

2023/24

U.D.2. LENGUAJE DE MARCAS. HTML.

2.1 Lenguaje de marcas: HTML5.

```
23 <meta property="og:title" content="Kostenloses Bild auf Pixabay - Code, Programmierer und mehr"/>
24 <meta name="twitter:card" content="photo">
25 <meta name="twitter:site" content="@pixabay">
26 <meta name="twitter:title" content="Kostenloses Bild auf Pixabay - Code, Programmierer und mehr">
27 <meta name="twitter:image" content="https://cdn.pixabay.com/photo/2017/06/23/10/48/code-2434271_960x.jpg">
28 <meta name="twitter:image:width" content="640">
29 <meta name="twitter:image:height" content="360">
30 <style>#header_inner, #footer_inner { max-width: 1620px; }
31
32 </head>
33 <body>
34   <div id="wrapper">
35     <div id="header"><div id="header_inner">
36       <div class="pure-menu pure-menu-horizontal pure-menu-open">
37         <ul><li class="pure-menu-item pure-menu-selected pure-menu-open"><a href="/de/editors_choice/">Explore</a>
38           <div class="pure-menu pure-menu-horizontal pure-menu-open">
39             <a id="logo" href="/" class="hover_opacity"></a>
40           </div>
41           <form class="media_search header_search" action="/de/photos/" method="get"><div><input type="text" value="Search..." /></div><button type="submit" value="Search"></button>
42           </form>
43         </ul>
44       <div id="content" class="clearfix">
45         <div id="media_show" class="photo">
46           <div class="left"><div class="inside">
47             <div style="text-align:center">
48               <div id="media_container" class="init itemscope itemtype="schema.org/ImageObject">
49                 <meta itemprop="license" content="https://creativecommons.org/licenses/publicdomain/">
50                 <img itemprop="contentURL" srcset="https://cdn.pixabay.com/photo/2017/06/23/10/48/code-2434271_960x.jpg" alt="A dark image showing a person's face, likely a placeholder or a low-quality image." data-size="1920x1080" data-type="image"/>
51                 <div class="overlay"><em>x</em></div>
52                 <div style="height:100%"><table border="1">
53                   <tr>
54                     <td><div id="img_click_overlay"></div></td>
55                     <td><div id="like_buttons_clone"></div></td>
56                     <td><div id="download_menu_clone"></div></td>
57                   </tr>
58                 </table>
59               </div>
60             </div>
61           <div><h1>Foto: Pixabay</h1></div>
62           <span data-href="/de/service/report/photo/2434271/*" data-size="150" data-type="link">Reportar</span>
63         </div>
64       </div>
65     </div>
66   </div>
67 
```



ÍNDICE

1. Lenguaje de marcas. HTML.....	4
1.1 Breve historia HTML.....	5
1.2 Etiquetas de apertura y de cierre.....	6
1.3 Etiquetas vacías o de autocierre	6
2. Introducción a HTML5	7
2.1 Estructura básica.....	7
2.2 Etiquetas obsoletas	8
2.3 Nuevas etiquetas.....	9
3. Etiquetas básicas HTML.....	12
3.1 Cabeceras	12
3.2 Párrafos	12
3.3 Contenido incrustado.....	13
3.3.1 Etiquetas para incrustar contenido	13
3.3.2 Imágenes	14
3.3.3 Vídeo	16
3.3.4 Audio	17
3.3.5 Incrustar un mapa de Google	18
3.4 Rutas a archivos.....	19
3.5 Enlaces.....	19
3.5.1 Atributo download.....	20
3.5.2 Imagen con enlace	20
3.6 Botones	21
3.6.1 Botón con enlace.....	21
3.7 Listas	21
3.7.1 Listas de descripción.....	23
3.8 Salto de línea.....	23
3.9 Línea horizontal.....	24
3.10 Estilos	24
3.11 Formato	25
3.12 Entrecorbillado y citas.....	25



3.13 Comentarios	26
3.14 Colores	27
3.15 Bloques y elementos en línea.....	28
3.16 Clases	28
3.17 Identificadores	29
3.18 Cabecera HTML	31
3.18.1 La etiqueta <meta>	33
3.19 Details y summary	35
3.20 Caracteres especiales	35
4. Layouts	37
5. Formularios	42
5.1 Elementos más comunes	43
5.2 El elemento <input>.....	47
5.3 Atributo pattern y expresiones regulares	54
6. Tablas	58
6.1 Atributos border, align y bgcolor	60
7. Marcos	61
8. Configuración metaetiqueta viewport, web responsive	63
8.1 Ejemplo de la etiqueta <i>viewport</i> que no permite escalar la página	65
8.2 Ejemplo etiqueta <i>viewport</i> que permite escalar la página.....	65
9. Test de verificación HTML y herramientas útiles.....	67
9.1 Validador HTML, W3C.....	67
9.2 HTMLHint, extensión para Visual Studio Code.....	67
9.3 Firefox developer edition	68
9.4 Web developer	68
9.5 CodePen.....	69
10. Resumen de etiquetas HTML	70
10.1 Elementos semánticos	70
10.2 Elementos raíz.....	70
10.3 Metadatos.....	70
10.4 Scripting.....	71



10.5 Secciones	71
10.6 Contenido	72
10.7 Etiquetas de texto	72
10.8 Formulario	73
10.9 Tablas de contenido	74
10.10 Contenido incrustado	75
10.11 Elementos en bloque o block	76
10.12 Elementos en línea o inline	76



1. Lenguaje de marcas. HTML.

Los sitios web están compuestos por páginas que están escritas en algún lenguaje. En la actualidad, debido a que el desarrollo de páginas web está cada vez más presente, existe un creciente desarrollo del sector profesional relacionado con el diseño web. Por lo tanto, comprender, conocer y manejar correctamente el lenguaje de marcas asociado al desarrollo de páginas web es recomendable.

El **lenguaje de marcas** o **lenguaje de marcado** es un tipo de lenguaje que **no permite la implementación de algoritmos y no dispone de funciones aritméticas ni de variables**. En este lenguaje **se combina el texto con etiquetas que contienen información sobre la estructura o presentación del texto**. Algunos **ejemplos** son:

- **HTML** (*HyperText Markup Language*). Es el lenguaje de marcado de hipertexto utilizado en las **páginas web**, el cual está **regulado y estandarizado** por la **W3C** (*World Wide Web Consortium*). Los documentos escritos en este lenguaje sirven para describir la **estructura** de una página web (título, párrafos de texto normal, enumeraciones, definiciones, citas, etc).

