alternativo de los iconos. Saber que esto sería un ícono si estuviera allí no ayuda a transmitir su propósito real.

```
<!-- Correct -->
 An anonymous user wrote:
<blockquote>Lorem ipsum dolor sed.</blockquote>
<a href="https://google.com/"></a> /
<a href="https://google.com/"></a>
```

Un usuario anónimo escribió:



Lorem ipsum dolor sed.

[Editar](#) / [Eliminar](#)

## Notas al pie

<sup>1</sup> Hay una diferencia semántica entre incluir un atributo alt vacío y excluirlo por completo. Un atributo alt vacío indica que la imagen *no* es una parte clave del contenido (como es cierto en este caso, es solo una imagen aditiva que no es necesaria para comprender el resto) y, por lo tanto, puede omitirse de la representación. Sin embargo, la falta de un atributo alt indica que la imagen es una parte clave del contenido y que simplemente no hay un equivalente textual disponible para la representación.

### Imagen sensible utilizando el atributo srcset

## Usando srcset con tamaños

```

```

`sizes` son como las consultas de medios, que describen la cantidad de espacio que ocupa la imagen de la ventana gráfica.

- Si la ventana gráfica es más grande que 1200 px, la imagen es exactamente 580 px (por ejemplo, nuestro contenido está centrado en un contenedor que tiene un ancho máximo de 1200 px. La imagen ocupa la mitad de los márgenes menos).
- Si la ventana gráfica está entre 640 px y 1200 px, la imagen ocupa el 48% de la ventana gráfica (por ejemplo, la imagen se escala con nuestra página y ocupa la mitad del ancho de la ventana gráfica menos los márgenes).
- si la ventana gráfica es de otro tamaño, en nuestro caso menos de 640 px, la imagen toma el 98% de la ventana gráfica (por ejemplo, la imagen se amplía con nuestra página y ocupa todo el ancho de la ventana gráfica menos los márgenes). **La condición del medio se debe omitir para el último elemento.**

`srcset` solo le dice al navegador qué imágenes tenemos disponibles y cuáles son sus tamaños.

- `img/hello-300.jpg` tiene 300px de ancho,
- `img/hello-600.jpg` tiene 600px de ancho,
- `img/hello-900.jpg` tiene 900px de ancho,
- `img/hello-1200.jpg` tiene 1200px de ancho

`src` siempre es fuente de imagen obligatoria. En caso de usar con `srcset`, `src` mostrará una imagen alternativa en caso de que el navegador no sea compatible con `srcset`.

## Usando `srcset` sin tallas

```

```

`srcset` proporciona una lista de imágenes disponibles, con descriptor `x` proporción de píxeles de dispositivo.

- si la proporción de píxeles del dispositivo es 1, use `img/hello-300.jpg`
- si la proporción de píxeles del dispositivo es 2, use `img/hello-600.jpg`
- si la proporción de píxeles del dispositivo es 3, use `img/hello-1200.jpg`

`src` siempre es fuente de imagen obligatoria. En caso de usar con `srcset`, `src` mostrará una imagen alternativa en caso de que el navegador no sea compatible con `srcset`.

## Imagen sensible utilizando un elemento de imagen.

### Código

```
<picture>
  <source media="(min-width: 600px)" srcset="large_image.jpg">
  <source media="(min-width: 450px)" srcset="small_image.jpg">
  
</picture>
```

### Uso

Para mostrar diferentes imágenes con diferente ancho de pantalla, debe incluir todas las imágenes que usen la etiqueta de origen en una etiqueta de imagen como se muestra en el ejemplo anterior.

### Resultado

- En pantallas con un ancho de pantalla > 600px, muestra `large_image.jpg`
- En pantallas con ancho de pantalla > 450px, muestra `small_image.jpg`
- En pantallas con otro ancho de pantalla, muestra `default_image.jpg`

Lea Imágenes en línea: <https://riptutorial.com/es/html/topic/587/imagenes>

# Capítulo 30: Incluir código JavaScript en HTML

## Sintaxis

- `<script type="text/javascript"> //some code </script>`
- `<script type="text/javascript" src="URL"></script>`
- `<script type="text/javascript" src="URL" async>//async code</script>`

## Parámetros

Atributo	Detalles
<code>src</code>	Especifica la ruta de acceso a un archivo JavaScript. Ya sea una URL relativa o absoluta.
<code>type</code>	Especifica el tipo MIME. Este atributo es obligatorio en HTML4, pero opcional en HTML5.
<code>async</code>	Especifica que el script se ejecutará de forma asíncrona (solo para scripts externos). Este atributo no requiere ningún valor (excepto de XHTML).
<code>defer</code>	Especifica que el script se ejecutará cuando la página haya finalizado el análisis (solo para scripts externos). Este atributo no requiere ningún valor (excepto de XHTML).
<code>charset</code>	Especifica la codificación de caracteres utilizada en un archivo de script externo, por ejemplo, UTF-8
<code>crossorigin</code>	Cómo el elemento maneja las solicitudes de origen cruzado.
<code>nonce</code>	Nonce criptográfico utilizado en <i>la Política de seguridad de contenido</i> comprueba <a href="#">CSP3</a>

## Observaciones

Si el código JavaScript incrustado (archivo) se utiliza para manipular <http://stackoverflow.com/documentation/javascript/503/document-object-model-dom> Elements, coloque sus etiquetas `<script></script>` justo **antes del cierre** `</body>` **etiquete** o use métodos o bibliotecas de JavaScript (como [jQuery](#) para manejar una variedad de navegadores) que aseguran que el DOM esté leído y listo para ser manipulado.

## Examples

## Enlace a un archivo JavaScript externo

```
<script src="example.js"></script>
```

El atributo `src` funciona como el atributo `href` en los anclajes: puede especificar una URL absoluta o relativa. El ejemplo anterior se vincula a un archivo dentro del mismo directorio del documento HTML. Esto generalmente se agrega dentro de las etiquetas `<head>` en la parte superior del documento html

## Incluyendo directamente el código JavaScript

En lugar de vincular a un archivo externo, también puede incluir el código JS tal como está en su HTML:

```
<script>
// JavaScript code
</script>
```

## Incluyendo un archivo JavaScript ejecutándose de forma asíncrona.

```
<script type="text/javascript" src="URL" async></script>
```

## Manejo de Javascript deshabilitado

Es posible que el navegador del cliente no admita Javascript o tenga la ejecución de Javascript deshabilitada, quizás por razones de seguridad. Para poder decirles a los usuarios que un script debe ejecutarse en la página, se puede usar la etiqueta `<noscript>`. El contenido de `<noscript>` se muestra cada vez que Javascript está deshabilitado para la página actual.

```
<script>
  document.write("Hello, world!");
</script>
<noscript>This browser does not support Javascript.</noscript>
```

Lea [Incluir código JavaScript en HTML en línea: https://riptutorial.com/es/html/topic/3719/incluir-codigo-javascript-en-html](https://riptutorial.com/es/html/topic/3719/incluir-codigo-javascript-en-html)

---

# Capítulo 31: Liza

## Introducción

HTML ofrece tres formas de especificar listas: listas ordenadas, listas no ordenadas y listas de descripción. Las listas ordenadas usan secuencias ordinales para indicar el orden de los elementos de la lista, las listas no ordenadas usan un símbolo definido como una viñeta para enumerar los elementos sin un orden designado, y las listas de descripción usan sangrías para enumerar los elementos con sus hijos. Este tema explica la implementación y combinación de estas listas en el formato HTML.

## Sintaxis

- `<ol> ordered list items </ol>`
- `<ul> unordered list items </ul>`
- `<li> list item (ordered and not) </li>`
- `<dl> description list items </dl>`
- `<dt> description list title </dt>`
- `<dd> description list description </dd>`

## Observaciones

### Ver también

Puede agregar una propiedad CSS de tipo estilo de lista a una etiqueta `<ul>` para cambiar el tipo de icono que se usa para marcar cada elemento de lista, por ejemplo, `<ul style="list-style-type:disc">` . Se admiten los siguientes tipos de estilo de lista:

- "list-style-type: disc" es el punto predeterminado.
- "list-style-type: circle" es un círculo sin rellenar.
- "list-style-type: square" es un cuadrado relleno.
- "list-style-type: none" no utiliza ninguna marca.

También puede agregar un atributo de tipo a una etiqueta `<ol>` para cambiar la forma en que se realiza la numeración, por ejemplo, `<ol type="1">` . Se permiten los siguientes tipos:

- `type = "1"` es la forma predeterminada.
- `type = "A"` usa letras mayúsculas en orden alfabético
- `type = "a"` usa letras minúsculas en orden alfabético
- `type = "I"` usa números romanos con letras mayúsculas
- `type = "i"` usa números romanos con letras minúsculas

## Examples

### Lista desordenada

Se puede crear una lista desordenada con la etiqueta `<ul>` y cada elemento de la lista se puede crear con la etiqueta `<li>` como se muestra en el siguiente ejemplo:

```
<ul>
  <li>Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ul>
```

Esto producirá una lista con viñetas (que es el estilo predeterminado):

- ítem
- Otro artículo
- Otro artículo más

Debe usar `ul` para mostrar una lista de elementos, donde el orden de los elementos no es importante. Si cambiar el orden de los elementos hace que la lista sea incorrecta, debe usar `<ol>`.

## Lista ordenada

Se puede crear una lista ordenada con la etiqueta `<ol>` y cada elemento de la lista se puede crear con la etiqueta `<li>` como en el siguiente ejemplo:

```
<ol>
  <li>Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

Esto producirá una lista numerada (que es el estilo predeterminado):

1. ítem
2. Otro artículo
3. Otro artículo más

---

## Cambiando manualmente los números.

Hay varias formas de jugar con los números que aparecen en los elementos de la lista en una lista ordenada. La primera forma es establecer un número de inicio, usando el atributo de `start`. La lista comenzará en este número definido y continuará incrementándose en uno como de costumbre.

```
<ol start="3">
  <li>Item</li>
  <li>Some Other Item</li>
  <li>Yet Another Item</li>
</ol>
```

Esto producirá una lista numerada (que es el estilo predeterminado):

3. í t
4. Algún otro artículo
5. Otro articulo mas

También puede establecer explícitamente un determinado elemento de lista en un número específico. Más elementos de la lista después de uno con un valor específico continuarán incrementándose en uno desde el valor de ese elemento de la lista, ignorando donde estaba la lista principal.

