

Manual de HTML 5

Teoría

Actualizado: Abril 2024

carontestudio.com

j.rivas@carontestudio.com

tfno: 625 40 91 95



ÍNDICE

1- Introducción	2
1.1 Qué es HTML	2
1.2. W3C y las especificaciones oficiales	3
1.3 HTML y CSS	4
2. Características elementales	5
2.2 Anatomía de HTML	5
2.3 Etiquetas y atributos	6
2.3.1 Atributos básicos	9
2.3.2 Atributos de eventos	9
2.3.3 Atributos de foco	10
2.3.4 Atributos de internacionalización	10
2.4. Elementos HTML	11
3. Etiquetas texto	13
3.1. Estructuración jerárquica del texto: texto corrido y secciones.	13
3.2. Espacios en blanco, nuevas líneas y texto preformateado	13
3.3. Marcado básico de texto	14
3.4. Marcado avanzado de texto	15
3.5. Ejercicio 1	16
3.6. Codificación de caracteres	16
4. Enlaces	16
4.1. Enlaces y URL	16
4.2. Enlaces "Script" y "Link"	20
4.3. Enlaces útiles y usos especiales	22
4.4. Ejercicio 2	22
5. Listas	23
5.1. Listas no ordenadas	23
5.2. Listas ordenadas	23
5.3. Listas de definición	24
5.4. Ejercicio 3	25
6. Imágenes y objetos	25
6.1. Imágenes	25
6.2. Mapas de imagen	26
6.3. Objetos	28
6.4. Ejercicio 4	29
7. Tablas	29
7.1. Uso correcto de las tablas	29
7.2. Tablas básicas	29

7.3. Tablas avanzadas	31
7.4. Ejercicio 5	32
8. Formularios	32
8.1. Formularios básicos	32
8.2. Elementos de los formularios	33
8.3. Desplegables y otros elementos de los formularios	36
8.5. Ejercicio 6	39
9. Estructurando un documento HTML	39
9.1 Contenedores	39
10. Metainformación	42
10.1 Estructura de la cabecera	42
10.2. Metadatos	42
10.2. Ejercicio 7	43
11. Otras etiquetas	43
11.1 Anotaciones	43
11.2. CSS "Id" y "class"	44
11.3. Iframes	45
12. Accesibilidad	46
12.1. Consejos para mejorar la accesibilidad	1
13. Validación	48
13.1. Importancia de la validación.	48
13.2. Validador del W3C	48

1- Introducción

1.1 Qué es HTML

En 1980 el CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos. El "hipertexto" permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaban visualizando por medio de enlaces similares a URLs. La idea se popularizó y dio nacimiento a la WorldWideWeb (W3).

El primer documento de HTML se publicó en 1991 HTML Tags (Etiquetas HTML) y la primera propuesta oficial para estandarizar HTML fue en 1993 por parte del organismo IETF (Internet Engineering Task Force).

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar el estándar HTML 2.0. aunque realmente fuera el primer estándar oficial de HTML.

De 1996 en adelante, los estándares de HTML los publica otro organismo de estandarización llamado W3C (World Wide Web Consortium) y su primera versión fue la 3.2 en 1997 con soporte para Java y texto que fluído alrededor de las imágenes.

En 1998 vio la luz HTML 4.0 que daba soporte a hojas de estilo CSS y scripts varios.

En 2004, Apple, Mozilla y Opera ante la aparente falta de interés del W3C en HTML decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group), centrados en HTML 5 y en 2007 el W3C continuó con HTML ayudando a estandarizar una versión final lanzada en 2008, vigente hasta hoy día.

1.2. W3C y las especificaciones oficiales

Es por tanto el organismo W3C el principal promotor de la estandarización de HTML y quien predica los cánones del lenguaje web de manera internacional, normas oficiales y gratuitas que podemos y debemos consultar en su web de manera periódica, puesto que como cualquier otro lenguaje, HTML requiere de una constante revisión ya que evoluciona en función de las necesidades de los usuarios web, dejando así obsoletas ciertas etiquetas, estructuras y/o herramientas, desaconsejando su uso.

Podemos mantenernos al día de estas directrices en el sitio web oficial del W3C:
<http://www.w3.org>

1.3 HTML y CSS

Como ya hemos comentado, originalmente las páginas en HTML eran extremadamente básicas y solo contenían texto, pero con el desarrollo y evolución de HTML este contenido fue ganando en complejidad, adquiriendo propiedades estéticas como colores, tamaño de fuente, tipos variados, formas e imágenes personalizables.

Con la compatibilidad con scripts como los de Java o flash, comenzaron a desarrollarse webs dinámicas con movimiento y efectos especiales. Dichas posibilidades hacían cada vez más difícil el mantenimiento regular de las páginas que ganaban peso y complejidad, haciendo necesaria la inversión de mayor tiempo revisando y corrigiendo una mayor cantidad de código que, por lo general, no pertenecían al mismo autor, complicando aún más si cabe la tarea.

Es gracias a las hojas de estilo CSS que se ven aligerados los documentos de HTML, viéndose así liberados de la información destinada a las características estéticas, pudiéndose centrar en su carácter estructural.

Si bien es cierto que este curso está orientado a HTML y abordaremos ciertas características de estilo que pueden ser atribuidas desde el mismo documento de HTML, haremos hincapié en la necesaria y correcta aplicación reservadas a las hojas de estilo CSS como recomiendan desde el W3C.

2. Características elementales

2.2 Anatomía de HTML

Aunque más adelante abordaremos varias de las estructuras más comunes de un documento HTML, lo que es totalmente indispensable en todos y cada uno de ellos, son las dos partes que forman el documento; la cabecera o *Head* y el cuerpo o *Body*.

Esta es la estructura básica de un HTML cualquiera:

```
<html>

<head>
<title>Mi documento HTML</title>
</head>

<body>
<p>Aquí va la información principal de la página..</p>
</body>

</html>
```

Así, las etiquetas esenciales en todo documento HTML tienen las siguientes funciones:

- **<html>**: Indica el principio y el fin del documento. A efectos prácticos y con leves excepciones, todo contenido ha de localizarse entre esta etiqueta. En ella vienen definidas las siguientes dos etiquetas.
- **<head>**: La cabecera contiene información sobre el propio documento HTML, como su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, salvo la etiqueta <title>, que se utiliza para indicar el título del documento y que aparece en la parte superior de la ventana del navegador.
- **<body>**: Delimita el cuerpo del documento HTML. El cuerpo contiene todos los contenidos que se muestran; texto, imágenes, tablas, etc... Al ser la etiqueta que contiene la mayor parte de la información de la web, suele ser mayor que <head>.