Sus características principales son las siguientes:

- Se usa para describir la estructura de una página web.
- Está formado por etiquetas interpretables por los navegadores las cuales indican cómo mostrar el contenido. Normalmente son dos etiquetas, una de apertura y otra de cierre.
- Estas etiquetas podrán incorporar atributos para describir alguna característica o modificar su comportamiento.
- No es sensible a mayúsculas, pero se recomienda usar minúsculas.

La última versión estable de HTML es **HTML5**, el cual veremos en detalle.

- **XML** (*eXtensible Markup Language*). Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. XML no es realmente un lenguaje en particular, es un metalenguaje extensible de etiquetas desarrollado por el W3C (*World Wide Web Consortium*).



A diferencia de HTML en el que los errores sintácticos no producen errores en el navegador y sólo pueden provocar variaciones en la presentación, XML es muy estricto en cuanto a sus normas de sintaxis.

- **XHTML** (*eXtensible HyperText Markup Language*) es un lenguaje que utiliza las mismas etiquetas y atributos que el HTML pero aplicando las reglas de sintaxis de XML. Por tanto, es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

1.1 Breve historia HTML

HTML fue desarrollado por **Tim Berners-Lee** en 1990 como parte del proyecto World Wide Web en el CERN (Consejo Europeo para la Investigación Nuclear). A lo largo de los años, HTML ha evolucionado con nuevas versiones y características para adaptarse a las necesidades cambiantes de la web.

La **última versión estable es HTML5**, que ofrece mejoras en multimedia, interactividad y accesibilidad. HTML ha sido **fundamental en el crecimiento y la expansión de Internet**, permitiendo la creación de contenido digital accesible a nivel mundial.

Las webs implementadas con lenguaje HTML5 puede que no sean correctamente visualizadas en los navegadores antiguos ya que esta versión incluye nuevas etiquetas que no son soportadas.

En el siguiente vídeo puedes conocer de forma breve la **historia de HTML**:
<https://youtu.be/EEtUcYhv30>

¿Sabías que...?

El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) comenzó su desarrollo. Sin embargo, no se publicó ningún documento HTML formal hasta el año 1991. Hoy en día, este documento puede ser consultado online:

w3.org/History/19921103hypertext/hypertext/WWW/MarkUp/Tags.html



1.2 Etiquetas de apertura y de cierre

Todas las etiquetas HTML disponen de los símbolos mayor que y menor que (< >), y normalmente se utilizan en parejas. Una al principio (etiqueta de apertura) y otra al final del texto al que afectan (etiqueta de cierre).

En el siguiente ejemplo se define un párrafo mediante la etiqueta `<p>`. Observa que la etiqueta de cierre del final incluye una barra inclinada.

```
<p>Párrafo</p>
```

1.3 Etiquetas vacías o de autocierre

Las etiquetas vacías o de autocierre en HTML son aquellas que no contienen contenido entre una etiqueta de apertura y cierre. Estas etiquetas se cierran automáticamente dentro de la etiqueta de apertura. Algunos ejemplos de etiquetas vacías o de autocierre son:

- `
`: Salto de línea o línea de quiebre.
- `<hr>`: Línea horizontal.
- ``: Imagen.
- `<input>`: Campo de entrada de datos.
- `<meta>`: Metadatos del documento.
- `<link>`: Enlace a un recurso externo.
- `<wbr>`: Salto de línea sugerido.
- `<area>`: Área de un mapa de imagen.
- `<base>`: URL base para enlaces relativos.

Estas etiquetas se utilizan para insertar elementos específicos en el documento HTML y no requieren una etiqueta de cierre separada



2. Introducción a HTML5

HTML5 es la última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje sólo pueden visualizarse correctamente en los navegadores más actuales, puesto que las versiones anteriores no son capaces de interpretar algunas de sus nuevas etiquetas.

Una de las principales **ventajas de HTML5** es la inclusión de **elementos semánticos o marcados semánticos**, que **nos ayudan a definir las distintas divisiones de una página web**.

2.1 Estructura básica

La **estructura mínima** de un documento HTML es la siguiente:

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8">
        <title>Titulo</title>
    </head>

