

HTML5

Apuntes para Profesionales

Chapter 12: Linking Resources

Attribute	Details
charset	Specifies the character encoding of the linked document.
crossorigin	Specifies how the element handles cross origin requests.
href	Specifies the location of the linked document.
hreflang	Specifies the language of the text in the linked document.
media	Specifies on what device the linked document will be displayed, often used with labels on the device to optimize.
rel	Specifies the relationship between the current document and the current document.
rev	Specifies the relationship between the linked document and the current document.
size	Specifies the size of the linked resource. Only when rel="stylesheet".
target	Specifies where the linked document is to be loaded.
type	Specifies the media type of the linked document.
integrity	Specifies a base64 encoded hash (sha256, sha384, or sha512) of the linked resource to verify its legitimacy.

Section 12.1: JavaScript

Synchronous
`<script src="path/to/js"></script>`

Standard practice is to place `<script>` tags just before the closing `</body>` tag to allow your site's visuals to show up more quickly and discourage your JavaScript elements that haven't loaded yet.

Asynchronous
`<script src="path/to/js" async></script>`

Another alternative, when the JavaScript code being loaded is not necessary to asynchronously speed up the page load, using `async` the browser will load and, once it is fully downloaded, will interrupt the HTML parsing in order to execute the script.

Deferred
`<script src="path/to/js" defer></script>`

Deferred scripts are like `async` scripts, with the exception that the parsing will not be interrupted until the script is fully loaded. Deferred scripts are guaranteed to be loaded in the order of their appearance in the document.

JavaScript
`<script></script>`

The `<script>` element defines content to be displayed if the user has scripts disabled or if the browser does not support using scripts. The `<script>` tag can be placed in either the `<head>` or the `<body>`.

Chapter 17: Input Control Elements

Parameter	Details
class	Indicates the Class of the input.
id	Indicates the ID of the input.
type	Identifies the type of input control to display. Acceptable values are: <code>button</code> , <code>checkbox</code> , <code>color</code> , <code>date</code> , <code>datetime-local</code> , <code>email</code> , <code>file</code> , <code>hidden</code> , <code>image</code> , <code>password</code> , <code>radio</code> , <code>range</code> , <code>reset</code> , <code>search</code> , <code>submit</code> , <code>tel</code> , <code>text</code> , <code>time</code> , <code>url</code> , <code>week</code> . Defaults to <code>text</code> if not specified. If the value is invalid, or if the browser does not support the type specified.
name	Indicates the name of the input.
disabled	Boolean value that indicates the input should be disabled. Disabled controls cannot be edited, are not sent on form submissions, and cannot receive focus.
checked	When the value of the type attribute is <code>checkbox</code> or <code>radio</code> , the presence of this Boolean attribute indicates that the control is selected by default; otherwise it is ignored.
multiple	HTML5 indicates multiple files or values can be passed (Applies only to <code>file</code> and <code>email</code> type inputs).
placeholder	HTML5 A hint to the user of what can be entered in the control. The placeholder text must not contain carriage returns or line-feeds.
autocomplete	HTML5 Indicates whether the value of the control can be automatically completed by the browser.
readonly	Boolean value that indicates the input is not editable. Readonly controls are still sent on form submissions, but will not receive focus. HTML5: This attribute is ignored when the value of type attribute is either <code>set</code> to <code>hidden</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> or <code>submit</code> .
required	HTML5 Indicates a value must be present or the element must be checked in order for the form to be submitted.
alt	An alternative text for images. In case they are not displayed.
autofocus	The <code><input></code> element should get the focus when page loads.
value	Specifies the value of <code><input></code> element.
step	The <code>step</code> attribute specifies the legal number intervals. It works with the following input types: <code>number</code> , <code>range</code> , <code>date</code> , <code>datetime-local</code> , <code>month</code> , <code>time</code> and <code>week</code> .

A key component of interactive web systems, input tags are HTML elements designed to take a specific form of input from users. Different types of input elements can regulate the data entered to fit a specified format and provide security to password entry.

Section 17.1: Text

The most basic input type and the default input if no type is specified. This input type defines a single-line text field with line-breaks automatically removed from the input value. All other characters can be entered into this. `<input type="text">` elements are used within `<form>` elements to declare input controls that allow users to input data.

Syntax

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

or without specifying a type, using the default attributes:

```
<input>
```

The default width of a text field input is 20 characters. This can be changed by specifying a value for the `width` attribute like this:

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

HTML5 Notes for Professionals

Chapter 20: Sectioning Elements

Section 20.1: Nav Element

The `<nav>` element is primarily intended to be used for sections that contain main navigation blocks for the website, this can include links to other parts of the web page (e.g. anchors for a table of contents) or other pages entirely.

Inline items

The following will display an inline set of hyperlinks:

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

Use list items when needed

If the content represents a list of items, use a list item to show this and enhance the user experience.

Note the `role="navigation"`, more on this below.

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

Avoid unnecessary usage

`<footer>` elements only have a list of links to other parts of the site (FAQ, T&C, etc.). The footer element alone is sufficient in this case, you don't need to further wrap your links with a `<nav>` element in the `<footer>`.

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

`<!-- The footer alone is sufficient -->`

```
<footer>  
  <a href="#">...</a>  
</footer>
```

Notes:

`<main>` `<section>` descendants are not allowed within a `<nav>`.

Adding a `role="navigation" data-bbox="606 618 863 657" style="border: 1px solid black; padding: 5px;">data-bbox="606 618 863 657"` to the `<nav>` element is advised to aid user agents that don't support HTML5 and also to provide more context for those that do.

HTML5 Notes for Professionals

Traducido por:

rortegag

100+ páginas

de consejos y trucos profesionales

Contenidos

Acerca de	1
Capítulo 1: Introducción a HTML.....	2
Sección 1.1: Hola Mundo	2
Capítulo 2: Doctypes.....	4
Sección 2.1: Añadir el Doctype	4
Sección 2.2: Doctype HTML 5.....	4
Capítulo 3: Encabezamientos	5
Sección 3.1: Uso de encabezamientos.....	5
Capítulo 4: Párrafos.....	6
Sección 4.1: Párrafos HTML.....	6
Capítulo 5: Formato del texto.....	7
Sección 5.1: Resaltado	7
Sección 5.2: Negrita, cursiva y subrayado	7
Sección 5.3: Abreviatura	8
Sección 5.4: Inserción, supresión o eliminación.....	8
Sección 5.5: Superíndices y subíndices	8
Capítulo 6: Anclajes e hipervínculos	9
Sección 6.1: Enlace a otro sitio.....	9
Sección 6.2: Enlace a un anclaje	10
Sección 6.3: Enlace a una página del mismo sitio	10
Sección 6.4: Enlace que marca un número.....	10
Sección 6.5: Abrir enlace en una nueva pestaña/ventana.....	11
Sección 6.6: Enlace que ejecuta JavaScript.....	11
Sección 6.7: Enlace que ejecuta el cliente de correo electrónico	12
Capítulo 7: Listas.....	13
Sección 7.1: Lista ordenada.....	13
Sección 7.2: Lista desordenada.....	14
Sección 7.3: Listas anidadas	15
Sección 7.4: Lista descriptiva	15
Capítulo 8: Tablas.....	17
Sección 8.1: Tabla simple.....	17
Sección 8.2: Extender columnas o filas.....	17
Sección 8.3: Grupos de columnas.....	18
Sección 8.4: Tabla con thead, tbody, tfoot y caption	19
Sección 8.5: Heading scope.....	19
Capítulo 9: Comentarios	21
Sección 9.1: Crear comentarios.....	21
Sección 9.2: Comentario de espacios en blanco entre elementos en línea	21

Capítulo 10: Clases e identificadores	23
Sección 10.1: Dar una clase a un elemento	23
Sección 10.2: Dar un identificador a un elemento	24
Sección 10.3: Valores aceptables	24
Sección 10.4: Problemas relacionados con los identificadores duplicados	26
Capítulo 11: Atributos de datos	27
Sección 11.1: Compatibilidad con navegadores antiguos	27
Sección 11.2: Uso de atributos de datos	27
Capítulo 12: Enlazar recursos	28
Sección 12.1: JavaScript	28
Sección 12.2: Hoja de estilos CSS externa	29
Sección 12.3: Favicon	29
Sección 12.4: CSS alternativo	29
Sección 12.5: Sugerencia sobre recursos: dns-prefetch, prefetch, prerender	29
Sección 12.6: Atributo 'media' del enlace	30
Sección 12.7: Anterior y Siguiente	30
Sección 12.8: Web Feed	30
Capítulo 13: Incluir código JavaScript en HTML	31
Sección 13.1: Manejo de JavaScript desactivado	31
Sección 13.2: Enlace a un archivo JavaScript externo	31
Sección 13.3: Inclusión directa de código JavaScript	31
Sección 13.4: Inclusión de un archivo JavaScript que se ejecuta de forma asíncrona	31
Capítulo 14: Uso de HTML con CSS	32
Sección 14.1: Uso de hojas de estilo externas	32
Sección 14.2: Hoja de estilo interna	32
Sección 14.3: Estilo en línea	32
Sección 14.4: Hojas de estilo múltiples	33
Capítulo 15: Imágenes	34
Sección 15.1: Creación de una imagen	34
Sección 15.2: Elección del texto alternativo	34
Sección 15.3: Imagen responsiva utilizando el atributo srcset	35
Sección 15.4: Imagen responsiva utilizando el elemento Picture	36
Capítulo 16: Mapas de imagen	37
Sección 16.1: Introducción a los mapas de imágenes	37
Capítulo 17: Elementos de control de entrada	39
Sección 17.1: Texto	39
Sección 17.2: Checkbox y botones de Radio	40
Sección 17.3: Validación de entrada	42
Sección 17.4: Color	43
Sección 17.5: Password	44

Sección 17.6: File.....	44
Sección 17.7: Button.....	45
Sección 17.8: Submit.....	46
Sección 17.9: Reset.....	46
Sección 17.10: Hidden.....	46
Sección 17.11: Tel.....	46
Sección 17.12: Email.....	47
Sección 17.13: Number.....	47
Sección 17.14: Range.....	47
Sección 17.15: Search.....	47
Sección 17.16: Image.....	47
Sección 17.17: Week.....	48
Sección 17.18: Url.....	48
Sección 17.19: DateTime-Local.....	48
Sección 17.20: Month.....	48
Sección 17.21: Time.....	48
Sección 17.22: DateTime (Global).....	49
Sección 17.23: Date.....	49
Capítulo 18: Formularios.....	50
Sección 18.1: Envíos.....	50
Sección 18.2: Atributo target en la etiqueta Form.....	50
Sección 18.3: Subir archivos.....	51
Sección 18.4: Agrupar algunos campos de entrada.....	51
Capítulo 19: Elemento Div.....	53
Sección 19.1: Uso básico.....	53
Sección 19.2: Anidación.....	53
Capítulo 20: Seccionar elementos.....	55
Sección 20.1: Elemento Nav.....	55
Sección 20.2: Elemento Article.....	56
Sección 20.3: Elemento Main.....	57
Sección 20.4: Elemento Header.....	58
Sección 20.5: Elemento Footer.....	58
Sección 20.6: Elemento Section.....	58
Capítulo 21: Barras de navegación.....	60
Sección 21.1: Barra de navegación básica.....	60
Sección 21.2: Barra de navegación HTML5.....	60
Capítulo 22: Elemento Label.....	61
Sección 22.1: Acerca del elemento Label.....	61
Sección 22.2: Uso básico.....	61
Capítulo 23. Elemento Output.....	62
Sección 23.1: Elemento Output mediante atributos for y form.....	62

Sección 23.2: Elemento Output con atributos.....	62
Capítulo 24: Elementos vacíos	63
Sección 24.1: Elementos vacíos.....	63
Capítulo 25: Elementos multimedia.....	64
Sección 25.1: Audio.....	64
Sección 25.2: Vídeo.....	64
Sección 25.3: Uso de los elementos ` <code><video></code> ` y ` <code><audio></code> ` para mostrar contenidos de audio/vídeo	64
Sección 25.4: Cabecera o fondo de vídeo	65
Capítulo 26: Elemento Progress	66
Sección 26.1: Progress.....	66
Sección 26.2: Cambiar el color de una barra de progreso.....	66
Sección 26.3: HTML Fallback.....	67
Capítulo 27. Controles del menú de selección	68
Sección 27.1: Seleccionar menú.....	68
Sección 27.2: Opciones.....	68
Sección 27.3: Grupos de opciones.....	69
Sección 27.4: Lista de datos.....	69
Capítulo 28: Embed	71
Sección 28.1: Uso básico	71
Sección 28.2: Definición del tipo MIME	71
Capítulo 29: IFrames	72
Sección 29.1: Conceptos básicos de un marco en línea	72
Sección 29.2: Entorno aislado.....	72
Sección 29.3: Ajustar el tamaño del marco	72
Sección 29.4: Uso del atributo "srcdoc"	73
Sección 29.5: Uso de anclajes con IFrames	73
Capítulo 30: Lenguajes de contenido	74
Sección 30.1: Lenguaje del documento base.....	74
Sección 30.2: Lenguaje del elemento	74
Sección 30.3: Elementos con varios idiomas.....	74
Sección 30.4: URL regionales	74
Sección 30.5: Manejo de atributos con lenguajes diferentes.....	74
Capítulo 31: SVG.....	75
Sección 31.1: SVG en línea	75
Sección 31.2: Incrustación de archivos SVG externos en HTML.....	75
Sección 31.3: Incrustación de SVG mediante CSS.....	76
Capítulo 32: Canvas.....	77
Sección 32.1: Ejemplo básico	77
Sección 32.2: Dibujar dos rectángulos en un <code><canvas></code>	78
Capítulo 33: Metadatos	79

Sección 33.1: Información de la página.....	79
Sección 33.2: Codificación de caracteres.....	79
Sección 33.3: Robots.....	80
Sección 33.4: Redes sociales.....	80
Sección 33.5: Control de la disposición móvil.....	81
Sección 33.6: Actualización automática.....	82
Sección 33.7: Reconocimiento de números de teléfono.....	82
Sección 33.8: Redirección automática.....	82
Sección 33.9: Web App.....	83
Capítulo 34: Marcado del código informático	84
Section 34.1: Bloque con <pre> y <code>.....	84
Section 34.2: En línea con <code>.....	84
Capítulo 35: Marcar citas.....	85
Section 35.1: En línea con <q>.....	85
Sección 35.2: Bloque con <blockquote>.....	85
Capítulo 36: Índice de tabulación.....	87
Sección 36.1: Añadir un elemento al orden de tabulación.....	87
Sección 36.2: Eliminar un elemento del orden de tabulación.....	87
Sección 36.3: Definir un orden de tabulación personalizado (no recomendado).....	87
Capítulo 37: Atributos globales	88
Sección 37.1: Atributo contenteditable.....	88
Capítulo 38: Caché de HTML 5.....	89
Sección 38.1: Ejemplo básico de caché HTML5.....	89
Capítulo 39: Atributos de eventos HTML	90
Sección 39.1: Eventos de formulario HTML.....	90
Sección 39.2: Eventos de teclado.....	90
Capítulo 40: Entidades de carácter	91
Sección 40.1: Entidades de carácter en HTML.....	91
Sección 40.2: Caracteres especiales comunes.....	91
Capítulo 41: ARIA	92
Sección 41.1: role="presentation".....	92
Sección 41.2: role="alert".....	92
Sección 41.3: role="alertdialog".....	92
Sección 41.4: role="application".....	92
Sección 41.5: role="article".....	92
Sección 41.6: role="banner".....	93
Sección 41.7: role="button".....	93
Sección 41.8: role="cell".....	93
Sección 41.9: role="checkbox".....	93
Sección 41.10: role="columnheader".....	93

Sección 41.11: role="combobox"	94
Sección 41.12: role="complementary"	94
Sección 41.13: role="contentinfo"	94
Sección 41.14: role="definition"	94
Sección 41.15: role="dialog"	94
Sección 41.16: role="directory"	95
Sección 41.17: role="document"	95
Sección 41.18: role="form"	95
Sección 41.19: role="grid"	96
Sección 41.20: role="gridcell"	96
Sección 41.21: role="group"	96
Sección 41.22: role="heading"	96
Sección 41.23: role="img"	96
Sección 41.24: role="link"	97
Sección 41.25: role="list"	97
Sección 41.26: role="listbox"	97
Sección 41.27: role="listitem"	97
Sección 41.28: role="log"	97
Sección 41.29: role="main"	98
Sección 41.30: role="marquee"	98
Sección 41.31: role="math"	98
Sección 41.32: role="menu"	98
Sección 41.33: role="menubar"	98
Sección 41.34: role="menuitem"	98
Sección 41.35: role="menuitemcheckbox"	99
Sección 41.36: role="menuitemradio"	99
Sección 41.37: role="navigation"	99
Sección 41.38: role="note"	99
Sección 41.39: role="option"	99
Sección 41.40: role="progressbar"	99
Sección 41.41: role="radio"	100
Sección 41.42: role="region"	100
Sección 41.43: role="radiogroup"	100
Sección 41.44: role="row"	100
Sección 41.45: role="rowgroup"	100
Sección 41.46: role="rowheader"	101
Sección 41.47: role="scrollbar"	101
Sección 41.48: role="search"	101
Sección 41.49: role="searchbox"	101
Sección 41.50: role="separator"	102
Sección 41.51: role="slider"	102

Sección 41.52: role="spinbutton"	102
Sección 41.53: role="status"	102
Sección 41.54: role="switch"	102
Sección 41.55: role="tab"	102
Sección 41.56: role="table"	103
Sección 41.57: role="tablist"	103
Sección 41.58: role="tabpanel"	103
Sección 41.59: role="textbox"	103
Sección 41.60: role="timer"	103
Sección 41.61: role="toolbar"	104
Sección 41.62: role="tooltip"	104
Sección 41.63: role="tree"	104
Sección 41.64: role="treegrid"	104
Sección 41.65: role="treeitem"	105
Créditos	106

Acerca de

Este libro ha sido traducido por rortegag.com

Si desea descargar el libro original, puede descargarlo desde:

<https://books.goalkicker.com/HTML5Book/>

Si desea contribuir con una donación, hazlo desde:

<https://www.buymeacoffee.com/GoalKickerBooks>

Por favor, siéntase libre de compartir este PDF con cualquier persona de forma gratuita, la última versión de este libro se puede descargar desde:

<https://goalkicker.com/HTML5Book>

Este libro HTML5 Apuntes para Profesionales está compilado a partir de la [Documentación de Stack Overflow](#), el contenido está escrito por la hermosa gente de Stack Overflow. El contenido del texto está liberado bajo Creative Commons BY-SA, ver los créditos al final de este libro quién contribuyó a los distintos capítulos. Las imágenes pueden ser copyright de sus respectivos propietarios a menos que se especifique lo contrario.

Este es un libro no oficial gratuito creado con fines educativos y no está afiliado con los grupo(s) o empresa(s) oficiales de HTML5 ni Stack Overflow. Todas las marcas comerciales y marcas registradas son propiedad de sus respectivos propietarios de la empresa.

No se garantiza que la información presentada en este libro sea correcta ni exacta. Utilícelo bajo su propia responsabilidad.

Envíe sus comentarios y correcciones a web@petercv.com

Capítulo 1: Introducción a HTML

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

Sección 1.1: Hola Mundo

Introducción

HTML (**Hypertext Markup Language**) utiliza un sistema de marcado compuesto por elementos que representan contenidos específicos. Markup significa que con HTML se declara lo que se presenta al espectador, no cómo se presenta. Las representaciones visuales se definen mediante [Cascading Style Sheets \(CSS\)](#) y son realizadas por los navegadores. [Hay elementos aún existentes que lo permiten](#), como p. ej. [font](#), "son totalmente obsoletos y no deben ser utilizados por los autores".[1]

A veces se dice que HTML es un lenguaje de programación, pero no tiene lógica, por lo que es un **lenguaje de marcado**. Las etiquetas HTML proporcionan significado semántico y legibilidad mecánica al contenido de la página.

Un elemento suele constar de una etiqueta de apertura (`<nombre_elemento>`), una etiqueta de cierre (`</nombre_elemento>`), que contienen el nombre del elemento rodeado de corchetes angulares, y el contenido entre ellos:

```
<nombre_elemento>...contenido...</nombre_elemento>
```

Hay algunos elementos HTML que no tienen etiqueta de cierre ni contenido. Son los denominados elementos vacíos. Los elementos vacíos incluyen ``, `<meta>`, `<link>` e `<input>`.

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

Una página HTML puede constar de cientos de elementos que un navegador lee, interpreta y convierte en contenido legible o audible en la pantalla.

Para este documento es importante señalar la diferencia entre elementos y etiquetas:

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

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

Información sobre los elementos

Analicemos una etiqueta...

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

Los elementos suelen tener una etiqueta de apertura y otra de cierre. La etiqueta de apertura contiene el nombre del elemento en ángulo corchetes (`<p>`). La etiqueta de cierre es idéntica a la de apertura con la adición de una barra oblicua (/) entre el corchete de apertura y el nombre del elemento (`</p>`).

El contenido puede ir entonces entre estas dos etiquetas: `<p>Este es un simple párrafo. </p>`.

Crear una página simple

El siguiente ejemplo HTML crea una sencilla página web ["Hola Mundo"](#).

Los archivos HTML pueden crearse con cualquier [editor de texto](#). Los archivos deben guardarse con una extensión .html o .htm[2] para que se reconozcan como archivos HTML.