2.3 Etiquetas y atributos

HTML 5 contiene más de 100 etiquetas que podemos consultar tanto en la web oficial del W3C como en varios otros sitios web como el siguiente:

https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos

<https://developer.mozilla.org/es/docs/Web/HTML/Element>

a, abbr, acronym, address, area, b, base, bdo, big, blockquote, body, br, button, caption, cite, code, col, colgroup, dd, del, dfn, div, dl, dt, em, fieldset, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, kbd, label, legend, li, link, map, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, samp, script, select, small, span, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, ul, var.

La lista de etiquetas evoluciona y varía en función de su utilidad y compatibilidad con los diferentes medios que se desarrollan a diario, y ahí radica la importancia de organismos como el W3C.

A pesar de ello, dichas etiquetas no son suficientes por sí mismas para desarrollar las complejas páginas web actuales, y muchas de ellas necesitan “etiquetas adicionales” que modifiquen sus valores llamados *Atributos*. Cada etiqueta tiene ciertos atributos compatibles, de esta manera, si atribuimos un valor incompatible con la etiqueta, el navegador lo ignorará. He aquí un ejemplo:

`<a>La etiqueta “a” sirve para enlazar, pero aquí no especifica a dónde.`

`Aquí sin embargo el enlace apunta a la página de Caronte Studio gracias al atributo “href”.`

De esta manera dividimos los atributos en cuatro grupos:

2.3.1 Atributos básicos

Compatibles casi con todas las etiquetas. Son realmente explotados cuando se trabaja con CSS conjuntamente.

Atributo	Descripción
id="texto"	Agrega un identificador único e irrepetible a un elemento dentro de una página HTML
class="texto"	Indica a una etiqueta la clase CSS aplicando los estilos de esta
style="texto"	Modifica directamente los estilos CSS de un elemento. No es la opción más recomendable.
title="texto"	Establece el título a un elemento. Se usa para mejorar la accesibilidad

2.3.2 Atributos de eventos

Los usados en webs dinámicas con JavaScripts. Cuando el usuario realiza acciones como pulsar teclas o pinchar en determinados lugares de la web se produce un evento dentro del navegador.

Atributo	Descripción	Elementos que pueden usarlo
onblur	Deseleccionar el elemento	<button>, <input>, <label>, <select>, <textarea>, <body>
onchange	Deseleccionar un elemento que se ha modificado	<input>, <select>, <textarea>
onclick	Pinchar y soltar el ratón	Todos los elementos
ondblclick	Pinchar dos veces seguidas con el ratón	Todos los elementos
onfocus	Seleccionar un elemento	<button>, <input>, <label>, <select>, <textarea>, <body>
onkeydown	Pulsar una tecla (sin soltar)	Elementos de formulario y <body>
onkeypress	Pulsar una tecla	Elementos de formulario y <body>
onkeyup	Soltar una tecla pulsada	Elementos de formulario y <body>

onload	La página se ha cargado completamente	<body>
onmousedown	Pulsar (sin soltar) un botón del ratón	Todos los elementos
onmousemove	Mover el ratón	Todos los elementos
onmouseout	El ratón "sale" del elemento (pasa por encima de otro elemento)	Todos los elementos
onmouseover	El ratón "entra" en el elemento (pasa por encima del elemento)	Todos los elementos
onmouseup	Soltar el botón que estaba pulsado en el ratón	Todos los elementos
onreset	Inicializar el formulario (borrar todos sus datos)	<form>
onresize	Se ha modificado el tamaño de la ventana del navegador	<body>
onselect	Seleccionar un texto	<input>, <textarea>
onsubmit	Enviar el formulario	<form>
onunload	Se abandona la página (por ejemplo al cerrar el navegador)	<body>

2.3.3 Atributos de foco

Compatibles casi con todas las etiquetas. Son realmente explotados cuando se trabaja con CSS conjuntamente.

Atributo	Descripción
accesskey="letra"	Establece una tecla de acceso rápido a un elemento HTML
tabindex="numero"	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32.767

onfocus, onblur

Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

2.3.4 Atributos de internacionalización

Empleados en las páginas con varios idiomas.

Atributo	Descripción
lang = "codigo de idioma"	Indica el idioma del elemento mediante un código predefinido
xml:lang = "codigo de idioma"	Indica el idioma del elemento mediante un código predefinido
dir	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

2.4. Elementos HTML

Las partes que componen el código de HTML son llamados *elementos*. Un elemento se compone de:

- **Etiqueta de apertura.**
- **Atributos.**
- **Texto contenido en la etiqueta.**
- **Etiqueta de cierre.**

```
<a href="https://www.carontestudio.com">Texto opcional. </a>.
```

Aquí podemos apreciar cómo dentro de la etiqueta de apertura se encuentra el atributo "href" y su valor, en este caso la dirección web. Después, entre las etiquetas de apertura y cierre, el texto que se

mostrará al usuario, y posteriormente la etiqueta de cierre que viene indicada con el nombre de la etiqueta y un "/" en su interior.

En HTML los elementos se distinguen en dos categorías; los **elementos en línea** (*inline elements*) y **elementos en bloque** (*block elements*). La principal diferencia radica en que los elementos en bloque comienzan en una nueva línea, y ocupan todo el espacio disponible hasta el final de esta, aunque sus contenidos no lo hagan. Por el contrario, los elementos en línea ocupan solo lo que ocupe su contenido. Por ejemplo, los elementos con etiqueta <a> son enlaces y son elementos en línea. Ocupan, por tanto, solamente lo que ocupe el texto contenido dentro de la etiqueta. Mientras que un elemento con etiqueta <p>, destinada a generar párrafos de texto, se considera de bloque al ocupar todo el ancho posible de la línea y generando una nueva.

Los elementos en línea en HTML son:

a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Los elementos de bloque:

address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul.

2.4.1 Sintaxis de las etiquetas

Aunque originalmente HTML era muy permisivo con la manera de escribir sus etiquetas, como por ejemplo, usando indistintamente minúsculas y mayúsculas, el protocolo actual es más estricto. Estas son sus claves:

- **Las etiquetas se abren y se cierran de manera inmediata:** he aquí un ejemplo de una mala praxis.

```
<p>El párrafo se está cerrando antes que el <a href="">enlace.</p></a>
```

- El valor de los atributos se expresa dentro de comillas.

3. Etiquetas texto

3.1. Estructuración jerárquica del texto: texto corrido y secciones.

Dado que la mayoría del contenido de una página web suele ser texto, es esencial dominar las etiquetas referentes al formato y estructuración de este, que nos ofrece el lenguaje HTML.

Saber identificar los distintos elementos de un texto ordenado, nos permitirá hacer más cómoda y atractiva la experiencia de navegación de nuestra web a los usuarios. De esta manera, habremos de dividir nuestros textos con titulares, secciones, párrafos bien definidos y amenos, texto destacado, citas, etc...