    <body>
        <header>
            <h1>Pagina HTML5</h1>
        </header>

    </body>
</html>
```



Página HTML5

Por tanto, podríamos decir que los **elementos estructurales** de un documento HTML5 son los siguientes:

- **<!DOCTYPE html>**: indica que el documento contiene código HTML. Es la característica principal de **HTML5**.
- **<html>**: elemento **raíz** de la página. Es una buena práctica indicar el idioma mediante el atributo **lang**.
- **<head>**: zona de **metainformación** de la página. Nos da información adicional de la página. Por tanto, es la parte del código donde:
 - Se **enlazan archivos CSS**.
 - Se **enlazan archivo js**.



- Se añaden **fuentes externas**.
 - Se incluyen **metadatos** (keywords, descriptions, robots, author, owner).
-
- **<title>**: representa el **título del documento**. Se muestra en la barra superior del navegador o en las pestañas de página.
 - **<link>**: enlaza documentos externos, por ejemplo CSS.
 - **<meta>**: define los metadatos que no pueden ser definidos usando otro elemento HTML.
 - **<style>**: usada para escribir CSS interno.
 - **<script>**: define un script interno o un enlace hacia un script externo de JavaScript.
 - **<body>**: **parte visible** de la página. Esta es la zona donde se creará el contenido web con el que podrán interactuar los usuarios.
 - Las etiquetas **suelen ir en parejas**, siendo la primera la de apertura y la segunda la de cierre, como se puede observar en **<h1> ... </h1>** y **<header> ... </header>**.
 - Entre las etiquetas se colocará el contenido, el cual se verá afectado por la acción de dichas etiquetas.

La norma HTML5 permite definir webs sin usar las etiquetas **<html>, **<body>** y **<head>**, aunque no se recomienda.**



Se puede encontrar un **manual completo de HTML5** en la página web de W3C:
<https://www.w3schools.com/html/>

2.2 Etiquetas obsoletas

Como ya se ha comentado, la característica principal de HTML5 es la cabecera **<!DOCTYPE html>**, **pero no es suficiente para tener un código en versión HTML5**, ya que HTML5 es



un conjunto de etiquetas que ha hecho que etiquetas antiguas queden obsoletas. Esto es debido, principalmente a que dichas etiquetas estaban relacionadas con la apariencia, visualización, ..., y de esto se encarga ahora CSS3.

Etiqueta HTML	Definición
<acronym>	Etiqueta que define un acrónimo, palabras clave que dan información a los navegadores o motores de búsqueda. (Utilizar <abbr>)
<applet>	Etiqueta que sirve para incrustar un applet en nuestra página. (Utilizar <object>)
<basefont>	Etiqueta que establece el valor predeterminado del formato de la fuente de todo el texto del documento. (Utilizar CSS)
<big>	Etiqueta que hace destacar el texto haciéndolo más grande respecto del que tiene al lado. (Utilizar CSS)
<center>	Etiqueta que alinea el texto o contenido en el centro de la página. (Utilizar CSS)
<dir>	Etiqueta que define una lista de directorios. (Utilizar)
	Etiqueta que establece el formato de la fuente de algún elemento en concreto. (Utilizar CSS)
<frame>	Etiqueta que define marcos o páginas más pequeñas dentro de nuestra página.
<frameset>	Etiqueta que define el conjunto de marcos o páginas más pequeñas dentro de nuestra página.
<isindex>	Etiqueta que crea un control de entrada de texto. (Utilizar <input>)
<noframes>	Etiqueta que contiene el contenido a mostrar en caso de que el navegador no soporte Frames.
<strike>	Etiqueta que muestra el texto tachado. (Utilizar) (Utilizar CSS)
<tt>	Etiqueta que establece texto de teletipo. (Utilizar CSS)

La utilización de código antiguo u obsoleto es penalizada por motores de búsqueda como Google, y por tanto penaliza en el posicionamiento SEO, por lo que es conveniente utilizar HTML5.

Además, HTML5 **ha eliminado el flash**, el cual, hacia las páginas mucho más lentes, y no permitía a los motores de búsqueda entender bien qué es lo que tenía la página web.



2.3 Nuevas etiquetas

Antes de HTML5, la inmensa mayoría de las páginas hacían uso de las etiquetas <**div**> o <**span**> para agrupar los distintos componentes de un documento o *Layout*. A través de atributos id o class podíamos especificar el propósito o tipo de bloque, dando lugar a identificadores típicos como header, content, sidebar, footer, etc.



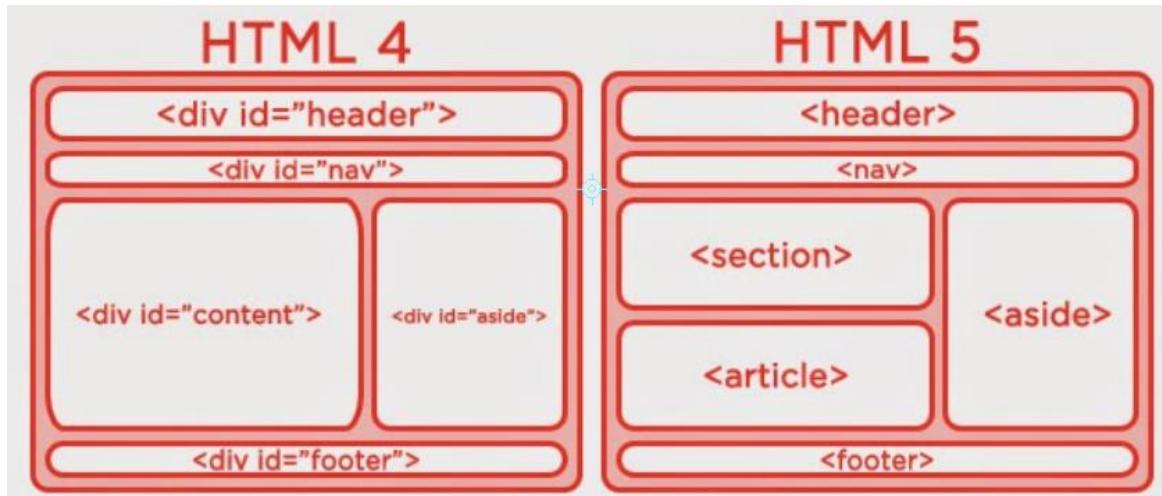
Todas estas secciones tan comunes hechas con bloques genéricos han dado lugar a las nuevas etiquetas HTML5. Estas permiten además que los motores de búsqueda analicen una página y sepan directamente lo que es la cabecera, el contenido principal y las secciones del sitio, entre otros aspectos.

Este sería un resumen de las nuevas etiquetas que nos permiten, además de agrupar una parte de la página, darle un **valor semántico**, es decir, la sección tiene un significado concreto: es una cabecera, una barra lateral, etc. Es por ello que se le **denominan etiquetas o elementos semánticos**

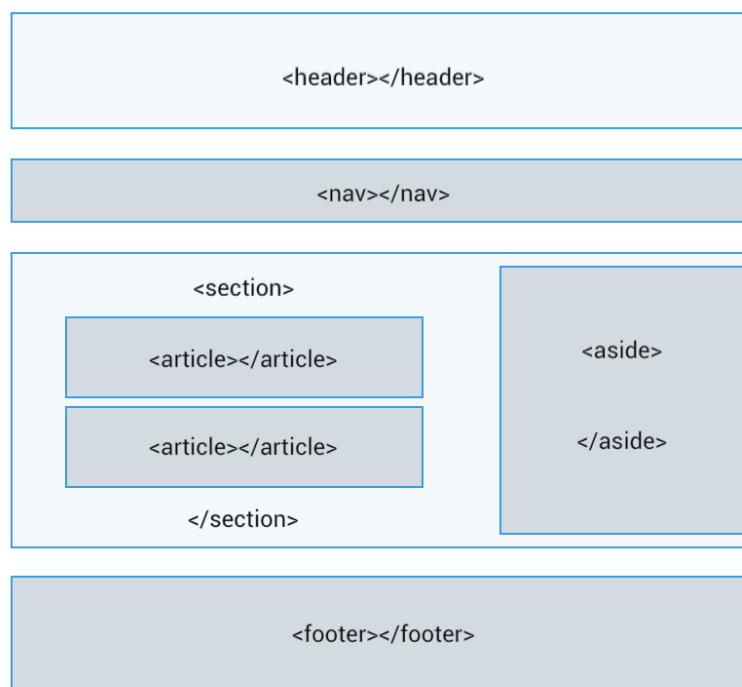
Elemento semántico	Descripción
<code><main></code>	Engloba el contenido principal de la página o de la aplicación. Solo puede haber una etiqueta <code><main></code> en el documento.
<code><header></code>	Representa la cabecera o inicio de la página donde se presenta la web, o bien puede ser la cabecera de una sección <code><section></code> o artículo <code><article></code> largo.
<code><footer></code>	Representa el pie o parte final de una página o de una sección o un artículo largo.
<code><nav></code>	Crea un menú de navegación , que enlaza con las diferentes partes del sitio web. Normalmente se sitúa justo debajo del encabezado o a la izquierda de la página.
<code><article></code>	Agrupa un contenido independiente de la página que podría ser extraído de la misma sin que se perdiera su significado . El ejemplo más típico sería un artículo de un blog. Puede estar subdividido en más partes: <ul style="list-style-type: none">• Cabecera <code><header></code>• Cuerpo <code><section></code>• Pie de artículo <code><footer></code>
<code><section></code>	Elemento genérico que permite agrupar distintas secciones de la página, generalmente acompañada de un <code><header></code> . También puede utilizarse para separar distintas partes de un <code><article></code> .
<code><time></code>	Representa la hora y la fecha en una página web.



<code><aside></code>	Elemento que nos permite indicar que una parte de la página es de menor importancia o que simplemente es complementaria a la parte principal. Define la barra lateral de una página web.
<code><figure></code> <code><figcaption></code>	Nos permite agrupar una imagen con su explicación.



Esquema comparativo entre una estructura HTML4 y HTML5.



Esquema básico de un documento HTML5



3. Etiquetas básicas HTML

En este apartado se verán las etiquetas básicas de HTML5. Recuerda que tienes el manual completo en: <https://www.w3schools.com/html/>

3.1 Cabeceras

Las cabeceras son las frases principales de la página web y se definen con las etiquetas que van desde `<h1>` hasta `<h6>`, de más grande a más pequeña.

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

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

</body>
</html>
```