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

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <title>Hola</title>
  </head>
  <body>
    <h1>Hola Mundo</h1>
    <p>Este es un simple párrafo. </p>
  </body>
</html>
```

Desglose de esta página simple

Estas son las etiquetas utilizadas en el ejemplo:

Etiqueta	Significado
<code><!DOCTYPE></code>	Define la versión de HTML utilizada en el documento. En este caso es HTML5. Consulte el tema doctype para obtener más información.
<code><html></code>	Abre la página. Después de la etiqueta de cierre (<code></html></code>) no debe aparecer ninguna marca. El atributo <code>lang</code> declara el idioma principal de la página utilizando los códigos de idioma ISO (en para inglés, <code>es</code> para español).
<code><head></code>	Para más información, consulte el tema Idioma del contenido . Abre la sección <code>head</code> , que no aparece en la ventana principal del navegador, pero contiene principalmente información <i>sobre</i> el documento HTML, denominada <i>metadatos</i> . También puede contener importaciones de hojas de estilo y scripts externos. La etiqueta de cierre es <code></head></code> .
<code><meta></code>	Proporciona al navegador algunos metadatos sobre el documento. El atributo <code>charset</code> declara la codificación de caracteres . Los documentos HTML modernos deberían utilizar siempre UTF-8 , aunque no sea un requisito. En HTML, la etiqueta <code><meta></code> no requiere una etiqueta de cierre. Consulta el tema Meta para obtener más información.
<code><title></code>	El título de la página. El texto escrito entre esta etiqueta de apertura y la de cierre (<code></title></code>) se mostrará en la pestaña de la página o en la barra de título del navegador.
<code><body></code>	Abre la parte del documento que se muestra a los usuarios, es decir, todo el contenido visible o audible de una página. No debe añadirse contenido a contenido después de la etiqueta de cierre <code></body></code> .
<code><h1></code>	Un encabezamiento de nivel 1 para la página. Consulte los encabezados para obtener más información.
<code><p></code>	Representa un párrafo común del texto.

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

Capítulo 2: Doctypes

Los doctypes - abreviatura de "tipo de documento" - ayudan 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 deben figurar en la parte superior del documento. Este tema, explica la estructura y declaración de varios doctypes en HTML.

Sección 2.1: Añadir el Doctype

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

Version ≥ 5

Véase Doctype HTML 5 para más detalles sobre el Doctype HTML 5.

```
<!DOCTYPE html>
```

Sección 2.2: Doctype HTML 5

HTML5 no se basa en SGML (Standard Generalized Markup Language), por lo que no requiere una referencia a una DTD (Document Type Definition).

Declaración Doctype HTML 5:

```
<!DOCTYPE html>
```

Insensibilidad a las mayúsculas y minúsculas

Según la [Especificación W3.org HTML 5 DOCTYPE](#):

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

1. Una cadena de caracteres que coincide con la cadena de caracteres `"<!DOCTYPE>"` **sin distinguir mayúsculas de minúsculas** en ASCII.

por lo que también son válidos los siguientes DOCTYPE:

```
<!doctype html>
```

```
<!dOCTyPe html>
```

```
<!DocTYpe html>
```

Este artículo de StackOverflow trata el tema en profundidad: [¿Doctype mayúsculas o minúsculas?](#)

Capítulo 3: Encabezamientos

HTML no sólo ofrece etiquetas de párrafo, sino también seis etiquetas de encabezado independientes para indicar títulos de distintos tamaños y grosores. Enumerados del epígrafe 1 al 6, el epígrafe 1 tiene el texto más grande y grueso, mientras que el epígrafe 6 es el más pequeño y fino, hasta el nivel de párrafo. Este tema detalla el uso correcto de estas etiquetas.

Sección 3.1: Uso de encabezamientos

Los encabezados pueden utilizarse para describir el tema al que preceden y se definen con las etiquetas `<h1>` a `<h6>`. Los encabezados admiten todos los atributos globales.

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

Definir un encabezado:

```
<h1>Encabezado 1</h1>
```

```
<h2>Encabezado 2</h2>
```

```
<h3>Encabezado 3</h3>
```

```
<h4>Encabezado 4</h4>
```

```
<h5>Encabezado 5</h5>
```

```
<h6>Encabezado 6</h6>
```

Una estructura correcta es importante

Los motores de búsqueda y otros agentes de usuario suelen indexar el contenido de las páginas basándose en los elementos de encabezamiento, por ejemplo, para crear una tabla de contenidos, por lo que es importante utilizar la estructura correcta para los encabezados.

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

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

```
<h1>Título principal</h1>
<p>Introducción</p>
  <h2>Motivos</h2>
    <h3>Motivo 1</h3>
    <p>Párrafo</p>
    <h3>Motivo 2</h3>
    <p>Párrafo</p>
  <h2>En conclusión</h2>
  <p>Párrafo</p>
```

Capítulo 4: Párrafos

Etiqueta	Descripción
<code><p></code>	Define un párrafo
<code>
</code>	Inserta un salto de línea
<code><pre></code>	Define un texto preformateado

Los párrafos son el elemento HTML más básico. Este tema explica y demuestra el uso del elemento párrafo en HTML.

Sección 4.1: Párrafos HTML

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

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

Visualización

No puede estar seguro de cómo se mostrará el HTML. Las pantallas grandes o pequeñas y las ventanas redimensionadas darán resultados diferentes. Con HTML, no se puede cambiar el resultado añadiendo espacios o líneas adicionales en el código HTML.

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

```
<p>Este    es otro    párrafo, los espacios adicionales serán eliminados por los navegadores</p>
```

Capítulo 5: Formato del texto

Aunque 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 tachado.

Sección 5.1: Resaltado

El elemento `<mark>` es nuevo en HTML5 y se utiliza para marcar o resaltar texto en un documento "debido a su relevancia en otro contexto".¹

El ejemplo más común sería el de los resultados de una búsqueda en la que el usuario ha introducido una consulta y los resultados se muestran resaltando la consulta deseada.

```
<p>Aquí hay algo de contenido de un artículo que contiene la <mark>consulta buscada</mark> que estamos buscando. Resaltando el texto será más fácil para el usuario encontrar lo que está buscando.</p>
```

Salida:

Aquí hay algo de contenido de un artículo que contiene la **consulta buscada** que estamos buscando. Resaltando el texto será más fácil para el usuario encontrar lo que está buscando.

Un formato estándar común es el texto negro sobre fondo amarillo, pero esto puede cambiarse con CSS.

Sección 5.2: Negrita, cursiva y subrayado

Texto en negrita

Para poner un texto en negrita, utilice las etiquetas `` o ``:

```
<strong>Texto en negrita aquí.</strong>
```

o

```
<b>Texto en negrita aquí.</b>
```

¿Cuál es la diferencia? La semántica. `` se utiliza para indicar que el texto es fundamental o semánticamente *importante* para el texto que lo rodea, mientras que `` no indica tal importancia y simplemente representa texto que debe ir en negrita.

Si utilizara ``, un programa de conversión de texto a voz no pronunciaría la(s) palabra(s) de forma diferente a las demás palabras que la(s) rodean, simplemente llama la atención sobre ellas sin añadirles ninguna importancia adicional. Sin embargo, al utilizar ``, el mismo programa querrá pronunciar esa(s) palabra(s) 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, utilice las etiquetas `` o `<i>`:

```
<em>Texto en cursiva aquí.</em>
```

o

```
<i>Texto en cursiva aquí.</i>
```

¿Cuál es la diferencia? La semántica. `` se utiliza para indicar que el texto debe tener un énfasis adicional que debe acentuarse, mientras que `<i>` simplemente representa texto que debe destacarse del texto normal que lo rodea. Por ejemplo, si se quiere destacar la acción dentro de una frase, se puede hacer enfatizándola en cursiva mediante ``: "¿Podrías *enviar* ya la edición?".

Pero si estuvieras identificando un libro o periódico que normalmente pondrías en cursiva estilísticamente, simplemente usarías `<i>`: "Me obligaron a leer *Romeo y Julieta* en el instituto."

Texto subrayado

Aunque el elemento `<u>` quedó obsoleto en HTML 4, se reintrodujo con un significado semántico alternativo en HTML 5: representar una anotación no articulada, no textual. Puede utilizarse para indicar texto mal escrito en la página o para marcar un nombre propio chino.

```
<p>Este párrafo contiene <u>texto mal escrito.</u></p>
```

Sección 5.3: Abreviatura

Para marcar una expresión como abreviatura, utilice la etiqueta `<abbr>`:

```
<p>Me gusta escribir <abbr title="Hypertext Markup Language">HTML</abbr>!</p>
```

Si está presente, el atributo `title` se utiliza para presentar la descripción completa de dicha abreviatura.

Sección 5.4: Inserción, supresión o eliminación

Para marcar un texto como insertado, utiliza la etiqueta `<ins>`:

```
<ins>Texto insertado. </ins>
```

Para marcar un texto como eliminado, utiliza la etiqueta ``:

```
<del>Texto eliminado. </del>
```

Para tachar texto, utiliza la etiqueta `<s>`:

```
<s>Texto tachado. </s>
```

Sección 5.5: Superíndices y subíndices

Para desplazar el texto hacia arriba o hacia abajo puede utilizar las etiquetas `<sup>` y `<sub>`.

Para crear un superíndice:

```
<sup>superíndice aquí</sup>
```

Para crear un subíndice:

```
<sub>subíndice aquí</sub>
```

Capítulo 6: Anclajes e hipervínculos

Parámetro	Detalles
<code>href</code>	Especifica la dirección de destino. Puede ser una URL absoluta o relativa, o el <code>name</code> de un anclaje. Una URL absoluta es la URL completa de un sitio web como http://example.com/ . 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 suelen devolver el documento "index.html" dentro de ese directorio.
<code>hreflang</code>	Especifica el idioma del recurso enlazado por el atributo <code>href</code> (que debe estar presente con este atributo). Utilice los valores de idioma de BCP 47 para HTML5 y RFC 1766 para HTML 4.
<code>rel</code>	Especifica la relación entre el documento actual y el documento vinculado. Para HTML5, los valores deben estar definidos en la especificación o registrados en la wiki Microformats .
<code>target</code>	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> (obsoleto). No se recomienda forzar este comportamiento, ya que viola el control del usuario sobre un sitio web.
<code>title</code>	Especifica información adicional sobre un enlace. La información suele mostrarse como texto de ayuda cuando el cursor se desplaza sobre el enlace. Este atributo no se limita a los enlaces, puede utilizarse en casi todas las etiquetas HTML.
<code>download</code>	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 hay restricciones sobre los valores permitidos, y el detectará automáticamente la extensión de archivo correcta y la añadirá al archivo (<code>.img</code> , <code>.pdf</code> , etc.). Si el valor se omite, se utiliza el nombre de archivo original.

Las etiquetas de anclaje se utilizan habitualmente para enlazar páginas web independientes, pero también pueden emplearse para enlazar entre distintos lugares de un mismo documento, a menudo dentro del índice o incluso para lanzar aplicaciones externas. Este tema explica la implementación y aplicación de las etiquetas de anclaje HTML en varios roles.

Sección 6.1: Enlace a otro sitio

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

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

Crea un hiperenlace a la URL `http://example.com/` especificada por el atributo `href` (referencia de hipertexto), con el texto de anclaje "Enlace a ejemplo.com". Tendría un aspecto similar al siguiente:

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

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

```
<a href="http://example.com/" rel="external">ejemplo de sitio </a>
```

Puede enlazar con un sitio que utilice un protocolo distinto de HTTP. Por ejemplo, para enlazar a un sitio FTP, puedes hacerlo.

```
<a href="ftp://example.com/">Esto podría ser un enlace a un sitio FTP </a>
```

En este caso, la diferencia es que esta etiqueta de anclaje está solicitando que el navegador del usuario se conecte a `example.com` utilizando 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](#)

Sección 6.2: Enlace a un anclaje

Los anclajes pueden utilizarse para saltar a etiquetas específicas de una página HTML. La etiqueta `<a>` puede apuntar a cualquier elemento que tenga un atributo `id`. Para saber más sobre los IDs, visite la documentación sobre Clases e IDs. Los anclajes se utilizan sobre todo para saltar a una subsección de una página y se utilizan junto con las etiquetas de encabezado.

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

```
<h2>Primer tema</h2>
<p>Contenido sobre el primer tema</p>
<h2>Segundo tema</h2>
<p>Contenido sobre el segundo tema</p>
```

Cuando tenga varias secciones, puede crear un índice en la parte superior de la página con enlaces rápidos (o marcadores) a secciones específicas.

Si asignas un atributo `id` a tus temas, podrás enlazar con ellos.

```
<h2 id="Tema1">Primer tema</h2>
<p>Contenido sobre el primer tema</p>
<h2 id="Tema2">Segundo tema</h2>
<p>Contenido sobre el segundo tema</p>
```

Ahora puede utilizar el anclaje en su índice:

```
<h1>Índice</h1>
<a href="#Tema1">Pulse para saltar al Primer Tema</a>
<a href="#Tema2">Pulse para saltar al Segundo Tema</a>
```

Estos anclajes también están vinculados a la página web en la que se encuentran (`page1.html`). Así que puede enlazar a través del sitio de una página a otra haciendo referencia a la página y al nombre del ancla.

Recuerda que siempre puedes `mirar atrás en el Primer Tema` para información complementaria.

Sección 6.3: Enlace a una página del mismo sitio

Puede utilizar una [ruta relativa](#) para enlazar a páginas del mismo sitio web.

```
<a href="/example">Texto Aquí</a>
```

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

Si este enlace estuviera en <http://example.com/>, los dos enlaces siguientes llevarían al usuario al mismo lugar.

```
<a href="/page">Texto Aquí</a>
<a href="http://example.com/page">Texto Aquí</a>
```

Ambos irían al archivo `page` en el directorio raíz de `example.com`.

Sección 6.4: Enlace que marca un número

Si el valor del atributo `href` empieza por `tel:`, tu dispositivo marcará el número cuando lo pulses. Funciona en dispositivos móviles o en ordenadores/tabletas con programas -como Skype o FaceTime- que pueden hacer llamadas telefónicas.

```
<a href="tel:11234567890">Llámanos</a>
```

La mayoría de los dispositivos y programas piden al usuario que confirme el número que va a marcar.

Sección 6.5: Abrir enlace en una nueva pestaña/ventana

```
<a href="example.com" target="_blank">Texto Aquí</a>
```

El atributo `target` especifica dónde abrir el enlace. Si se establece en `_blank`, se indica al navegador que lo abra en una nueva pestaña o ventana (según las preferencias del usuario).

¡AVISO DE VULNERABILIDAD DE SEGURIDAD!

El uso de `target="_blank"` da al sitio de apertura acceso parcial al objeto `window.opener` a través de JavaScript, lo que permite a esa página acceder y cambiar el `window.opener.location` de su página y redirigir potencialmente a los usuarios a sitios de malware o phishing.

Cuando utilice esta opción para páginas que no controle, añada `rel="noopener"` a su enlace para evitar que el objeto `window.opener` con la solicitud.

Actualmente, Firefox no es compatible con `noopener`, por lo que tendrá que utilizar `rel="noopener noreferrer"` para obtener el máximo efecto.

Sección 6.6: Enlace que ejecuta JavaScript

Basta con utilizar el protocolo `javascript:` para ejecutar el texto como JavaScript en lugar de abrirlo como un enlace normal:

```
<a href="javascript:miFuncion();">Ejecutar código</a>
```

También puede conseguir lo mismo utilizando el atributo `onclick`:

```
<a href="#" onclick="miFuncion(); return false;">Ejecutar código</a>
```

El `return false;` es necesario para evitar que su página se desplace a la parte superior cuando se hace clic en el enlace a `#`. Asegúrese de incluir todo el código que desea ejecutar antes de ella, como volver detendrá la ejecución de código adicional.

También cabe destacar que puede incluir un signo de exclamación `!` después del hashtag para evitar que la página se desplace a la parte superior. Esto funciona porque cualquier slug inválido hará que el enlace no se desplace a *ninguna parte* de la página, porque no podría localizar el elemento al que hace referencia (un elemento con `id="!"`). También puede utilizar cualquier (como `#scrollsNowhere`) para conseguir el mismo efecto. En este caso, `return false;` no es necesario:

```
<a href="#!" onclick="miFuncion();">Ejecutar código</a>
```

¿Deberías utilizar algo de esto?

La respuesta es casi seguro que **no**. Ejecutar JavaScript en línea con el elemento de esta manera es bastante mala práctica. Considere la posibilidad de utilizar soluciones JavaScript puras que busquen el elemento en la página y, en su lugar, vinculen una función. Escuchar un evento.

Considere también si este elemento es realmente un *botón* en lugar de un *enlace*. Si es así, debe utilizar `<button>`.

Sección 6.7: Enlace que ejecuta el cliente de correo electrónico

Uso básico

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

```
<a href="mailto:ejemplo@ejemplo.com">Enviar email</a>
```

Esto pondrá la dirección de correo electrónico `ejemplo@ejemplo.com` como destinatario para el correo electrónico recién creado.

CC y CCO

También puede añadir direcciones para destinatarios `cc` o `bcc` utilizando la siguiente sintaxis:

```
<a href="mailto:ejemplo@ejemplo.com?cc=juan@ejemplo.com&bcc=juana@ejemplo.com">Enviar email</a>
```

Asunto y cuerpo del texto

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

```
<a href="mailto:ejemplo@ejemplo.com?subject=Ejemplo+subject&body=Mensaje+text">Enviar email</a>
```

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

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

Capítulo 7: Listas

HTML ofrece tres formas de especificar listas: listas ordenadas, listas desordenadas y listas descriptivas. Las listas ordenadas utilizan secuencias ordinales para indicar el orden de los elementos de la lista, las listas desordenadas utilizan un símbolo definido, como una viñeta, para enumerar los elementos sin un orden determinado, y las listas descriptivas utilizan sangrías para enumerar los elementos con sus hijos. Este tema explica la implementación y combinación de estas listas en el marcado HTML.

Sección 7.1: Lista ordenada

Se puede crear una lista ordenada con la etiqueta `` y cada elemento de la lista con la etiqueta `` como en el ejemplo siguiente:

```
<ol>
  <li>Item</li>
  <li>Otro item</li>
  <li>Otro item más</li>
</ol>
```

Esto producirá una lista numerada (que es el estilo por defecto):

1. Item
2. Otro item
3. Otro item más

Modificar manualmente los números

Hay un par de formas de jugar con los números que aparecen en los elementos de una lista ordenada. La primera consiste en fijar un número inicial, utilizando el atributo `start`. La lista comenzará en este número definido, y continuará incrementando de uno en uno como de costumbre.

```
<ol start="3">
  <li>Item</li>
  <li>Algún otro item</li>
  <li>Otro item más</li>
</ol>
```

Esto producirá una lista numerada (que es el estilo por defecto):

3. Item
4. Algún otro item
5. Otro item más

También puede establecer explícitamente un determinado elemento de la lista a un número específico. Los elementos de lista posteriores a uno con un valor especificado continuarán incrementándose en uno a partir del valor de ese elemento de lista, ignorando el punto en el que se encontraba la lista principal.

```
<li value="7"></li>
```

También cabe destacar que, al utilizar el atributo `value` directamente en un elemento de lista, puede anular el sistema de numeración existente de una lista ordenada reiniciando la numeración en un valor inferior. Por lo tanto, si la lista principal ya ha alcanzado el valor 7 y se encuentra con un elemento de la lista en el valor 4, ese elemento de la lista seguirá mostrándose como 4 y continuará contando desde ese punto de nuevo.

```
<ol start="5">
  <li>Item</li>
  <li>Algún otro item</li>
  <li value="4">Un item reestablecido</li>
  <li>Otro item</li>
  <li>Algún otro item</li>
</ol>
```

Así, el ejemplo anterior producirá una lista que sigue el patrón de numeración de 5, 6, 4, 5, 6 - comenzando de nuevo en un número inferior al anterior y duplicando el número 6 en la lista.

Nota: Los atributos `start` y `value` sólo aceptan números, incluso si la lista ordenada está configurada para mostrarse en números romanos o letras. romanos o letras.

Version ≥ 5

Puede invertir la numeración añadiendo `reversed` en su elemento `ol`:

```
<ol reversed>
  <li>Item</li>
  <li>Algún otro item</li>
  <li value="4">Un item reestablecido</li>
  <li>Otro item</li>
  <li>Algún otro item</li>
</ol>
```

La numeración inversa es útil si añades continuamente elementos a una lista, como nuevos episodios de podcasts o presentaciones, y quieres que los más recientes aparezcan primero.

Modificar el tipo de número

Puede cambiar fácilmente el tipo de número mostrado en el marcador de elementos de la 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 (minúsculas)	a,b,c,d
A	Ordenados alfabéticamente (mayúsculas)	A,B,C,D
i	Números romanos (minúsculas)	i,ii,iii,iv
I	Números romanos (mayúsculas)	I,II,III,IV

Utilizar `ol` para mostrar una lista de elementos, donde los elementos se han ordenado intencionadamente y el orden debe enfatizarse. Si cambiar el orden de los elementos NO hace que la lista sea incorrecta, usa ``.

Sección 7.2: Lista desordenada

Una lista desordenada se puede crear con la etiqueta `` y cada elemento de la lista se puede crear con la etiqueta `` como se muestra en el siguiente ejemplo:

```
<ul>
  <li>Item</li>
  <li>Otro item</li>
  <li>Otro item más</li>
</ul>
```

Esto producirá una lista con viñetas (que es el estilo por defecto):

- Item
- Otro item
- Otro item más

Utiliza `ul` para mostrar una lista de elementos, donde el orden de los elementos no es importante. Si al cambiar el orden de los elementos la lista resulta incorrecta, utiliza ``.

Sección 7.3: 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>
```

- item 1
- item 2
 - sub-item 2.1
 - sub-item 2.2
- item 3

La lista anidada debe ser hija del elemento `li`.