Para estructurar el texto, la herramienta principal es la etiqueta **<p>**, que como hemos comentado en el punto anterior, se trata de una herramienta de bloque, haciéndola así muy útil de cara a la estructuración del documento HTML.

Para determinar la jerarquía del texto y ordenar los párrafos, están los títulos de sección o encabezados. HTML ofrece 6 etiquetas jerarquizadas para títulos, con un formato dado de manera predeterminada por el navegador. Estas son las **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>** y son de vital importancia, no solo en el aspecto estético, sino de cara al SEO y posicionamiento web. Como la etiqueta **<p>**, también son etiquetas en bloque. Aunque el formato lo determine el navegador, estas etiquetas son siempre modificadas por hojas de estilo CSS.

3.2. Espacios en blanco, nuevas líneas y texto preformateado

HTML considera un vacío en los espacios en blanco, los tabuladores y el carácter de nueva línea. Así, en este texto se mostrarán ambos párrafos de igual manera en el navegador:

```
<p>HTML no realiza saltos de línea si no se especifica con la etiqueta pertinente.</p>
```

```
<p>HTML no  
realiza saltos de línea  
si no se especifica con      la etiqueta  
pertinente.</p>
```

Para poder insertar un salto de línea, aparte de servirnos de los elementos de bloque, existe la etiqueta **
**, que tiene como peculiaridad el hecho de ser una etiqueta vacía. Es decir, no contiene ningún texto en su interior, por eso se debe abrir y cerrar consecutivamente. Es muy común ver solo la expresión `</br>` o `
`, abreviando así la expresión `
</br>`.

En el caso de los espacios en blanco, ha de incluirse el texto ** ** sin olvidar el “ampersand” y el “punto y coma” final. Esto se debe a la codificación de tal carácter. Se pueden usar de manera consecutiva tantas veces como se quiera.

Otras etiquetas de formato son **<pre>** y **<code>** que en esencia respetan el formato de origen del código. Así, si en el editor de código HTML se ha escrito un texto dentro de dicha etiqueta con espacios, tabulaciones o saltos de línea, en el navegador se mantendrá de manera idéntica. Estas etiquetas se usan con carácter didáctico y por ello no profundizaremos en ellas.

3.3. Marcado básico de texto

Cuando el texto se encuentre estructurado en párrafos y secciones, deberemos proceder a destacar elementos relevantes de los párrafos con pasajes en negrita o cursiva, citas o anotaciones, y para ello HTML nos ofrece las siguientes etiquetas:

- ****: De “Emphrased”, su función es semántica y muestra en cursiva el texto. Similar a **<i>**.
- ****: muestra en negrita el texto. Similar a **** pero con importancia SEO.
- **<ins>**: se usa para indicar texto añadido a uno original y mostrar así una modificación del contenido.
- ****: se emplea para indicar una modificación del texto original y mostrar un texto tachado pero visible para el lector.
- **<blockquote>**: Permite añadir citas a nuestro texto. Además, mediante el atributo cite podemos añadir una URL referente a dicha cita. Los navegadores añaden de manera automática un espacio a la izquierda de texto, diferenciándose así del texto corrido.

3.4. Marcado avanzado de texto

Aunque no son las etiquetas más comunes para marcar texto, hay ocasiones en las que las opciones anteriores no son suficientes y es necesario añadir ciertos matices. HTML nos ofrece herramientas tan útiles como las siguientes etiquetas:

- **<abbr>**: Se emplea para añadir abreviaturas.
- **<acronym>**: muestra un marco de texto sobre los acrónimos que especifiquemos.
- **<dfn>**: Se utiliza para marcar definiciones y explicar su significado.
- **<cite>**: Junto con **<blockquote>** sirve para añadir citas y sus autores. En sí, **<cite>**, marca el autor de la cita.
- **<blockquote>**: Junto a **<cite>**, completa la función de cita. **<blockquote>** contiene la cita en sí.
- **<hr>**: Genera una línea horizontal separadora.

3.5. Ejercicio 1

Realiza un documento HTML con las etiquetas necesarias para copiar de manera exacta el siguiente ejemplo. Hay que tener en cuenta que el primer texto en negrita ha de ser importante de cara al buscador web:

Miguel de Cervantes Saavedra

Miguel de Cervantes Saavedra (*Alcalá de Henares, 29 de septiembre de 1547-Madrid, 22 de abril de 1616*) fue un soldado, novelista, poeta y dramaturgo español.

Está considerado la máxima figura de la literatura española y es universalmente conocido por haber escrito **El ingenioso Hidalgo don Quijote de la Mancha** (conocida habitualmente como el Quijote), que muchos críticos han descrito como la primera novela moderna y una de las mejores obras de la literatura universal, además de ser el libro más editado y traducido de la historia, solo superado por la Biblia. Se le ha dado el sobrenombre de «Príncipe de los Ingenios».

Extracto del Quijote

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto della concluían sayo de velarte, calzas de velludo para las fiestas con sus pantuflos de lo mismo, los días de entre semana se honraba con su vellori de lo más fino.

Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la podadera. Frisaba la edad de nuestro hidalgo con los cincuenta años, era de complexión recia, seco de carnes, enjuto de rostro; gran madrugador y amigo de la caza. Quieren decir que tenía el sobrenombre de Quijada o Quesada (que en esto hay alguna diferencia en los autores que deste caso escriben), aunque por conjeturas verosímiles se deja entender que se llama Quijana; pero esto importa poco a nuestro cuento; basta que en la narración dél no se salga un punto de la verdad.

"Ninguna ciencia, en cuanto a ciencia, engaña; el engaño está en quien no la sabe."

Miguel de Cervantes Saavedra.

[Averiguar más sobre Miguel de Cervantes](#)

3.6. Codificación de caracteres

Como hemos podido comprobar, HTML requiere de una sintaxis muy concreta. Sin embargo, esta no es su único requerimiento. Cabe destacar que HTML posee una serie de caracteres reservados para su estructura, como “<”, o “>”, entre otros. Es por eso que, para poder emplearlos como caracteres en nuestro código, en vez de emplearlos de manera estructural, HTML hace uso de una codificación concreta para cada uno de ellos. Estos son los caracteres más empleados y sus códigos.

Entidad	Carácter	Traducción
<	<	Menor que
>	>	Mayor que
&	&	Ampersand
"	"	Comillas
 	(espacio en blanco)	Espacio en blanco
'	'	Apóstrofo

4. Enlaces

4.1. Enlaces y URL

La mayor virtud de HTML es constituirse a base de hipertexto, lo que permite la interconectividad entre documentos o páginas. De esta manera, por medio de enlaces o hipervínculos HTML nos permite añadir infinidad de contenido a nuestras webs.