Cabecera 1
Cabecera 2
Cabecera 3
Cabecera 4
Cabecera 5
Cabecera 6

3.2 Párrafos

Los párrafos de texto se definen con la etiqueta `<p>`, aunque podemos usar `<pre>`, en caso de que queramos que el contenido se muestre de forma idéntica a como se preformatea en el fichero html.

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

<p>Esto es un parrafo.</p>
<p>Esto es otro parrafo</p>

<pre>
  esto es
  un parrafo
  preformatizado
</pre>
```

Esto es un parrafo.
Esto es otro parrafo
esto es
un parrafo
preformatizado



```
</body>  
</html>
```

3.3 Contenido incrustado

El **contenido incrustado** se utiliza para **mostrar recursos externos** como, por ejemplo, mapas, previsiones meteorológicas, fórmulas matemáticas, vídeos y audios, entre otros. Este método permite utilizar un servicio o interfaz de terceros.

3.3.1 Etiquetas para incrustar contenido

Algunas de las etiquetas más importantes para incrustar contenido son las siguientes:

Elemento	Descripción
<code></code>	Representa una imagen.
<code><iframe></code>	Representa un contexto anidado de navegación, es decir, un documento HTML embebido.
<code><embed></code>	Representa un punto de integración para una aplicación o contenido interactivo externo que por lo general no es HTML.
<code><object></code>	Representa un recurso externo, que será tratado como una imagen, un sub-dокументo HTML o un recurso externo a ser procesado por un plugin.
<code><param></code>	Define parámetros para el uso por los plugins invocados por los elementos <code><object></code> .
<code><video></code>	Representa un vídeo, y sus archivos de audio y canciones asociadas, con la interfaz necesaria para reproducirlos.
<code><audio></code>	Representa un sonido o <i>stream de audio</i> .
<code><source></code>	Permite a autores especificar recursos multimedia alternativos para los elementos multimedia como <code><video></code> o <code><audio></code> .
<code><track></code>	Permite a autores especificar una pista de texto temporizado para elementos multimedia como <code><video></code> o <code><audio></code> .
<code><canvas></code>	Representa un área de mapa de bits en el que se pueden utilizar scripts para renderizar gráficos.
<code><map></code>	En conjunto con <code><area></code> , define un mapa de imagen.
<code><area></code>	En conjunto con <code><map></code> , define un mapa de imagen.
<code><svg></code>	Define una imagen vectorial embebida.
<code><math></code>	Define una fórmula matemática.



A continuación, se muestra un ejemplo que utiliza alguna de las etiquetas vistos en el cuadro anterior.

```
<!DOCTYPE html>
<html>
<head>
    <title>Ejemplo de Incrustación de Contenido</title>
</head>
<body>
    <h1>Ejemplo de Incrustación de Contenido</h1>

    <!-- Incrustación de una imagen -->
    

    <!-- Incrustación de un video -->
    <video src="video.mp4" controls></video>

    <!-- Incrustación de un audio -->
    <audio src="audio.mp3" controls></audio>

    <!-- Incrustación de una página web externa -->
    <iframe src="https://www.ejemplo.com"></iframe>

    <!-- Incrustación de contenido externo utilizando plugins -->
    <embed src="contenido.swf">

    <!-- Incrustación de contenido multimedia o plugins -->
    <object data="contenido.swf" type="application/x-shockwave-flash"></object>

    <!-- Incrustación de un lienzo para dibujar gráficos -->
    <canvas id="miCanvas" width="200" height="200"></canvas>

    <!-- Incrustación de gráficos vectoriales escalables -->
    <svg width="400" height="180">
        <rect x="50" y="20" width="300" height="100" style="fill:blue" />
    </svg>
</body>
</html>
```

3.3.2 Imágenes

Para **insertar imágenes** en una web se usa la etiqueta ****, **sin etiqueta de cierre**, junto con los atributos:

- **src**: especifica la ruta o URL de la imagen.
- **alt**: texto alternativo que se mostrará en caso de no poder cargar la imagen.

Esta es importante de cara al posicionamiento SEO.



- **width**: ancho de la imagen en píxeles o como un porcentaje del ancho disponible.
- **height**: alto de la imagen en píxeles o como un porcentaje de la altura disponible.
- **title**: proporciona texto descriptivo que se muestra al pasar el cursor sobre la imagen.
- **border**: especifica el ancho del borde alrededor de la imagen.
- **align**: especifica la alineación de la imagen con respecto al texto circundante.

Si la imagen no está en el mismo directorio que la página habrá que especificar una ruta hacia dicha imagen. En caso de que la imagen esté en otro servidor, se especificará la URL. Se recomienda guardar todas las imágenes en un único directorio, denominado **/imágenes**, o similar.

La etiqueta **** permite gifs estáticos y animados.

A continuación, se muestran un par de ejemplos:

Ejemplo 1.

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



</body>
</html>
```

Se mostraría la foto indicada con el tamaño indicado en pixels. Si la foto no se pudiera cargar, se mostraría la cadena “www.fotosplayas.com”

Ejemplo 2. La imagen se carga desde el archivo “imagen.jpg”. Si la imagen no se puede cargar, se mostrará el texto “Descripción de la imagen”. Al pasar el cursor sobre la imagen, se mostrará el texto “Título de la imagen”. La imagen tendrá un ancho de 300 píxeles y una altura de 200 píxeles. Habrá un borde de ancho 1 alrededor de la imagen. La imagen estará alineada a la izquierda del texto circundante.

```

```



Para **enmarcar** una foto en un documento se utiliza el elemento **<figure>**, junto con **<figcaption>** para darle un **subtítulo** a la foto.