```
<li value="7"></li>
```

También vale la pena señalar que, al utilizar el atributo de `value` directamente en un elemento de la lista, puede anular el sistema de numeración existente de una lista ordenada al reiniciar la numeración a un valor inferior. Por lo tanto, si la lista principal ya tenía el valor 7 y se encontró con un elemento de la lista con el valor 4, ese elemento de la lista aún se mostrará como 4 y continuará contando desde ese punto nuevamente.

```
<ol start="5">
  <li>Item</li>
  <li>Some Other Item</li>
  <li value="4">A Reset Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

Por lo tanto, el ejemplo anterior producirá una lista que sigue el patrón de numeración de 5, 6, 4, 5, 6, comenzando nuevamente en un número inferior al anterior y duplicando el número 6 en la lista.

**Nota:** Los atributos de `start` y `value` solo aceptan un número, incluso si la lista ordenada está configurada para mostrarse como números romanos o letras.

5

Puede revertir la numeración agregando `reversed` en su elemento `ol` :

```
<ol reversed>
  <li>Item</li>
  <li>Some Other Item</li>
  <li value="4">A Reset Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

La numeración inversa es útil si está agregando continuamente a una lista, por ejemplo, con nuevos episodios de podcast o presentaciones, y desea que los elementos más recientes aparezcan primero.

## Cambiando el tipo de numeral

Puede cambiar fácilmente el tipo de número que se muestra en el marcador del elemento de lista utilizando el atributo `type`

```
<ol type="1|a|A|i|I">
```

Tipo	Descripción	Ejemplos
1	Valor por defecto - números decimales	1,2,3,4
a	Ordenados alfabéticamente (en minúsculas)	a B C D
A	Ordenados alfabéticamente (en mayúsculas)	A B C D
i	Números romanos (minúsculas)	i, ii, iii, iv
I	Números romanos (mayúsculas)	I, II, III, IV

Debe usar `ol` para mostrar una lista de artículos, donde los artículos han sido ordenados intencionalmente y el orden debe ser enfatizado. Si cambiar el orden de los elementos NO hace que la lista sea incorrecta, debe usar `<ul>` .

## Descripción de la lista

Con el elemento `dl` se puede crear una lista de descripción (o *lista de definiciones* , como se llamó antes de HTML5). Consiste en grupos de nombre-valor, donde el nombre se da en el elemento `dt` , y el valor se da en el elemento `dd` .

```
<dl>
  <dt>name 1</dt>
  <dd>value for 1</dd>
  <dt>name 2</dt>
  <dd>value for 2</dd>
</dl>
```

### Demo en vivo

Un grupo nombre-valor puede tener más de un nombre y / o más de un valor (que representan alternativas):

```
<dl>

  <dt>name 1</dt>
  <dt>name 2</dt>
  <dd>value for 1 and 2</dd>

  <dt>name 3</dt>
  <dd>value for 3</dd>
  <dd>value for 3</dd>
```



```
</dl>
```

[Demo en vivo](#)

## Listas anidadas

Puede anidar listas para representar subelementos de un elemento de lista.

```
<ul>
  <li>item 1</li>
  <li>item 2
    <ul>
      <li>sub-item 2.1</li>
      <li>sub-item 2.2</li>
    </ul>
  </li>
  <li>item 3</li>
</ul>
```

- Objeto 1
- item 2
  - subelemento 2.1
  - subelemento 2.2
- item 3

La lista anidada debe ser un elemento secundario del elemento `li` .

También puede anidar diferentes tipos de lista:

```
<ol>
  <li>Hello, list!</li>
  <li>
    <ul>
      <li>Hello, nested list!</li>
    </ul>
  </li>
</ol>
```

Lea Liza en línea: <https://riptutorial.com/es/html/topic/393/liza>

# Capítulo 32: Lona

## Parámetros

Atributo	Descripción
altura	Especifica la altura del lienzo.
anchura	Especifica el ancho del lienzo.

## Observaciones

- Esta etiqueta no es compatible con versiones de Internet Explorer de menos de 9. Consulte [caniuse.com](http://caniuse.com) para ver la compatibilidad del navegador.
- `canvas` es solo un contenedor para gráficos, y el dibujo real de gráficos se realiza mediante JavaScript.

## Examples

### Ejemplo básico

El elemento `canvas` se introdujo en HTML5 para dibujar gráficos.

```
<canvas id="myCanvas">
  Cannot display graphic. Canvas is not supported by your browser (IE<9)
</canvas>
```

Lo anterior creará un elemento HTML transparente `<canvas>` de 300 x 150 px de tamaño.

Puedes usar el elemento del **lienzo** para dibujar cosas sorprendentes como formas, gráficos, manipular imágenes, crear juegos atractivos, etc. con **JavaScript**.

El objeto de superficie de *capa* `CanvasRenderingContext2D` 2D del `canvas` se conoce como `CanvasRenderingContext2D`; o de un `HTMLCanvasElement` usando el `.getContext("2d")`:

```
var ctx = document.getElementById("myCanvas").getContext("2d");
// now we can refer to the canvas's 2D layer context using `ctx`

ctx.fillStyle = "#f00";
ctx.fillRect(0, 0, ctx.canvas.width, ctx.canvas.height); // x, y, width, height

ctx.fillStyle = "#000";
ctx.fillText("My red canvas with some black text", 24, 32); // text, x, y
```

[Ejemplo de jsFiddle](#)

## Dibujando dos rectángulos en una

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Draw two rectangles on the canvas</title>
  <style>
    canvas{
      border:1px solid gray;
    }
  </style>
  <script async>
    window.onload = init; // call init() once the window is completely loaded
    function init(){
      // #1 - get reference to <canvas> element
      var canvas = document.querySelector('canvas');

      // #2 - get reference to the drawing context and drawing API
      var ctx = canvas.getContext('2d');

      // #3 - all fill operations are now in red
      ctx.fillStyle = 'red';

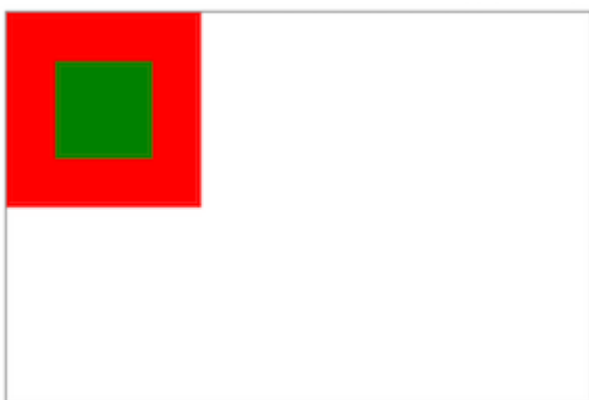
      // #4 - fill a 100x100 rectangle at x=0,y=0
      ctx.fillRect(0,0,100,100);

      // #5 - all fill operations are now in green
      ctx.fillStyle = 'green';

      // #6 - fill a 50x50 rectangle at x=25,y=25
      ctx.fillRect(25,25,50,50);

    }
  </script>
</head>
<body>
  <canvas width=300 height=200>Your browser does not support canvas.</canvas>
</body>
</html>
```

Este ejemplo se ve así:



Lea Lona en línea: <https://riptutorial.com/es/html/topic/1162/lona>

# Capítulo 33: Mapas de imagen

## Sintaxis

- `<img usemap="#[map-name]">`
- `<map name="[map-name]"></map>`
- `<area>`

## Parámetros

Etiqueta / Atributo	Valor
<code>&lt;img&gt;</code>	A continuación se muestran los atributos específicos del mapa de imagen para usar con <code>&lt;img&gt;</code> . Se aplican los atributos <code>&lt;img&gt;</code> regulares.
<code>usemap</code>	El <code>name</code> del mapa con un símbolo de hash precedido. Por ejemplo, para un mapa con <code>name="map"</code> , la imagen debe tener <code>usemap="#map"</code> .
<code>&lt;map&gt;</code>	
<code>name</code>	El nombre del mapa para identificarlo. Para ser utilizado con el atributo <code>usemap</code> la imagen.
<code>&lt;area&gt;</code>	A continuación se muestran los atributos específicos de <code>&lt;area&gt;</code> . Cuando se especifica <code>href</code> , haciendo de <code>&lt;area&gt;</code> un enlace, <code>&lt;area&gt;</code> también admite todos los atributos de la etiqueta de ancla ( <code>&lt;a&gt;</code> ) excepto <code>ping</code> . Véalos en los <a href="#">documentos de MDN</a> .
<code>alt</code>	El texto alternativo para mostrar si las imágenes no son compatibles. Esto solo es necesario si <code>href</code> también se configura en <code>&lt;area&gt;</code> .
<code>coords</code>	Las coordenadas que describen el área seleccionable. Cuando <code>shape="polygon"</code> , esto se debe establecer en una lista de "x, y" pares separados por comas (es decir, <code>shape="polygon" coords="x1, y1, x2, y2, x3, y3, ..."</code> ). Cuando <code>shape="rectangle"</code> , esto debe configurarse a <code>left, top, right, bottom</code> . Cuando <code>shape="circle"</code> , esto debe establecerse en <code>centerX, centerY, radius</code> .
<code>href</code>	La URL del hipervínculo, si se especifica. Si se omite, entonces <code>&lt;area&gt;</code> no representará un hipervínculo.
<code>shape</code>	La forma de la <code>&lt;area&gt;</code> . Se puede establecer en <code>default</code> para seleccionar toda la imagen (no hay <code>coords</code> atributo necesario), <code>circle</code> o <code>circ</code> para un círculo, <code>rectangle</code> o <code>rect</code> para un rectángulo, y <code>polygon</code> o <code>poly</code> para un área poligonal especificada por puntos de esquina.

## Observaciones

- La lista de parámetros anterior se modifica desde los documentos MDN: `<map>` y `<area>` .
- Es posible crear las coordenadas de un mapa de imagen para una imagen con formas más simples (como en el ejemplo introductorio anterior) con un editor de imágenes que muestra las coordenadas (como GIMP). Sin embargo, podría ser más fácil en general usar un generador de mapas de imágenes, como [este](#) .