Las URL (*Uniform Resource Locator*) son identificadores que actúan como punteros que señalan a determinado contenido. Son únicas, concisas y concretas, por lo que apuntan a un solo recurso. Se dividen en tres partes:

- **Protocolo:** (http://) o (https://), necesario para el navegador.
- **Servidor:** (www.carontestudio.com) en esencia es donde se almacena la página y su contenido.
- **Ruta:** (/lecciones/.html).es un localizador específico dentro de la web.

Los enlaces pueden ser internos o externos dependiendo de si estos conducen fuera de la web o permitan navegar por las distintas secciones de esta. Hay varios tipos de enlaces dependiendo del tipo de URL de que se trate.

- **URL Absolutas:** Incluyen todas las partes de una URL (protocolo, servidor y ruta). Por ejemplo:
`http://www.carontestudio.com/clases/photoshop.html`
- **URL Relativas:** Sólo muestran la ruta, y por ello ha de determinarse mejor el origen del enlace.

Son abreviaciones de URL mayores. Dependiendo de la cantidad de `../` que encuentre el navegador, éste sabe cuantos niveles inferiores o superiores ha de buscar para hallar su origen. Por ejemplo, aquí, dos niveles: `../../clases/photoshop.html`

Los enlaces HTML se crean con la etiqueta `<a>` (*anchor, ancla*) y su elemento más importante es su atributo **href** del que ya hemos hablado antes. Por ejemplo:

```
<a href="http://www.carontestudio.com">Página principal de Jon Rivas</a>
```

Al pinchar en el texto del navegador “Página principal de Jon Rivas”, este nos dirigirá a la dirección especificada. Href también puede apuntar a una imagen, documento PDF, doc, ...

```
<a href="http://www.carontestudio.com/documento.doc">Descargar documento</a>
```

Uno de los recursos más utilizados es apuntar al inicio para hacer botones que nos lleven a la página principal de una web de la siguiente manera:

```
<a href="#">Pagina principal</a>
```

Otro atributo relevante es **<name>** que nos sirve para definir enlaces dentro de una misma página, y navegar de sección en sección:

```
<a name="seccion_primera"></a>  
<a name="seccion_segunda"></a>
```

O de la misma manera, enlazar directamente a una sección concreta de una página en concreto mediante el uso del símbolo # para especificar la parte de la página en la que se encuentra el contenido que queremos enlazar:

```
<a href="http://www.carontestudio.com/tema1.html#seccion_octava">Enlace a la octava  
sección de la página del tema 1</a>
```

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

La etiqueta **<a>** cuenta con varios atributos muy específicos y ciertamente importantes. No todos son de uso común, pero conviene conocerlos:

- **type:** Indica previamente al navegador el tipo de contenido al que apunta el enlace. Por ejemplo: "image/jpg" o "txt/css".
- **rel:** Indican la relación que tiene la página con la que se enlaza. El más común es el usado para las hojas de estilo. Por ejemplo: "stylesheet". También pueden indicar mediante "alternate" que existe una versión alternativa a la página en cuestión, quizá para dispositivos móviles o impresoras, ...
- **Hreflang:** Indica el idioma del recurso enlazado.
- **charset:** Hace referencia a codificaciones de idioma para especificar que el enlace al que apunta, el navegador ha de decodificarlo usando determinado estándar. Por ejemplo:
`Página principal de Jon Rivas`

Aquí se ha especificado que el lenguaje es en español mediante el atributo **hreflang**, el tipo mediante **type** y la codificación a usar mediante **charset**.

La principal ventaja de los enlaces realizados con la etiqueta `<a>` es sin duda que permiten realizar todo tipo de enlaces sin que el recurso llegue a cargarse antes de pinchar en el enlace.

4.2. Enlaces “Script” y “Link”

Las etiquetas **<script>** y **<link>** son enlaces que a diferencia de **<a>**, HTML carga de manera automática junto a la página en cuestión.

- **Script:** Nos permite enlazar código de otros lenguajes de programación, normalmente de Java, enlazando así el HTML a un recurso externo. Estos archivos suelen cargarse por defecto en la cabeza del HTML **<head>**, para cargar junto a la web nada más entrar en ella y se enlazan de la siguiente manera:

```
<head>
  <script type="text/javascript"
    src="http://www.carontestudio.com/js/archivo_java.js"></script>
</head>
```

Esta etiqueta tiene el atributo **src** que actúa como un **href**, y un **type** para indicar que es un archivo de tipo JavaScript. En este caso se puede comprobar que la URL indica que el archivo de Java (los *.js) se encuentra en la carpeta JS, dedicada por norma general a contener todos los archivos de este tipo de las webs, pero podría corresponderse a un enlace a un recurso externo.

Esta etiqueta también nos permite insertar un extracto de código de Java por ejemplo, en medio del código, sin necesidad de apuntar a un archivo externo, tal que así:

```
<head>
  <script type="text/javascript">
    window.onload = function() { alert("La página se muestra correctamente"); }
  </script>
</head>
```

- **link:** Sirve para enlazar relaciones entre nuestro documento HTML y otros recursos. Solo se puede incluir en la cabeza <head>. El caso más habitual es el de las hojas de estilo CSS. De esta manera relacionamos nuestro HTML con un recurso externo como es una hoja de estilos:

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

Como en el caso de los JavaScripts, los CSS suelen tener una ruta similar a esta, en la que se especifica que una carpeta llamada CSS aglomera todas las hojas de estilos de nuestra web.

4.3. Enlaces útiles y usos especiales

En muchas de las webs que visitemos podremos observar ciertos enlaces de uso muy común de que si bien es cierto que son sencillos, aportan soluciones muy útiles a problemas habituales.

- **Enlace a favicon:** El favicon es el diminuto icono propio de cada página que aparece en la barra de título o en la de URL. Son archivos .ico o .png. Por ejemplo:

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

- **Enlace a email:** Permite abrir directamente nuestro gestor de correo predeterminado junto a la dirección que indiquemos:

```
<a href="mailto:nombre@direccion.com" title="Dirección de email">
```