También puedes anidar distintos tipos de listas:

```
<ol>
  <li>¡Hola, lista!</li>
  <li>
    <ul>
      <li>¡Hola, lista anidada!</li>
    </ul>
  </li>
</ol>
```

Sección 7.4: Lista descriptiva

Se puede crear una lista de descripción (o *lista de definición*, como se llamaba antes de HTML5) con el elemento `dl`. Consiste en grupos nombre-valor, donde el nombre se da en el elemento `dt`, y el valor se da en el elemento `dd`.

```
<dl>
  <dt>nombre 1</dt>
  <dd>valor para 1</dd>
  <dt>nombre 2</dt>
  <dd>valor para 2</dd>
</dl>
```

Un grupo nombre-valor puede tener más de un nombre y/o más de un valor (que representan alternativas):

```
<dl>  
  <dt>nombre 1</dt>  
  <dt>nombre 2</dt>  
  <dd>valor para 1 y 2</dd>  
  <dt>nombre 3</dt>  
  <dd>valor para 3</dd>  
  <dd>valor para 3</dd>  
</dl>
```

Capítulo 8: Tablas

El elemento HTML `<table>` permite a los autores web mostrar datos de tabla (como texto, imágenes, enlaces, otras tablas, etc.) en una tabla bidimensional con filas y columnas de celdas.

Sección 8.1: Tabla simple

```
<table>
  <tr>
    <th>Encabezado 1/Columna 1</th>
    <th>Encabezado 2/Columna 2</th>
  </tr>
  <tr>
    <td>Datos Fila 1 Columna 1</td>
    <td>Datos Fila 1 Columna 2</td>
  </tr>
  <tr>
    <td>Datos Fila 2 Columna 1</td>
    <td>Datos Fila 2 Columna 2</td>
  </tr>
</table>
```

Esto mostrará una `<table>` que consta de tres filas totales (`<tr>`): una fila de celdas de encabezado (`<th>`) y dos filas de celdas de contenido (`<td>`). Los elementos `<th>` son *encabezados de tabla* y los elementos `<td>` son *datos de tabla*. Puede poner lo que quiera dentro de un `<td>` o `<th>`.

Encabezado 1/Columna 1	Encabezado 2/Columna 2
Datos Fila 1 Columna 1	Datos Fila 1 Columna 2
Datos Fila 2 Columna 1	Datos Fila 2 Columna 2

Sección 8.2: Extender columnas o filas

Las celdas de una tabla pueden abarcar varias columnas o filas mediante los atributos `colspan` y `rowspan`. Estos atributos pueden aplicarse a los elementos `<th>` y `<td>`.

```
<table>
  <tr>
    <td>fila 1 columna 1</td>
    <td>fila 1 columna 2</td>
    <td>fila 1 columna 3</td>
  </tr>
  <tr>
    <td colspan="3">Esta segunda fila abarca las tres columnas</td>
  </tr>
  <tr>
    <td rowspan="2">Esta celda abarca dos filas</td>
    <td>fila 3 columna 2</td>
    <td>fila 3 columna 3</td>
  </tr>
  <tr>
    <td>fila 4 columna 2</td>
    <td>fila 4 columna 3</td>
  </tr>
</table>
```

Tendrá como resultado

fila 1 columna 1	fila 1 columna 2	fila 1 columna 3
Esta segunda fila abarca las tres columnas		
Esta celda abarca dos filas	fila 3 columna 2	fila 3 columna 3
	fila 4 columna 2	fila 4 columna 3

Ten en cuenta que no debe diseñar una tabla en la que las filas y las columnas se solapen, ya que se trata de HTML no válido y el resultado es tratado de forma diferente por los distintos navegadores web.

`rowspan` = Número entero no negativo que especifica el número de filas que abarca una celda. El valor por defecto 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` = Número entero no negativo que especifica el número de columnas que abarca la celda actual. El valor de este atributo es uno (1). Un valor de cero (0) significa que la celda se extenderá desde la columna actual hasta la última columna del grupo de columnas `<colgroup>` en el que está definida la celda.

Sección 8.3: Grupos de columnas

A veces es posible que desee aplicar un estilo a una columna o a un grupo de columnas. O por motivos semánticos, agrupar las columnas. Para ello, utilice los elementos `<colgroup>` y `<col>`.

La etiqueta opcional `<colgroup>` permite agrupar columnas. Los elementos `<colgroup>` deben ser elementos hijos de `<table>` y deben ir después de cualquier elemento `<caption>` y antes de cualquier contenido de 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 a un rango de columnas sin aplicar una lógica de lógica. Los elementos `<col>` son opcionales, pero si están presentes, deben estar dentro de un elemento `<colgroup>`.

```
<table>
  <colgroup>
    <col id="MiColumnaEspecial" />
    <col />
  </colgroup>
  <colgroup>
    <col class="ColumnaGenial" />
    <col class="ColumnaBonita" span="2" />
  </colgroup>
  ...
</table>
```

Los siguientes estilos CSS pueden aplicarse a los elementos `<colgroup>` y `<col>`:

- `border`
- `background`
- `width`
- `visibility`
- `display` (como `display: none`)
 - `display: none`; eliminará las columnas de la pantalla, haciendo que la tabla se muestre como si no existieran esas celdas. esas celdas no existirían.

Para más información, vea [Datos de tabla HTML5](#).

Sección 8.4: Tabla con `thead`, `tbody`, `tfoot` y `caption`

HTML también proporciona las tablas con los elementos `<thead>`, `<tbody>`, `<tfoot>` y `<caption>`. Estos elementos adicionales son útiles para añadir valor semántico a sus tablas y para proporcionar un lugar para estilos CSS independientes.

Al imprimir una tabla que no cabe en una página (de papel), la mayoría de los navegadores repiten el contenido de `<thead>` en cada página.

Hay un orden específico que hay que seguir y ser conscientes de que no todos los elementos encajan en su sitio como uno esperaría. El siguiente ejemplo muestra cómo deben colocarse nuestros 4 elementos.

```
<table>
  <caption>Título tabla</caption> <!-- caption es el primer hijo de la tabla -->
  <thead> <!-- thead está después de caption -->
    <tr>
      <th> Contenido del encabezado 1</th>
      <th> Contenido del encabezado 2</th>
    </tr>
  </thead>
  <tbody> <!-- tbody está después de thead -->
    <tr>
      <td> Contenido del cuerpo 1</td>
      <td> Contenido del cuerpo 2</td>
    </tr>
  </tbody>
  <tfoot>
    <!-- tfoot puede colocarse antes o después de tbody, pero no en un grupo de tbody. -->
    <!-- Independientemente de dónde se encuentre tfoot en el marcado, se muestra en la parte inferior -->
    <tr>
      <td> Contenido de pie de página 1</td>
      <td> Contenido de pie de página 2</td>
    </tr>
  </tfoot>
</table>
```

Los resultados del siguiente ejemplo se muestran dos veces: la primera tabla carece de estilos y la segunda tiene aplicadas algunas propiedades CSS: `background-color`, `color` y `border*`. Los estilos se proporcionan como una guía visual y no es un aspecto esencial del tema que nos ocupa.

Table Title	
Header content 1	Header content 2
Body content 1	Body content 2
Footer content 1	Footer content 2

Table Title	
Header content 1	Header content 2
Body content 1	Body content 2
Footer content 1	Footer content 2

Elemento	Estilos Aplicados
<code><caption></code>	Texto amarillo sobre fondo negro.
<code><thead></code>	Texto en negrita sobre fondo morado.
<code><tbody></code>	Texto sobre fondo azul.
<code><tfoot></code>	Texto sobre fondo verde.
<code><th></code>	Bordes naranjas.
<code><td></code>	Bordes rojos.

Sección 8.5: Heading scope

`th` se utiliza muy a menudo para indicar los encabezamientos de las filas y columnas de una tabla, por ejemplo:

```

<table>
  <thead>
    <tr>
      <td></td>
      <th> Encabezamiento de columna 1</th>
      <th> Encabezamiento de columna 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th> Encabezamiento de fila 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th> Encabezamiento de fila 2</th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>

```

Esto puede mejorarse para la accesibilidad mediante el uso del atributo `scope`. El ejemplo anterior se modificaría de la siguiente manera como sigue:

```

<table>
  <thead>
    <tr>
      <td></td>
      <th scope="col"> Encabezamiento de columna 1</th>
      <th scope="col"> Encabezamiento de columna 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row"> Encabezamiento de fila 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th scope="row"> Encabezamiento de fila 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 conjunto 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>

Capítulo 9: Comentarios

Al igual que en otros lenguajes de programación, marcado y Markdown, los comentarios en HTML proporcionan a otros desarrolladores información específica sobre el desarrollo sin afectar a la interfaz de usuario. Sin embargo, a diferencia de otros lenguajes, los comentarios HTML pueden utilizarse para especificar elementos HTML sólo para Internet Explorer. Este tema explica cómo escribir comentarios HTML y sus aplicaciones funcionales. Este tema explica cómo escribir comentarios HTML y sus aplicaciones funcionales.

Sección 9.1: Crear comentarios

Los comentarios HTML pueden utilizarse para dejarse notas a uno mismo o a otros desarrolladores sobre un punto concreto del código. Pueden iniciarse con `<!--` y concluirse con `-->`, así:

```
<!-- ¡Soy un comentario HTML! -->
```

Pueden incorporarse en línea dentro de otros contenidos:

```
<h1>Esta parte se mostrará <!-- mientras que este no se mostrará -->.</h1>
```

También pueden abarcar varias líneas para proporcionar más información:

```
<!-- Esto es un comentario HTML multilínea.  
Lo que esté aquí no será renderizado por el navegador.  
Puede "comentar" secciones enteras de código HTML.  
-->
```

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

```
<h1 <!-- testAttribute="algo" -->>Esto no va a funcionar</h1>
```

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

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

Sección 9.2: Comentario de 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 en el documento. Para evitar líneas muy largas en el marcado (que son difíciles de leer) y espacios en blanco no intencionados (que afectan al formato), los espacios en blanco pueden comentarse. involuntarios (que afectan al formato), los espacios en blanco pueden comentarse.

```
<!-- Utilice un comentario HTML para anular el carácter de nueva línea a continuación: -->  
<a href="#">¡Espero que no haya más espacios en blanco!</a><!-- --><button>Foo</button>
```

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

Código de ejemplo:

```
<!-- Utilice un comentario HTML para anular el carácter de nueva línea a continuación: -->  
<a href="#">¡Espero que no haya más espacios en blanco!</a><!-- --><button>Foo</button>  
<hr>  
<!-- Sin él, se nota una pequeña diferencia de formato: -->  
<a href="#">¡Espero que no haya más espacios en blanco!</a> <button>Foo</button>
```

Salida:

¡Espero que no haya más espacios en blanco! Foo

¡Espero que no haya más espacios en blanco! Foo

Capítulo 10: Clases e identificadores

Parámetro	Detalles
class	Indica la clase del elemento (no único)
id	Indica el identificador del elemento (único en el mismo contexto)

Las clases y los ID facilitan la referencia a elementos HTML desde scripts y hojas de estilo. El atributo class puede en una o varias etiquetas y es utilizado por CSS para aplicar estilos. Sin embargo, los identificadores están pensados para referirse a un único elemento, lo que significa que el mismo identificador nunca debe utilizarse dos veces. Los ID se utilizan generalmente con JavaScript y los enlaces internos del documento, y se desaconsejan en CSS. Este tema contiene explicaciones y ejemplos útiles sobre el uso correcto de los atributos class e ID en HTML.

Sección 10.1: Dar una clase a un elemento

Las clases son identificadores de los elementos a los que se asignan. Utilice el atributo class para asignar una clase a un elemento.

```
<div class="clase-ejemplo "></div>
```

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

```
<div class="clase1 clase2"></div>
```

Uso de clases en CSS

Las clases pueden utilizarse para dar estilo a determinados elementos sin cambiar todos los elementos de ese tipo. Por ejemplo, estos dos elementos span pueden tener estilos completamente diferentes:

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

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

Por ejemplo, tenemos dos elementos, ambos con la clase resaltar:

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

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

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

Sin embargo, si sólo queremos apuntar a div con la clase resaltar entonces podemos añadir especificidad como a continuación:

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

No obstante, al aplicar estilos con CSS, en general se recomienda utilizar sólo clases (por ejemplo, .resaltar) en lugar de elementos con clases (por ejemplo, div.resaltar).

Como con cualquier otro selector, las clases pueden anidarse:

```
.main .resaltar { color: red; } /* Combinador descendente */  
.footer > .resaltar { color: blue; } /* Combinador hijo*/
```

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

```
<div class="especial left menu"> Este texto será rosa</div>
```

Si queremos colorear este texto de rosa, podemos hacer lo siguiente en nuestro CSS:

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

Sección 10.2: Dar un identificador a un elemento

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 enlaza (utilizando un ancla), scripting, o estilo (con CSS).

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

No debe haber dos elementos con el mismo ID en el mismo documento, incluso si los atributos se adjuntan a dos tipos diferentes de elementos. Por ejemplo, el siguiente código es incorrecto:

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

Los navegadores harán todo lo posible para mostrar este código, pero pueden producirse comportamientos inesperados al aplicar estilos con CSS o añadir funcionalidades con JavaScript.

Para hacer referencia a elementos por su ID en CSS, anteponga # al ID.

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

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

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

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

Sección 10.3: Valores aceptables

Para una identificación

Version \geq 5

Las únicas restricciones al valor de un `id` son:

1. debe ser único en el documento
2. no debe contener espacios en blanco
3. debe contener al menos un carácter

Así que el valor puede ser todo dígitos, sólo un dígito, sólo caracteres de puntuación, incluir caracteres especiales, lo que sea. Sólo sin espacios en blanco.

Por tanto, 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="c"> ... </div>
<div id="Ⓜ.☆€~\ "> ... </div>
```

Esto no es válido:

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

Esto tampoco es válido cuando se incluye en el mismo documento:

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

Un valor de `id` debe empezar por una letra, que sólo puede ir seguida de:

- letras (A-Z/a-z)
- dígitos (0-9)
- guiones ("-")
- guiones bajos ("_")
- dos puntos (":")
- puntos (".")

En relación con el primer grupo de ejemplos de la sección HTML5 anterior, sólo uno es válido:

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

También son válidos:

```
<div id="textomodelo"> ... </div>
<div id="texto-modelo"> ... </div>
<div id="texto_modelo"> ... </div>
<div id="texto:modelo"> ... </div>
<div id="texto.modelo"> ... </div>
```

De nuevo, si no empieza por una letra (mayúscula o minúscula), no es válido.

Para una clase

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

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

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

Esto es perfectamente correcto:

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

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

Tenga en cuenta que las reglas y ejemplos anteriores se aplican en el contexto de HTML.

El uso de números, signos de puntuación o caracteres especiales en el valor de un identificador o una clase puede causar problemas en otros contextos, como CSS, JavaScript y expresiones regulares.

Por ejemplo, aunque el siguiente `id` es válido en HTML5:

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

... no es válido en CSS:

4.1.3 Caracteres y casos

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

En la mayoría de los casos podrá escapar de los caracteres en contextos en los que tengan restricciones o un significado especial.

Referencias del W3C

- [3.2.5.1 El atributo id](#)
- [3.2.5.7 Atributo class](#)
- [6.2 Tipos básicos SGML](#)

Sección 10.4: Problemas relacionados con los identificadores duplicados

Tener más de un elemento con el mismo ID es un problema difícil de solucionar. El analizador HTML normalmente intenta renderizar la página, en cualquier caso. Normalmente no se produce ningún error. Pero el ritmo podría acabar en una página web con mal comportamiento.

En este ejemplo:

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

Los selectores CSS aún funcionan

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

Pero JavaScript no maneja ambos elementos:

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

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

Capítulo 11: Atributos de datos

Valor	Descripción
algunvalor	Especifica el valor del atributo (en forma de cadena de caracteres)

Sección 11.1: Compatibilidad con navegadores antiguos

Los atributos de datos se introdujeron en HTML5, que es compatible con todos los navegadores modernos, pero los navegadores anteriores a 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, es decir, atributos que no forman parte de las especificaciones HTML. Por ejemplo, el atributo `value` de la línea inferior se considera un atributo no estándar porque las especificaciones de la etiqueta `` no tienen un atributo `value` y no es un atributo global:

```

```

Esto significa que, aunque los atributos de datos no están soportados en los navegadores antiguos, siguen funcionando y se pueden establecer y recuperar utilizando los mismos métodos genéricos de JavaScript `setAttribute` y `getAttribute`, pero no se puede utilizar la propiedad nueva `dataset`, que sólo es compatible con los navegadores modernos.

Sección 11.2: Uso de atributos de datos

Los atributos `data-*` de HTML5 ofrecen una forma cómoda de almacenar datos en elementos HTML. Los datos almacenados pueden leerse o modificarse mediante JavaScript.

```
<div data-submitted="yes" class="perfil_usuario">  
  ... algún contenido ...  
</div>
```

- La estructura de los atributos de datos es `data-*`, es decir, el nombre del atributo de datos va después de la parte `data-`. Con este nombre, se puede acceder al atributo.
- Los datos en formato cadena de caracteres (incluido `json`) pueden almacenarse utilizando el atributo `data-*`.

Capítulo 12: Enlazar recursos

Atributo	Detalles
charset	Especifica la codificación de caracteres del documento enlazado
crossorigin	Especifica cómo gestiona el elemento las solicitudes de origen cruzado
href	Especifica la ubicación del documento vinculado
hreflang	Especifica el idioma del texto del documento enlazado
media	Especifica en qué dispositivo se mostrará el documento vinculado, a menudo se utiliza con la selección de hojas de estilo en función del dispositivo
rel	Obligatorio. Especifica la relación entre el documento actual y el documento vinculado.
rev	Especifica la relación entre el documento vinculado y el documento actual
sizes	Especifica el tamaño del recurso enlazado. Sólo cuando <code>rel="icon"</code>
target	Especifica dónde debe cargarse el documento vinculado
type	Especifica el tipo de soporte del documento enlazado
integrity	Especifica un hash codificado en base64 (sha256, sha384 o sha512) del recurso enlazado que permite al navegador verificar su legitimidad. navegador verificar su legitimidad.

Aunque muchas secuencias de comandos, iconos y hojas de estilo pueden escribirse directamente en el marcado HTML, la mejor práctica y más eficaz consiste en incluir estos recursos en su propio archivo y vincularlos al documento. Este tema trata de la vinculación de recursos externos, como hojas de estilo y scripts, a un documento HTML.

Sección 12.1: JavaScript

Síncrono

```
<script src="ruta/hacia.js"></script>
```

La práctica habitual es colocar las etiquetas `<script>` de JavaScript justo antes de la etiqueta de cierre `</body>`. Cargar los scripts en último lugar permite que los elementos visuales del sitio se muestren más rápidamente y evita que el JavaScript intente interactuar con elementos que aún no se han cargado.

Asíncrono

```
<script src="ruta/hacia.js" async></script>
```

Otra alternativa, cuando el código Javascript que se está cargando no es necesario para la inicialización de la página, puede cargarse 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 sintáctico del HTML para analizar el archivo Javascript.

Diferido

```
<script src="ruta/hacia.js" defer></script>
```

Los scripts diferidos son como los scripts asíncronos, con la excepción de que el análisis sólo se realizará una vez que el HTML esté completamente analizado. Se garantiza que las secuencias de comandos diferidas se carguen en el orden de declaración, al igual que las secuencias de comandos síncronas.

```
<noscript>
```

```
<noscript>JavaScript disabled</noscript>
```

El elemento `<noscript>` define el contenido que se mostrará si el usuario tiene desactivados los scripts o si el navegador no soporta el uso de scripts. La etiqueta `<noscript>` puede colocarse en el `<head>` o en el `<body>`.

Sección 12.2: Hoja de estilos CSS externa

```
<link rel="stylesheet" href="ruta/hacia.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 forma, el CSS se cargará primero y se aplicará a la página mientras se carga, en lugar de mostrar el HTML sin estilo hasta que se cargue el CSS. El atributo `type` no es necesario en HTML5, porque HTML5 suele soportar CSS.

```
<link rel="stylesheet" href="ruta/hacia.css" type="text/css">
```

y

```
<link rel="stylesheet" href="ruta/hacia.css">
```

... hacen lo mismo en HTML5.

Otra práctica, aunque menos común, es utilizar una declaración `@import` dentro de CSS directo. Así:

```
<style type="text/css">
  @import("path/to.css")
</style>
<style>
  @import("path/to.css")
</style>
```

Sección 12.3: Favicon

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

Utilice el tipo MIME `image/png` para los archivos PNG e `image/x-icon` para los archivos de iconos (`*.ico`). Para ver la diferencia, consulte [esta pregunta de StackOverflow](#).

Un archivo llamado `favicon.ico` en la raíz de su sitio web normalmente se cargará y se aplicará automáticamente, sin la necesidad de una etiqueta `<link>`. Si este archivo cambia alguna vez, los navegadores pueden ser lentos y obstinados a la hora de actualizar su caché.

Sección 12.4: CSS alternativo

```
<link rel="alternate stylesheet" href="ruta/hacia/style.css" title="tuTitulo">
```

Algunos navegadores permiten aplicar hojas de estilo alternativas si se ofrecen. Por defecto no se aplicarán, pero normalmente se pueden cambiar a través de la configuración del navegador:

Firefox permite al usuario seleccionar la hoja de estilos utilizando el submenú Ver > Estilo de página, Internet Explorer también admite esta función (a partir de IE 8), a la que también se accede desde Ver > Estilo de página (al menos a partir de 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 que el usuario cambie de estilo.

(Fuente: [MDN Docs](#))

Sección 12.5: Sugerencia sobre recursos: dns-prefetch, prefetch, prerender

Preconnect

La relación `preconnect` es similar a `dns-prefetch` en que resolverá el DNS. Sin embargo, también realizará el handshake TCP, y la negociación TLS opcional. Esta función es experimental.

```
<link rel="preconnect" href="URL">
```

dns-prefetch

Informa a los navegadores para que resuelvan el DNS de 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 determinado recurso debe precargarse para que pueda cargarse más rápidamente.