## Examples

### Introducción a los mapas de imagen

---

## Descripción

Un mapa de imagen es una imagen con áreas en las que se puede hacer clic que generalmente actúan como hipervínculos.

La imagen está definida por la etiqueta `<img>` , y el mapa está definido por una etiqueta `<map>` con etiquetas `<area>` para indicar cada área en la que se puede hacer clic. Utilice los `usemap` y `name` para enlazar la imagen y el mapa.

---

## Ejemplo básico

Para crear un mapa de imágenes de modo que se pueda hacer clic en cada una de las formas de la imagen a continuación:



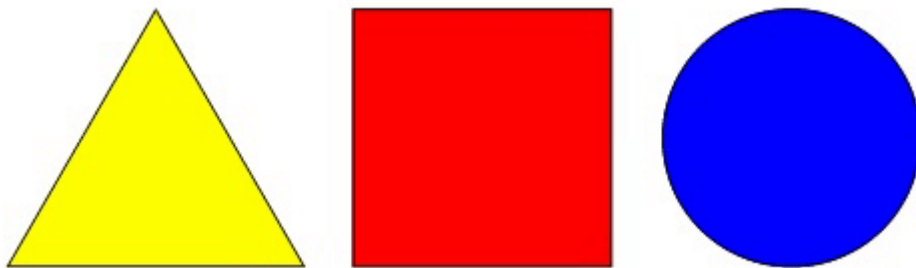
El código sería el siguiente:

```

<map name="shapes">
  <area shape="polygon" coords="79,6,5,134,153,134">
  <area shape="rectangle" coords="177,6,306,134">
  <area shape="circle" coords="397,71,65">
</map>
```

Debería ver que el navegador reconoce las áreas cuando el cursor se convierte en un puntero.

Vea una [demostración](#) en [vivo](#) en JSFiddle, o vea una demostración a continuación:



Lea Mapas de imagen en línea: <https://riptutorial.com/es/html/topic/3819/mapas-de-imagen>

---

# Capítulo 34: Marcando código informático

## Sintaxis

- `<pre>Formatted text</pre>`
- `<code>Inline Code</code>`

## Observaciones

El elemento de `code` se debe utilizar para cualquier tipo de "cadena que una computadora reconocería" ( [HTML5](#) ), por ejemplo:

- código fuente
- términos de marcado / lenguajes de programación (nombres de elementos, nombres de funciones, etc.)
- nombres de archivos

## Elementos relacionados

Para las variables, se puede utilizar el [elemento](#) `var` .

Para salida de computadora, se puede usar el [elemento](#) `samp` .

Para la entrada del usuario, se puede utilizar el [elemento](#) `kbd` .

## Examples

### En línea con

Si una oración contiene código de computadora (por ejemplo, el nombre de un elemento HTML), use el elemento de `code` para marcarlo:

```
<p>The <a> element creates a hyperlink.</p>
```

### Bloque con

```
y
```

Si el formato (espacio en blanco, nuevas líneas, sangría) del código importa, use el elemento `pre` en combinación con el elemento de `code` :

```
<pre>
  <code>
x = 42
if x == 42:
    print "x is ...      ... 42"
```

```
</code>  
</pre>
```

Aún tiene que escapar los caracteres con un significado especial en HTML (como < con `&lt;` ), por lo que para *mostrar* un bloque de código HTML ( `<p>This is a paragraph.</p>` ), podría tener este aspecto:

```
<pre>  
  <code>  
    &lt;p>This is a paragraph.&lt;/p>  
  </code>  
</pre>
```

Lea **Marcando código informático en línea**: <https://riptutorial.com/es/html/topic/1836/marcando-codigo-informatico>



---

# Capítulo 35: Mesas

## Introducción

El elemento HTML `<table>` permite a los autores web mostrar datos tabulares (como texto, imágenes, enlaces, otras tablas, etc.) en una tabla bidimensional con filas y columnas de celdas.

## Sintaxis

- `<table></table>`
- `<thead></thead>`
- `<tbody></tbody>`
- `<tfoot></tfoot>`
- `<tr></tr>`
- `<th></th>`
- `<td></td>`

## Observaciones

Los diversos elementos de la tabla y sus atributos de contenido juntos definen el modelo de tabla. El elemento `<table>` es el elemento contenedor para los modelos de tabla / datos tabulares. Las tablas tienen filas, columnas y celdas dadas por sus descendientes. Las filas y columnas forman una cuadrícula; las celdas de una tabla deben cubrir completamente la cuadrícula sin superponerse. La siguiente lista describe los diversos elementos en el modelo de tabla:

- `<table>` - El elemento contenedor para modelos de tablas / datos tabulares. `<table>` representa datos con más de una dimensión en forma de tabla.
- `<caption>` - Título o título de la tabla (como una `figcaption` a una `figure` )
- `<col>` - Una columna (un elemento sin contenido)
- `<colgroup>` - una agrupación de columnas
- `<thead>` - encabezado de tabla (solo uno)
- `<tbody>` - Cuerpo / contenido de la tabla (múltiples están bien)
- `<tfoot>` - pie de tabla (solo uno)
- `<tr>` - fila de la tabla
- `<th>` - Celda de encabezado de tabla
- `<td>` - Celda de datos de tabla

Semánticamente, las tablas están destinadas a contener datos tabulares. Puede pensarlo como una forma de mostrar y describir datos que tendrían sentido en una hoja de cálculo (columnas y filas).

No se recomienda el uso de tablas para el diseño. En su lugar, use las reglas de CSS para el diseño y el formato, incluida la `display: table`

Una excepción notable que generalmente se muestra en la industria con respecto al uso del diseño de `<table>` es el correo electrónico HTML: algunos clientes de correo electrónico, incluido

Outlook, regresan a los motores de renderización más antiguos después de que Microsoft pierda su caso de monopolio frente a la UE. Para que Microsoft haga que IE no sea parte del sistema operativo, simplemente hicieron retroceder el motor de renderizado de Outlook a una versión anterior de Trident. Esta reversión simplemente no es compatible con las tecnologías web modernas, por lo que el uso de diseños basados en `<table>` para correo electrónico HTML es la única manera de garantizar la compatibilidad entre navegadores / plataforma / cliente.

## Examples

### Mesa simple

```
<table>
  <tr>
    <th>Heading 1/Column 1</th>
    <th>Heading 2/Column 2</th>
  </tr>
  <tr>
    <td>Row 1 Data Column 1</td>
    <td>Row 1 Data Column 2</td>
  </tr>
  <tr>
    <td>Row 2 Data Column 1</td>
    <td>Row 2 Data Column 2</td>
  </tr>
</table>
```

Esto generará una `<table>` consta de tres filas totales ( `<tr>` ): una fila de celdas de encabezado ( `<th>` ) y dos filas de celdas de contenido ( `<td>` ). `<th>` elementos `<th>` son *encabezados tabulares* y los elementos `<td>` son *datos tabulares* . Puedes poner lo que quieras dentro de `<td>` o `<th>` .

Título 1 / Columna 1	Título 2 / Columna 2
Fila 1 Columna de datos 1	Fila 1 Columna de datos 2
Fila 2 Columna de datos 1	Fila 2 Columna de datos 2

### Abarcando columnas o filas

Las celdas de la tabla pueden abarcar varias columnas o filas usando los atributos `colspan` y `rowspan` . Estos atributos se pueden aplicar a los elementos `<th>` y `<td>` .

```
<table>
  <tr>
    <td>row 1 col 1</td>
    <td>row 1 col 2</td>
    <td>row 1 col 3</td>
  </tr>
  <tr>
    <td colspan="3">This second row spans all three columns</td>
  </tr>
</table>
```

```

        <td rowspan="2">This cell spans two rows</td>
        <td>row 3 col 2</td>
        <td>row 3 col 3</td>
    </tr>
    <tr>
        <td>row 4 col 2</td>
        <td>row 4 col 3</td>
    </tr>
</table>

```

Resultará en

row 1 col 1	row 1 col 2	row 1 col 3
This second row spans all three columns		
This cell spans two rows	row 3 col 2	row 3 col 3
	row 4 col 2	row 4 col 3

Tenga en cuenta que no debe diseñar una tabla en la que tanto las filas como las columnas se superpongan, ya que este no es un código HTML válido y el resultado se maneja de manera diferente por diferentes navegadores web.

`rowspan` = Un entero no negativo que especifica el número de filas que ocupa una celda. El valor predeterminado de este atributo es uno ( 1 ). Un valor de cero ( 0 ) significa que la celda se extenderá desde la fila actual hasta la última fila de la tabla ( `<thead>` , `<tbody>` o `<tfoot>` ).

`colspan` = Un entero no negativo que especifica el número de columnas abarcadas por la celda actual. El valor predeterminado de este atributo es uno ( 1 ). Un valor de cero ( 0 ) significa que la celda se extenderá desde la corriente hasta la última columna del grupo de columnas `<colgroup>` en el que se define la celda.

## Tabla con `thead`, `tbody`, `tfoot` y subtítulo

HTML también proporciona las tablas con los elementos `<thead>` , `<tbody>` , `<tfoot>` y `<caption>` . Estos elementos adicionales son útiles para agregar valor semántico a sus tablas y para proporcionar un lugar para el estilo CSS separado.

Al imprimir una tabla que no cabe en una página (papel), la mayoría de los navegadores repiten el contenido de `<thead>` en cada página.

Hay un orden específico que debe cumplirse, y debemos ser conscientes de que no todos los elementos encajan en su lugar como uno esperaría. El siguiente ejemplo muestra cómo se deben colocar nuestros 4 elementos.

```

<table>
  <caption>Table Title</caption> <!--| caption is the first child of table |-->
  <thead> <!--| head is after caption |-->
    <tr>
      <th>Header content 1</th>
      <th>Header content 2</th>

```

```

</tr>
</thead>

<tbody> <!--=====| tbody is after thead |-->
  <tr>
    <td>Body content 1</td>
    <td>Body content 2</td>
  </tr>
</tbody>

<tfoot><!--| tfoot can be placed before or after tbody, but not in a group of tbody. |-->
<!--| Regardless where tfoot is in markup, it's rendered at the bottom. |-->

  <tr>
    <td>Footer content 1</td>
    <td>Footer content 2</td>
  </tr>
</tfoot>

</table>

```

Los resultados del siguiente ejemplo se muestran dos veces: la primera tabla carece de estilos, la segunda tabla tiene algunas propiedades CSS aplicadas: `background-color` , `color` y `border` \*. Los estilos se proporcionan como una guía visual y no son un aspecto esencial del tema en cuestión.