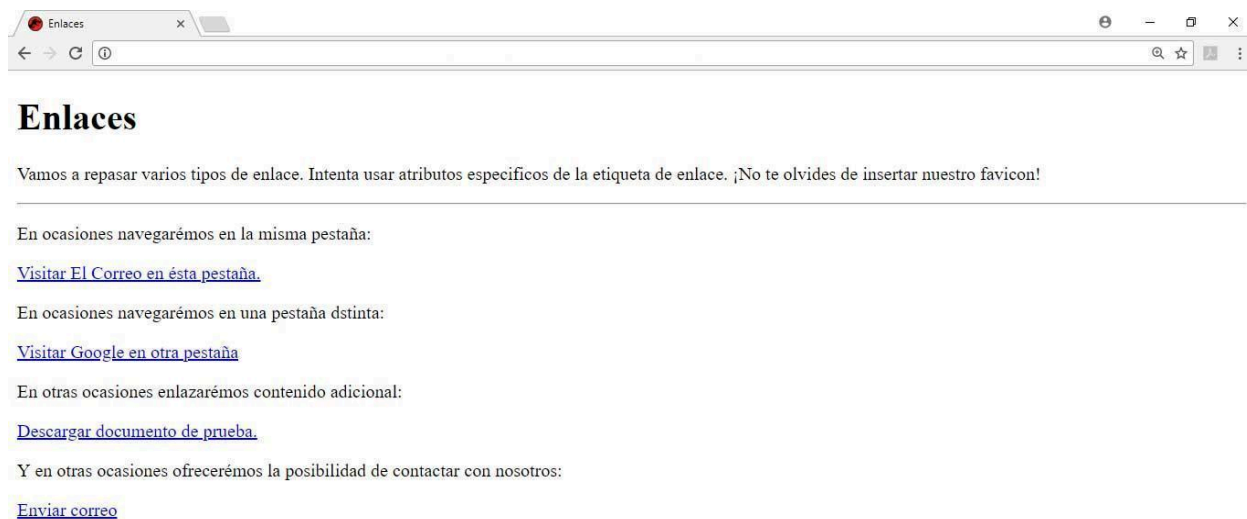
Esríbenos

``

Aunque aún es fácil encontrar este sistema, no es recomendable debido a la cantidad de robots que desde hace tiempo escanean constantemente la red en busca de direcciones de email a la que poder enviar correo pernicioso o spam.

4.4. Ejercicio 2

Crea un documento HTML exacto al que se muestra a continuación:



5. Listas

5.1. Listas no ordenadas

Se trata de un conjunto de elementos relacionados entre sí, pero sin un orden determinado. Son las más habituales y se usan bajo la etiqueta `` (*unordered list*). Es un elemento de bloque y por lo tanto se usa mucho de manera estructural. La unidad de las listas no ordenadas es la misma que para las ordenadas, la etiqueta ``. Por ejemplo:

```
<h1>Menú</h1>

<ul>
  <li>Inicio</li>
  <li>Blog</li>
  <li>Contacto</li>
</ul>
```

El navegador muestra por defecto la lista con una tabulación y viñeta predeterminada, pero este estilo se puede editar posteriormente mediante reglas CSS.

5.2. Listas ordenadas

Son como las no ordenadas pero con un orden especificado. Se usan bajo la etiqueta `` (*ordered list*). También es un elemento de bloque y por lo tanto se usa de igual manera que la `` aunque aprovechando su carácter ordenado, como por ejemplo en índices o listas numeradas. He aquí un ejemplo:

```
<h1>Índice</h1>
```

```
<ol>
```

```
<li>Prehistoria</li>
```

```
<li>Edad de bronce</li>
```

```
<li>Historia moderna</li>
```

```
</ol>
```

El navegador muestra por defecto la lista con una tabulación y una secuencia numerada predeterminada, pero se puede editar mediante reglas CSS.

5.3. Listas de definición

Son como las y están destinadas a listas explicativas o de definición. Se usan con la etiqueta <dl>, y sus unidades son <dt> y <dd>, la primera especifica el término, y la segunda la definición:

```
<h1>Animales</h1>
```

```
<dl>
```

```
<dt>Perro</dt>
```

```
<dd>El perro es un mamífero carnívoro de la familia de los cánidos.</dd>
```

```
</dl>
```

5.4. Ejercicio 3

Crea un documento HTML exactamente igual al que se muestra a continuación:



Ranking de la temporada 2018

• Goleadores

- F.C.Barcelona
 - 1. *Lionel Messi*
 - 2. *Luis Suarez*
 - 3. *Paulinho*
- Real Madrid
 - 1. *Cristiano Ronaldo*
 - 2. *Karim Benzema*
 - 3. *Gareth Bale*

• Asistentes

- F.C.Barcelona
 - 1. *Lionel Messi*
 - 2. *Ivan Rakitic*
 - 3. *Jordi Alba*
- Real Madrid
 - 1. *Cristiano Ronaldo*
 - 2. *Luca Modric*
 - 3. *Toni Kroos*

6. Imágenes y objetos

6.1. Imágenes

Después del texto, las imágenes son lo más importante de una web. En HTML se insertan mediante la etiqueta **** de la siguiente manera:

```

```

La etiqueta **** dispone de varios atributos que hay que tener en cuenta:

- **src:** Como en otras etiquetas, es el que indica a qué archivo de imagen apunta la URL.
- **alt:** Es un campo de texto que explica el contenido de la imagen. Es muy útil por motivos de accesibilidad puesto que aunque una imagen no se llegue a cargar, el texto nos explica lo que debería aparecer. Además, es texto que el narrador de los Sistemas Operativos ofrece a gente con dificultades en la vista. También es un factor a tener en cuenta de cara al SEO y posicionamiento web.
- **height:** Es una unidad de medida, en pixels, em, %, puntos, ... Indica la altura con la que se va a mostrar una imagen.
- **width:** Es una unidad de medida, en pixels, em, %, puntos, ... Indica la anchura con la que se va a mostrar una imagen.

6.2. Mapas de imagen

Un mapa de imagen se define con la etiqueta **<map>** y permite definir zonas que se pueden pinchar dentro de una imagen. Dicha zona está especificada por medio de unas coordenadas dadas por la etiqueta **<area>**. La etiqueta **<area>** se puede configurar en forma con los atributos **rect** (*rectangle*), **circle** (*círculo*), o **poly** (*polígono*). Después se especifican las coordenadas con el atributo **coords**.