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

<h1>The figure and figcaption element</h1>

<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy.
</figcaption>
</figure>

</body>
</html>
```

The figure and figcaption element



Fig.1 - Trulli, Puglia, Italy.

3.3.3 Vídeo

Mediante la etiqueta **<video>** podremos insertar vídeos en las páginas web.

- ⇒ El atributo **controls** permitirá añadir los **controles** típicos de **pausa**, **play** o **volumen**.
- ⇒ Se recomienda el uso de los atributos **width** y **height** para evitar parpadeos mientras el vídeo carga.
- ⇒ El elemento **<source>** permite especificar un vídeo alternativo el cual podrá ser elegido por el navegador en caso del que el vídeo principal tenga un formato no válido.
- ⇒ Se puede incluir el atributo **autoplay** para que el vídeo comience automáticamente.

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

  <video width="320" height="240"
controls autoplay>
  <source src="movie.mp4"
type="video/mp4">
  <source src="movie.ogv"
type="video/ogg">
</video>
</body>
</html>
```





Al utilizar vídeos internos, puede ser que no se carguen en el navegador web del usuario, puesto que cada navegador tiene códecs distintos. Es por ello que se recomienda utilizar vídeos de **Youtube** siempre y cuando sea posible, ya que de esta manera nos aseguramos de que vayan a funcionar siempre, además de ayudarnos con el posicionamiento SEO.

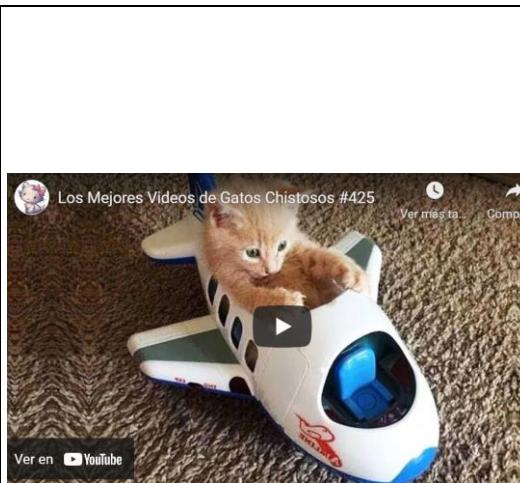
- ⇒ Podemos insertar vídeos desde ubicaciones web usando la etiqueta **<iframe>**.
- ⇒ **Youtube** proporciona el **<iframe>** ya creado para copiarlo y pegarlo directamente a través de la opción **compartir → insertar**.

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

    <iframe width="560" height="315"
src="https://www.youtube.com/embed/47bP_RTO
KQQ" title="YouTube video player"
frameborder="0" allow="accelerometer;
autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
allowfullscreen>

    </iframe>

</body>
</html>
```



3.3.4 Audio

Mediante la etiqueta **<audio>** podremos insertar audio en las páginas.

- ⇒ **<source>, controls** y **autoplay** funcionan como en el caso del elemento **<video>**.

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

    <audio controls>
        <source src="sonido.ogg" type="audio/ogg">
        <source src="sonido.mp3" type="audio/mpeg">
    </audio>
```





```
</body>  
</html>
```

3.3.5 Incrustar un mapa de Google

Para insertar un mapa de Google, se utiliza la etiqueta **<iframe>**, al igual que se hace cuando queremos incrustar un vídeo de Youtube.

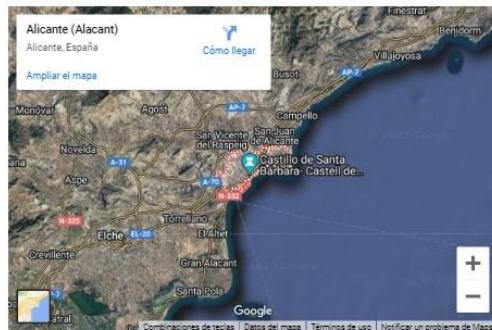
1. Accede a las indicaciones, al mapa o a la imagen de Street View que quieras insertar.
2. En la parte superior izquierda, haz **clic en Menú** .
3. Haz clic en **Compartir o insertar el mapa**.
4. Haz clic en **Insertar un mapa**.
5. A la izquierda del cuadro de texto, haz **clic en el ícono de la flecha hacia abajo**  para elegir el tamaño que deseas.
6. Copia el texto en el cuadro y pégalo en el HTML de tu sitio web o blog.

En el siguiente código se muestra cómo incrustar una imagen, un mapa de Google Maps y un vídeo de Youtube.

```
<!DOCTYPE html>  
<html>  
<body>  
<h3>Etiqueta iframe</h3>  
  
<iframe  
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d154214.5103364689!  
2d-  
0.5427351669515812!3d38.35795447718404!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.  
1!3m3!1m2!1s0xd6235da3b9dab4b%3A0x1d7da872ac0b81e3!2sAlicante+(Alacant)%2C+Al  
icante%2C+Espa%C3%B1a!5e1!3m2!1ses!2sus!4v1564299065614!5m2!1ses!2sus"  
width="600" height="400" frameborder="0" style="border:0"  
allowfullscreen></iframe>
```



Etiqueta iframe



3.4 Rutas a archivos

La ruta a un archivo se puede especificar de dos maneras:

- **absoluta**: indicando toda la ruta desde la raíz de la estructura de directorios hasta el archivo en cuestión.

```

```

- **relativa**: indicando la ruta desde la carpeta actual en la que se encuentra la página que hace la referencia al archivo buscado. Este es el modo de referenciar archivos recomendado, ya que no estará ligado a una URL específica.

```

```

3.5 Enlaces

Los **enlaces o hipervínculos** se definen con la etiqueta `<a>`, modificada a través de su atributo `href`.

Los enlaces pueden ser a páginas web, si ponemos la **URL absoluta**, o a un elemento local, si usamos una **URL relativa** (sin <http://www>).