Table Title	
<b>Header content 1</b>	<b>Header content 2</b>
Body content 1	Body content 2
Footer content 1	Footer content 2

Table Title	
<b>Header content 1</b>	<b>Header content 2</b>
Body content 1	Body content 2
Footer content 1	Footer content 2

Elemento	Estilos Aplica
<caption>	Texto amarillo sobre fondo negro.
<thead>	Texto en negrita en el fondo púrpura.
<tbody>	Texto sobre fondo azul.
<tfoot>	Texto sobre fondo verde.
<th>	Bordes naranjas.
<td>	Bordes rojos.

## Grupos de columnas

A veces es posible que desee aplicar un estilo a una columna o grupo de columnas. O para fines semánticos, puede agrupar columnas. Para hacer esto, use los elementos `<colgroup>` y `<col>` .

La etiqueta opcional `<colgroup>` permite agrupar columnas. `<colgroup>` elementos `<colgroup>` deben ser elementos secundarios de una `<table>` y deben ir después de cualquier elemento `<caption>` y antes del contenido de la tabla (por ejemplo, `<tr>` , `<thead>` , `<tbody>` , etc.).

```
<table>
  <colgroup span="2"></colgroup>
  <colgroup span="2"></colgroup>
  ...
</table>
```

La etiqueta opcional `<col>` permite hacer referencia a columnas individuales o un rango de columnas sin aplicar una agrupación lógica. `<col>` elementos `<col>` son opcionales, pero si están presentes, deben estar dentro de un elemento `<colgroup>` .

```
<table>
  <colgroup>
    <col id="MySpecialColumn" />
    <col />
  </colgroup>
  <colgroup>
    <col class="CoolColumn" />
    <col class="NeatColumn" span="2" />
  </colgroup>
  ...
</table>
```

Los siguientes estilos CSS se pueden aplicar a los elementos `<colgroup>` y `<col>` :

- `border`
- `background`
- `width`
- `visibility`
- `display` (como en la `display: none` )
  - `display: none`; realmente eliminará las columnas de la pantalla, lo que hará que la tabla se muestre como si esas celdas no existieran

Para más información, vea [HTML5 Tabular data](#) .

## Ámbito de aplicación

`th` elementos se usan con mucha frecuencia para indicar encabezados de filas y columnas de tablas, de esta manera:

```
<table>
  <thead>
```

```

    <tr>
      <td></td>
      <th>Column Heading 1</th>
      <th>Column Heading 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th>Row Heading 2</th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>

```

Esto se puede mejorar para accesibilidad mediante el uso del atributo de `scope`. El ejemplo anterior se enmendaría como sigue:

```

<table>
  <thead>
    <tr>
      <td></td>
      <th scope="col">Column Heading 1</th>
      <th scope="col">Column Heading 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th scope="row">Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>

```

`scope` se conoce como un *atributo enumerado*, lo que significa que puede tener un valor de un conjunto específico de valores posibles. Este set incluye:

- col
- row
- colgroup
- rowgroup

#### Referencias:

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/th#attr-scope>
- <https://www.w3.org/TR/WCAG20-TECHS/H63.html>

Lea Mesas en línea: <https://riptutorial.com/es/html/topic/274/mesas>

---

# Capítulo 36: Meta información

## Introducción

Las etiquetas meta en los documentos HTML proporcionan información útil sobre el documento, que incluye una descripción, palabras clave, autor, fechas de modificaciones y alrededor de otros 90 campos. Este tema cubre el uso y propósito de estas etiquetas.

## Sintaxis

- `<meta name="metadata name" content="value">`
- `<meta http-equiv="pragma directive" content="value">`
- `<meta charset="encoding label">`

## Observaciones

La etiqueta meta es una etiqueta HTML utilizada para establecer los metadatos del documento HTML. Las etiquetas meta deben ir en el elemento principal. Una página puede tener cualquier número de etiquetas meta.

Las `keywords` metaetiqueta no son utilizadas típicamente por los robots. La mayoría de los motores de búsqueda determinan qué palabras clave encajan con el contenido de las páginas web. Dicho esto, nada dice que ya no se debe incluir la metaetiqueta de palabras clave.

Los metadatos de una página son utilizados principalmente por el navegador (como la escala de un documento) y las arañas de rastreo web utilizadas por los motores de búsqueda (Google, Yahoo !, Bing).

La especificación proporciona varios [nombres de metadatos estandarizados](#) para usar con `<meta name>` y [directivas pragma de metadatos estandarizados](#) para usar con `<meta http-equiv>`. Sin embargo, muchos servicios a través de Internet (rastreadores web, herramientas de creación, servicios para compartir en redes sociales, etc.) utilizan el formulario `<meta name>` como un punto de extensión genérico para los metadatos. Algunos de estos están listados [en la página wiki de la especificación](#).

## Examples

### Codificación de caracteres

El atributo `charset` especifica la codificación de caracteres para el documento HTML y debe ser una codificación de caracteres válida (los ejemplos incluyen `windows-1252`, `ISO-8859-2`, `Shift_JIS` y `UTF-8`). `UTF-8` (Unicode) es el más utilizado y debe usarse para cualquier proyecto nuevo.



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

```
<meta charset="ISO-8859-1">
```

Todos los navegadores siempre han reconocido el formulario `<meta charset>` , pero si por alguna razón necesita que su página sea HTML 4.01 válido, puede usar lo siguiente:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

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

Consulte también el [Estándar de codificación](#) , para ver todas las etiquetas de codificación de caracteres disponibles que reconocen los navegadores.

## Actualización automática

Para actualizar la página cada cinco segundos, agregar esta `meta` elemento de la `head` elemento:

```
<meta http-equiv="refresh" content="5">
```

**¡PRECAUCIÓN!** Si bien este es un comando válido, se recomienda que no lo use debido a sus efectos negativos en la experiencia del usuario. Actualizar la página con demasiada frecuencia puede hacer que deje de responder y, a menudo, se desplaza a la parte superior de la página. Si alguna información en la página necesita actualizarse continuamente, hay maneras mucho mejores de hacerlo solo actualizando una parte de la página.

## Control de diseño móvil

Los sitios comunes optimizados para dispositivos móviles utilizan la etiqueta `<meta name="viewport">` así:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

El elemento de `viewport` le da instrucciones al navegador sobre cómo controlar las dimensiones de la página y la escala en función del dispositivo que está utilizando.

En el ejemplo anterior, `content="width=device-width"` significa que el navegador mostrará el ancho de la página en el ancho de su propia pantalla. Entonces, si esa pantalla tiene un `480px wide` , la ventana del navegador tendrá un `480px wide` . `initial-scale=1` representa que el zoom inicial (que es 1 en este caso, significa que no hace zoom).

A continuación se muestran los atributos que admite esta etiqueta:

Atributo	Descripción
<code>width</code>	El ancho de la ventana virtual del dispositivo. Valores <sup>1</sup> : <code>device-width</code> o ancho real en píxeles, como <code>480</code>

Atributo	Descripción
height	La altura de la ventana virtual del dispositivo. Valores <sup>2</sup> : <code>device-height</code> o el ancho real en píxeles, como <code>600</code>
initial-scale	El zoom inicial cuando se carga la página. <code>1.0</code> no hace zoom.
minimum-scale	La cantidad mínima que el visitante puede hacer zoom en la página. <code>1.0</code> no hace zoom.
maximum-scale	La cantidad máxima que el visitante puede hacer zoom en la página. <code>1.0</code> no hace zoom.
user-scalable	Permite al dispositivo acercarse y alejarse. Los valores son <code>yes</code> o <code>no</code> . Si se establece en <code>no</code> , el usuario no puede ampliar la página web. El valor predeterminado es <code>sí</code> . La configuración del navegador puede ignorar esta regla.

### Notas:

<sup>1</sup> La propiedad de `width` se puede especificar en *píxeles* ( `width=600` ) o por *ancho de dispositivo* ( `width=device-width` ) que representa el ancho físico de la pantalla del dispositivo.

<sup>2</sup> Del mismo modo, la propiedad de `height` puede especificarse en *pixels* ( `height=600` ) o por *device-height* ( `height=device-height` ) que representa la altura física de la pantalla del dispositivo.

## Información de la página

### application-name

Dar el nombre de la aplicación web que representa la página.

```
<meta name="application-name" content="OpenStreetMap">
```

Si no es una aplicación web, no se debe utilizar la metaetiqueta de `application-name` la `application-name`.

### author

Establecer el autor de la página:

```
<meta name="author" content="Your Name">
```

Sólo se puede dar un nombre.

### description

Establecer la descripción de la página:

```
<meta name="description" content="Page Description">
```

La metaetiqueta de `description` puede ser utilizada por varios motores de búsqueda mientras indexa su página web con fines de búsqueda. Por lo general, la descripción contenida en la etiqueta meta es el breve resumen que se muestra debajo del título principal de la página / sitio web en los resultados del motor de búsqueda. Google usualmente usa solo las primeras 20-25 palabras de su descripción.

#### generator

```
<meta name="generator" content="HTML Generator 1.42">
```

Identifica uno de los paquetes de software utilizados para generar el documento. Solo para ser utilizado para páginas donde el marcado se genera automáticamente.

#### keywords

Establecer palabras clave para los motores de búsqueda (separados por comas):

```
<meta name="keywords" content="Keyword1, Keyword2">
```

Las etiquetas meta de `keywords` veces son utilizadas por los motores de búsqueda para conocer la consulta de búsqueda que es relevante para su página web.

Como regla general, probablemente sea una buena idea no agregar demasiadas palabras, ya que la mayoría de los motores de búsqueda que usan esta etiqueta meta para indexar solo indexarán las primeras ~ 20 palabras. Asegúrese de poner primero las palabras clave más importantes.

## Los robots

El atributo `robots`, soportado por varios motores de búsqueda principales, controla si las arañas de los motores de búsqueda pueden indexar una página o no y si deben seguir los enlaces de una página o no.