Se ha de especificar en una etiqueta `` que es un mapa con el atributo “usemap” y el código se mostraría tal que así:

```

<map name="mapa">
  <area shape="rect" coords="20,25,84,113" href="rectangulo.html" />
</map>
```

6.3. Objetos

En HTML no solo podemos insertar imágenes, sino también todo tipo de elementos como animaciones flash, videos en diversos formatos, mapas ... todo gracias al uso de plugins externos. Para ello usaremos la etiqueta **<object>** y sus atributos:

- **type**: Indica el tipo de contenido.
- **classid, codetype, codebase**: Información específica del elemento que insertamos.
- **data**: Indica la URL del contenido.

```
<object data="video_ejemplo.mpeg" type="application/mpeg" />
```

Uno de los principales inconvenientes de la etiqueta `<object>` es que en algunos navegadores el contenido no se muestra hasta que no es descargado por completo, haciendo la navegación más costosa y lenta. Para ello se puede recurrir a la etiqueta **<embed>** como la que nos facilitan páginas como Youtube o Vimeo.

```
<object width="400" height="350">
  <param name="movie" value="https://www.youtube.com/embed/YS2-v6nqYQg"></param>
  <param name="wmode" value="transparent"></param>
  <embed src="https://www.youtube.com/embed/YS2-v6nqYQg"
    type="application/x-shockwave-flash" wmode="transparent" width="400"
    height="350"></embed>
</object>
```

En la actualidad el uso de la etiqueta `<embed>` como tal no está aconsejada por dar errores de validación, y muchos de los proveedores de contenido multimedia facilitan el uso de la etiqueta `<iframe>` que actúa de manera similar como en el siguiente ejemplo:

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/YS2-v6nqYQg"
frameborder="0" allow="autoplay; encrypted-media" allowfullscreen></iframe>
```

6.4. Ejercicio 4

Reproduce el siguiente documento de HTML. Ten en cuenta que la relación de aspecto de las imágenes entre sí se han de mantener siempre constantes:

Gatitos

El gato o gato doméstico y coloquialmente llamado minino, michino, micho, mizo, miz, morrongo o morroño; es una subespecie de mamífero carnívoro de la familia Felidae.



7. Tablas

7.1. Uso correcto de las tablas

Las tablas se basan en un sistema de filas y columnas con unos títulos encabezando cada una de las columnas. Son muy útiles pero peligrosas si no se usan bien. Hace años era habitual usarlas para estructurar todo el diseño del sitio web, pero no debemos olvidar que HTML ofrece herramientas mucho mejores para dicha tarea. Es por ello que debemos destinar las tablas únicamente a cumplir su función: la de recoger información que precise de su estructura.

7.2. Tablas básicas

Las tablas en HTML se definen con la etiqueta **<table>** y sus unidades son **<tr>** y **<td>**, la primera para hacer filas, y la segunda para columnas.

```
<table>
<tr>
  <td><strong>Curso</strong></td>
  <td><strong>Horas</strong></td>
  <td><strong>Días</strong></td>
</tr>
<tr>
  <td>HTML 5</td>
  <td>30</td>
  <td>Martes y jueves</td>
</tr>
<tr>
  <td>CSS</td>
  <td>20</td>
  <td>Lunes y viernes</td>
</tr>
```


</table>

Otros atributos muy útiles a la hora de hacer tablas son los siguientes:

- **colspan**: Es un valor numérico que indica cuántas columnas ocupa dicha celda.
- **rowspan**: Es un valor numérico que indica cuántas líneas ocupa dicha celda.
- **scope**: Determina que es la cabecera de una línea o columna.
- **abbr**: Es una descripción del contenido de la celda. Muy útil de cara a accesibilidad y gente con problemas de visión.

7.3. Tablas avanzadas

En ocasiones es indispensable el uso de tablas complejas para ordenar correctamente la información, y cuando las tablas simples no son suficientes, éstas se pueden ampliar con recursos como cabeceras o pies. Para ello HTML nos ofrece los siguientes atributos para <table>:

- **<thead>**: Define una cabecera.
- **<tbody>**: Define el cuerpo de la tabla.
- **<tfoot>**: Define un pie de la tabla.

Cuando se definan una cabecera y un pie, siempre han de ir delante del conjunto que forme el cuerpo. De esta manera, <thead> y <tfoot> siempre han de ir antes del grueso que conforme el <tbody>.

7.4. Ejercicio 5

Reproduce el siguiente documento HTML con ayuda de lo aprendido hasta ahora:

Led Zeppelin

Led Zeppelin fue un grupo británico de hard rock fundado en 1968 por el guitarrista Jimmy Page, quien había pertenecido a The Yardbirds. La banda estuvo integrada por John Paul Jones como bajista y teclista, al vocalista Robert Plant y a John Bonham a la batería.



Miembros

- Jimmy Page
- Robert Plant
- John Paul Jones
- John Bonham

Mejores temas

*Top 3 canciones más
escuchadas*

Año	Tema	Escuchas
1971	Stairway to Heaven	15.236.200
1969	Whole lotta love	12.200.000
1970	The Immigrant song	9.500.000
Datos extraídos de la web de Spotify.		

Saber más acerca de Led Zeppelin

www.ledzeppelin.com

8. Formularios

8.1. Formularios básicos

Los formularios se definen por la etiquetas `<form>` e `<input>`. HTML contiene todo lo necesario para realizarlo, ya sean cuadros de texto, listas desplegables o botones. Mediante los atributos de la etiqueta `<form>` podemos configurarlo:

```
<form action="http://www.carontestudio.com/formulario.php" method="post">
  Escribe tu nombre:
  <input type="text" name="nombre" value="Nombre" />
  <input type="submit" value="Enviar" />
</form>
```

De la siguiente manera el usuario vería el texto "Escribe tu nombre:" value un espacio en blanco donde poder escribir y un botón submit con el texto "enviar". El atributo `method` establece la forma en la que se envían los datos al servidor. El método `post` admite más información que el método `get` pero ambos son válidos.

8.2. Elementos de los formularios

Los elementos como botones y cuadros de texto de un formulario son configurables. La etiqueta `<input>` tiene atributos que es conveniente especificar:

- **type:** Indica el tipo de acción del input. (button, reset, submit, checkbox, password, text, ...=).
- **name:** el nombre del input. Es estrictamente necesario.
- **value:** valor inicial.
- **size:** tamaño.
- **checked:** si hay un checkbox o radiobutton indica cual está de por sí seleccionado.
- **disabled:** si el campo no se puede usar.
- **src:** para botones con imágenes.
- **alt:** descripción.

Los cuadros de texto son los campos vacíos a rellenar por el usuario. Se estructura con la etiqueta `<input>` y se definen los atributos mencionados anteriormente de la siguiente manera:

```
<input type="text" name="codigopostal" value="" />
```

Para crear casillas de verificación o “checkbox” y poder seleccionar opciones no excluyentes usaremos el type checkbox:

```
<input type="checkbox" name="comida" value="hamburguesa" />  
<input type="checkbox" name="comida" value="patatas" />  
<input type="checkbox" name="comida" value="bebida" />
```

Mediante el “type” “radiobutton” podemos ofrecer opciones excluyentes de tal manera que el usuario solo pueda escoger una opción, como en el siguiente caso:

```
<input type="radio" name="origen" value="norte" checked="checked" /> Norte  
<input type="radio" name="origen" value="sur" /> Sur
```

Una de las opciones más útiles es la de vaciar el formulario o resetearlo. HTML permite crear un botón en el formulario mediante el atributo `<type>` con esta función de la siguiente manera:

```
<input type="reset" name="limpiar" value="Borrar datos" />
```

En ocasiones podemos requerir que el usuario adjunte algún tipo de archivo y para ello podemos solicitarlo con el atributo `<type>` “file”:

```
<input type="file" name="adjuntar"/>
```

Para este caso en particular se requiere añadir a la etiqueta `<form>` el atributo `<enctype>` de la siguiente manera:

```
<form action="..." method="post" enctype="multipart/form-data">
...
<input type="file" name="adjuntar"/>
...
</form>
```

8.3. Desplegables y otros elementos de los formularios

HTML nos ofrece la posibilidad de completar nuestros formularios con opciones que facilitan el uso de éstos al usuario. Para ello habremos de usar las siguientes etiquetas:

- **<fieldset>**: Agrupa los campos a nuestro gusto, congregando así los controles del formulario con un borde resaltado.
- **<legend>**: Define el título o leyenda de los campos de un formulario agrupados por `<fieldset>`. Se localiza dentro de `<fieldset>` y sirve como título para ésta.

De esta manera podemos agrupar el contenido de un formulario tal que así:

```
<form action="formulario_ejemplo.php" method="post">
  <fieldset>
    <legend>Datos personales</legend>
    Nombre <br/>
    <input type="text" name="nombre" value="" />
    <br/>
    Apellidos <br/>
    <input type="text" name="apellidos" value="" />
    <br/>
    DNI <br/>
    <input type="text" name="DNI" value="" size="10" maxlength="9" />
  </fieldset>

  <fieldset>
    <legend>Datos de usuario</legend>
    Nombre de usuario<br/>
    <input type="text" name="nombre" value="" maxlength="10" />
    <br/>
    Password<br/>
    <input type="password" name="password" value="" maxlength="10" />
    <br/>
    Repite el password<br/>
    <input type="password" name="password_2" value="" maxlength="10" />
  </fieldset>
</form>
```

También podemos organizar nuestro formulario con la etiqueta **<label>** que sirve para definir el título o leyenda de campos determinados. Se usa con el atributo “for” de la siguiente manera:

```
<label for="nombre">Nombre</label> <br/>
<input type="text" id="nombre" name="nombre" value="" />

<label for="apellidos">Apellidos</label> <br/>
<input type="text" id="apellidos" name="apellidos" value="" />
```

Las listas desplegables son otro de los recursos más utilizados y por lo general no se pueden crear con la etiqueta **<input>**. Para ello nos valdremos de las etiquetas **<select>**, **<textarea>** y **<option>**:

- **textarea:** Son los campos vacíos de un formulario y se manejan mediante el atributo “cols” que hace referencia a las columnas que usa, definiendo así el máximo de caracteres que se pueden escribir en cada fila. La altura se controla mediante el atributo “rows” y especifica el número de líneas visibles.
- **select:** Se usa para crear una lista desplegable. Es configurable mediante los atributos “size”, “multiple” para elegir más de una opción, ...
- **option:** Define cada elemento de una lista desplegable. el atributo “selected” define cual aparece seleccionado al iniciar la página.

```
<label for="equipo_f">Equipo de futbol</label> <br/>
<select id="equipo_f" name="equipo">
  <option value="" selected="selected">- selecciona -</option>
  <option value="Athletic">Athletic de Bilbao</option>
  <option value="Real">Real Sociedad</option>
  <option value="Eibar">Eibar</option>
  <option value="Alavés">Alavés</option>
</select>

<label for="equipo_b">Equipo de baloncesto</label> <br/>
<select id="equipo_b" name="equipo_b" size="5">
  <option value="Baskonia" selected="selected">Baskonia</option>
  <option value="BBB">Bilbao Bizkaia Basket</option>
  <option value="Unicaja">Unicaja Málaga</option>
  <option value="otro">Otro</option>
</select>
```


8.5. Ejercicio 6

Reproduce el siguiente documento HTML con ayuda de lo aprendido hasta ahora:

Perfil de usuario

Datos personales

Nombre

Introduce tu nombre

Apellidos

Introduce tus apellidos

Contraseña

Contraseña de 8 caracteres

DNI

Introduce tu DNI

E-mail

Introduce tu email

☒ Recibir notificaciones

Sexo

☒ Hombre

☐ Mujer

Adjuntar foto

Seleccionar archivo

Ningún archivo seleccionado

Intereses

☒ HTML

☐ Música

☐ Deporte

☐ Aventura

Enviar

Reset

9. Estructurando un documento HTML

9.1 Contenedores

Ya hemos hablado de la anatomía de un HTML y hemos determinado que consta de una cabeza o `<head>` y un cuerpo o `<body>`, y es precisamente en este `<body>` donde se diseña toda la estructura de la web y para ello tenemos una etiqueta que nos facilita mucho la labor a la hora de organizar el contenido de nuestra web. Esta etiqueta es la **`<div>`**, que agrupa elementos en bloque.

La etiqueta `<div>` suele ir casi siempre acompañada de un “id” o identificador de dicho contenedor o `<div>`, o un “class” si dicho contenedor va a compartir propiedades atribuidas más tarde en una hoja de estilos CSS con otros en la misma página.

En esencia, se puede observar que la gran mayoría de páginas web comparten ciertos elementos estructurales en forma de `<div>`, de ahí que haya una especie de consenso a la hora de llamar diferentes partes de una web:

- **header (cabecera):** Suele incluir los elementos más estáticos y comunes como logos de la web, un banner o un cuadro de búsqueda).
- **menu (menú):** Destinado a agrupar el menú y botones de navegación.
- **contenedor (wrapper):** Suele contener la mayoría de contenido de la página.
- **content (contenido):** El contenido principal del sitio, sus artículos, productos, noticias,...).
- **lateral (sidebar):** algunas pueden incluir ahí el menú, o enlaces a artículos destacados, etc...
- **pie (footer):** es similar al header pero menos visual y relevante al ser lo último en ser visto. Suele contener las políticas de privacidad, copyrights, términos de uso, etc...).

Con la aparición de HTML 5, las diferentes secciones habituales de una página web dejaron de ser meros nombres o id de decenas de `<div>` por toda la página para tener su propia etiqueta, mejorando así la lectura y comprensión del documento por parte del desarrollador web.

`<header>` Contenido de introducción, cabecera.

`<nav>` Grupo de enlaces para la navegación.

`<section>` Sección de contenido que es una parte diferenciada. Es muy similar a `<article>` y por eso son de las que más dudas producen.

`<article>` Contenido que tiene sentido por sí mismo.

`<main>` Contenido principal de la página. Solo puede haber uno por cada página.

`<aside>` Contenido adicional, notas, enlaces relacionados... Generalmente se trata de una zona lateral que complementa a la información principal.

`<footer>` Pié de página.

`<div>` Contenedor genérico, para unir contenido sin necesidad de tener sentido semántico.

10. Metainformación

10.1 Estructura de la cabecera

La metainformación es el conjunto de información de la web indispensable para su funcionamiento y que no se muestra al usuario. Ésta siempre está anidada en la cabeza o `<head>` y por defecto suele contener esta información.

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<title>El título del documento</title>
<link rel="stylesheet" href="#" type="text/css" media="screen" />
<script src="#" type="text/javascript"></script>
</head>
```

En primer lugar se puede observar la codificación usada (UTF-8). Después el título de la página web. Luego un enlace a una hoja de estilos CSS y otro a un JavaScript.