```
<link rel="prefetch" href="URL">
```

DNS-Prefetch sólo resuelve el nombre de dominio, mientras que prefetch descarga/almacena los recursos especificados.

prerender

Informa a los navegadores para que obtengan y rendericen la URL en segundo plano, de modo que puedan ser entregadas al usuario instantáneamente cuando el usuario navega a esa URL. Esta función es experimental.

```
<link rel="prerender" href="URL">
```

Sección 12.6: Atributo 'media' del enlace

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

Media especifica qué hoja de estilo debe utilizarse para cada tipo de medio. Utilizando el valor `print` sólo se mostraría esa hoja de estilo para las páginas de impresión.

El valor de este atributo puede ser cualquiera de los valores de tipo `media` (similar a una consulta de medios CSS).

Sección 12.7: Anterior y Siguiente

Cuando una página forma parte de una serie de artículos, por ejemplo, se pueden utilizar `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">
```

Sección 12.8: Web Feed

Utilice el atributo `rel="alternate"` para permitir el descubrimiento de sus fuentes Atom/RSS.

```
<link rel="alternate" type="application/atom+xml" href="http://ejemplo.com/feed.xml" />  
<link rel="alternate" type="application/rss+xml" href="http://ejemplo.com/feed.xml" />
```

Capítulo 13: Incluir código JavaScript en HTML

Atributo	Detalles
<code>src</code>	Especifica la ruta a un archivo JavaScript. Puede ser 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 (sólo para scripts externos). Este atributo no requiere ningún valor (excepto en XHTML).
<code>defer</code>	Especifica que el script se ejecutará cuando la página haya terminado de parsear (sólo para scripts externos). Este atributo no requiere ningún valor (excepto en XHTML).
<code>charset</code>	Especifica la codificación de caracteres utilizada en un archivo de script externo, por ejemplo, UTF-8
<code>crossorigin</code>	Cómo gestiona el elemento las solicitudes de origen cruzado
<code>nonce</code>	Criptográfico utilizado en las comprobaciones de la política de seguridad de contenidos CSP3

Sección 13.1: Manejo de JavaScript desactivado

Es posible que el navegador del cliente no soporte Javascript o tenga desactivada la ejecución de Javascript, quizás por razones de seguridad. Para poder indicar a los usuarios que un script debe ejecutarse en la página, puede utilizarse la etiqueta `<noscript>`. El contenido de `<noscript>` se muestra siempre que Javascript esté desactivado para la página actual.

```
<script>
  document.write("¡Hola, mundo!");
</script>
<noscript> Este navegador no soporta Javascript. </noscript>
```

Sección 13.2: Enlace a un archivo JavaScript externo

```
<script src="ejemplo.js"></script>
```

El atributo `src` funciona como el atributo `href` en los anclajes: puede especificar una URL absoluta o relativa. El ejemplo anterior enlaza a un archivo dentro del mismo directorio del documento HTML. Normalmente se añade dentro de las etiquetas `<head>` en la parte superior del documento html.

Sección 13.3: Inclusión directa de código JavaScript

En lugar de enlazar a un archivo externo, también puede incluir el código JS tal cual en su HTML:

```
<script>
// código JavaScript
</script>
```

Sección 13.4: Inclusión de un archivo JavaScript que se ejecuta de forma asíncrona

```
<script type="text/javascript" src="URL" async></script>
```

Capítulo 14: Uso de HTML con CSS

CSS proporciona estilos a los elementos HTML de la página. El estilo en línea implica el uso del atributo `style` en las etiquetas, y se desaconseja totalmente. Las hojas de estilo internas utilizan la etiqueta `<style>` y se emplean para declarar reglas para partes dirigidas de la página. Las hojas de estilo externas pueden utilizarse mediante una etiqueta `<link>` que toma un archivo externo de CSS y aplica las al documento. Este tema cubre el uso de los tres métodos de fijación.

Sección 14.1: Uso de hojas de estilo externas

Utilice el atributo `link` en la cabecera del documento:

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

También puede utilizar hojas de estilo proporcionadas por sitios web a través de una red de distribución de contenidos (CDN). (por ejemplo, Bootstrap):

```
<head>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
BVYiISIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
</head>
```

Generalmente, puede encontrar soporte CDN para un framework en su sitio web.

Sección 14.2: Hoja de estilo interna

También puede incluir elementos CSS internamente utilizando la etiqueta `<style>`:

```
<head>
  <style type="text/css">
    body {
      background-color: gray;
    }
  </style>
</head>
```

También se pueden incluir varias 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>
```

Sección 14.3: Estilo en línea

Puede aplicar estilo a un elemento específico utilizando el atributo `style`:

```
<span style="color: red"> El texto aparecerá en rojo. </span>
```

Nota: Trate de evitar esto - el punto de CSS es separar el contenido de la presentación.

Sección 14.4: Hojas de estilo múltiples

Es posible cargar varias hojas de estilo:

```
<head>
  <link rel="stylesheet" type="text/css" href="general.css">
  <link rel="stylesheet" type="text/css" href="specific.css">
</head>
```

Ten en cuenta que **los archivos y declaraciones posteriores anularán los anteriores**. Así que si `general.css` contiene:

```
body {
  background-color: red;
}
```

y `specific.css` contiene:

```
body {
  background-color: blue;
}
```

sí se utilizan ambos, el fondo del documento será azul.

Capítulo 15: Imágenes

Parámetros	Detalles
<code>src</code>	Especifica la URL de la imagen
<code>srcset</code>	Imágenes para utilizar 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 gestiona el elemento 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 servidor
<code>alt</code>	Texto alternativo que debería mostrarse si por alguna razón la imagen no pudiera mostrarse.
<code>width</code>	Especifica el ancho de la imagen (opcional)
<code>height</code>	Especifica la altura de la imagen (opcional)

Sección 15.1: Creación de una imagen

Para añadir una imagen a una página, utilice la etiqueta `image`.

Las etiquetas de imagen (`img`) no tienen etiquetas de cierre. Los dos atributos principales que se dan a la etiqueta `img` son `src`, la fuente de la imagen, y `alt`, que es un texto alternativo que describe la imagen.

```

```

También puedes obtener imágenes de una URL web:

```

```

Nota: técnicamente, las imágenes no se insertan en una página HTML, sino que se vinculan a páginas HTML. La etiqueta `` crea un espacio de retención para la imagen referenciada.

También es posible incrustar imágenes directamente dentro de la página utilizando base64:

```

```

Consejo: para enlazar una imagen a otro documento, basta con anidar la etiqueta `` dentro de etiquetas `<a>`.

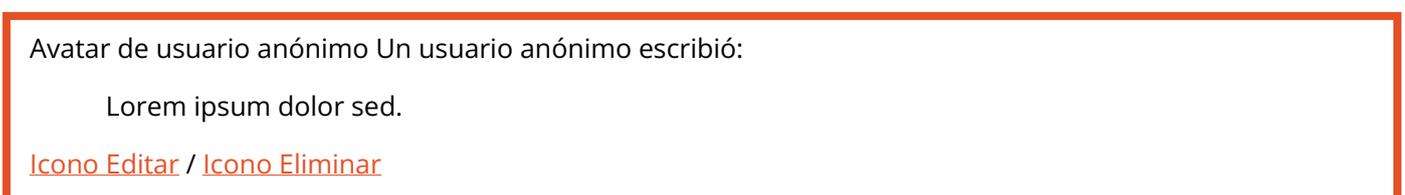
Sección 15.2: Elección del texto alternativo

Los lectores de pantalla y los motores de búsqueda utilizan el texto alternativo. Por eso es importante escribir un buen texto alternativo para las imágenes.

El texto debería verse correcto incluso si sustituye la imagen por su atributo `alt`. Por ejemplo:

```
<!-- Incorrecto -->
 Un usuario anónimo escribió:
<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:



Para corregirlo:

- Elimina el texto alternativo del avatar. Esta imagen añade 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. ¹
- Elimine el "icono" del texto alternativo de los iconos. Saber que sería un icono si estuviera allí no ayuda a transmitir su propósito real.

```
<!-- Correcto -->
 Un usuario anónimo escribió:
<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 a pie de página

1 Existe 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 ocurre en este caso: es sólo una imagen añadida que no es necesaria para entender el resto) y, por tanto, puede omitirse en 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.

Sección 15.3: Imagen responsiva utilizando el atributo `srcset`

Uso de `srcset` con tamaños

```

```

`sizes` son como media queries, que describen cuánto espacio ocupa la imagen en el viewport.

- si la ventana es mayor que 1200px, la imagen es exactamente 580px (por ejemplo, nuestro contenido se centra en el contenedor que tiene un ancho máximo de 1200px. La imagen ocupa la mitad menos los márgenes).
- si el viewport está entre 640px y 1200px, la imagen ocupa el 48% del viewport (por ejemplo, la imagen se escala con nuestra y ocupa la mitad del ancho de la ventana menos los márgenes).
- si el viewport es de cualquier otro tamaño, en nuestro caso inferior a 640px, la imagen ocupa el 98% del viewport (por ejemplo, la imagen se escala con nuestra página y ocupa todo el ancho del viewport menos los márgenes). **La condición de medios debe omitirse para último elemento.**

`srcset` es sólo decirle 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` es siempre la fuente de imagen obligatoria. En el caso de usar `srcset`, `src` servirá la imagen fallback en el caso de que el navegador no soporte `srcset`.

Uso de `srcset` sin tamaños

```

```

`srcset` proporciona la lista de imágenes disponibles, con la relación dispositivo-píxel x descriptor.

- si la relación dispositivo-píxel es 1, utilice `img/hello-300.jpg`,
- si la relación dispositivo-píxel es 2, utilice `img/hello-600.jpg`,
- si la relación dispositivo-píxel es 3, utilice `img/hello-1200.jpg`.

`src` es siempre la fuente de imagen obligatoria. En el caso de usar `srcset`, `src` servirá la imagen fallback en el caso de que el navegador no soporte `srcset`.

Sección 15.4: Imagen responsiva utilizando el elemento `Picture`

Código

```
<picture>
  <source media="(min-width: 600px)" srcset="large_image.jpg">
  <source media="(min-width: 450px)" srcset="small_image.jpg">
  
</picture>
```

Uso

Para mostrar distintas imágenes en distintos anchos de pantalla, debe incluir todas las imágenes que utilicen la etiqueta `source` en una etiqueta como se muestra en el ejemplo anterior.

Resultado

- En pantallas con ancho de pantalla >600px, se muestra `large_image.jpg`
- En pantallas con ancho de pantalla >450px, se muestra `small_image.jpg`
- En pantallas con otro ancho de pantalla, muestra `default_image.jpg`

Capítulo 16: Mapas de imagen

Etiqueta/Atributo	Valor
<code></code>	A continuación, se muestran los atributos específicos del mapa de imagen que se deben utilizar con <code></code> . Se aplican los atributos <code></code> normales.
<code>usemap</code>	El <code>name</code> del mapa con un símbolo hash añadido. Por ejemplo, para un mapa con <code>name="map"</code> , la imagen debe tener <code>usemap="#map"</code> .
<code><map></code>	
<code>name</code>	El nombre del mapa para identificarlo. Debe utilizarse con el atributo <code>usemap</code> de la imagen.
<code><area></code>	A continuación, se indican los atributos específicos de <code><area></code> . Cuando se especifica <code>href</code> , convirtiendo el <code><area></code> en un enlace, <code><area></code> también admite todos los atributos de la etiqueta de anclaje (<code><a></code>) excepto ping. Consulta en los documentos MDN .
<code>alt</code>	El texto alternativo que se mostrará si no se admiten imágenes. Esto sólo es necesario si <code>href</code> también se establece en el <code><area></code> .
<code>coords</code>	Las coordenadas que delimitan el área seleccionable. Cuando <code>shape="polygon"</code> , debe establecerse como una lista de pares "x, y" separados por comas (es decir, <code>shape="polygon" coords="x1, y1, x2, y2, x3, y3, ..."</code>). Cuando <code>shape="rectangle"</code> , esto debe establecerse 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, el <code><area></code> no representará un hipervínculo.
<code>shape</code>	La forma de la <code><area></code> . Puede establecerse por defecto para seleccionar toda la imagen (no es necesario el atributo <code>coords</code>), <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 para un área poligonal especificada por puntos de esquina.

Sección 16.1: Introducción a los mapas de imágenes

Descripción

Un mapa de imagen es una imagen con zonas en las que se puede hacer clic y que suelen actuar como hipervínculos.

La imagen se define mediante la etiqueta ``, y el mapa se define mediante una etiqueta `<map>` con etiquetas `<area>` para denotar cada zona en la que se puede hacer clic. Utilice los atributos `usemap` y `name` para vincular la imagen y el mapa.

Ejemplo básico

Para crear un mapa de imagen de forma que se pueda hacer clic en cada una de las formas de la imagen inferior:



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>
```

Capítulo 17: Elementos de control de entrada

Parámetro	Detalles
<code>class</code>	Indica la clase de la entrada
<code>id</code>	Indica el identificador de la entrada
<code>type</code>	Identifica el tipo de control de entrada que se mostrará. Los valores aceptables son <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.
<code>name</code>	Indica el nombre de la entrada
<code>disabled</code>	Valor booleano que indica que la entrada debe estar desactivada. Los controles desactivados no pueden editarse no se envían al enviar el formulario y no pueden recibir foco.
<code>checked</code>	Cuando el valor del atributo <code>type</code> es <code>radio</code> o <code>checkbox</code> , la presencia de este atributo booleano indica que el control está seleccionado por defecto; en caso contrario, se ignora.
<code>multiple</code>	HTML5 Indica que se pueden pasar varios archivos o valores (Sólo se aplica a las entradas de tipo <code>file</code> y <code>email</code>).
<code>placeholder</code>	HTML5 Indicación al usuario de lo que puede introducir en el control. El texto del marcador de posición no debe contener retornos de carro ni saltos de línea.
<code>autocomplete</code>	HTML5 Indica si el valor del control puede ser completado automáticamente por el navegador.
<code>readonly</code>	Valor booleano que indica que la entrada no es editable. Los controles de sólo lectura se siguen enviando al enviar el formulario, pero no reciben el foco. HTML5 : Este atributo se ignora cuando el valor del atributo <code>type</code> se establece como <code>hidden</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> o <code>button</code> .
<code>required</code>	HTML5 Indica que un valor debe estar presente o que el elemento debe estar marcado para que el formulario se envíe se envíe
<code>alt</code>	Un texto alternativo para las imágenes, en caso de que no se muestren.
<code>autofocus</code>	El elemento <code><input></code> debe obtener el foco cuando se carga la página.
<code>value</code>	Indica el valor del elemento <code><input></code> .
<code>step</code>	El atributo paso 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> .

Las etiquetas de entrada son un componente clave de los sistemas web interactivos de entrada de los usuarios. Los distintos tipos de elementos de entrada pueden regular los datos introducidos para que se ajusten a un formato determinado y proporcionar seguridad a la introducción de contraseñas.

Sección 17.1: Texto

El tipo de entrada más básico y la entrada por defecto 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 introducido. Todos los demás caracteres pueden introducirse en él. Los elementos `<input>` se utilizan dentro de un elemento `<form>` para declarar controles de entrada que permiten a los usuarios introducir datos.

Sintaxis

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

o (sin especificar un tipo, utilizando el atributo por defecto):

```
<input>
```

La anchura por defecto de un campo de texto es de 20 caracteres. Esto puede cambiarse especificando un valor para el atributo `size` de la siguiente manera:

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

El atributo `size` es claramente diferente de establecer un ancho con CSS. El uso de `size` define un valor específico (en número de píxeles, porcentaje del elemento padre, etc.) que la entrada debe tener siempre. El uso del `size` calcula la cantidad de anchura que debe asignarse en función de la fuente utilizada y de la anchura normal de los caracteres.

Nota: La utilización del atributo `size` no limita intrínsecamente el número de caracteres que pueden introducirse en el cuadro, sino sólo la anchura de éste. Para limitar la longitud, ve a Validación de entradas.

Un campo de entrada sólo permite una línea de texto. Si necesita una entrada de texto multilinea para una cantidad considerable de texto, utilice en su lugar un elemento `<textarea>`.

Sección 17.2: Checkbox y botones de Radio

Resumen

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

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

Un único elemento de casilla de verificación independiente se utiliza 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* desmarca las otras casillas de verificación del grupo de casillas de verificación.

Los botones de radio suelen venir en grupos (si no está agrupado con otro botón de radio, probablemente querías usar una casilla de verificación en su lugar) identificados mediante el uso del mismo atributo `name` en todos los botones dentro de ese grupo. La selección de botones de opción es *mutuamente excluyente*, lo que significa que el usuario sólo puede seleccionar una opción de un grupo de botones de opción. Cuando se marca un botón de opción, se desmarca cualquier otro botón de opción con el mismo `name` que estuviera marcado anteriormente. desmarcado.

Por ejemplo:

```
<input type="radio" name="color" id="rojo" value="#F00">
<input type="radio" name="color" id="verde" value="#0F0">
<input type="radio" name="color" id="azul" value="#00F">
```

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

Atributos

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

`value`

Como cualquier otro elemento de entrada, el atributo `value` especifica el valor de la cadena de caracteres que se asociará al botón en caso de que de envío del formulario. Sin embargo, las casillas de verificación y los botones de opción tienen la particularidad de que, cuando se omite el valor, se activa por defecto a `on` al enviarse, en lugar de enviar un valor en blanco. El atributo `value` no se refleja en la apariencia del botón del botón.

checked

El atributo `checked` especifica el estado inicial de una casilla de verificación o botón de radio. Se trata de un atributo booleano y puede omitirse.

Cada una de ellas es una forma válida y equivalente 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

Labels

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

Por 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="rojo">
<label for="rojo">Rojo</label>
```

Grupo de botones

Dado que cada botón de opción afecta a los demás del grupo, es habitual proporcionar una etiqueta o contexto para todo el grupo de botones de opción. grupo de botones de opción.

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

Por ejemplo:

```
<fieldset>
  <legend>Color del tema:</legend>
  <p>
    <input type="radio" name="color" id="rojo" value="#F00">
    <label for="rojo">Rojo</label>
  </p>
  <p>
    <input type="radio" name="color" id="verde" value="#0F0">
    <label for="verde">Verde</label>
  </p>
  <p>
    <input type="radio" name="color" id="azul" value="#00F">
    <label for="azul">Azul</label>
  </p>
</fieldset>
```

Las casillas de verificación también se pueden agrupar de forma similar, con un conjunto de campos y una leyenda que identifiquen 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 son mutuamente excluyentes. Si se hace esto, el formulario enviará múltiples valores para la misma clave y no todos los lenguajes de servidor manejan esto de la misma manera (comportamiento indefinido). Cada casilla de verificación debe tener un nombre único o utilizar un conjunto de corchetes ([]) para indicar que el formulario debe enviar un array de valores para esa clave. El método que elija dependerá de cómo vaya a manejar los datos del formulario en el lado del cliente o en el 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 del conjunto de campos.

Sección 17.3: Validación de entrada

La validación de entradas HTML la realiza automáticamente el navegador basándose en atributos especiales del elemento de entrada. Podría sustituir parcial o totalmente a la validación de entradas de JavaScript. Este tipo de validación puede ser eludida por el usuario a través de peticiones HTTP especialmente diseñadas, por lo que no sustituye a la validación de entrada del lado del servidor. La validación sólo se produce al intentar 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 utilice JavaScript). Tenga en cuenta que las entradas desactivadas o de sólo lectura no se validarán. la validación.

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

Version \geq 5

Required

Utilice el atributo `required` para indicar que un campo debe rellenarse para pasar la validación.

```
<input required>
```

Longitud mínima / máxima

Utilice los atributos `minlength` y `maxlength` para indicar los requisitos de longitud. La mayoría de los navegadores impedirán que el usuario escriba más caracteres de los *máximos* en la casilla, evitando así que su entrada sea inválida incluso antes de que de intentar enviarla.

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

Especificar un rango

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

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

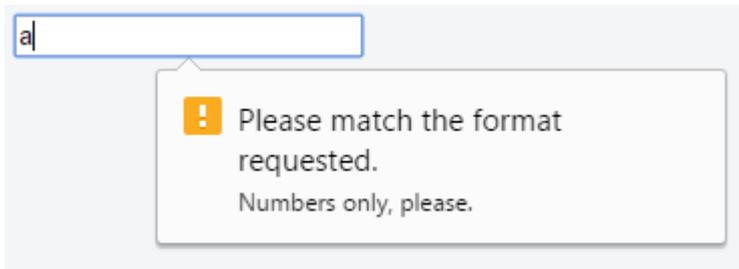
Version \geq 5

Coincidir con un patrón

Para un mayor control, utilice el atributo `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="Solo números, por favor.">
```

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



No todos los navegadores muestran un mensaje en caso de patrones no válidos, aunque la mayoría de los navegadores modernos son totalmente compatibles.

Compruebe el último soporte en [CanIUse](#) y aplíquelo en consecuencia.

Version \geq 5

Aceptar tipo de archivo

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

```
<input type="file" accept="image/*" title="Sólo se admiten imágenes">
```

Se pueden especificar varios valores con una coma, por ejemplo

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

Nota: Añadir el atributo `novalidate` al elemento formulario o el atributo `formnovalidate` al botón `submit`, impide la validación de los elementos del 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="Publicar" > <!-- se validará el formulario -->
  <input type="submit" value="Guardar" formnovalidate> <!-- el formulario NO será validado -->
</form>
```

El formulario contiene campos obligatorios para "publicar" el borrador, pero no para "guardarlo".

Sección 17.4: Color

Version \geq 5

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

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



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



Para los navegadores que no admiten este tipo de entrada, se utiliza una entrada normal `type=text`.

Sección 17.5: Password

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

El elemento `input` con un atributo `type` cuyo valor es `password` crea un campo de texto de una sola línea similar al `input type=text`, salvo que el texto no se muestra a medida que el usuario lo introduce.