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

Este ejemplo indica a todos los motores de búsqueda que no muestren la página en los resultados de búsqueda. Otros valores permitidos son:

Valor / Directiva	Sentido
all	<b>Defecto.</b> Equivalente a <code>index, follow</code> . Vea la nota abajo.
noindex	No indexar la página en absoluto.
nofollow	No sigas los enlaces de esta página.
follow	Los enlaces en la página pueden seguirse. Vea la nota abajo.
none	Equivalente a <code>noindex, nofollow</code> .

Valor / Directiva	Sentido
<code>noarchive</code>	No haga una versión en caché de esta página disponible en los resultados de búsqueda.
<code>nocache</code>	Sinónimo de <code>noarchive</code> utilizado por algunos bots como Bing.
<code>nosnippet</code>	No mostrar un fragmento de esta página en los resultados de búsqueda.
<code>noodp</code>	No utilice los metadatos de esta página del <a href="#">proyecto de Open Directory</a> para títulos o fragmentos en los resultados de búsqueda.
<code>notranslate</code>	No ofrezca traducciones de esta página en los resultados de búsqueda.
<code>noimageindex</code>	No indexar imágenes en esta página.
<code>unavailable_after [RFC-850 date/time]</code>	No mostrar esta página en los resultados de búsqueda después de la fecha / hora especificada. La fecha / hora debe especificarse en el <a href="#">formato RFC 850</a> .

**Nota:** Definir explícitamente el `index` y / o el `follow` , mientras que los valores válidos, no es necesario, ya que prácticamente todos los motores de búsqueda asumirán que pueden hacerlo si no se les impide hacerlo explícitamente. De manera similar a como opera el archivo robots.txt, los motores de búsqueda generalmente solo buscan cosas que *no* tienen *permitido* hacer. Solo declarar cosas que un motor de búsqueda no tiene permitido hacer también evita que se declaren opuestos (como `index`, ..., `noindex` ) que no todos los motores de búsqueda tratarán de la misma manera.

## Reconocimiento de números de teléfono

Las plataformas móviles como iOS reconocen automáticamente los números de teléfono y los convierten en `tel:` links. Si bien la función es muy práctica, el sistema a veces detecta los códigos ISBN y otros números como números de teléfono.

Para que Safari móvil y algunos otros navegadores móviles basados en WebKit desactiven el reconocimiento y el formato automáticos del número de teléfono, necesita esta etiqueta meta:

```
<meta name="format-detection" content="telephone=no">
```

## Medios de comunicación social

Open Graph es un estándar para metadatos que extiende la información normal contenida en el marcado de encabezado de un sitio. Esto permite que los sitios web como Facebook muestren información más profunda y rica sobre un sitio web en un formato estructurado. Esta información se muestra automáticamente cuando los usuarios comparten enlaces a sitios web que contienen

metadatos OG en Facebook.

## Facebook / Open Graph

```
<meta property="fb:app_id" content="123456789">
<meta property="og:url" content="https://example.com/page.html">
<meta property="og:type" content="website">
<meta property="og:title" content="Content Title">
<meta property="og:image" content="https://example.com/image.jpg">
<meta property="og:description" content="Description Here">
<meta property="og:site_name" content="Site Name">
<meta property="og:locale" content="en_US">
<meta property="article:author" content="">
<!-- Facebook: https://developers.facebook.com/docs/sharing/webmasters#markup -->
<!-- Open Graph: http://ogp.me/ -->
```

- [Facebook Open Graph Markup](#)
- [Protocolo de gráfico abierto](#)

## Facebook / Artículos instantáneos

```
<meta charset="utf-8">
<meta property="op:markup_version" content="v1.0">

<!-- The URL of the web version of your article -->
<link rel="canonical" href="http://example.com/article.html">

<!-- The style to be used for this article -->
<meta property="fb:article_style" content="myarticlestyle">
```

- [Facebook Instant Articles: Creación de artículos](#)
- [Artículos instantáneos: Referencia de formato](#)

Twitter utiliza su propio marcado para metadatos. Estos metadatos se utilizan como información para controlar cómo se muestran los tweets cuando contienen un enlace al sitio.

## Gorjeo

```
<meta name="twitter:card" content="summary">
<meta name="twitter:site" content="@site_account">
<meta name="twitter:creator" content="@individual_account">
<meta name="twitter:url" content="https://example.com/page.html">
<meta name="twitter:title" content="Content Title">
<meta name="twitter:description" content="Content description less than 200 characters">
<meta name="twitter:image" content="https://example.com/image.jpg">
```

- [Tarjetas de Twitter: Guía de inicio](#)
- [Validador de tarjetas de Twitter](#)

## Google+ / Schema.org

```
<link href="https://plus.google.com/+YourPage" rel="publisher">
<meta itemprop="name" content="Content Title">
<meta itemprop="description" content="Content description less than 200 characters">
<meta itemprop="image" content="https://example.com/image.jpg">
```

## Redireccionamiento automatico

A veces su página web necesita una redirección automática.

Por ejemplo, para redirigir a `example.com` después de 5 segundos:

```
<meta http-equiv="refresh" content="5;url=https://www.example.com/" />
```

Esta línea lo enviará al sitio web designado (en este caso `example.com` después de 5 segundos).

Si necesita cambiar la demora de tiempo antes de una redirección, simplemente cambiando el número justo antes de su `;url=` modificará la demora de tiempo.

## Aplicación Web

Puede configurar su aplicación web o sitio web para agregar un ícono de acceso directo a la pantalla de inicio de un dispositivo y hacer que la aplicación se inicie en el "modo de aplicación" de pantalla completa usando el elemento del menú ["Agregar a la pantalla de inicio"](#) de Chrome para Android.

Las siguientes etiquetas meta (s) abrirán la aplicación web en modo de pantalla completa (sin barra de direcciones).

### Android Chrome

```
<meta name="mobile-web-app-capable" content="yes">
```

### IOS

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

También puede establecer el color para la barra de estado y la barra de direcciones en la etiqueta meta.

### Android Chrome

```
<meta name="theme-color" content="black">
```

### IOS

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

Lea Meta informacion en línea: <https://riptutorial.com/es/html/topic/1264/meta-informacion>

# Capítulo 37: Párrafos

## Introducción

Los párrafos son el elemento HTML más básico. Este tema explica y demuestra el uso del elemento de párrafo en HTML.

## Parámetros

Columna	Columna
<code>&lt;p&gt;</code>	Define un párrafo
<code>&lt;br&gt;</code>	Inserta un solo salto de línea
<code>&lt;pre&gt;</code>	Define el texto preformateado.

## Examples

### Párrafos HTML

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

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

Monitor-

No puede estar seguro de cómo se mostrará HTML.

Las pantallas grandes o pequeñas y las ventanas redimensionadas crearán resultados diferentes.

Con HTML, no puede cambiar la salida agregando espacios extra o líneas adicionales en su código HTML.

El navegador eliminará cualquier espacio adicional y líneas adicionales cuando se muestre la página:

```
<p>This is          another      paragraph, extra spaces  will be  removed by
browsers</p>
```

Lea Párrafos en línea: <https://riptutorial.com/es/html/topic/7997/parrafos>



# Capítulo 38: Recursos de enlace

## Introducción

Si bien muchos scripts, íconos y hojas de estilo se pueden escribir directamente en el formato HTML, es una práctica recomendada y más eficiente incluir estos recursos en su propio archivo y vincularlos a su documento. Este tema trata sobre la vinculación de recursos externos, como hojas de estilo y scripts, en un documento HTML.

## Sintaxis

- `<link rel="link-relation" type="mime-type" href="url">`
- `<script src="path-to-script"></script>`

## Parámetros

Atributo	Detalles
<code>charset</code>	Especifica la codificación de caracteres del documento vinculado.
<code>crossorigin</code>	Especifica cómo el elemento maneja las solicitudes de origen cruzado
<code>href</code>	Especifica la ubicación del documento vinculado.
<code>hreflang</code>	Especifica el idioma del texto en el documento vinculado.
<code>media</code>	Especifica en qué dispositivo se mostrará el documento vinculado, que se utiliza a menudo con la selección de hojas de estilo basadas en el dispositivo en cuestión
<code>rel</code>	<b>Requerido</b> Especifica la relación entre el documento actual y el documento vinculado.
<code>rev</code>	Especifica la relación entre el documento vinculado y el documento actual.
<code>sizes</code>	Especifica el tamaño del recurso vinculado. Solo cuando <code>rel="icon"</code>
<code>target</code>	Especifica dónde se cargará el documento vinculado
<code>type</code>	Especifica el tipo de medio del documento vinculado.
<code>integrity</code>	Especifica un hash codificado en base64 (sha256, sha384 o sha512) del recurso vinculado que permite al navegador verificar su legitimidad.

## Examples

## Hoja de estilo CSS externa

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

La práctica estándar es colocar etiquetas CSS `<link>` dentro de la etiqueta `<head>` en la parte superior de su HTML. De esta manera, primero se cargará el CSS y se aplicará a su página mientras se carga, en lugar de mostrar HTML sin estilo hasta que se cargue el CSS. El atributo de `type` no es necesario en HTML5, porque HTML5 generalmente es compatible con CSS.

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

y

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

... hacer lo mismo en HTML5.

Otra práctica, aunque menos común, es usar una declaración `@import` dentro de CSS directo. Me gusta esto:

```
<style type="text/css">
  @import("path/to.css")
</style>

<style>
  @import("path/to.css")
</style>
```

## JavaScript

### Sincrónico

```
<script src="path/to.js"></script>
```

La práctica estándar es colocar etiquetas `<script>` JavaScript justo antes de la etiqueta de cierre `</body>`. La última carga de sus scripts permite que las imágenes de su sitio se muestren más rápidamente y desanima a su JavaScript de intentar interactuar con elementos que aún no se han cargado.

### Asincrónico

```
<script src="path/to.js" async></script>
```

Otra alternativa, cuando el código de Javascript que se está cargando no es necesario para la inicialización de la página, se puede cargar de forma asíncrona, acelerando la carga de la página.

Usando `async` el navegador cargará el contenido del script en paralelo y, una vez que esté completamente descargado, interrumpirá el análisis HTML para analizar el archivo Javascript.