El atributo `target` permite especificar dónde se abrirá el enlace. Las opciones más comunes son:

- `_blank`: en una nueva ventana o pestaña.
- `_self`: (por defecto). En la ventana o pestaña actual.



```
<!DOCTYPE html>
<html>
<body>
    <h2>Enlaces HTML</h2>
    <p>Uso de la etiqueta "a"</p>

    <a href="http://www.google.es" target="_blank">Enlace a google</a>
</body>
</html>
```

Enlaces HTML

Uso de la etiqueta "a"

[Enlace a google](http://www.google.es)

3.5.1 Atributo download

El atributo **download** en HTML se utiliza para especificar que un enlace debe ser descargado al hacer clic en él en lugar de navegar hacia él. Permite proporcionar un nombre de archivo sugerido para el archivo descargado. Así es como funciona el atributo **download**:

```
<a href="ruta/al/archivo.pdf" download="miarchivo.pdf">Descargar PDF</a>
```

En este ejemplo, cuando el usuario hace clic en el enlace «Descargar PDF», en lugar de abrirse en el navegador, se descargará el archivo especificado en el atributo «href» (<ruta/al/archivo.pdf>). El atributo «download» especifica el nombre de archivo sugerido para el archivo descargado ([miarchivo.pdf](#)).

3.5.2 Imagen con enlace

En el ejemplo siguiente, como no existe la imagen indicada, se muestra el texto alternativo.

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

    <p style="font-size:20">Image Links</p>
    <p>La imagen es un enlace.</p>

    <a href="http://www.google.com" >
        
    </a>
```

Image Links

La imagen es un enlace.





```
</body>  
</html>
```

3.6 Botones

Se pueden **insertar botones** en una web mediante la etiqueta **<button>**.

```
<!DOCTYPE html>  
<html>  
<body>  
  <h2> Este es un boton simple </h2>  
  <button>Haz click aqui</button>  
</body>  
</html>
```

Este es un boton simple

Haz click aqui

3.6.1 Botón con enlace

```
<!DOCTYPE html>  
<html>  
<body>  
  
  <h2> Este es un boton con enlace a  
google </h2>  
  <button onclick="document.location =  
'http://www.google.com'">Ir a  
google</button>  
  
</body>  
</html>
```

Este es un boton con enlace a google

Ir a google

3.7 Listas

Las dos **etiquetas principales** para listas son:

- 1) **** para listas **sin orden numérico**.
- 2) **** para listas **con orden numérico**.

⇒ Las listas **se pueden anidar**.

⇒ Para los **elementos** que forman las listas se usará la etiqueta ****.



⇒ Con el atributo **type** podemos definir el **tipo de numeración** que aparecerá al lado de los elementos de la lista, como números comunes, romanos (**type="i"**; **type="I"**), etc.

Atributos “type” más comunes	
Listas ordenadas 	<ul style="list-style-type: none">«1»: Numeración decimal (1, 2, 3, ...).«A»: Letras mayúsculas (A, B, C, ...).«a»: Letras minúsculas (a, b, c, ...).«I»: Números romanos mayúsculos (I, II, III, ...).«i»: Números romanos minúsculos (i, ii, iii, ...).
Listas no ordenadas 	<ul style="list-style-type: none">«disc»: Puntos sólidos (●, ●, ●, ...).«circle»: Círculos (○, ○, ○, ...).«square»: Cuadrados (■, ■, ■, ...).

- ⇒ Con el atributo **start** (para listas ordenadas) podemos indicar en qué número queremos que comience la numeración.
- ⇒ Con el atributo **reversed** (para listas ordenadas) invertimos el orden de numeración de la lista.

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

    <h1>Lista sin orden numérico</h1>

    <ul>
        <li>Elemento A</li>
        <li>Elemento B</li>
        <li>Elemento C</li>
    </ul>

    <h1>Lista ordenada
    numéricamente</h1>

    <ol>
        <li>Elemento A</li>
        <li>Elemento B</li>
        <li>Elemento C</li>
    </ol>
```

Lista sin orden numérico

- Elemento A
- Elemento B
- Elemento C

Lista ordenada

1. Elemento A
2. Elemento B
3. Elemento C



```
</body>  
</html>
```

3.7.1 Listas de descripción

Las **listas de descripción** son listas formadas por términos junto con la **definición o descripción de cada término**. Para crearlas, se utilizan las etiquetas **<dl>** y, **<dt>** y **<dd>**.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<dl>  
  <dt>Dia</dt>  
  <dd>- sale el sol</dd>  
  <dt>Noche</dt>  
  <dd>- no sale el sol</dd>  
</dl>  
  
</body>  
</html>
```

Dia
- sale el sol
Noche
- no sale el sol

3.8 Salto de línea

**
** es la etiqueta de salto de línea y **no tiene etiqueta de cierre**. También se puede usar de esta forma **
** o incluso de esta, **</br>**.

```
<!DOCTYPE html>  
<html>  
<body>  
  
  <p>Esto es una linea <br> y esto  
otra.</p>  
  
</body>  
</html>
```

Esto es una linea
y esto otra.



3.9 Línea horizontal.

<hr> permite insertar una línea horizontal. **No tiene etiqueta de cierre.**

<pre><!DOCTYPE html> <html> <body> <p title="txt1">texto</p> <hr> <p title="txt2">mas texto.....</p> </body> </html></pre>	texto
	mas texto.....

En el ejemplo anterior, atributo **title** mostrará un mensaje al pasar el ratón por encima del párrafo.

3.10 Estilos

Los estilos permiten cambiar los atributos relacionados con el **diseño visual** de una web.

Aunque la forma recomendada es a través de **archivos .css**, también se puede usar, junto con la etiqueta **html**, el atributo **style** en línea y el par **atributo:valor**

```
<etiqueta style="atributo:valor;">
```

Algunos **atributos** de estilo son:

- **color**: color del texto.
- **background-color**: color de fondo de un elemento.
- **font-family**: fuente de texto.
- **font-size**: tamaño de la fuente.
- **text-align**: alineamiento del texto.

<pre><!DOCTYPE html> <html> <body> <p style="color:red;">rojo</p> <p style="font-size:50px;">Tam de</pre>	rojo
--	------



```
fuente 36 pixels</p>  
  
</body>  
</html>
```

Tam de fuente 36 pixels

3.11 Formato

Las etiquetas de formato permiten modificar la **apariencia visual del texto**:

⇒ ****: negrita.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>The b element</h1>  
  
<p>This is normal text - <b>and this is bold text</b>.</p>  
  
</body>  
</html>
```

The b element

This is normal text - **and this is bold text**.

⇒ ****: texto importante. Equivalente a negrita, pero añadiendo importancia semántica.

⇒ **<i>**: cursiva.

⇒ ****: texto enfatizado. Equivalente a cursiva, pero añadiendo importancia semántica.

⇒ **<mark>**: texto subrayado.

⇒ **<small>**: texto pequeño.

⇒ ****: texto tachado

⇒ **<ins>**: texto subrayado.

⇒ **<sub>**: subíndice.

⇒ **<sup>**: superíndice.