```
<input type="password" name="contrasena" placeholder="Contraseña">
```

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

Nota: Algunos navegadores y sistemas modifican el comportamiento por defecto del campo de contraseña para mostrar también el carácter más recientemente tecleado durante un breve espacio de tiempo, de esta forma:

Sección 17.6: File

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

Las entradas de archivos permiten a los usuarios seleccionar un archivo de su sistema de archivos local para utilizarlo con la página actual. Si se utilizan junto con un elemento `form`, pueden usarse para permitir a los usuarios subir archivos a un servidor (para más información, véase Subir archivos).

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

```
<form action="subir_archivo.php" method="post" enctype="multipart/form-data">
  Select file to upload:
  <input type="file" name="archivoEnvio" id="archivoEnvio">
  <input type="submit" value="Sube tu archivo" name="enviar">
</form>
```

Múltiples archivos

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

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

Sección 17.7: Button

```
<input type="button" value="Texto del botón">
```

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

```
<button type="button">Texto del botón</button>
```

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

```
<input type="button" onclick="alert(';hola mundo!')" value="Haz clic">
```

o

```
<button type="button" onclick="alert(';hola mundo!')">Haz clic</button>
```

Atributos

[name]

El nombre del botón que se envía con los datos del formulario.

[type]

El `type` del 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 comportamiento por defecto. Puede tener scripts del lado del cliente asociados a los eventos del elemento, que se activan cuando se producen los eventos.

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

[value]

El valor inicial del botón.

Version ≥ 5

Atributos adicionales de los 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 antecesor (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 deben codificarse los datos del formulario al enviarlos al servidor mediante este botón. Sólo puede utilizarse con <code>formmethod="post"</code> .
<code>formmethod</code>	Especifica el método HTTP a utilizar (POST o GET) al enviar datos de formulario mediante este botón.
<code>formnovalidate</code>	Especifica que los datos del formulario no deben validarse al enviarse.
<code>formtarget</code>	Especifica dónde mostrar la respuesta que se recibe después de enviar el formulario utilizando este botón.

Sección 17.8: Submit

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

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

También puede utilizar el elemento `<button>` si necesita un botón de envío al que se le pueda aplicar un estilo más fácilmente o que contenga otros elementos:

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

Sección 17.9: Reset

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

Una entrada de tipo `reset` crea un botón que, cuando se pulsa, restablece todas las entradas del formulario que contiene a su estado por defecto.

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

Nota: Un botón de reinicio debe estar dentro o unido a (mediante el atributo `form`) un elemento `<form>` para que tenga algún efecto. El botón sólo restablecerá los elementos dentro de este formulario.

Sección 17.10: Hidden

```
<input type="hidden" name="entradaNombre" value="entradaNombre">
```

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

Sección 17.11: Tel

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

El elemento `input` con un atributo `type` cuyo valor es `tel` representa un control de edición de texto plano de una línea para introducir un número de teléfono.

Sección 17.12: Email

Version \geq 5

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

```
<form>
  <label>Correo electrónico: <label>
  <input type="email" name="email">
</form>
```

La dirección de correo electrónico puede validarse automáticamente al enviarla en función de la compatibilidad del navegador.

Sección 17.13: Number

Version \geq 5

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

El elemento `input` con un atributo `type` cuyo valor es `number` representa un control preciso para establecer el valor del elemento del elemento a una cadena de caracteres que representa un número.

Tenga en cuenta que este campo no garantiza que el número sea correcto. Sólo permite todos los símbolos que ser utilizados en cualquier número real, por ejemplo, el usuario podrá introducir un valor como `e1e-`, `0`.

Sección 17.14: Range

Version \geq 5

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

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

Atributo	Descripción	Valor por defecto
<code>min</code>	Valor mínimo del intervalo	0
<code>max</code>	Valor máximo del alcance	100
<code>step</code>	Importe a incrementar en cada incremento.	1

Sección 17.15: Search

Version \geq 5

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

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

Sección 17.16: Image

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

Una imagen. Debe utilizar el atributo `src` para definir la fuente de la imagen y el atributo `alt` para definir el texto alternativo. Puede utilizar los atributos `height` y `width` para definir el tamaño de la imagen en píxeles.

Sección 17.17: Week

Version \geq 5

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

Dependiendo de la compatibilidad del navegador, aparecerá un control para introducir un número de semana-año y un número de semana sin zona horaria.

Sección 17.18: Url

Version \geq 5

```
<input type="url" name="Pagina de inicio">
```

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 al enviarlo.

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

Sección 17.19: DateTime-Local

Version \geq 5

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

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

Sección 17.20: Month

Version \geq 5

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

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

Sección 17.21: Time

Version \geq 5

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

El `time` marca este elemento como aceptando una cadena de caracteres que representa una hora. El formato se define 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 `type="time"`. Las versiones más recientes de Android Browser, específicamente 4.4 y superiores lo soportan. Safari para iOS ofrece una compatibilidad parcial, ya que no admite los atributos `min`, `max`, y `step`.

Sección 17.22: DateTime (Global)

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

```
<fieldset>
  <p><label> Hora de la reunión: <input type=datetime name="reunion.inicio"></label>
</fieldset>
```

Atributos permitidos:

- atributos globales
- `name`
- `disabled`
- `form`
- `type`
- `autocomplete`
- `autofocus`
- `list`
- `min y max`
- `step (decimal)`
- `readonly`
- `valor required`

Sección 17.23: Date

Version \geq 5

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

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

Capítulo 18: Formularios

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 activado o desactivado el autocompletado.
<code>enctype</code>	Especifica cómo deben codificarse los datos del formulario cuando se envían al servidor (sólo para <code>method="post"</code>).
<code>method</code>	Especifica el método HTTP a utilizar cuando se envían datos de formulario (POST o GET).
<code>name</code>	Especifica el nombre del formulario.
<code>novalidate</code>	Especifica que el formulario no debe validarse al enviarse.
<code>target</code>	Especifica dónde mostrar la respuesta que se recibe tras enviar el formulario.

Para agrupar los elementos de entrada y envío de datos, HTML utiliza un elemento de formulario para encapsular los elementos de entrada y envío de datos. Estos formularios se encargan de enviar los datos del método especificado a una página gestionada por un servidor o manejador. Este tema explica y demuestra el uso de formularios HTML para recoger y enviar datos de entrada.

Sección 18.1: Envíos

El atributo Action

El atributo `action` define la acción que se realizará cuando se envíe el formulario, que normalmente lleva a un script que recoge la información enviada y trabaja con ella. Si lo dejas en blanco, lo enviará al mismo archivo.

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

El atributo Method

El atributo `method` se utiliza para definir el método HTTP del formulario, que puede ser GET o POST.

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

El método GET se utiliza sobre todo para obtener datos, por ejemplo, para recibir una entrada por su ID o nombre, o para enviar una consulta de búsqueda. El método GET añadirá los datos del formulario a la URL especificada en el atributo `action`.

```
www.ejemplo.com/accion.php?nombre=Mickey&apellido=Mouse
```

El método POST se utiliza cuando se envían datos a un script. El método POST no añade los datos del formulario a la URL de la acción, sino que los envía utilizando el cuerpo de la solicitud.

Para enviar correctamente los datos del formulario, debe especificarse un nombre de atributo `name`.

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

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

Más atributos

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

Sección 18.2: Atributo target en la etiqueta Form

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

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

De etiqueta con un atributo de destino:

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

Valores de los atributos

Valor	Descripción
<code>_blank</code>	La respuesta se muestra en una nueva ventana o pestaña
<code>_self</code>	La respuesta se muestra en el mismo marco (por defecto)
<code>_parent</code>	La respuesta se muestra en el marco principal
<code>_top</code>	La respuesta se muestra en el cuerpo completo de la ventana
<code>framename</code>	La respuesta se muestra en un <code>iframe</code>

Nota: El atributo `target` quedó **obsoleto** en **HTML 4.01**. El atributo `target` es **compatible** con **HTML5**.

Los marcos y los conjuntos de marcos no están soportados en **HTML5**, por lo que los valores `_parent`, `_top` y `framename` ahora se utilizan sobre todo con `iframes`.

Sección 18.3: Subir archivos

Las imágenes y los archivos pueden cargarse/enviarse al servidor estableciendo el atributo `enctype` de la etiqueta `form` como `multipart/form-data`. `enctype` especifica cómo se codificarán los datos del formulario al enviarse al servidor.

Ejemplo

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

Sección 18.4: Agrupar 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 puede hacerse utilizando la etiqueta. Aquí tienes un ejemplo de uso.

Para cada conjunto de campos, puede establecer una leyenda para el conjunto utilizando la etiqueta `<legend>`.

Ejemplo

```
<form>
  <fieldset>
    <legend>1er set de campo:</legend>
    Campo uno:<br>
    <input type="text"><br>
    Campo dos:<br>
    <input type="text"><br>
  </fieldset><br>
  <fieldset>
    <legend>2º set de campo:</legend>
    Campo tres:<br>
    <input type="text"><br>
    Campo cuatro:<br>
    <input type="text"><br>
  </fieldset><br>
  <input type="submit" value="Enviar">
</form>
```

Resultado

1er set de campo: _____

Campo uno:

Campo dos:

2º set de campo: _____

Campo tres:

Campo cuatro:

Compatibilidad con navegadores

Las últimas versiones de Chrome, IE, Edge, Firefox, Safari y Opera también admiten la etiqueta.

Capítulo 19: Elemento Div

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

Sección 19.1: Uso básico

El elemento `<div>` no suele tener un significado semántico específico por sí mismo, simplemente representa una división, y se suele utilizar para agrupar y encapsular otros elementos dentro de un documento HTML y separarlos de otros grupos de contenido. Como tal, cada `<div>` se describe mejor por su contenido.

```
<div>
  <p> ¡Hola! Esto es un párrafo. </p>
</div>
```

El elemento `div` es típicamente un elemento a 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 CSS por defecto:

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

El Consorcio World Wide Web (W3C) recomienda encarecidamente considerar el elemento `div` como un elemento de último recurso para cuando no hay otro elemento adecuado. El uso de elementos más apropiados en lugar del elemento `div` mejora la accesibilidad para los lectores y facilita el mantenimiento para los autores.

Por ejemplo, una entrada de blog se marcaría usando `<article>`, un capítulo usando `<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 con fines estilísticos o para envolver varios párrafos de una sección que deben ser anotados de forma similar.

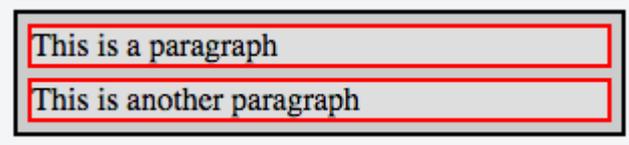
Sección 19.2: Anidación

Es una práctica común colocar varios `<div>` dentro de otro `<div>`. Esto suele denominarse "anidamiento" de elementos y permite dividir aún más los elementos en subsecciones o ayudar a los desarrolladores con el estilo CSS.

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

```
<div class="outer-div">
  <div class="inner-div">
    <p>Este es un párrafo</p>
  </div>
  <div class="inner-div">
    <p>Este es otro párrafo</p>
  </div>
</div>
```

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



Anidamiento de elementos en línea y de bloque

Al anidar elementos, debe tener en cuenta que existen elementos en línea y de bloque. Mientras que los elementos de bloque "añaden un salto de línea en el fondo", lo que significa que otros elementos anidados se muestran automáticamente en la línea siguiente, los elementos en línea pueden colocarse unos junto a otros por defecto.

Evitar la anidación profunda de `<div>`

Un diseño de contenedores anidados profundo y de uso frecuente muestra un mal estilo de codificación.

Las esquinas redondeadas o algunas funciones similares suelen crear un código HTML de este tipo. Para la mayoría de los navegadores existen equivalentes en CSS3. Intente utilizar la menor cantidad posible de elementos HTML para aumentar la relación contenido/etiqueta y reducir la carga de la página, lo que se traducirá en una mejor clasificación en los motores de búsqueda.

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

Capítulo 20: Seccionar elementos

Sección 20.1: Elemento Nav

El elemento `<nav>` está pensado principalmente para ser utilizado en secciones que contengan **bloques de navegación principales** para el sitio web, esto puede incluir enlaces a otras partes de la página web (*por ejemplo, anclajes para una tabla de contenidos*) o a otras páginas completamente.

Elementos en línea

Lo siguiente mostrará un conjunto de hipervínculos en línea.

```
<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>
```

Utilizar los elementos de la lista cuando sea necesario

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

Fíjese en el `role="navigation"`, más sobre esto más adelante.

```
<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

Los elementos `<footer>` pueden tener una lista de enlaces a otras partes del sitio (FAQ, T&C, etc.). El elemento de pie de página por sí solo es suficiente en este caso, no necesitas envolver más tus enlaces con un elemento `<nav>` en el `<footer>`.

```
<!-- el <nav> no es obligatorio en el <footer> -->
<footer>
  <nav>
    <a href="#">...</a>
  </nav>
</footer>
<!-- El pie de página es suficiente -->
<footer>
  <a href="#">...</a>
</footer>
```

Notas:

No se permiten descendientes del elemento `<main>` dentro de un `<nav>`.

Se aconseja añadir un `rol ARIA` `role="navigation"` al elemento `<nav>` para ayudar a los agentes de usuario que no soportan HTML5 y también para proporcionar más contexto a los que sí lo soportan.

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

Lectores de pantalla: (*software que permite a los usuarios ciegos o con problemas de visión navegar por el sitio*)

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

- Podría dar al elemento `<nav>` una mayor prioridad al renderizar la página
- Podría retrasar la representación del elemento
- Podría adaptar la página de una forma específica a las necesidades del usuario, *por ejemplo*: hacer que los enlaces de texto de los elementos `<nav>` sean más grandes para las personas con problemas de visión.

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

Sección 20.2: Elemento Article

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

- Cuando se anidan elementos de artículo, el contenido del nodo de artículo interior debe estar relacionado con el elemento de artículo exterior exterior.

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

```
<section>
  <!-- Cada entrada individual del blog es un <articulo> -->
  <article>
    <header>
      <h1>Entrada de blog </h1>
      <time datetime="2016-03-13">13 de marzo de 2016</time>
    </header>
    <p>El elemento artículo representa un artículo o documento independiente. </p>
    <p>El elemento sección representa una agrupación de contenidos. </p>
    <section>
      <h2>Comentarios <small>relacionados con "Entrada del blog" </small></h2>
      <!-- El comentario relacionado también es un artículo independiente -->
      <article id="user-comment-1">
        <p>¡Excelente! </p>
        <footer><p>...</p><time>...</time></footer>
      </article>
    </section>
  </article>
  <!-- ./repite: <article>-->
</section>
<!-- El contenido no relacionado con el blog o las entradas debe estar fuera de la sección. -->
<footer>
  <p>Este contenido no debe estar relacionado con el 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 `<article>` en favor del elemento `<main>`.

```
<article>
  <p>Esto no tiene sentido, este artículo no tiene `contexto` real. </p>
</article>
```

En su lugar, sustituya el artículo por un elemento `<main>` para indicar que se trata del contenido principal de esta página.

```
<main>
  <p> Soy el contenido principal, no necesito pertenecer a un artículo. </p>
</main>
```

Si utiliza otro elemento, asegúrese de especificar el rol ARIA `<main>` para una correcta interpretación y renderización a través de múltiples dispositivos y navegadores que no sean HTML5.

```
<section role="main">
  <p> Esta sección es el contenido principal de esta página. </p>
</section>
```

Notas:

No se permiten descendientes del elemento `<main>` dentro de un `<article>`

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

Sección 20.3: Elemento Main

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

- El elemento `<main>` sólo debe utilizarse como máximo una vez en cada página.
- El elemento `<main>` no debe incluirse como descendiente de un elemento `article`, `aside`, `footer`, `header` o `nav`.

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

```
<body>
  <header>
    <nav>...</nav>
  </header>
  <main>
    <h1> Entrada de blog individual </h1>
    <p> Una introducción para el puesto. </p>
    <article>
      <h2> Referencias </h2>
      <p>...</p>
    </article>
    <article>
      <h2> Comentarios </h2> ...
    </article>
  </main>
  <footer>...</footer>
</body>
```

La entrada del blog está contenida dentro del elemento `<main>` para indicar que se trata del contenido principal de esta página (y, por tanto, único en todo el sitio web).

Las etiquetas `<header>` y `<footer>` son *hermanas* del elemento `<main>`.

Notas:

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

Atributos de rol ARIA: `main` (por defecto), `presentation`

Se aconseja añadir un 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 soportan HTML5 y también para proporcionar más contexto a los que sí lo soportan.

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

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

Sección 20.4: Elemento Header

El elemento `<header>` representa el contenido introductorio para su ancestro más cercano contenido de sección o elemento raíz de sección. Un `<header>` suele contener un grupo de ayudas introductorias o de navegación.

Nota: El elemento de cabecera no está seccionando el contenido; no introduce una nueva sección.

Ejemplos:

```
<header>
  <p>Bienvenido a...</p>
  <h1>Voidwars!</h1>
</header>
```

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

```
<article>
  <header>
    <h1>Flexbox: La guía definitiva </h1>
  </header>
  <p>Se suponía que la guía sobre Flexbox iba a estar aquí, pero resultó que Wes tampoco era un experto en Flexbox. </p>
</article>
```

[Recomendación propuesta del W3C](#)

Sección 20.5: Elemento Footer

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

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

```
<footer>
  <p>Todos los derechos reservados</p>
</footer>
```

Sección 20.6: Elemento Section

El elemento `<section>` representa una sección genérica para agrupar temáticamente contenidos. Por lo general, cada sección debe poder identificarse con un elemento de encabezamiento como hijo de la sección.

- Puede utilizar el elemento `<section>` dentro de un `<article>` y viceversa.
- Cada sección debe tener un *tema* (un elemento de encabezamiento que identifique esta región)

- No utilice el elemento `<section>` como “contenedor” de estilo general. Si necesita un contenedor para aplicar estilos, utilice un `<div>` en su lugar.

En el siguiente ejemplo, mostramos una **única entrada de blog** con varios capítulos: cada capítulo es una sección (*un conjunto de contenidos agrupados temáticamente, que pueden identificarse por los elementos de encabezamiento de cada sección*).

```
<article>
  <header>
    <h2>Entrada de blog </h2>
  </header>
  <p>Una introducción a la entrada. </p>
  <section>
    <h3>Capítulo 1</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Capítulo 2</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Comentarios</h3> ...
  </section>
</article>
```

Notas:

Los desarrolladores deben utilizar el elemento `article` cuando tenga sentido syndicar el contenido del elemento.

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

Capítulo 21: Barras de navegación

Sección 21.1: 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="#">Inicio</a></li>
  <li><a href="#">Acerca de</a></li>
  <li><a href="#">Contacto</a></li>
</ul>
```

Sección 21.2: Barra de navegación HTML5

Para crear una barra de navegación utilizando el elemento `nav` de HTML5, encierra los enlaces dentro de la etiqueta `nav`.

```
<nav>
  <a href="#">Inicio</a>
  <a href="#">Acerca de</a>
  <a href="#">Contacto</a>
</nav>
```

Capítulo 22: Elemento Label

Atributos	Descripción
for	Referencia al elemento ID de destino. Por ejemplo: <code>for="apellido"</code>
form	HTML5, [Obsoleto] Referencia al formulario que contiene el elemento de destino. Se espera que los elementos Label dentro de un elemento <code><form></code> . Si se indica <code>form="algunFormId"</code> , podrá colocar la etiqueta en cualquier lugar del documento.

Sección 22.1: Acerca del elemento Label

El elemento `<label>` se utiliza para hacer referencia a un elemento de acción del formulario. En el ámbito de la **Interfaz de Usuario** se utiliza para facilitar el objetivo / selección de elementos como tipo `radio` o `checkbox`.

`<label>` como envoltorio

Puede incluir el elemento de acción deseado

```
<label>
  <input type="checkbox" name="Gatos">
  ¡Me gustan los gatos!
</label>
```

(Al hacer clic en el texto de la entrada de destino cambiará su estado / valor)

`<label>` como referencia

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

```
<input id="gatos" type="checkbox" name="Gatos">
<label for="gatos" >¡Me gustan los gatos!</label>
```

Nota

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

Sección 22.2: Uso básico

Formulario simple con etiquetas...

```
<form action="/inicio_sesion" method="POST">
  <label for="usuario">Usuario:</label>
  <input id="usuario" type="text" name="usuario" />
  <label for="contra">Contraseña:</label>
  <input id="contra" type="password" name="usuario" />
  <input type="submit" name="enviar" />
</form>
```

Version ≥ 5

```
<form id="mi-formulario" action="/inicio_sesion" method="POST">
  <label for="usuario" form="mi-formulario">Usuario:</label>
  <input id="usuario" type="text" name="usuario" />
  <label for="contra">Constraseña:</label>
  <input id="contra" type="password" name="contra" />
  <input type="submit" name="enviar" />
</form>
```

Capítulo 23. Elemento Output

Atributo	Descripción
Global	Atributos disponibles para cualquier elemento HTML5. Para una documentación completa de estos atributos: Atributos globales MDN
name	Cadena de caracteres que representa el nombre de una salida. Como elemento de formulario, la salida puede ser referenciada por su nombre utilizando la propiedad <code>document.forms</code> . Este atributo también se utiliza para recoger valores en el envío de un formulario.
for	Una lista separada por espacios de ids de elementos de formulario (por ejemplo, <code><inputs id="inp1"></code> para el valor es "inp1") que la debe mostrar cálculos.
form	Una cadena de caracteres que representa el <code><form></code> que se asocia a la salida. Si la salida está realmente fuera del <code><form></code> , este atributo se asegurará de que la salida sigue perteneciendo al <code><form></code> y sujeta a colecciones y envíos de dicho <code><form></code> .