## Diferido

```
<script src="path/to.js" defer></script>
```

Los scripts diferidos son como los scripts asíncronos, con la excepción de que el análisis solo se realizará una vez que el HTML esté completamente analizado. Se garantiza que los scripts diferidos se carguen en el orden de la declaración, de la misma manera que los scripts síncronos.

## <noscript>

```
<noscript>JavaScript disabled</noscript>
```

El elemento `<noscript>` define el contenido que se mostrará si el usuario tiene los scripts deshabilitados o si el navegador no admite el uso de scripts. La etiqueta `<noscript>` se puede colocar en el `<head>` o en el `<body>` .

## Favicon

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

Utilice la `image/png` tipo mime para los archivos PNG y el `image/x-icon` para los archivos de iconos ( `*.ico` ). Para la diferencia, vea [esta pregunta SO](#) .

Un archivo llamado `favicon.ico` en la raíz de su sitio web normalmente se cargará y aplicará automáticamente, sin la necesidad de una etiqueta `<link>` . Si este archivo cambia, los navegadores pueden ser lentos y obstinados en cuanto a la actualización de su caché.

## CSS alternativo

```
<link rel="alternate stylesheet" href="path/to/style.css" title="yourTitle">
```

Algunos navegadores permiten que se apliquen hojas de estilo alternativas si se ofrecen. De forma predeterminada, no se aplicarán, pero generalmente se pueden cambiar a través de la configuración del navegador:

Firefox le permite al usuario seleccionar la hoja de estilo usando el submenú Ver > Estilo de página, Internet Explorer también admite esta función (que comienza con IE 8), también accesible desde Ver > Estilo de página (al menos desde IE 11), pero Chrome requiere una extensión para utilizar la función (a partir de la versión 48). La página web también puede proporcionar su propia interfaz de usuario para permitir

que el usuario cambie de estilos.

(Fuente: los [documentos MDN](#) )

## Alimentación web

Use el `rel="alternate"` para permitir la detección de sus fuentes Atom / RSS.

```
<link rel="alternate" type="application/atom+xml" href="http://example.com/feed.xml" />
<link rel="alternate" type="application/rss+xml" href="http://example.com/feed.xml" />
```

Consulte la documentación de MDN para [las fuentes RSS](#) y [RSS atómica](#) .

## Enlace del atributo 'media'

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

Medios especifica qué hoja de estilo debe usarse para qué tipo de medios. El uso del valor de `print` solo mostraría esa hoja de estilo para las páginas impresas.

El valor de este atributo puede ser cualquiera de los [mediatype valores](#) (similar a un CSS [consulta de medios](#) ).

## Anterior y Siguiente

Cuando una página es parte de una serie de artículos, por ejemplo, uno puede usar `prev` y `next` para señalar las páginas que vienen antes y después.

```
<link rel="prev" href="http://stackoverflow.com/documentation/java/topics">
<link rel="next" href="http://stackoverflow.com/documentation/css/topics">
```

## Sugerencia de recursos: dns-prefetch, prefetch, prerender

### Preconectarse

La relación `preconnect` la `preconnect` es similar a `dns-prefetch` ya que resolverá el DNS. Sin embargo, también hará que el protocolo de enlace TCP y la negociación TLS opcional. Esta es una característica experimental.

```
<link rel="preconnect" href="URL">
```

### DNS-Prefetch

Informa a los navegadores para que resuelvan el DNS para una URL, de modo que todos los

activos de esa URL se carguen más rápido.

```
<link rel="dns-prefetch" href="URL">
```

---

## Prefetch

Informa a los navegadores de que un recurso dado debe ser predefinido para que pueda cargarse más rápidamente.

```
<link rel="prefetch" href="URL">
```

DNS-Prefetch resuelve solo el nombre de dominio mientras que prefetch descarga / almacena los recursos especificados.

---

## Prerender

Informa a los navegadores para que obtengan y representen la URL en segundo plano, de modo que puedan enviarse al usuario de forma instantánea mientras el usuario navega a esa URL. Esta es una característica experimental.

```
<link rel="prerender" href="URL">
```

Lea Recursos de enlace en línea: <https://riptutorial.com/es/html/topic/712/recursos-de-enlace>

---

# Capítulo 39: SVG

## Introducción

SVG significa gráficos vectoriales escalables. SVG se utiliza para definir gráficos para la web.

El elemento HTML `<svg>` es un contenedor para gráficos SVG.

SVG tiene varios métodos para dibujar trazados, cuadros, círculos, texto e imágenes gráficas.

## Observaciones

SVG es un lenguaje basado en XML para crear imágenes vectoriales escalables. Puede escribirse directamente en un documento HTML o incrustarse desde archivos SVG externos. Inline SVG puede ser rediseñado y modificado usando CSS y JavaScript respectivamente.

El soporte del navegador para SVG varía, pero se puede determinar [aquí](#) .

Para obtener información más completa, consulte la [documentación de SVG](#) .

## Examples

### Incrustar archivos SVG externos en HTML

Puede usar los elementos `<img>` o `<object>` para incrustar elementos SVG externos. Establecer la altura y el ancho es opcional, pero es altamente recomendable.

### Usando el elemento de imagen

```

```

El uso de `<img>` no le permite diseñar el SVG utilizando CSS o manipularlo utilizando JavaScript.

### Usando el elemento objeto

```
<object type="image/svg+xml" data="attention.svg" width="50" height="50">
```

A diferencia de `<img>` , `<object>` importa directamente el SVG en el documento y, por lo tanto, se puede manipular utilizando Javascript y CSS.

### SVG en línea

SVG puede escribirse directamente en un documento HTML. Inline SVG se puede diseñar y manipular usando CSS y JavaScript.

```

<body>
  <svg class="attention" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 1000 1000" >
    <path id="attention" d="m571,767l0,-106q0,-8,-5,-13t-12,-5l-108,0q-7,0,-12,5t-
5,13l0,106q0,8,5,13t12,6l108,0q7,0,12,-6t5,-13Zm-1,-208l10,-257q0,-6,-5,-10q-7,-6,-14,-6l-
122,0q-7,0,-14,6q-5,4,-5,12l9,255q0,5,6,9t13,3l103,0q8,0,13,-3t6,-9Zm-7,-522l428,786q20,35,-
1,70q-10,17,-26,26t-35,10l-858,0q-18,0,-35,-10t-26,-26q-21,-35,-1,-70l429,-786q9,-17,26,-
27t36,-10t36,10t27,27Z" />
  </svg>
</body>

```

El SVG en línea anterior se puede diseñar utilizando la clase CSS correspondiente:

```

.attention {
  fill: red;
  width: 50px;
  height: 50px;
}

```

El resultado se ve así:



## Incrustar SVG usando CSS

Puede agregar archivos SVG externos usando la propiedad de `background-image`, como haría con cualquier otra imagen.

HTML:

```
<div class="attention"></div>
```

CSS:

```

.attention {
  background-image: url(attention.svg);
  background-size: 100% 100%;
  width: 50px;
  height: 50px;
}

```

También puede incrustar la imagen directamente en un archivo css usando una url de datos:

```

background-image:
url(data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsvg%22%20xmlns%3Axlink%3D%
106q0%2C-8%2C-5%2C-13t-12%2C-5l-108%2C0q-7%2C0%2C-12%2C5t-
5%2C13l0%2C106q0%2C8%2C5%2C13t12%2C6l108%2C0q7%2C0%2C12%2C-6t5%2C-13Zm-1%2C-208l10%2C-
257q0%2C-6%2C-5%2C-10q-7%2C-6%2C-14%2C-6l-122%2C0q-7%2C0%2C-14%2C6q-5%2C4%2C-
5%2C12l9%2C255q0%2C5%2C6%2C9t13%2C3l103%2C0q8%2C0%2C13%2C-3t6%2C-9Zm-7%2C-
522l428%2C786q20%2C35%2C-1%2C70q-10%2C17%2C-26%2C26t-35%2C10l-858%2C0q-18%2C0%2C-35%2C-10t-
26%2C-26q-21%2C-35%2C-1%2C-70l429%2C-786q9%2C-17%2C26%2C-27t36%2C-

```

10t36%2C10t27%2C27Z%22%20%2F%3E%0D%0A%3C%2Fsvg%3E);

Lea SVG en línea: <https://riptutorial.com/es/html/topic/1183/svg>



# Capítulo 40: Tabindex

## Parámetros

Valor	Sentido
negativo	El elemento será enfocable, pero no debería ser accesible a través de la navegación secuencial del teclado
0	El elemento será enfocable y accesible a través de la navegación secuencial del teclado, pero su orden relativo está definido por la convención de la plataforma
positivo	el elemento debe ser enfocable y accesible a través de la navegación secuencial del teclado; su orden relativo se definirá por el valor del atributo: la secuencia sigue el número creciente de <code>tabindex</code>

## Observaciones

El valor máximo para `tabindex` no debe exceder de 32767 según la sección 17.11.1 de W3C, a menos que el valor predeterminado especificado sea -1

Un elemento con un valor de 0, un valor no válido o ningún valor de `tabindex` debe colocarse después de los elementos con un índice positivo en el orden secuencial de navegación del teclado.

## Examples

### Agregar un elemento al orden de tabulación

```
<div tabindex="0">Some button</div>
```

**Nota** : intente utilizar un `button` HTML nativo o a etiqueta cuando sea apropiado.

### Eliminar un elemento del orden de tabulación

```
<button tabindex="-1">This button will not be reachable by tab</button>
```

El elemento se eliminará del orden de tabulación, pero aún será enfocable.

### Definir un orden de tabulación personalizado (no recomendado)

```
<div tabindex="2">Second</div>  
<div tabindex="1">First</div>
```

Los valores positivos insertarán el elemento en la posición de orden de tabulación de su valor respectivo. Los elementos sin preferencia (es decir, `tabindex="0"` o elementos nativos como el `button` y `a`) se agregarán después de aquellos con preferencia.

Los valores positivos **no se recomiendan** ya que interrumpen el comportamiento esperado de tabulación y pueden confundir a las personas que dependen de los lectores de pantalla. Intente crear un orden natural reorganizando su estructura DOM.