3.12 Entrecomillado y citas

Para el entrecomillado y citas, utilizamos las siguientes **etiquetas**:



- **<q>**: inserta comillas dobles.

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

<h1>The q element</h1>

<p>WWF's goal is to:<br/>
<q>Build a future where people live in harmony with<br/>
nature.</q><br/>
We hope they succeed.</p>

</body>
</html>
```

- **<blockquote>**: marca una sección obtenida de alguna fuente. El texto puede aparecer con sangría.

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

<h1>The blockquote element</h1>

<p>Here is a quote from WWF's website:</p>

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

</body>
</html>
```

The q element

WWF's goal is to: "Build a future where people live in harmony with nature." We hope they succeed.

The blockquote element

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

3.13 Comentarios

Los comentarios permiten comentar el código html. El navegador los obvia a la hora de generar la página.

Se considera un comentario todo lo que esté encerrado entre **<!-- -->**

```
<!-- Esto es un comentario dentro de un documento HTML -->

<!--
   /$$  /$$ /$$$$$$$$$ /$$      /$$ /$$
 | $$ | $$|__ $$__/| $$$ /$$$| $$
 | $$ | $$ | $$ | $$$ /$$$| $$
 |$$$$$$$ | $$ | $$ $$/$$ $$| $$
```



```
| $$__ $$ | $$ | $$ $$$| $$| $$  
| $$ | $$ | $$ | $$\ $ | $$| $$  
| $$ | $$ | $$ | $$ \/ | $$|$$$$$$$  
| __/ | __/ | __/ | __/ | __/ | ____/  
-->
```

<!--¿Te están entrando ganas de hacer ASCII Art con tu marca? Puedes hacerlo en: <http://patorjk.com/software/taag> -->

3.14 Colores

Para dar color se debe hacer con CSS, aunque en HTML existen formas comunes de especificar los colores en html:

- **Hexadecimal:** comienzan con el símbolo **#**, seguido de **6 dígitos hexadecimales**.
- **RGB:** **rgb (compR, compG, compB)**. Cada componente de 0 a 255.
- **Nombres:** Se nombra directamente le color, **red, blue**, etc.

```
<!DOCTYPE html>  
<html>  
<body>  
  
    <p style="color:blue">Esto es un parrafo  
azul nombrado</p>  
    <p style="color:#ff024a">Parrafo con color  
en hexadecimal</p>  
    <p style="color:rgb(30,240,200)">Parrafo  
con color RGB</p>  
  
</body>  
</html>
```

Esto es un parrafo azul nombrado

Parrafo con color en hexadecimal

Parrafo con color RGB



3.15 Bloques y elementos en línea

Todos los elementos HTML tienen una forma de **mostrarse por defecto**, dependiendo del tipo de elemento que sea. Las dos formas en las que se pueden mostrar los elementos son:

- **bloque.**
- **en línea.**

Un **bloque** comienza siempre en una nueva línea y ocupa todo el ancho disponible. Los bloques genéricos se declaran con la etiqueta **<div>**.

Un elemento **en línea** no comienza una nueva línea y solo ocupará el espacio que necesita. Los elementos en línea genéricos se declaran con la etiqueta ****.

- ⇒ Se usa normalmente **dentro de un párrafo** para luego destacar su contenido con estilos.
- ⇒ **El contenido es lineal**, sin saltos de línea, por ello no se comporta como un bloque independiente, sino que se integra en el texto.

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

    <div>
        <p>London is the
            <span style="color:red">capital city of
        England</span>. It is the most populous
        city in the United Kingdom, with a metropolitan
        area of over 13 million inhabitants.</p>
    </div>

</body>
</html>
```

London is the **capital city of England**. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

3.16 Clases

El atributo **class** se usa para establecer los **mismos estilos a elementos diferentes**, pero pertenecientes a la misma clase. De esta forma, a dos elementos HTML distintos, definidos con la misma clase, se les aplicará los mismos estilos.



El nombre que se da a una clase debe empezar por el carácter punto ‘.’ y **es sensible a mayúsculas**.

```
<!DOCTYPE html>
<html>
<head>
<style>
.bloque {
background-color: black;
color: white;
margin: 20px;
padding: 20px;
}

.grande{
font-size: 36px;
}

</style>
</head>

<body>
<div class="bloque">
<p class="grande">Londres</p>
<p>Londres es la capital de Inglaterra</p>
</div>

<div class="bloque">
<p class="grande">Paris</p>
<p>Paris es la <span
class="grande">capital</span> de Francia.</p>
</div>

</body>
</html>
```

Londres

Londres es la capital de Inglaterra

Paris

Paris es la capital de Francia.

3.17 Identificadores

Se pueden asignar **IDs** a elementos HTML de forma que estos queden **identificados de forma única**. Para ello se usará el atributo ***id***.

El nombre que se da a un identificador debe empezar por el carácter ‘#’ y **es sensible a mayúsculas**.



La **diferencia** entre las clases y los identificadores es que **un identificador está asociado a un único elemento de la página mientras que una clase puede estar asociada a varios elementos**.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        #parrafol {
            background-color: lightblue;
            color: black;
            padding: 40px;
            text-align: left;
        }

        .estilo1 {
            background-color: tomato;
            color: white;
            padding: 10px;
        }
    </style>
</head>

<body>
    <pre id="parrafol">
Este parrafo tiene el id:
parrafol y ningun otro elemento
podra usar este ide</pre>
    <pre class="estilo1">
Este parrafo pertenece a la clase estilo1,
la cual puede ser usado por otros elementos
    <p class="estilo1">Este otro
parrafo utiliza tambien la clase
estilo1</p>

</body>
</html>
```

Este parrafo tiene el id:
parrafol y ningun otro elemento
podra usar este ide

Este parrafo pertenece a la clase estilo1,
la cual puede ser usado por otros elementos

Este otro parrafo utiliza tambien la clase estilo1

También se pueden usar los **ids para crear marcas en zonas de las páginas** a las que se pueda ir directamente a través de un link.