Sección 23.1: Elemento Output mediante atributos for y form

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

Demostración en vivo

```
<!-- form1 recogerá los valores de in1 e in2 en el evento 'input'. -->
<!-- el valor out1 será la suma de los valores in1 e in2. -->
<form id="form1" name="form1" oninput="out1.value = parseInt(in1.value, 10) +
parseInt(in2.value, 10)">
  <fieldset>
    <legend>Ejemplo Output</legend>
    <input type="number" id="in1" name="in1" value="0">
    <br/>
    +
    <input type="number" id="in2" name="in2" value="0">
  </fieldset>
</form>
<!-- El atributo [for] permite que out1 muestre cálculos para in1 e in2.-->
<!-- El atributo [form] designa a form1 como propietario del formulario de out1 aunque no sea
descendiente.-->
<output name="out1" for="in1 in2" form="form1">0</output>
```

Sección 23.2: Elemento Output con atributos

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

Capítulo 24: Elementos vacíos

No todas las etiquetas HTML tienen la misma estructura. Mientras que la mayoría de los elementos requieren una etiqueta de apertura, una etiqueta de cierre y contenido, algunos elementos - conocidos como elementos vacíos - sólo requieren una etiqueta de apertura, ya que no contienen ningún elemento. Este tema explica y demuestra el uso correcto de los elementos vacíos en HTML.

Sección 24.1: Elementos vacíos

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

- `area` - área definida en una imagen sobre la que se puede hacer clic
- `base` - especifica una URL base a partir de la cual se basan todos los enlaces
- `br` - salto de línea
- `col` - columna en una tabla [obsoleto]
- `hr` - regla horizontal (línea)
- `img` - imagen
- `input` - campo donde los usuarios introducen datos
- `link` - enlaza un recurso externo al documento
- `meta` - proporciona información sobre el documento
- `param` - define parámetros para los plugins

Las normas HTML 5 incluyen todas las etiquetas no obsoletas 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 las fuentes multimedia para los elementos de imagen, audio y vídeo

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

```
<div>
  <a href="http://stackoverflow.com/">
    <h3>¡Haz clic aquí para visitar <i>Stack Overflow! </i></h3>
  </a>
  <button onclick="alert('¡Hola!');"> ¡Diga Hola! </button>
  <p> Mi lenguaje favorito es <b>HTML</b>. Estos son mis otros:<p>
  <ol>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>PHP</li>
  </ol>
</div>
```

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

```

<br>
<hr>
<input type="number" placeholder="Introduzca su número favorito">
```

A excepción de la etiqueta `img`, todos estos elementos vacíos sólo tienen una etiqueta de apertura. La etiqueta `img`, a diferencia de cualquier otra etiqueta, tiene un auto cierre / antes del signo mayor que de la etiqueta de apertura. Se recomienda dejar un espacio antes de la barra.

Capítulo 25: Elementos multimedia

Atributo	Detalles
width	Establece la anchura del elemento en píxeles.
height	Establece la altura del elemento en píxeles.
<source>	Define los recursos de los archivos de audio o vídeo
track	Define la pista de texto para los elementos multimedia
controls	Controles de visualización
autoplay	Iniciar automáticamente la reproducción multimedia
loop	Reproduce los medios en un ciclo repetido
muted	Reproduce el contenido multimedia sin sonido
poster	Asigna una imagen para mostrar hasta que se cargue un vídeo

Sección 25.1: 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">
Su navegador no soporta el elemento audio.
</audio>
```

Sección 25.2: Vídeo

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

```
<video width="500" height="700" controls>
  <source src="video.mp4" type="video/mp4">
Su navegador no soporta la etiqueta video.
</video>
```

Sección 25.3: Uso de los elementos `<video>` y `<audio>` para mostrar contenidos de audio/vídeo

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

Ejemplo de etiqueta `audio`:

```
<!-- Ejemplo de vídeo sencillo -->
<video src="videofile.webm" autoplay poster="posterimage.jpg">
Lo sentimos, tu navegador no admite vídeos incrustados, pero no te preocupes, puedes <a
href="videofile.webm">descargarlo </a> y ¡verlo con tu reproductor de vídeo favorito!
</video>
<!-- Vídeo con subtítulos -->
<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>
<!-- Ejemplo de vídeo sencillo -->
<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">
Tu navegador no soporta la etiqueta video HTML5.
</video>
```

Ejemplo de etiqueta `audio`:

```
<!-- Reproducción de audio sencilla -->
<audio src="http://developer.mozilla.org/@api/deki/files/2926/=AudioTest_(1).ogg" autoplay>
Su navegador no soporta el elemento <code>audio</code>. </audio>
<!-- Reproducción de audio con subtítulos -->
<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>
```

Sección 25.4: Cabecera o fondo de vídeo

Añadir un vídeo que se reproducirá automáticamente en bucle y no tiene controles ni sonido. Perfecto para una cabecera o fondo de vídeo.

```
<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 una alternativa si el vídeo no se puede cargar. Tenga en cuenta que es recomendable utilizar el primer fotograma del vídeo como póster `video.jpg`.

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

Capítulo 26: Elemento Progress

Parámetro	Valor
max	Cuánto trabajo requiere la tarea en total
value	Qué parte del trabajo ya se ha realizado
position	Este atributo devuelve la posición actual del elemento <code><progress></code> .
labels	Este atributo devuelve una lista de etiquetas del elemento <code><progress></code> (si las hay)

Sección 26.1: Progress

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 22%.

Sección 26.2: Cambiar el color de una barra de progreso

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

Este ejemplo da a 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 estilizar.

Chrome / Safari / Opera

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

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

Ahora podemos aplicar estilo al propio contenedor.

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

Firefox

Firefox aplica estilos diferentes a la barra de progreso. Tenemos que utilizar estos estilos.

```
progress[value] {  
    -moz-appearance: none;  
    appearance: none;  
    border: none; /* Firefox también muestra un borde */  
}
```

Internet Explorer

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

```
progress[value] {
  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none;
  border: none; /* Eliminar el borde de Firefox */
  width: 250px;
  height: 20px;
  color: blue;
}
```

Sección 26.3: HTML Fallback

Para los navegadores que no admiten el elemento `progress`, puede utilizar esta solución.

```
<progress max="100" value="20">
  <div class="progress-bar">
    <span style="width: 20%;">Progreso: 20%</span>
  </div>
</progress>
```

Los navegadores que admiten la etiqueta `progress` ignorarán el `div` anidado en su interior. Los navegadores heredados que no puedan identificar la etiqueta `progress` mostrarán el `div` en su lugar.

Capítulo 27. Controles del menú de selección

Sección 27.1: Seleccionar menú

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

```
<select name="">
  <option value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
</select>
```

Cambiar el tamaño

Puede cambiar el tamaño del menú de selección con el atributo `size`. Un tamaño de 0 o 1 muestra el menú desplegable estándar. Un tamaño superior a 1 convertirá el desplegable en un cuadro que mostrará ese número de líneas, con una opción por línea y una barra de desplazamiento para recorrer las opciones disponibles.

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

Menús de selección de varias opciones

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

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

Cuando se utiliza el atributo `multiple`, existe una diferencia entre utilizar 0 y 1 para el tamaño, mientras que no existe ninguna diferencia cuando no se utiliza el atributo. El uso de 0 hará que el navegador se comporte de la manera predeterminada que está programado. Si se utiliza 1, el tamaño de la caja resultante se fijará explícitamente en una sola fila de alto.

Sección 27.2: Opciones

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

```
<option>Alguna Opción</option>
```

Sin embargo, es importante tener en cuenta que el texto dentro del propio elemento `<option>` no siempre se utiliza, y esencialmente se convierte en el valor por defecto de los atributos que no se especifican.

Los atributos que controlan el aspecto real y la función de la opción son el `value` y `label`. El `label` representa el texto que se mostrará en el menú desplegable (lo que está viendo y sobre lo que hará clic para seleccionarlo). El `value` representa el texto que se enviará junto con el envío del formulario. Si se omite alguno de estos valores, se utilizará el texto del interior del elemento como valor. Así que el ejemplo que hemos dado antes podría "ampliarse" a esto:

```
<option label="Alguna Opcion" value="Alguna Opcion">
```

Nótese la omisión de la etiqueta dentro y al final del texto, que no son necesarias para construir realmente una opción dentro del menú. Si se incluyeran, el texto interior se ignoraría porque ambos atributos ya están especificados y no es necesario. Sin embargo, es probable que no haya mucha gente que los escriba así. La

forma más común de escribirlo es con un valor que se enviará al servidor, junto con el texto interior que finalmente se convierte en el atributo `label`, así:

```
<option value="option1">Alguna Opción</option>
```

Seleccionar una opción por defecto

También puede especificar una determinada opción para que se seleccione en el menú por defecto adjuntándole el atributo `selected`. Por defecto, si no se especifica ninguna opción como seleccionada en el menú, se seleccionará la primera opción del menú al renderizado. Si más de una opción tiene asociado el atributo `selected`, la última opción presente en el menú con el atributo será la seleccionada por defecto.

```
<option value="option1" selected>Alguna Opción</option>
```

Si utiliza el atributo en un menú de selección de varias opciones, se seleccionarán por defecto todas las opciones que tengan el atributo, y no se seleccionará ninguna si ninguna opción tiene el atributo.

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

Sección 27.3: Grupos de opciones

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

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

```
<select name="">
  <option value="leche">Leche</option>
  <optgroup label="Frutas">
    <option value="platano">Plátanos</option>
    <option value="fresa">Fresas</option>
  </optgroup>
  <optgroup label="Verduras" disabled>
    <option value="zanahoria">Zanahoria</option>
    <option value="calabacin">Calabacín</option>
  </optgroup>
</select>
```

Cuando se utilizan grupos de opciones, no es necesario que todas las opciones estén contenidas en un grupo. Además, la desactivación de un grupo de opciones desactivará todas las opciones del grupo, y no es posible volver a activar manualmente una sola opción de un grupo desactivado.

Sección 27.4: Lista de datos

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

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

Compatibilidad con navegadores

Chrome	Edge	Mozilla	Safari	Opera
20.0	10.0	4.0	No soportado	9.0

Capítulo 28: Embed

Parámetros

src
type
width
height

Detalles

Dirección del recurso
Tipo de recurso incrustado
Dimensión horizontal
Dimensión vertical

Sección 28.1: Uso básico

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

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

Sección 28.2: Definición del tipo MIME

El tipo MIME debe definirse mediante el atributo `type`.

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

Capítulo 29: IFrames

Atributo	Detalles
<code>name</code>	Establece el nombre del elemento, que se utilizará con una etiqueta <code><a></code> para cambiar el <code>src</code> del <code>iframe</code> .
<code>width</code>	Establece la anchura 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 soporte. El contenido debe ser HTML válido.
<code>sandbox</code>	Cuando se establece, el contenido del <code>iframe</code> se trata como si fuera de un único origen y se desactivan funciones como scripts, plugins, formularios y ventanas emergentes. Las restricciones se pueden relajar selectivamente añadiendo una lista de valores separados por espacios. Consulte la tabla en Observaciones para conocer los valores posibles.
<code>allowfullscreen</code>	Permitir o no que el contenido del <code>iframe</code> utilice <code>requestFullscreen()</code>

Sección 29.1: Conceptos básicos de un marco en línea

El término "IFrame" significa marco en línea. Se puede utilizar para incluir otra página en su página. Esto producirá un pequeño frame que muestra el contenido exacto de `base.html`.

```
<iframe src="base.html"></iframe>
```

Sección 29.2: Entorno aislado

A continuación, se incrusta una página web no fiable con todas las restricciones activadas.

```
<iframe sandbox src="http://ejemplo.com/"></iframe>
```

Para permitir que la página ejecute scripts y envíe formularios, añade `allow-scripts` y `allow-forms` al atributo `sandbox`.

```
<iframe sandbox="allow-scripts allow-forms" src="http://ejemplo.com/"></iframe>
```

Si hay contenido no fiable (como comentarios de usuarios) en el mismo dominio que la página web principal, se puede utilizar un `iframe` para desactivar los scripts y permitir al mismo tiempo que el documento principal interactúe con su contenido utilizando JavaScript.

```
<iframe sandbox="allow-same-origin allow-top-navigation"
src="http://ejemplo.com/sinconfiar/comentarios/pagina2">
```

El documento principal puede añadir escuchadores de eventos y redimensionar el `IFrame` para ajustar su contenido. Esto, junto con `allow-top-navigation`, puede hacer que el `iframe` aislado parezca formar parte del documento principal.

Este espacio aislado no sustituye a la limpieza de la entrada, pero puede utilizarse como parte de una estrategia de [defensa en profundidad](#).

También tenga en cuenta que esta caja de arena puede ser subvertida por un atacante convenciendo a un usuario para que visite la fuente del `iframe` directamente. La cabecera HTTP [Content Security Policy](#) puede utilizarse para mitigar este ataque.

Sección 29.3: Ajustar el tamaño del marco

El `IFrame` se puede redimensionar utilizando los atributos `width` y `height`, donde los valores se representan en píxeles (HTML 4.01 permitía valores porcentuales, pero HTML 5 sólo permite valores en píxeles CSS).

```
<iframe src="base.html" width="800" height="600"></iframe>
```

Sección 29.4: Uso del atributo "srcdoc"

El atributo `srcdoc` puede utilizarse (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 molan!".

```
<iframe srcdoc="<p> ¡Los IFrames molan! </p>"></iframe>
```

Si el navegador no admite el atributo `srcdoc`, el `Iframe` volverá a utilizar el atributo `src`, pero si los atributos `src` y `srcdoc` están presentes y son compatibles con el navegador, `srcdoc` tendrá prioridad.

```
<iframe srcdoc="<p> ¡Los IFrames molan! </p>" src="base.html"></iframe>
```

En el ejemplo anterior, si el navegador no admite el atributo `srcdoc`, mostrará en su lugar el contenido de la página `base.html`.

Sección 29.5: Uso de anclajes con IFrames

Normalmente, un cambio de página web dentro de un `Iframe` se inicia desde dentro del `Iframe`, por ejemplo, haciendo clic en un enlace dentro de del `Iframe`. Sin embargo, es posible cambiar el contenido de un `Iframe` desde fuera del mismo. Puede utilizar una etiqueta de anclaje cuyo atributo `href` sea la URL deseada y cuyo atributo `target` sea el atributo `name` del `iframe`.

```
<iframe src="paginaweb.html" name="miIframe"></iframe>  
<a href="paginaweb_diferente.html" target="miIframe"> Cambia el contenido del Iframe a  
paginaweb_diferente.html</a>
```

Capítulo 30: Lenguajes de contenido

Sección 30.1: Lenguaje del documento base

Es una buena práctica declarar el idioma principal del documento en el elemento `html`:

```
<html lang="es">
```

Si no se especifica ningún otro atributo `lang` en el documento, significa que todo (es decir, el contenido de los elementos y los valores de texto de los atributos) está en ese idioma.

Si el documento contiene partes en otros idiomas, estas partes deben obtener sus propios atributos `lang` para "sobrescribir" la declaración de idioma.

Sección 30.2: Lenguaje del elemento

El atributo `lang` se utiliza para especificar el idioma del contenido de los elementos y los valores de texto de los atributos:

```
<p lang="es">El contenido de este elemento está en español. </p>
<p lang="es" title="El valor de este atributo también está en español">El contenido de este
elemento está en español. </p>
```

La declaración de idioma se hereda:

```
<div lang="es">
  <p>Este elemento contiene contenidos en español. </p>
  <p title="Este atributo, también">Lo mismo ocurre con este elemento. </p>
</div>
```

Sección 30.3: Elementos con varios idiomas

Puede "sobrescribir" una declaración de idioma:

```
<p lang="es">Esta frase española contiene la palabra inglesa <span lang="en">Hello</span>. </p>
```

Sección 30.4: URL regionales

Es posible añadir el atributo `hreflang` a los elementos `a` y `area` que crean hipervínculos. Dicho atributo especifica el idioma del recurso enlazado. El idioma definido debe ser una etiqueta de idioma [BCP 47](#) [1] válida.

```
<p>
  <a href="example.org" hreflang="en">example.org</a> es uno de los dominios de ejemplo de
IANA.
</p>
```

1. ↑ Grupo de Trabajo en Red del IETF: RFC 5646 [Etiquetas para identificar Idiomas](#), IETF, septiembre de 2009

Sección 30.5: Manejo de atributos con lenguajes diferentes

Puede "sobrescribir" la declaración de idioma de un elemento padre 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>
```

Capítulo 31: SVG

SVG son las siglas de **S**calable **V**ector **G**raphics (gráficos vectoriales escalables). SVG se utiliza para definir gráficos para la Web.

El elemento HTML `<svg>` es un contenedor para gráficos SVG.

SVG dispone de varios métodos para dibujar trazados, cajas, círculos, texto e imágenes gráficas.

Sección 31.1: SVG en línea

SVG puede escribirse directamente en un documento HTML. El SVG en línea puede estilizarse y manipularse mediante CSS y JavaScript.

```
<body>
  <svg class="atencion" 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,-10t27,27Z" />
  </svg>
</body>
```

El SVG en línea anterior puede estilizarse utilizando la clase CSS correspondiente:

```
.atencion {
  fill: red;
  width: 50px;
  height: 50px;
}
```

El resultado tiene este aspecto:



Sección 31.2: Incrustación de archivos SVG externos en HTML

Puede utilizar los elementos `` u `<object>` para incrustar elementos SVG externos. Establecer la altura y la anchura es opcional, pero es muy recomendable.

Uso del elemento `img`

```

```

El uso de `` no permite aplicar estilo al SVG con CSS ni manipularlo con JavaScript.

Uso del elemento `object`

```
<object type="image/svg+xml" data="atencion.svg" width="50" height="50">
```

A diferencia de ``, `<object>` importa directamente el SVG en el documento y, por lo tanto, se puede manipular utilizando Javascript y CSS.

Sección 31.3: Incrustación de SVG mediante CSS

Puede añadir archivos SVG externos utilizando la propiedad `background-image`, igual que haría con cualquier otra imagen.

HTML

```
<div class="atencion"></div>
```

CSS

```
.attention {  
    background-image: url(atencion.svg);  
    background-size: 100% 100%;  
    width: 50px;  
    height: 50px;  
}
```

También puede incrustar la imagen directamente en un archivo css utilizando una url de datos:

```
background-image:  
url(data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsvg%22%20xmlns%3A%2F%2Fwww.w3.org%2F1999%2Fxmlns%3D%22http%3A%2F%2Fwww.w3.org%2F1999%2F%20viewBox%3D%220%200%20100%20100%22%20%3E%0D%0A%3Cpath%20id%3D%22attention%22%20d%3D%22m571%2C76710%2C-106q0%2C-8%2C-5%2C-13t-12%2C-51-108%2C0q-7%2C0%2C-12%2C5t-5%2C1310%2C106q0%2C8%2C5%2C13t12%2C61108%2C0q7%2C0%2C12%2C-6t5%2C-13Zm-1%2C-208110%2C-257q0%2C-6%2C-5%2C-10q-7%2C-6%2C-14%2C-61-122%2C0q-7%2C0%2C-14%2C6q-5%2C4%2C-5%2C1219%2C255q0%2C5%2C6%2C9t13%2C31103%2C0q8%2C0%2C13%2C-3t6%2C-9Zm-7%2C-5221428%2C786q20%2C35%2C-1%2C70q-10%2C17%2C-26%2C26t-35%2C101-858%2C0q-18%2C0%2C-35%2C-10t-26%2C-26q-21%2C-35%2C-1%2C-701429%2C-786q9%2C-17%2C26%2C-27t36%2C-10t36%2C10t27%2C27Z%22%20%2F%3E%0D%0A%3C%2Fsvg%3E);
```

Capítulo 32: Canvas

Atributo	Descripción
height	Especifica la altura del canvas
width	Especifica la anchura del canvas

Sección 32.1: Ejemplo básico

El elemento `canvas` se introdujo en HTML5 para dibujar gráficos.

```
<canvas id="miCanvas">  
  No se puede mostrar el gráfico. Canvas no es compatible con su navegador (IE<9)  
</canvas>
```

Lo anterior creará un elemento HTML `<canvas>` transparente de 300×150 px de tamaño.

Usted puede utilizar el elemento `canvas` para dibujar cosas increíbles como formas, gráficos, manipular imágenes, crear atractivos juegos, etc. con **JavaScript**.

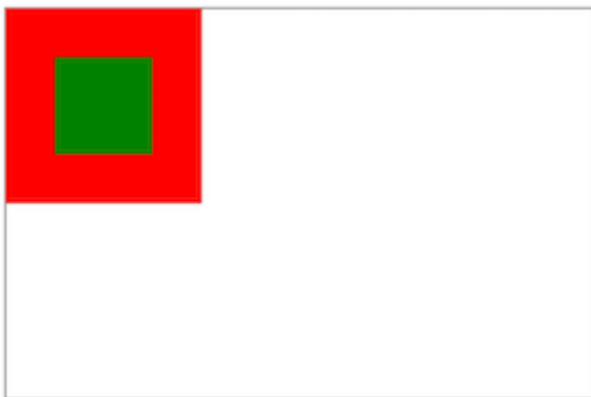
El Objeto de superficie de la *capa dibujable* 2D de `canvas` se denomina `CanvasRenderingContext2D`; o desde un `HTMLCanvasElement` utilizando el método `.getContext("2d")`:

```
var ctx = document.getElementById("miCanvas").getContext("2d");  
// ahora podemos hacer referencia al contexto de capa 2D del canvas utilizando `ctx`.  
ctx.fillStyle = "#f00";  
ctx.fillRect(0, 0, ctx.canvas.width, ctx.canvas.height); // x, y, anchura, altura  
ctx.fillStyle = "#000";  
ctx.fillText("Mi canvas rojo con texto en negro ", 24, 32); // texto, x, y
```

Sección 32.2: Dibujar dos rectángulos en un <canvas>

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Dibuja dos rectángulos en el canvas</title>
    <style>
      canvas {
        border:1px solid gray;
      }
    </style>
    <script async>
      window.onload = init; // llamar a init() una vez que la ventana está completamente
      cargada
      function init() {
        // #1 - obtener referencia al elemento <canvas>.
        var canvas = document.querySelector('canvas');
        // #2 - obtener referencia al contexto de dibujo y a la API de dibujo
        var ctx = canvas.getContext('2d');
        // #3 - todas las operaciones de relleno aparecen ahora en rojo
        ctx.fillStyle = 'red';
        // #4 - rellenar un rectángulo de 100x100 en x=0,y=0
        ctx.fillRect(0,0,100,100);
        // #5 - todas las operaciones de relleno aparecen ahora en verde
        ctx.fillStyle = 'green';
        // #6 - rellenar un rectángulo de 50x50 en x=25,y=25
        ctx.fillRect(25,25,50,50);
      }
    </script>
  </head>
  <body>
    <canvas width=300 height=200> Tu navegador no soporta canvas. </canvas>
  </body>
</html>
```

Este ejemplo tiene este aspecto:



Capítulo 33: Metadatos

Las metaetiquetas de los documentos HTML proporcionan información útil sobre el documento, como una descripción, palabras clave autor, fechas de modificación y unos 90 campos más.