Lea Tabindex en línea: <https://riptutorial.com/es/html/topic/2594/tabindex>

# Capítulo 41: Usando HTML con CSS

## Introducción

**CSS** proporciona estilos a los elementos HTML en la página. El estilo en línea implica el uso del atributo de estilo en las etiquetas, y no se recomienda. Las hojas de estilo internas usan la etiqueta `<style>` y se usan para declarar reglas para partes dirigidas de la página. Las hojas de estilo externas se pueden usar a través de una etiqueta `<link>` que toma un archivo externo de CSS y aplica las reglas al documento. Este tema cubre el uso de los tres métodos de apego.

## Sintaxis

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

## Examples

### Uso externo de la hoja de estilo

Utilice el atributo de `link` en la `head` del documento:

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

También puede usar las hojas de estilo proporcionadas desde sitios web a través de una red de distribución de contenido, o CDN para abreviar. (por ejemplo, Bootstrap):

```
<head>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiisSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
</head>
```

En general, puede encontrar soporte CDN para un marco en su sitio web.

### Hoja de estilo interna

También puede incluir elementos CSS internamente usando la etiqueta `<style>` :

```
<head>
  <style type="text/css">
    body {
      background-color: gray;
    }
  </style>
```

```
</head>
```

También se pueden incluir múltiples hojas de estilo internas en un programa.

```
<head>
  <style type="text/css">
    body {
      background-color: gray;
    }
  </style>

  <style type="text/css">
    p {
      background-color: blue;
    }
  </style>
</head>
```

## Estilo en línea

Puede diseñar un elemento específico utilizando el atributo de `style` :

```
<span style="color: red">This text will appear in red.</span>
```

Nota: intente evitar esto: el punto de CSS es separar el contenido de la presentación.

## Múltiples hojas de estilo

Es posible cargar múltiples hojas de estilo:

```
<head>
  <link rel="stylesheet" type="text/css" href="general.css">
  <link rel="stylesheet" type="text/css" href="specific.css">
</head>
```

Tenga en cuenta que los **archivos y las declaraciones posteriores anularán los anteriores** . Así que si `general.css` contiene:

```
body {
  background-color: red;
}
```

y `specific.css` contiene:

```
body {
  background-color: blue;
}
```

Si se usan ambos, el fondo del documento será azul.

Lea Usando HTML con CSS en línea: <https://riptutorial.com/es/html/topic/4536/usando-html-con>



# Creditos

S. No	Capítulos	Contributors
1	Empezando con HTML	<a href="#">4444</a> , <a href="#">Abhishek Pandey</a> , <a href="#">aea2002</a> , <a href="#">ahmednawazbutt</a> , <a href="#">Alexander Wigmore</a> , <a href="#">Alexandre N.</a> , <a href="#">Amanda Ahn</a> , <a href="#">amflare</a> , <a href="#">Amitay Stern</a> , <a href="#">animuson</a> , <a href="#">Anthony Pham</a> , <a href="#">Boris</a> , <a href="#">bwegs</a> , <a href="#">Callan Heard</a> , <a href="#">ChrisD</a> , <a href="#">CocoaBean</a> , <a href="#">Community</a> , <a href="#">Dave Everitt</a> , <a href="#">Dinidu</a> , <a href="#">dippas</a> , <a href="#">dtyler</a> , <a href="#">duskwuff</a> , <a href="#">Eric Dobbs</a> , <a href="#">Firix</a> , <a href="#">FlyingPiMonster</a> , <a href="#">geek1011</a> , <a href="#">George Bailey</a> , <a href="#">Gerold Broser</a> , <a href="#">H Mirza</a> , <a href="#">H. Pauwelyn</a> , <a href="#">Harish Gyanani</a> , <a href="#">Hemant Kumar</a> , <a href="#">hillary.fraley</a> , <a href="#">Hudson Taylor</a> , <a href="#">ihavemorealts</a> , <a href="#">intboolstring</a> , <a href="#">Isak Combrinck</a> , <a href="#">Jeffrey Lin</a> , <a href="#">JHS</a> , <a href="#">jmmarco</a> , <a href="#">joe_young</a> , <a href="#">John Slegers</a> , <a href="#">Jon Chan</a> , <a href="#">JonasCz</a> , <a href="#">JPB</a> , <a href="#">kelvinlove</a> , <a href="#">Krii</a> , <a href="#">Kurniawantaari</a> , <a href="#">Lahiru Ashan</a> , <a href="#">Lambda Ninja</a> , <a href="#">Léo Martin</a> , <a href="#">Leonidas Menendez</a> , <a href="#">Malcolm</a> , <a href="#">Matt</a> , <a href="#">Matt</a> , <a href="#">MC93</a> , <a href="#">Michael Moriarty</a> , <a href="#">mnoronha</a> , <a href="#">Muntasir</a> , <a href="#">Nishchay</a> , <a href="#">Ortomala Lokni</a> , <a href="#">Persijn</a> , <a href="#">Prateek</a> , <a href="#">Pyloid</a> , <a href="#">Ryan Hilbert</a> , <a href="#">Shannon Young</a> , <a href="#">sideshowbarker</a> , <a href="#">stark</a> , <a href="#">Stelian Matei</a> , <a href="#">Sunny R Gupta</a> , <a href="#">the12</a> , <a href="#">tmg</a> , <a href="#">unor</a> , <a href="#">user3130333</a> , <a href="#">Valor Naram</a> , <a href="#">Willi</a> , <a href="#">Wolfgang</a> , <a href="#">Zaz</a> , <a href="#">zygimantus</a> , <a href="#">Zze</a>
2	Anclas e hipervínculos	<a href="#">Al.G.</a> , <a href="#">animuson</a> , <a href="#">Anselm Urban</a> , <a href="#">Anthony Pham</a> , <a href="#">ban17</a> , <a href="#">DawnPaladin</a> , <a href="#">Emil</a> , <a href="#">FlyingPiMonster</a> , <a href="#">insertusernamehere</a> , <a href="#">J F</a> , <a href="#">JHS</a> , <a href="#">joe_young</a> , <a href="#">Jojodmo</a> , <a href="#">Jones Joseph</a> , <a href="#">Lambda Ninja</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Nathan Tuggy</a> , <a href="#">Pranav</a> , <a href="#">Prateek</a> , <a href="#">Raystafarian</a> , <a href="#">Robert Columbia</a> , <a href="#">Squidward</a> , <a href="#">Steyn van Esveld</a> , <a href="#">Thomas Gerot</a> , <a href="#">unor</a> , <a href="#">Wolfgang</a>
3	ARIA	<a href="#">Bhavya Singh</a> , <a href="#">Paul Sweatte</a> , <a href="#">Shannon Young</a> , <a href="#">Travis</a> , <a href="#">unor</a> , <a href="#">user30796</a>
4	Atributos de datos	<a href="#">animuson</a> , <a href="#">Community</a> , <a href="#">Faegy</a> , <a href="#">Infuzed guy</a> , <a href="#">James Donnelly</a> , <a href="#">Manish</a> , <a href="#">Nathan Tuggy</a> , <a href="#">Nhan</a> , <a href="#">Racil Hilan</a> , <a href="#">rajarshig</a> , <a href="#">swatchai</a> , <a href="#">unor</a> , <a href="#">Yasir T</a>
5	Atributos de evento HTML	<a href="#">Paresh Maghodiya</a>
6	Atributos globales	<a href="#">animuson</a> , <a href="#">Becca</a> , <a href="#">unor</a> , <a href="#">Zange-chan</a>
7	Barras de navegación	<a href="#">Community</a>
8	Clases e identificaciones	<a href="#">Angelos Chalaris</a> , <a href="#">animuson</a> , <a href="#">brandaemon</a> , <a href="#">Caleb Kleveter</a> , <a href="#">Community</a> , <a href="#">Duh-Wayne-101</a> , <a href="#">Emil</a> , <a href="#">Epodax</a> , <a href="#">Evan</a> ,

		GoatsWearHats, Ingrid Stevens, jhnance, JHS, John Slegers, lexith, Luca Putzu, Michael_B, Natalie, Nhan, Richard Hamilton, Simone Carletti, Thomas Gerot, Timothy, Tyler Zika, unor, Wolfgang, xims
9	Comentarios	Ani Menon, animuson, Ashwin Ramaswami, bdkopen, ChrisD, Epodax, JHS, jkdev, RamenChef, Robert Grant, Soaring Code, Squazz, Thomas Gerot, Ulrich Schwarz, Wolfgang
10	Controles del menú de selección	Ali Almoullim, amflare, animuson, GentlePurpleRain, Ilyas karim, Mosh Feu, Tot Zam
11	Cotizaciones de mercado	Content Solutions, mnoronha, unor
12	Doctypes	Akshay Anand, Al.G., Angelos Chalaris, Ani Menon, animuson, Chris, Content Solutions, heerfk, mnoronha, pinjasaur, Right leg, Sumner Evans, Thomas Gerot, tmg, unor
13	Elemento de etiqueta	Anthony Pham, fredden, Josiah Keller, m_callens, Roko C. Buljan
14	Elemento de progreso	animuson, Content Solutions, Richard Hamilton
15	Elemento de salida	J F, Stephen Leppik, zer00ne
16	Elemento div	animuson, Araknid, Black Mamba, Chris, Content Solutions, D M, Faust, feeela, Gal Ratzkin, JHS, MySpeed, S.L. Barth, SuperStormer, Thomas Gerot
17	Elementos de control de entrada	Abhishek Pandey, Al.G., Alohci, amflare, Amitay Stern, Angelos Chalaris, Ani Menon, animuson, bhansa, Bob, Charlie H, Christophe Strobbe, CN, Community, cone56, Daniel, DawnPaladin, Dipen Shah, Domenic, Druzion, Edvin Tenovimas, Epodax, Franck Dernoncourt, gabe3886, geeksal, H. Pauwelyn, Henrique Barcelos, Huy Nguyen, J F, John Slegers, Kashyap Jha, Lahiru Ashan, Lankymart, Magisch, Marvin, Matas Vaitkevicius, Matt, Maximillian Laumeister, Mike McCaughan, morewry, Mosh Feu, Nathan Arthur, Nil Llisterri, Nishchay, niyasc, NoobCoder, Optimiser, Ortomala Lokni, Pi Programs, Pimgd, Prateek, Prav, Praveen Kumar, Psaniko, QoP, RamenChef, Ranjit Singh, Richard Hamilton, Robert Columbia, Roko C. Buljan, SeinopSys, Sharavnan Kv, Shivangi Chaurasia, SJDS, Squazz, Stephen Leppik, Stewartside, Sunny R Gupta, sv3k, the12, think123, Thomas Gerot, Timon, tmg, Tot Zam, trungk18, Undo, vladdobra, zzzzBov
18	Elementos de los	feeela, Isak Combrinck, LisaMM, Shiva, Yossi Aharon