10.2. Metadatos

Además de los elementos mostrados anteriormente también se puede especificar información adicional bajo la etiqueta **<meta>** como el nombre del autor, un apunte para futuros diseñadores de nuestro código, etc...

```
<meta name="autor" content="Jon Rivas" />
```

Definir la codificación de caracteres del documento:

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

Añadir una breve descripción del sitio:

```
<meta name="description" content="Curso de HTML 5" />
```

10.2. Ejercicio 7

Crea un documento HTML con la siguiente estructura y metadatos requeridos:

 <ul style="list-style-type: none">• Inicio• Galería• Noticias• Contacto	Cabecera o menú
Título principal: Estructura y Metadatos <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam, debitis nesciunt a fuga error reiciendis aperiam voluptatem quia, possimus, minima maiores sequi nemo cupiditate deserunt ducimus in labore. Mollitia, velit.</p> Título secundario. <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam, debitis nesciunt a fuga error reiciendis aperiam voluptatem quia, possimus, minima maiores sequi nemo cupiditate deserunt ducimus in labore. Mollitia, velit.</p>	Contenido principal
Sidebar <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus molestiae ullam dolorem dolores repellendus cum at iure quaerat molestias. Aliquid iste atque incidunt modi excepturi placeat consequuntur laborum dolorem tempore.</p>	Contenido lateral
Curso de HTML. 2018 Política de privacidad y Cookies.	Pie de página

En los metadatos ha de verse recogida la siguiente información:

1. Título.
2. Temática de la web, optimizada para motores de búsqueda.
3. Keywords.
4. Enlaces si los hubiera.

11. Otras etiquetas

11.1 Anotaciones

Una herramienta indispensable en el diseño de HTML son los comentarios. Estos comentarios son anotaciones que añadiremos a lo largo del desarrollo de nuestro HTML para que en un futuro el que haya de modificar nuestro código pueda orientarse con mayor facilidad. Toda la información explicativa es poca si viene dada de un modo claro y conciso.

Su funcionamiento es sencillo, al igual que el resto de etiquetas, funciona con un signo de apertura, un contenido y un signo de cierre por lo que procederemos tal que así:

```
<!--Esto es contenido invisible para el usuario final-->
```

11.2. CSS “Id” y “class”

Los atributos “id” y “class” están relacionados de manera directa con CSS.

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

El atributo “id” dota a cada elemento una identificación única en toda la página, así el “id” prueba es único y las propiedades que especifiquemos en un CSS se le atribuyen únicamente a dicho elemento, siendo incorrecto darle ese mismo “id” a otro elemento de la página.

```
<div class="prueba"></div>
<p class="prueba"> Texto de ejemplo</p>
```

En el caso de las clases o "class", su relación es igual con CSS, salvo que los estilos dados a una clase se pueden aplicar a los elementos que deseemos y compartirán las todas las características que les sean posibles .

11.3. Iframes

Ya los mencionamos con anterioridad. La etiqueta <iframe> permite integrar un HTML dentro de otro. Es como ver a través de una ventana en nuestra web parte de otra. Es así como funcionan varias de las integraciones de Google como Google maps o Youtube. Estos son sus atributos más comunes:

- **src**: la URL del contenido.
- **height**: la altura de la ventana o iframe.
- **width**: la anchura.
- **name**: identificador del iframe.
- **longdesc**: una descripción del contenido.
- **scrolling**: (yes, no or auto) determina si se puede hacer scroll en el iframe.

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/YS2-v6nqYQg"
frameborder="0" allow="autoplay; encrypted-media" allowfullscreen>
</iframe>
```

12. Accesibilidad

12.1. Consejos para mejorar la accesibilidad

El objetivo de la accesibilidad web es acercar a cualquier usuario el acceso a los contenidos de nuestra web y facilitarle la navegación pese a las posibles discapacidades que pudiera sufrir.

La meta de un diseñador web ha de ser procurar la mayor compatibilidad con todo tipo de usuarios y dispositivos. Procurando así la posibilidad de llegar a la mayor cantidad de usuarios posibles. Es por eso que hay que pensar en las diferencias entre usuarios, ya sean idiomas, soportes, dispositivos, fisionomía, ...

La W3C aboga por la total accesibilidad en HTML y como no dejamos de hacer hincapié durante el resto del curso, es muy conveniente mantenerse informado de la actualidad en su web, donde podemos encontrar las mejores ayudas para hacer de nuestra página web un ejemplo de accesibilidad. Varios de los requisitos que conforman esta lista del W3C son los siguientes:

- Una gama cromática que facilite la lectura a usuarios con daltonismo o deficiencias visuales.
- La posibilidad de que la web sea totalmente legible sin necesidad de las hojas de estilo CSS.
- La web no debe contener elementos intermitentes o con parpadeos.
- El contenido ha de estar ordenado, jerarquizado y sencillo para su óptima comprensión.
- Especificar los atributos "alt" y "longdesc" y el resto de elementos no textuales.
- Identificar los "iframes" si los hubiera.
- Asegurarse de que la página funciona sin applets o scripts que pudieran no estar disponibles.
- Incluir descripciones sobre videos y audio si los hubiera.

13. Validación

13.1. Importancia de la validación.

La validación es el proceso de comprobación de un documento que determina si existen restricciones o errores en nuestro código. En muchos casos, sobre todo en webs muy complejas, la validación negativa no implica que ésta no funcione, pero se entiende que hay partes del código que se pueden mejorar.

Algunos editores como Adobe Dreamweaver o Visual Studio incorporan sus propios validadores altamente configurables. También los hay de carácter gratuito.

13.2. Validador del W3C

El W3C pone a servicio de los diseñadores de manera gratuita un validador que nos facilitará el correcto diseño de nuestras webs en tan solo un par de pasos.

<http://validator.w3.org/>

- Validate by URI: permite escribir la URL a validar.
- Validate by File Upload: Permite cumplimentar una solicitud para subir una web que aún no se haya subido a la red para ser comprobada.
- Validate by Direct: permite validar el código HTML de forma directa. Permite introducir el código completo o por trozos y detalla los errores.