Sección 33.1: 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 se trata de una aplicación Web, no debe utilizarse la metaetiqueta `application-name`.

author

Establece el autor de la página:

```
<meta name="author" content="Tu nombre">
```

Sólo se puede dar un nombre.

description

Establezca la descripción de la página:

```
<meta name="description" content="Descripción de la página">
```

La metaetiqueta de descripción puede ser utilizada por varios motores de búsqueda al indexar su página web con fines de búsqueda. Normalmente, la descripción contenida en la metaetiqueta es el breve resumen que aparece bajo el título principal de la página o sitio web en los resultados del motor de búsqueda. Google suele utilizar sólo 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. Sólo debe utilizarse para páginas en las que se genera automáticamente.

keywords

Establezca las palabras clave para los motores de búsqueda (separadas por comas):

```
<meta name="keywords" content="PalabraClave1, PalabraClave2">
```

La metaetiqueta de palabras clave es utilizada a veces 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 añadir demasiadas palabras, ya que la mayoría de los motores de búsqueda que utilizan esta etiqueta meta sólo indexarán las primeras 20 palabras. Asegúrese de poner en primer lugar las palabras clave más importantes.

Sección 33.2: Codificación de caracteres

El atributo `charset` especifica la codificación de caracteres del documento HTML y debe ser una codificación de caracteres válida (por ejemplo, `windows-1252`, `ISO-8859-2`, `Shift_JIS` y `UTF-8`). `UTF-8` (Unicode) es la más utilizada y debería utilizarse en cualquier proyecto nuevo.

Version = 5

```
<meta charset="UTF-8">
<meta charset="ISO-8859-1">
```

Todos los navegadores han reconocido siempre la forma `<meta charset>`, pero si por alguna razón necesita que su página sea HTML 4.01 válido, puede utilizar lo siguiente en su lugar:

```
<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 la [Norma de codificación](#), para ver todas las etiquetas de codificación de caracteres disponibles que reconocen los navegadores.

Sección 33.3: Robots

El atributo `robots`, admitido por varios de los principales motores de búsqueda, 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	Significado
<code>all</code>	Por defecto. Equivale a <code>index</code> , <code>follow</code> . Véase la nota siguiente.
<code>noindex</code>	No indexar la página en absoluto.
<code>nofollow</code>	No seguir los enlaces de esta página
<code>follow</code>	Puede seguir los enlaces de la página. Véase la nota siguiente.
<code>none</code>	Equivalente a <code>noindex</code> , <code>nofollow</code> .
<code>noarchive</code>	No ponga a disposición una versión en caché de esta página 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 metadatos de esta página del proyecto Open Directory 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</code> <code>[RFC-850 date/time]</code>	No mostrar esta página en los resultados de búsqueda después de la fecha/hora especificada. La fecha/hora de fecha/hora debe especificarse en el formato RFC 850 .

Nota: Definir explícitamente `index` y/o `follow`, aunque son valores válidos, no es necesario ya que casi todos los motores de búsqueda asumirán que pueden hacerlo si no se les impide explícitamente. De forma similar a cómo funciona el archivo `robots.txt`, los motores de búsqueda generalmente sólo buscan cosas que *no tienen permitido* hacer. Indicar sólo lo que un motor de búsqueda no puede hacer también evita indicar accidentalmente cosas opuestas (como `index`, ..., `noindex`) que no todos los motores de búsqueda tratarán de la misma manera.

Sección 33.4: Redes sociales

Open Graph es un estándar para metadatos que amplía la información normal contenida en la cabecera de un sitio web. Esto permite a sitios web como Facebook mostrar información más detallada y rica sobre un sitio web en un formato estructurado. Esta información se muestra automáticamente cuando los usuarios comparten en Facebook enlaces a sitios web que contienen metadatos OG. 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="sitio web">
<meta property="og:title" content="Título del contenido">
<meta property="og:image" content="https://example.com/image.jpg">
<meta property="og:description" content="Descripción">
<meta property="og:site_name" content="nombre del sitio">
<meta property="og:locale" content="es_ES">
<meta property="article:author" content="">
<!-- Facebook: https://developers.facebook.com/docs/sharing/webmasters#markup -->
<!-- Open Graph: http://ogp.me/ -->
```

- [Facebook Open Graph Markup](#)
- [Open Graph protocol](#)

Facebook / Instant Articles

```
<meta charset="utf-8">
<meta property="op:markup_version" content="v1.0">
<!-- La URL de la versión web de tu artículo -->
<link rel="canonical" href="http://example.com/article.html">
<!-- El estilo que se utilizará para este artículo -->
<meta property="fb:article_style" content="myarticlestyle">
```

- [Facebook Instant Articles: Creación de artículos](#)
- [Instant Articles: Formato de referencia](#)

Twitter utiliza su propio marcado para los metadatos. Estos metadatos se utilizan como información para controlar la forma en que los tweets se cuándo contienen un enlace al sitio.

Twitter

```
<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="Título del contenido">
<meta name="twitter:description" content="Descripción del contenido inferior a 200 caracteres">
<meta name="twitter:image" content="https://example.com/image.jpg">
```

- [Twitter Cards: Guía de introducción](#)

Google+ / Schema.org

```
<link href="https://plus.google.com/+YourPage" rel="publisher">
<meta itemprop="name" content="Título del contenido">
<meta itemprop="description" content="Descripción del contenido inferior a 200 caracteres">
<meta itemprop="image" content="https://example.com/image.jpg">
```

Sección 33.5: Control de la disposición móvil

Los sitios optimizados para móviles suelen utilizar la etiqueta `<meta name="viewport">` de la siguiente manera:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

El elemento `viewport` da instrucciones al navegador sobre cómo controlar las dimensiones y el escalado de la página en función del dispositivo que se esté utilizando. en función del dispositivo utilizado.

En el ejemplo anterior, `content="width=device-width"` significa que el navegador representará el ancho de la página a la anchura de su propia pantalla. Así que, si esa pantalla tiene `480px wide`, la ventana del navegador tendrá `480px wide.initialscale=1` representa que el zoom inicial (que es 1 en este caso, significa que no hace zoom).

A continuación, se indican los atributos que admite esta etiqueta:

Atributo	Descripción
<code>width</code>	La anchura de la ventana virtual del dispositivo. Valores1: <code>device-width</code> o la anchura real en píxeles, como <code>480</code> .
<code>height</code>	La altura de la ventana virtual del dispositivo. Valores2: <code>device-height</code> o la anchura real en píxeles, como <code>600</code> .
<code>initial-scale</code>	El zoom inicial cuando se carga la página. <code>1.0</code> no hace zoom.
<code>minimum-scale</code>	La cantidad mínima que el visitante puede ampliar en la página. <code>1.0</code> no hace zoom.
<code>maximum-scale</code>	La cantidad máxima que el visitante puede ampliar en la página. <code>1.0</code> no hace zoom.
<code>user-scalable</code>	Permite acercar y alejar el dispositivo. Los valores son <code>yes</code> o <code>no</code> . Si se establece en <code>no</code> , el usuario no podrá hacer zoom en la página web. El valor predeterminado es <code>sí</code> . La configuración del navegador puede ignorar esta regla.

Notas:

1 La propiedad `width` puede especificarse en *píxeles* (`width=600`) o mediante *device-width* (`width=device-width`), que representa la anchura física de la pantalla del dispositivo.

2 Del mismo modo, la propiedad `height` puede especificarse en *píxeles* (`height=600`) o mediante *device-height* (`height=device-height`), que representa la altura física de la pantalla del dispositivo.

Sección 33.6: Actualización automática

Para actualizar la página cada cinco segundos, añade este elemento meta en el elemento `head`:

```
<meta http-equiv="refresh" content="5">
```

¡ATENCIÓN! Aunque se trata de un comando válido, se recomienda no utilizarlo 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 al parte superior de la página. Si alguna información en la página necesita ser actualizada continuamente, hay formas mucho mejores de hacerlo refrescando sólo una parte de la página.

Sección 33.7: Reconocimiento de números de teléfono

Las plataformas móviles como iOS reconocen automáticamente los números de teléfono y los convierten en enlaces `tel:`. Aunque esta función es muy práctica, el sistema detecta a veces códigos ISBN y otros números como números de teléfono.

Para Safari móvil y algunos otros navegadores móviles basados en WebKit para desactivar el reconocimiento automático del número de teléfono y el formato, necesita esta metaetiqueta:

```
<meta name="format-detection" content="telephone=no">
```

Sección 33.8: Redirección automática

A veces su página web necesita una redirección automática.

Por ejemplo, para redirigir a ejemplo.com después de 5 segundos:

```
<meta http-equiv="refresh" content="5;url=https://www.example.com/" />
```

Esta línea le enviará a la página web designada (en este caso `example.com` después de 5 segundos).

Si necesita cambiar el tiempo de retardo antes de una redirección, simplemente cambiando el número justo antes de su `;url=` alterará el tiempo de retardo. el tiempo de retardo.

Sección 33.9: Web App

Puedes configurar tu aplicación web o sitio web para que se añada un icono de acceso directo a la aplicación en la pantalla de inicio de un dispositivo, y hacer que la aplicación se inicie en "modo aplicación" a pantalla completa mediante la opción de menú "Añadir a la pantalla de inicio" de Chrome para Android.

La(s) metaetiqueta(s) siguiente(s) abrirá(n) 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 definir el color de la barra de estado y la barra de direcciones en la metaetiqueta.

Android Chrome

```
<meta name="theme-color" content="black">
```

IOS

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

Capítulo 34: Marcado del código informático

Section 34.1: Bloque con `<pre>` y `<code>`

Si el formato (espacio en blanco, nuevas líneas, sangría) del código es importante, utilice el elemento `pre` en combinación con el elemento `code`:

```
<pre>
  <code>
    x = 42
    if x == 42:
      print "x is ... .. 42"
  </code>
</pre>
```

Todavía tiene que escapar caracteres con significado especial en HTML (como `<` con `<`), por lo que para mostrar un bloque de código HTML (`<p>Esto es un párrafo. </p>`), podría verse así:

```
<pre>
  <code>
    &lt;p> Esto es un párrafo. &lt;/p>
  </code>
</pre>
```

Section 34.2: En línea con `<code>`

Si una frase contiene código informático (por ejemplo, el nombre de un elemento HTML), utilice el elemento `code` para marcarla:

```
<p> El elemento <code>a</code> crea un hipervínculo. </p>
```

Capítulo 35: Marcar citas

Section 35.1: En línea con <q>

El **elemento q** puede utilizarse para una cita que forme parte de una frase:

```
<p>Escribió <q>La respuesta es 42. </q> y todos estuvieron de acuerdo. </p>
```

Comillas de las citas

Version ≤ 4.01

No deben añadirse comillas de las citas. Los agentes de usuario deberían (en HTML 4.01) renderizarlas automáticamente.

Version = 5

No deben añadirse comillas de las citas. Los agentes de usuario las mostrarán automáticamente.

URL de origen (atributo cite)

El **atributo cite** puede utilizarse para hacer referencia a la URL de origen citada:

```
<p>Escribió <q cite="http://example.com/blog/hello-world">La respuesta es 42. </q> y todos  
estuvieron de acuerdo. </p>
```

Tenga en cuenta que los navegadores no suelen mostrar esta URL, por lo que, si la fuente es relevante, debe añadir un hipervínculo (elemento a) además.

Sección 35.2: Bloque con <blockquote>

El **elemento blockquote** puede utilizarse para una cita (a nivel de bloque):

```
<blockquote>  
  <p>La respuesta es 42.</p>  
</blockquote>
```

URL de origen (atributo cite)

El atributo cite puede utilizarse para hacer referencia a la URL de origen citada:

```
<blockquote cite="http://example.com/blog/hello-world">  
  <p>La respuesta es 42.</p>  
</blockquote>
```

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

Cita/Atribución

Version ≤ 4.01

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

```
<blockquote cite="http://example.com/blog/hello-world">  
  <p>La respuesta es 42. </p>  
</blockquote>  
<p>Fuente: <cite><a href="http://example.com/blog/hello-world" rel="external">Hola  
Mundo</a></cite></p>
```

Puede añadir un elemento **div** para agrupar la cita y la atribución, pero no existe ninguna forma de asociarlas semánticamente.

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

La cita/atribución (por ejemplo, el hipervínculo que da la URL de la fuente) puede estar dentro de la `blockquote`, pero en ese caso debe estar dentro de un elemento `cite` (para atribuciones dentro del texto) o un elemento `footer`:

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>La respuesta es 42. </p>
<footer>
  <p>Fuente: <cite><a href="http://example.com/blog/hello-world" rel="external">Hola
    Mundo</a></cite></p>
</footer>
</blockquote>
```

El elemento `cite` puede utilizarse para la referencia de la fuente citada o para el nombre del autor de la cita.

Capítulo 36: Índice de tabulación

Valor	Significado
negative	será enfocable, pero no debe ser accesible a través de la navegación secuencial del teclado
0	será enfocable y alcanzable a través de la navegación secuencial del teclado, pero su orden relativo está definido por la convención de la plataforma
positive	debe ser enfocable y accesible mediante navegación secuencial por teclado; su orden relativo vendrá definido por el valor del atributo: el secuencial sigue el número creciente del <code>tabindex</code>

Sección 36.1: Añadir un elemento al orden de tabulación

```
<div tabindex="0">Some button</div>
```

Nota: Intente utilizar un botón HTML nativo o una etiqueta a cuando proceda.

Sección 36.2: 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 seguirá siendo enfocable.

Sección 36.3: 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 `button` y `a`) se añadirán después de los elementos con preferencia.

No se recomiendan los valores positivos, ya que alteran el comportamiento esperado del tabulador y pueden confundir a las personas que utilizan lectores de pantalla. Intente crear un orden natural reorganizando su estructura DOM.

Capítulo 37: Atributos globales

Atributo	Descripción
<code>class</code>	Define uno o más nombres de clase para un elemento. Vea Clases e ID.
<code>contenteditable</code>	Establece si se puede editar el contenido de un elemento.
<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 dentro de un elemento.
<code>draggable</code>	Establece si un elemento puede ser arrastrado.
<code>hidden</code>	Oculto un elemento que no se utiliza actualmente en la página.
<code>id</code>	Define un identificador único para un elemento. Vea Clases e identificadores.
<code>lang</code>	Define el idioma del contenido de un elemento y los valores de sus atributos de texto. Vea Contenido Idiomas.
<code>spellcheck</code>	Establece si se revisa la ortografía/gramática del 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 se navega por los elementos de una página mediante el atajo de teclado tabulador.
<code>title</code>	Define información adicional sobre un elemento, generalmente en forma de texto de ayuda al pasar el ratón sobre él. al pasar el ratón.
<code>translate</code>	Define si se traduce el contenido de un elemento.

Sección 37.1: Atributo `contenteditable`

```
<p contenteditable> Este es un párrafo editable. </p>
```

Al hacer clic en el párrafo, su contenido puede editarse de forma similar a un campo de entrada de texto.

Cuando el atributo `contenteditable` no está establecido en un elemento, éste lo heredará de su padre. Así que todo el texto hijo de un elemento editable de contenido también será editable, pero *puedes* desactivarlo para texto específico, así:

```
<p contenteditable>
  Este es un párrafo editable.
  <span contenteditable="false"> Pero esto no. </span>
</p>
```

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

Capítulo 38: Caché de HTML 5

Sección 38.1: Ejemplo básico de caché HTML5

Este es nuestro archivo index.html

```
<!DOCTYPE html>
<html manifest="index.appcache">
<body>
<p>Contenido</p>
</body>
</html>
```

Entonces crearemos el archivo index.appcache con los siguientes códigos.

```
CACHE MANIFEST
index.html
```

Escribir los archivos que desea que se almacenen en caché, cargará el index.html a continuación, activa un modo offline y volverá a cargar la pestaña.

Nota: En este ejemplo, los dos archivos deben estar en la misma carpeta.

Capítulo 39: Atributos de eventos HTML

Sección 39.1: 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
<code>onblur</code>	Se dispara en el momento en que el elemento pierde el foco.
<code>onchange</code>	Se dispara en el momento en que se cambia el valor del elemento
<code>oncontextmenu</code>	Script que se ejecuta cuando se activa un menú contextual
<code>onfocus</code>	Se dispara en el momento en que el elemento se centra
<code>oninput</code>	Script que se ejecuta cuando un elemento recibe una entrada del usuario
<code>oninvalid</code>	Script a ejecutar cuando un elemento no es válido
<code>onreset</code>	Se activa cuando se pulsa el botón Reiniciar de un formulario
<code>onsearch</code>	Se dispara cuando el usuario escribe algo en un campo de búsqueda (para <code><input=" search "></code>)
<code>onselect</code>	Se activa después de seleccionar un texto en un elemento
<code>onsubmit</code>	Se activa cuando se envía un formulario

Sección 39.2: Eventos de teclado

Atributo	Descripción
<code>onkeydown</code>	Se activa cuando un usuario pulsa una tecla
<code>onkeypress</code>	Se activa cuando un usuario pulsa una tecla
<code>onkeyup</code>	Se activa cuando un usuario suelta una tecla

Capítulo 40: Entidades de carácter

Sección 40.1: Entidades de carácter en HTML

Muchos símbolos y caracteres especiales son necesarios al desarrollar una página web en HTML, pero como sabemos que a veces el uso de caracteres directamente puede interferir con el código HTML real que tienen ciertos caracteres reservados y también ciertos caracteres que no están disponibles en el teclado. Así, para evitar el conflicto y al mismo tiempo poder utilizar diferentes símbolos en nuestro código w3.org nos proporciona 'Entidades de Carácter'.

Las Entidades de Carácter están predefinidas con 'Nombre de Entidad' - &entity_name; y 'Número de Entidad' - &entity_number; así que necesitamos usar cualquiera de los dos para que el símbolo requerido sea renderizado en nuestra página.

La lista de algunas Entidades de Carácter puede consultarse en <https://dev.w3.org/html5/html-author/charref>.

Un ejemplo sencillo con el uso de entidad de caracteres para 'lupa':

```
<input type="text" placeholder=" 🔍 Buscar" />
```

que se representa como

Sección 40.2: Caracteres especiales comunes

Algunos caracteres pueden estar reservados para HTML y no pueden utilizarse directamente, ya que podrían 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. Del mismo modo, es posible que los espacios en blanco escritos en el código fuente no se muestren como se espera en el código HTML de salida. Algunos, como 📞, no están disponibles en el juego de caracteres ASCII.

Para ello, se crean entidades de carácter. Tienen la forma &entity_name; o &entity_number;. En siguientes son algunas de las entidades HTML disponibles.

Carácter	Descripción	Nombre de entidad	Número de entidad
" "	espacio sin ruptura	 	
"<"	menos que	<	<
">"	mayor que	>	>
"&"	signo ampersand	&	&
" - "	barra baja	—	—
" _ "	guión	–	–
"©"	copyright	©	©
"®"	marca registrada	®	®
"™"	marca	™	™
" 📞 "	teléfono	☎	☎

Por lo tanto, para escribir

© 2016 Stack Exchange Inc.

se utiliza el siguiente código HTML:

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

Capítulo 41: ARIA

Sección 41.1: role="presentation"

Un elemento cuya semántica de rol nativa implícita no se mapeará a la API de accesibilidad.

```
<div style="float:left;"> Algunos contenidos a la izquierda. </div>
<div style="float:right;"> Algunos contenidos a la derecha. </div>
<div role="presentation" style="clear:both;"></div>
```

Sección 41.2: role="alert"

Un mensaje con información importante y, por lo general, urgente.

```
<div role="alert" aria-live="assertive"> Su sesión caducará en 60 segundos. </div>
```

Observa que se ha incluido tanto `role="alert"` como `aria-live="assertive"` al mismo tiempo. Son atributos sinónimos, pero algunos lectores de pantalla sólo admiten uno u otro. Por tanto, al utilizar ambos simultáneamente maximizamos las posibilidades de que funcione como se espera.

Sección 41.3: role="alertdialog"

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

```
<div role="alertdialog">
  <h1>Advertencia</h1>
  <div role="alert"> Su sesión caducará en 60 segundos. </div>
</div>
```

Sección 41.4: role="application"

Una región declarada como aplicación web, en contraposición a un documento web. En este ejemplo, la aplicación es una simple calculadora que puede sumar dos números.

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

Sección 41.5: role="article"

Sección de una página que consiste en una composición que forma parte independiente de un documento, página o sitio.

Establecer un rol ARIA y/o un atributo aria-* que coincida con la semántica ARIA implícita por defecto es innecesario y no se recomienda, ya que estas propiedades ya están establecidas por el navegador.