	medios	
19	Elementos de seccionamiento	<a href="#">Andrew Brooke</a> , <a href="#">Anil</a> , <a href="#">animuson</a> , <a href="#">Hanif Formoly</a> , <a href="#">nalply</a> , <a href="#">Shannon Young</a> , <a href="#">SuperBiasedMan</a> , <a href="#">SuperStormer</a> , <a href="#">Yossi Aharon</a>
20	Elementos vacíos	<a href="#">4444</a> , <a href="#">ChrisD</a> , <a href="#">Thomas Gerot</a> , <a href="#">unor</a>
21	Empotrar	<a href="#">Alexandre N.</a>
22	Encabezados	<a href="#">Ani Menon</a> , <a href="#">animuson</a> , <a href="#">dippas</a> , <a href="#">Evan</a> , <a href="#">Infuzed guy</a> , <a href="#">joe_young</a> , <a href="#">MervS</a> , <a href="#">Nathan Arthur</a> , <a href="#">Pseudonym Patel</a> , <a href="#">sasha</a> , <a href="#">Thomas Gerot</a> , <a href="#">unor</a> , <a href="#">V-Kopio</a>
23	Entidades de carácter	<a href="#">animuson</a> , <a href="#">MervS</a> , <a href="#">stack-learner</a>
24	Formas	<a href="#">Ali Almoullim</a> , <a href="#">Ani Menon</a> , <a href="#">animuson</a> , <a href="#">Aown Muhammad</a> , <a href="#">Chris Rutherford</a> , <a href="#">Cullub</a> , <a href="#">Gabriel Chi Hong Lee</a> , <a href="#">Greg T</a> , <a href="#">j08691</a> , <a href="#">Kimmmax</a> , <a href="#">Luca langella</a> , <a href="#">Niek Brouwer</a> , <a href="#">Thomas Gerot</a>
25	Formato de texto	<a href="#">animuson</a> , <a href="#">Ben Rhys-Lewis</a> , <a href="#">Emil</a> , <a href="#">gustavohenke</a> , <a href="#">J F</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Peter L.</a> , <a href="#">Raystafarian</a> , <a href="#">Stephen Leppik</a> , <a href="#">Thomas Gerot</a> , <a href="#">unor</a> , <a href="#">Wolfgang</a>
26	HTML 5 Caché	<a href="#">Farhad</a> , <a href="#">TricksfortheWeb</a> , <a href="#">Valor Naram</a>
27	Idiomas de contenido	<a href="#">animuson</a> , <a href="#">FelipeAls</a> , <a href="#">Gerold Broser</a> , <a href="#">Isak Combrinck</a> , <a href="#">Muntasir</a> , <a href="#">Shannon Young</a> , <a href="#">unor</a>
28	IFrames	<a href="#">Adjit</a> , <a href="#">Alexandre N.</a> , <a href="#">animuson</a> , <a href="#">ChrisD</a> , <a href="#">dorukayhan</a> , <a href="#">Duh-Wayne-101</a> , <a href="#">Emanuel Vintilă</a> , <a href="#">J F</a> , <a href="#">Ojen</a> , <a href="#">Wojciech Kazior</a>
29	Imágenes	<a href="#">Alex</a> , <a href="#">Alexandre N.</a> , <a href="#">andreaem</a> , <a href="#">animuson</a> , <a href="#">Boysenb3rry</a> , <a href="#">Caleb Kleveter</a> , <a href="#">Gabriel Chi Hong Lee</a> , <a href="#">Infuzed guy</a> , <a href="#">ivn</a> , <a href="#">Kake_Fisk</a> , <a href="#">Maximillian Laumeister</a> , <a href="#">Mohd Samir Khan</a> , <a href="#">Mr Lister</a> , <a href="#">Nhan</a> , <a href="#">Shivangi Chaurasia</a> , <a href="#">Stephen Leppik</a> , <a href="#">Wolfgang</a>
30	Incluir código JavaScript en HTML	<a href="#">Alexandre N.</a> , <a href="#">andreaem</a> , <a href="#">animuson</a> , <a href="#">Anselm Urban</a> , <a href="#">Charles</a> , <a href="#">Marjorie Pickard</a> , <a href="#">MervS</a> , <a href="#">Roko C. Buljan</a> , <a href="#">Sildoreth</a> , <a href="#">SuperStormer</a>
31	Liza	<a href="#">animuson</a> , <a href="#">BiscuitBaker</a> , <a href="#">Daniel Käfer</a> , <a href="#">Grace Note</a> , <a href="#">H. Pauwelyn</a> , <a href="#">Jon Ericson</a> , <a href="#">kcpike</a> , <a href="#">Marvin</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">platy11</a> , <a href="#">Prateek</a> , <a href="#">Pseudonym Patel</a> , <a href="#">Richard Hamilton</a> , <a href="#">Right leg</a> , <a href="#">Sayakiss</a> , <a href="#">Stewartside</a> , <a href="#">Thomas Gerot</a> , <a href="#">tmg</a> , <a href="#">Tom Johnson</a> , <a href="#">unor</a> , <a href="#">Zack</a>
32	Lona	<a href="#">cone56</a> , <a href="#">Richard Hamilton</a> , <a href="#">Roko C. Buljan</a> , <a href="#">tonethar</a> , <a href="#">Trevor Clarke</a> , <a href="#">user4040648</a>



33	Mapas de imagen	<a href="#">animuson</a> , <a href="#">Lambda Ninja</a> , <a href="#">RamenChef</a>
34	Marcando código informático	<a href="#">4444</a> , <a href="#">Naveen Gogineni</a> , <a href="#">Shannon Young</a> , <a href="#">SuperStormer</a> , <a href="#">Tot Zam</a> , <a href="#">unor</a>
35	Mesas	<a href="#">albert</a> , <a href="#">Alexandre N.</a> , <a href="#">animuson</a> , <a href="#">Cedric Zoppolo</a> , <a href="#">Eduardo Molteni</a> , <a href="#">Grant Palin</a> , <a href="#">J F</a> , <a href="#">j08691</a> , <a href="#">JHS</a> , <a href="#">joe_young</a> , <a href="#">Lambda Ninja</a> , <a href="#">Mottie</a> , <a href="#">Mr Lister</a> , <a href="#">Nijin22</a> , <a href="#">Prateek</a> , <a href="#">PrAtik Lochawala</a> , <a href="#">Praveen Kumar</a> , <a href="#">Sildoreth</a> , <a href="#">svarog</a> , <a href="#">Ted Goas</a> , <a href="#">tehciolo</a> , <a href="#">Thomas Landauer</a> , <a href="#">zer00ne</a>
36	Meta informacion	<a href="#">Abhishek Pandey</a> , <a href="#">Akshit Soota</a> , <a href="#">Alexander Wigmore</a> , <a href="#">Angelos Chalaris</a> , <a href="#">Ani Menon</a> , <a href="#">animuson</a> , <a href="#">Anselm Urban</a> , <a href="#">Bálint</a> , <a href="#">bdkopen</a> , <a href="#">Bookeater</a> , <a href="#">Boris</a> , <a href="#">coliff</a> , <a href="#">Domenic</a> , <a href="#">geek1011</a> , <a href="#">Habel Philip</a> , <a href="#">Hafidz Ilham Aji Permana</a> , <a href="#">Himanshu Vaghela</a> , <a href="#">insertusernamehere</a> , <a href="#">jhoanna</a> , <a href="#">JHS</a> , <a href="#">kelvinelove</a> , <a href="#">m_callens</a> , <a href="#">Matt S</a> , <a href="#">Michael Moriarty</a> , <a href="#">Mr. Alien</a> , <a href="#">Nishchay</a> , <a href="#">Ortomala Lokni</a> , <a href="#">Peter O.</a> , <a href="#">Safoor Safdar</a> , <a href="#">Senjuti Mahapatra</a> , <a href="#">Shannon Young</a> , <a href="#">Stas Christiansen</a> , <a href="#">Stephen Leppik</a> , <a href="#">Ted Goas</a> , <a href="#">Thomas Gerot</a> , <a href="#">timmyRS</a> , <a href="#">tmg</a> , <a href="#">unor</a> , <a href="#">VatsalSura</a> , <a href="#">xims</a>
37	Párrafos	<a href="#">Abrar Jahin</a> , <a href="#">Thomas Gerot</a> , <a href="#">Valor Naram</a>
38	Recursos de enlace	<a href="#">AA2992</a> , <a href="#">animuson</a> , <a href="#">Anselm Urban</a> , <a href="#">Aravind Suresh</a> , <a href="#">Callan Heard</a> , <a href="#">Chris Rutherford</a> , <a href="#">cone56</a> , <a href="#">DawnPaladin</a> , <a href="#">Domenic</a> , <a href="#">feeela</a> , <a href="#">Henrique Barcelos</a> , <a href="#">Infuzed guy</a> , <a href="#">JHS</a> , <a href="#">Lambda Ninja</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Nhan</a> , <a href="#">Thomas Gerot</a> , <a href="#">unor</a> , <a href="#">V4karian</a> , <a href="#">vladdobra</a>
39	SVG	<a href="#">andreas</a> , <a href="#">Black Mamba</a> , <a href="#">ChrisD</a> , <a href="#">HerrSerker</a> , <a href="#">Patrickdev</a> , <a href="#">Timothy Miller</a> , <a href="#">w5m</a>
40	Tabindex	<a href="#">Content Solutions</a> , <a href="#">Psaniko</a>
41	Usando HTML con CSS	<a href="#">animuson</a> , <a href="#">bdkopen</a> , <a href="#">Christian Ternus</a> , <a href="#">Community</a> , <a href="#">Euan Williams</a> , <a href="#">feeela</a> , <a href="#">Jones Joseph</a> , <a href="#">Michael Moriarty</a> , <a href="#">Thomas Gerot</a> , <a href="#">thousten</a>