```
<article>
  <h1>Mi primer artículo</h1>
  <p>Lorem ipsum...</p>
</article>
```

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

```
<div role="article">
  <h1>Mi primer artículo</h1>
  <p>Lorem ipsum...</p>
</div>
```

Entrada W3C para [role=article](#)

Sección 41.6: role="banner"

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

```
<div role="banner">
  <h1>Mi sitio</h1>
  <ul>
    <li><a href="/">Inicio</a></li>
    <li><a href="/about">Acerca de</a></li>
    <li><a href="/contact">Contacto</a></li>
  </ul>
</div>
```

Sección 41.7: role="button"

Una entrada que permite acciones desencadenadas por el usuario cuando se hace clic en ella o se pulsa.

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

Sección 41.8: role="cell"

Una celda de un contenedor de tabla.

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

Sección 41.9: role="checkbox"

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

```
<p>
  <input type="checkbox" role="checkbox" aria-checked="false">
  Acepto los términos
</p>
```

Sección 41.10: role="columnheader"

Celda que contiene la información de cabecera de una columna.

```

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

```

Sección 41.11: role="combobox"

Una presentación de una selección; normalmente similar a un cuadro de texto donde los usuarios pueden teclear adelante para seleccionar una opción, o teclear para introducir un texto arbitrario como nuevo elemento de la lista.

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

Por lo general, se utiliza JavaScript para crear el resto de la funcionalidad de la selección de lista.

Sección 41.12: role="complementary"

Sección de apoyo del documento, diseñada para complementar el contenido principal en un nivel similar de la jerarquía DOM, pero que sigue teniendo sentido cuando se separa del contenido principal.

```

<div role="complementary">
  <h2>Más artículos</h2>
  <ul>
    <!-- etc -->
  </ul>
</div>

```

Sección 41.13: role="contentinfo"

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

```

<p role="contentinfo">
  Autor: Albert Einstein<br>
  Publicado en: 15 de agosto de 1940
</p>

```

Sección 41.14: role="definition"

La definición de un término o concepto.

```

<span role="term" aria-labelledby="def1">Amor</span>
<span id="def1" role="definition">un intenso sentimiento de profundo afecto. </span>

```

Sección 41.15: role="dialog"

Un cuadro de diálogo es una ventana de aplicación diseñada para interrumpir el proceso actual de una aplicación con el fin de solicitar al usuario que introduzca información o que dé una respuesta.

```

<div role="dialog">
  <p>¿Estás seguro? </p>
  <button role="button">Si</button>
  <button role="button">No</button>
</div>

```

Sección 41.16: role="directory"

Una lista de referencias a los miembros de un grupo, como un índice estático.

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

Sección 41.17: role="document"

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

```
<div role="document">
  <h1>La vida de Albert Einstein </h1>
  <p>Lorem ipsum...</p>
</div>
```

Sección 41.18: role="form"

Región de referencia que contiene una colección de elementos y objetos que, en su conjunto, se combinan para crear una forma.

Utilizar el elemento HTML semánticamente correcto `<form>` implica semántica ARIA por defecto, lo que significa que `role=form` no es necesario ya que no se debe aplicar un rol de contraste a un elemento que ya es semántico, ya que añadir un rol anula la semántica nativa de un elemento.

Establecer un rol ARIA y/o un atributo aria-* que coincida con la semántica ARIA implícita por defecto es innecesario y no se recomienda, ya que estas propiedades ya están establecidas por el navegador.

```
<form action="">
  <fieldset>
    <legend>Formulario de acceso </legend>
    <div>
      <label for="username">Tu usuario</label>
      <input type="text" id="username" aria-describedby="username-tip" required />
      <div role="tooltip" id="username-tip">Tu nombre de usuario es tu dirección de
      correo electrónico </div>
    </div>
    <div>
      <label for="password">Tu contraseña</label>
      <input type="text" id="password" aria-describedby="password-tip" required />
      <div role="tooltip" id="password-tip">Se le envió por correo electrónico cuando
      se inscribió </div>
    </div>
  </fieldset>
</form>
```

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

```
<div role=form>
  <input type="email" placeholder="Tu dirección de correo electrónico">
  <button>Registrarse</button>
</div>
```

Sección 41.19: role="grid"

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

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

Sección 41.20: role="gridcell"

Celda de 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>
```

Sección 41.21: role="group"

Conjunto de objetos de la interfaz de usuario que no están destinados a ser incluidos en un resumen de página o índice por tecnologías de asistencia.

```
<div role="group">
  <button role="button">Anterior</button>
  <button role="button">Siguiente</button>
</div>
```

Sección 41.22: role="heading"

Título de una sección de la página.

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

Sección 41.23: role="img"

Contenedor de una colección de elementos que forman una imagen.

```
<figure role="img">
  
  <figcaption>Este es mi gato, Albert. </figcaption>
</figure>
```

Sección 41.24: role="link"

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

Establecer un rol ARIA y/o un atributo aria-* que coincida con la [semántica ARIA implícita por defecto](#) es innecesario y no se recomienda, ya que estas propiedades ya están establecidas por el navegador.

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

Sección 41.25: role="list"

Un grupo de elementos de lista no interactivos.

```
<ul role="list">
  <li role="listitem">Uno</li>
  <li role="listitem">Dos</li>
  <li role="listitem">Tres</li>
</ul>
```

Sección 41.26: role="listbox"

Widget que permite al usuario seleccionar uno o varios elementos de una lista de opciones.

```
<ul role="listbox">
  <li>Uno</li>
  <li>Dos</li>
  <li>Tres</li>
</ul>
```

Normalmente, se utiliza JavaScript para crear la función de selección múltiple.

Sección 41.27: role="listitem"

Un único elemento de una lista o directorio.

```
<ul role="list">
  <li role="listitem">Uno</li>
  <li role="listitem">Dos</li>
  <li role="listitem">Tres</li>
</ul>
```

Sección 41.28: role="log"

Un tipo de región viva en la que la información nueva se añade en un orden significativo y la antigua puede desaparecer.

```
<ul role="log">
  <li>Usuario 1 conectado. </li>
  <li>Usuario 2 conectado. </li>
  <li>Usuario 3 conectado. </li>
</ul>
```

Sección 41.29: role="main"

Contenido principal de un documento.

```
<!-- encabezado y navegación aquí -->
<div role="main">
  <p>Lorem ipsum...</p>
</div>
<!-- pie de página aquí -->
```

Sección 41.30: role="marquee"

Un tipo de región viva en la que 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>
```

Sección 41.31: role="math"

Contenido que representa una expresión matemática.

```

```

Sección 41.32: role="menu"

Tipo de widget que ofrece una lista de opciones al usuario.

```
<ul role="menu">
  <li role="menuitem">Nuevo</li>
  <li role="menuitem">Abrir</li>
  <li role="menuitem">Guardar</li>
  <li role="menuitem">Cerrar</li>
</ul>
```

Sección 41.33: role="menubar"

Presentación de menú que suele permanecer visible y se suele presentar horizontalmente.

```
<ul role="menubar">
  <li role="menuitem">Archivo</li>
  <li role="menuitem">Editar</li>
  <li role="menuitem">Ver</li>
  <li role="menuitem">Ayuda</li>
</ul>
```

Sección 41.34: role="menuitem"

Opción de un grupo de opciones contenidas en un menú o barra de menús.

```
<ul role="menubar">
  <li role="menuitem">Archivo</li>
  <li role="menuitem">Editar</li>
  <li role="menuitem">Ver</li>
  <li role="menuitem">Ayuda</li>
</ul>
```

Sección 41.35: role="menuitemcheckbox"

Un elemento de menú comprobable que tiene tres valores posibles: verdadero, falso o mixto.

```
<ul role="menu">
  <li role="menuitem">Consola</li>
  <li role="menuitem">Diseño</li>
  <li role="menuitemcheckbox" aria-checked="true">Ajuste de palabras</li>
</ul>
```

Sección 41.36: role="menuitemradio"

Elemento de menú que se puede marcar en un grupo de elementos de menú con funciones de radio, de los cuales sólo se puede marcar uno a la vez.

```
<ul role="menu">
  <li role="menuitemradio" aria-checked="true">Izquierda</li>
  <li role="menuitemradio" aria-checked="false">Centro</li>
  <li role="menuitemradio" aria-checked="false">Derecho</li>
</ul>
```

Sección 41.37: role="navigation"

Conjunto de elementos de navegación (normalmente enlaces) para navegar por el documento o por documentos relacionados.

```
<ul role="navigation">
  <li><a href="/">Inicio</a></li>
  <li><a href="/about">Acerca de</a></li>
  <li><a href="/contact">Contacto</a></li>
</ul>
```

Sección 41.38: role="note"

Sección cuyo contenido es parentético o accesorio al contenido principal del recurso.

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

Sección 41.39: role="option"

Elemento seleccionable en una lista de selección.

```
<ul role="listbox">
  <li role="option">Opción 1</li>
  <li role="option">Opción 2</li>
  <li role="option">Opción 3</li>
</ul>
```

Sección 41.40: role="progressbar"

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

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

Sección 41.41: role="radio"

Una entrada comprobable en un grupo de roles de radio, sólo uno de los cuales puede ser comprobado a la vez.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> Uno<br>
  <input role="radio" type="radio" aria-checked="false"> Dos<br>
  <input role="radio" type="radio" aria-checked="false"> Tres
</div>
```

Sección 41.42: role="region"

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

```
<div role="region">
  Equipo local: 4<br>
  Equipo visitante: 2
</div>
```

Sección 41.43: role="radiogroup"

Un grupo de botones de radio.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> Uno<br>
  <input role="radio" type="radio" aria-checked="false"> Dos<br>
  <input role="radio" type="radio" aria-checked="false"> Tres
</div>
```

Sección 41.44: role="row"

Una fila de celdas en un contenedor de tabla.

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

Sección 41.45: role="rowgroup"

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

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

Sección 41.46: role="rowheader"

Celda que contiene información de cabecera para una fila de una cuadrícula.

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

Sección 41.47: role="scrollbar"

Un objeto gráfico que controla el desplazamiento del contenido dentro de un área de visualización, independientemente de si el contenido está 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>
```

Sección 41.48: role="search"

Región de referencia que contiene una colección de elementos y objetos que, en su conjunto, se combinan para crear una búsqueda.

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

Sección 41.49: role="searchbox"

Tipo de cuadro de texto destinado a especificar criterios de búsqueda.

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

Sección 41.50: role="separator"

Divisor que separa y distingue secciones de contenido o grupos de elementos de menú.

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

Sección 41.51: role="slider"

Entrada en la que el usuario selecciona un valor dentro de un intervalo determinado.

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

Sección 41.52: role="spinbutton"

Una forma de gama 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">
```

Sección 41.53: role="status"

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

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

Sección 41.54: role="switch"

Tipo de casilla de verificación que representa valores de activado/desactivado, en contraposición a valores de marcado/desmarcado.

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

Sección 41.55: role="tab"

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

```
<ul role="tablist">
  <li role="tab">Introducción</li>
  <li role="tab">Capítulo 1</li>
  <li role="tab">Capítulo 2</li>
</ul>
```

Sección 41.56: role="table"

Sección que contiene datos organizados en filas y columnas. El rol de tabla está pensado para contenedores de tabla que son no interactivos.

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

Sección 41.57: role="tablist"

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

```
<ul role="tablist">
  <li role="tab">Introducción</li>
  <li role="tab">Capítulo 1</li>
  <li role="tab">Capítulo 2</li>
</ul>
```

Sección 41.58: role="tabpanel"

Un contenedor para los recursos asociados a una pestaña, donde cada pestaña está contenida en una lista de pestañas.

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

Sección 41.59: role="textbox"

Entrada que permite texto libre como valor.

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

Sección 41.60: role="timer"

Tipo de región en directo que contiene un contador numérico que indica la cantidad de tiempo transcurrido desde un punto inicial, o el tiempo restante hasta un punto final.

```
<p>
  <span role="timer">60</span> segundos restantes.
</p>
```

Sección 41.61: role="toolbar"

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

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

Sección 41.62: role="tooltip"

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

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

Normalmente, la información sobre herramientas estaría oculta. Utilizando JavaScript, la información sobre herramientas se mostraría tras un retardo cuando el usuario sobre el elemento que describe.

Sección 41.63: role="tree"

Tipo de lista que puede contener grupos anidados de subniveles que pueden contraerse y expandirse.

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

Sección 41.64: role="treegrid"

Una cuadrícula cuyas filas pueden expandirse y contraerse del mismo modo que un árbol.

Sección 41.65: role="treeitem"

Un elemento de opción de un árbol. Es un elemento dentro de un árbol que puede expandirse o colapsarse si contiene un subnivel de elementos del árbol.

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

Créditos

Muchas gracias a todas las personas de Stack Overflow Documentation que ayudaron a proporcionar este contenido, más cambios pueden ser enviados a web@petercv.com para que el nuevo contenido sea publicado o actualizado.

Traductor al español

[rortegag](#)

AA2992	Capítulo 12
Abhishek Pandey	Capítulos 17 y 33
Abrar Jahin	Capítulo 4
Adjit	Capítulo 29
ahmednawazbutt	Capítulo 1
Al.G.	Capítulos 2, 6 y 17
albert	Capítulo 8
Alex	Capítulo 15
Alexander Wigmore	Capítulo 33
Alexandre N.	Capítulos 8, 13, 15, 28 y 29
Ali Almoullim	Capítulos 18 y 27
Amanda Ahn	Capítulo 1
amflare	Capítulos 1 y 27
Amitay Stern	Capítulos 1 y 17
andreaem	Capítulo 13
andreas	Capítulo 31
Andrew Brooke	Capítulo 20
Angelos Chalaris	Capítulos 2, 10, 17 y 33
Ani Menon	Capítulos 2, 3, 9, 17, 18 y 33
Anil	Capítulo 20
animuson	Capítulos 1, 2, 5, 6, 7, 8, 9, 15, 17, 18, 27, 29, 33, 37 y 40
Anselm	Capítulos 6, 12 y 13
Anthony Pham	Capítulo 6
Aown Muhammad	Capítulo 18
Araknid	Capítulo 19
Aravind Suresh	Capítulo 12
Ashwin Ramaswami	Capítulo 9
Axe	Capítulo 33
Bálint	Capítulo 33
ban17	Capítulo 6
bdkopen	Capítulos 9, 14 y 33
Becca	Capítulo 37
Ben Rhys	Capítulo 5
bhansa	Capítulo 17
Bhavya Singh	Capítulo 41
Bookeater	Capítulo 33
Boris	Capítulo 33
Boysenb3rry	Capítulo 15
brandaemon	Capítulo 10
Caleb Kleveter	Capítulos 10 y 15
Callan Heard	Capítulos 1 y 12
Cedric Zoppolo	Capítulo 8
Charles	Capítulo 13
Chris	Capítulos 2 y 19
Chris Rutherford	Capítulo 12

ChrisD	Capítulos 1, 9, 24, 29 y 31
Christian Ternus	Capítulo 14
Christophe Strobbe	Capítulo 17
cone56	Capítulos 12 y 32
Content Solutions	Capítulos 2, 19, 26, 35 y 36
Cullub	Capítulo 18
CHUBByNINJA	Capítulo 17
D M	Capítulo 19
Daniel	Capítulo 17
Daniel Käfer	Capítulo 7
Dave Everitt	Capítulo 1
DawnPaladin	Capítulos 6, 12 y 17
Dipen Shah	Capítulo 17
dippas	Capítulo 3
Domenic	Capítulos 12 y 33
Druzion	Capítulo 17
Emanuel Vintilă	Capítulo 29
Emil	Capítulos 5, 6 y 10
Evan	Capítulo 3
Farhad	Capítulo 38
feeela	Capítulos 12 y 25
FelipeAls	Capítulo 30
FlyingPiMonster	Capítulo 6
gabe3886	Capítulo 17
Gabriel Chi Hong Lee	Capítulos 15 y 18
Gal Ratzkin	Capítulo 19
geek1011	Capítulos 1 y 33
geeksal	Capítulo 17
Gerold Broser	Capítulos 1 y 30
GoatsWearHats	Capítulo 10
Grace Note	Capítulo 7
Grant Palin	Capítulo 8
gustavohenke	Capítulo 5
Gytis Tenovimas	Capítulo 17
H Mirza	Capítulo 1
H. Pauwelyn	Capítulo 1
Habel Philip	Capítulo 33
Hafidz Ilham Aji Permana	Capítulo 33
Hanif Formoly	Capítulo 20
heerfk	Capítulo 2
Henrique Barcelos	Capítulos 12 y 17
Huy Nguyen	Capítulo 17
Infuzion	Capítulos 3 y 12
insertusernamehere	Capítulo 6
intboolstring	Capítulo 1
Isak Combrinck	Capítulos 25 y 30
ivn	Capítulo 15
JF	Capítulos 5 y 8
j08691	Capítulos 8 y 18
jhnance	Capítulo 10
jhoanna	Capítulo 33
JHS	Capítulos 6, 8, 9, 12 y 19
jkdev	Capítulo 9
joe_young	Capítulos 1, 3 y 8
John Slegers	Capítulos 10 y 17
Jojodmo	Capítulo 6

Jon Ericson	Capítulo 7
Jonathan Lam	Capítulos 1, 6, 12 y 16
Jones Joseph	Capítulos 6 y 14
Kake_Fisk	Capítulo 15
kcpike	Capítulo 7
kelvinlove	Capítulos 1 y 33
Kimmax	Capítulo 18
Lahiru Ashan	Capítulo 17
Leonidas Menendez	Capítulo 1
Luca langella	Capítulo 18
Luca Putzu	Capítulo 10
Marjorie Pickard	Capítulo 13
Marvin	Capítulos 7 y 17
Matas Vaitkevicius	Capítulos 6, 7, 12 y 17
Matt	Capítulo 1
Maximillian Laumeister	Capítulos 15 y 17
MervS	Capítulos 13 y 40
Michael Moriarty	Capítulo 33
Michael_B	Capítulo 10
mnoronha	Capítulo 1
Mohd Samir Khan	Capítulo 15
morewry	Capítulo 17
Mosh Feu	Capítulo 27
Mottie	Capítulo 8
Mr Lister	Capítulo 8
Mr. Alien	Capítulo 33
Muntasir	Capítulo 30
MySpeed	Capítulo 19
m_callens	Capítulos 22 y 33
nalply	Capítulo 20
Natalie	Capítulo 10
Nathan Tuggy	Capítulo 6
Nhan	Capítulos 12 y 15
Niek Brouwer	Capítulo 18
Nijin22	Capítulo 8
Nil Llisterri	Capítulo 17
Nishchay	Capítulos 1 y 33
NoobCoder	Capítulo 17
Ojen	Capítulo 29
Ortomala Lokni	Capítulos 17 y 33
Paresh Maghodiya	Capítulo 39
Paul Sweatte	Capítulo 41
Persijn	Capítulo 1
Peter L.	Capítulo 5
Pi Programs	Capítulo 17
pinjasaur	Capítulo 2
platy11	Capítulo 7
Pranav	Capítulo 6
Prateek	Capítulos 6, 7, 8 y 17
PrAtik Lochawala	Capítulo 8
Praveen Kumar	Capítulo 8
Psaniko	Capítulos 17 y 36
Pseudonym Patel	Capítulos 3 y 7
Racil Hilan	Capítulo 11
rajarshig	Capítulo 11
RamenChef	Capítulos 16 y 17

Ranjit Singh	Capítulo 17
Raystafarian	Capítulos 5 y 6
Richard Hamilton	Capítulos 7, 10, 26 y 32
Robert Columbia	Capítulo 6
Roko C. Buljan	Capítulos 13, 17, 22 y 32
Safoor Safdar	Capítulo 33
sasha	Capítulo 3
SeinopSys	Capítulo 17
Senjuti Mahapatra	Capítulo 33
Shannon Young	Capítulos 20 y 41
Sharavnan Kv	Capítulo 17
Shiva	Capítulo 25
Shivangi Chaurasia	Capítulos 15 y 17
Sildoreth	Capítulos 8 y 13
Simone Carletti	Capítulo 10
SJDS	Capítulo 17
stack	Capítulo 40
Stephen Leppik	Capítulos 17 y 33
Stewartside	Capítulos 7 y 17
Steyn van Esveld	Capítulo 6
Sumner Evans	Capítulo 2
Sunny R Gupta	Capítulos 1 y 17
SuperStormer	Capítulos 13, 19 y 34
sv3k	Capítulo 17
svarog	Capítulo 8
Ted Goas	Capítulos 8 y 33
the12	Capítulos 1 y 17
think123	Capítulo 17
Thomas Gerot	Capítulo 24
Thomas Landauer	Capítulo 8
thouusten	Capítulo 14
Timon	Capítulo 17
Timothy Miller	Capítulo 31
tmg	Capítulos 2, 7 y 17
Tom Johnson	Capítulo 7
tonethar	Capítulo 32
Tot Zam	Capítulos 17 y 27
Travis	Capítulo 41
Trevor Clarke	Capítulo 32
trungk18	Capítulo 17
Tyler Zika	Capítulo 10
Ulrich Schwarz	Capítulo 9
Undo	Capítulo 17
unor	Capítulos 1, 5, 6, 7, 12, 30, 33, 34, 35 y 37
user3130333	Capítulo 1
user5389107	Capítulo 17
V4karian	Capítulo 12
Valor Naram	Capítulos 1, 4 y 38
vkopio	Capítulo 3
vladdobra	Capítulos 12 y 17
w5m	Capítulo 31
Wolfgang	Capítulos 6, 9 y 15
xims	Capítulo 33
Yasir T	Capítulo 11
Yossi Aharon	Capítulos 20 y 25
Zack	Capítulo 7

[Zange](#)
[Zaz](#)
[zer00ne](#)
[zygimantus](#)
[Zze](#)
[zzzzBov](#)

Capítulo 37
Capítulo 1
Capítulos 8 y 23
Capítulo 1
Capítulo 1
Capítulo 17