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

<p><a href="#C6" target="blank">Capitulo 6</a></p>
<p><a href="#C8">Capitulo 8</a></p>

<pre>
.
.
.
</pre>

<p id="C6">Capitulo 6</p>
<pre>
.
.
.
</pre>

<p id="C8">Capitulo 8</p>

</body>
</html>
```

3.18 Cabecera HTML

Como ya se comentó anteriormente, la cabecera de un documento HTML es la zona de los **metadatos o metainformación** y está marcada por la etiqueta **<head>**.

La **metainformación** en HTML se utiliza para **proporcionar información adicional** sobre un documento HTML. Esta información **no se muestra directamente en la página web**, pero puede ser utilizada por los **motores de búsqueda, los navegadores y otros sistemas** para entender y clasificar el contenido del documento.

En esta zona podremos usar las siguientes **etiquetas**. Es importante tener en cuenta que las **metatags pueden variar** dependiendo de las necesidades y estándares específicos de cada sitio.



- **<title>**: título de la página. Aparecerá en la barra de título del navegador y en los resultados de búsqueda.
- **<style>**: zona de estilos definidos en la propia página.
- **<link>**: zona para enlazar diferentes archivos, como CSS o Javascript.
- **<meta>**: zona para proporcionar información tipo autor, palabras clave, etc. Puede ser usada por los motores de búsqueda.
- **<script>**: zona para funciones Javascript definidas en la propia página.

```
<!DOCTYPE html>
<html>

    <head>
        <title>Titulo de la pagina</title>

        <style>
            body {background-color: powderblue; }
            h1 {color: red; }
            p {color: blue; }
        </style>

        <link rel="holaDeEstilos" href="misEstilos.css">

        <meta charset="UTF-8">
        <meta name="description" content="Pagina de pruebas HTML">
        <meta name="keywords" content="HTML,CSS,XML,JavaScript">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <meta name="author" content="Pepe Perez">

        <script>
            function myFunction {
document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
        </script>

    </head>

</html>
```

- **<base>**: especifica la URL base y/o el destino de todas las URL relativas de un documento. La etiqueta **<base>** debe tener un atributo **href** o un atributo **target**, o ambos. **Sólo puede haber un único elemento <base>** en un documento web.



```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
    <base href="imagenes/" target="_blank" >
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>base</title>
    <link rel="stylesheet" type="text/css" href="css/
estilos.css">
</head>

<body>
    <h2>Base Tag</h2>
    
</body>
</html>
```

3.18.1 La etiqueta <meta>

Algunos de los **atributos** más comunes utilizados en la etiqueta **<meta>** son los siguientes:

- 1) **<meta charset="UTF-8">**. Especifica la codificación de caracteres utilizada en el documento. UTF-8 es la codificación más comúnmente utilizada para admitir caracteres multilingües.
- 2) **<meta name="description" content="Descripción del sitio">**. Proporciona una **breve descripción del contenido** del sitio web. Es utilizado por los motores de búsqueda para **mostrar información** sobre el sitio en los resultados de búsqueda.
- 3) **<meta name="keywords" content="palabra clave1, palabra clave2">**. Especifica **palabras clave relacionadas con el contenido** del sitio web. Aunque no es tan relevante para los motores de búsqueda modernos, aún puede ser utilizado para **conocer la temática** del sitio web.
- 4) **<meta name="viewport" content="width=device-width, initial-scale=1.0">**. Permite definir el área visible de la web para visualizarlas en móviles o tablets, correctamente, en caso de que las páginas no estén optimizadas para dichos dispositivos.



dispositivos. **Es importante para lograr un diseño responsive**. Dado a su importancia, más adelante veremos todas sus opciones.

- 5) `<meta name="author" content="Nombre del autor">`. Especifica el nombre del autor del documento.
- 6) `<meta name="copyright" content="Año de copyright, nombre del titular de los derechos">`. Indica los derechos de autor del documento.
- 7) `<meta name="robots" content="index,follow">`. Indica a los motores de búsqueda si deben indexar y seguir los enlaces dentro del documento.

A continuación, se muestra un **ejemplo** en donde se usa la etiqueta `<meta>` con diferentes atributos.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="description" content="Descripción del sitio">
    <meta name="keywords" content="palabra clave1, palabra clave2">
    <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
    <meta name="author" content="Nombre del autor">
    <meta name="copyright" content="2023, Nombre del titular de los
    derechos">
    <meta name="robots" content="index,follow">

    <title>Metainformación del Sitio</title>
</head>
<body>
    <!-- Contenido del sitio web -->
</body>
</html>
```



3.19 Details y summary

La etiqueta `<details>` tiene como finalidad mostrar información adicional de algún elemento de la página web. El funcionamiento de esta etiqueta consiste en un widget que oculta y muestra información, similar a un menú con efecto acordeón.

La etiqueta `<summary>` incorpora un sumario de la etiqueta `<details>`, es decir, añade información complementaria.

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

<h1>The details element</h1>

<details>
  <summary>Epcot Center</summary>
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</details>

</body>
</html>
```

The details element

▼ Epcot Center

Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.

3.20 Caracteres especiales

Dentro de una página web no podemos poner directamente los caracteres «<» Y «>» porque el navegador los confundiría con los caracteres que se utilizan para la creación de etiquetas HTML.

Para cierto tipo de caracteres, es necesario utilizar otra sintaxis o codificación de caracteres. A continuación, veremos la representación ASCII de los símbolos mediante entidades, código decimal y hexadecimal.

1. **Entidades HTML:** Las entidades HTML son secuencias de caracteres que representan símbolos especiales en HTML. Comienzan con un ampersand (&), seguido de un nombre o número y terminan con un punto y coma (;). Las entidades HTML son fáciles de leer y entender, lo que las hace convenientes para usar en el código HTML.

2. **Códigos decimales:** Los códigos decimales son valores numéricos que representan símbolos en la tabla ASCII. Estos códigos se escriben en la forma «&#x;», donde «x» es el número decimal correspondiente al símbolo. Los códigos



decimales son útiles cuando no se puede utilizar una entidad HTML específica o cuando se necesita una representación numérica precisa del símbolo.

3. **Códigos hexadecimales:** Los códigos hexadecimales son valores numéricos en base 16 que representan símbolos en la tabla ASCII. Los códigos hexadecimales son similares a los códigos decimales, pero utilizan una representación en base 16 en lugar de base 10.

En la siguiente tabla se muestran algunos símbolos y su correspondiente código decimal y su entidad HTML:

Símbolo	Código decimal	Entidad
&	&	&
<	<	<
"	"	"
>	>	>
©	©	©
÷	÷	÷
€	€	€
←	←	←

Aquí puedes ver una tabla ASCII estándar con los códigos y los nombres de las entidades en HTML: ascii.cl/es/codigos-html.htm. En el siguiente enlace puedes ver más símbolos especiales webusable.com/CharsExtendedTable.htm

A continuación, se presenta un ejemplo de cómo se puede representar un carácter o símbolo especial de diferentes formas:

```
&#8592; <!--*Representa una flecha hacia  
la izquierda mediante código -->  
  
&larr; <!--*Representa una flecha hacia  
la izquierda mediante una entidad HTML-->  
  
← <!--*Representa una flecha mediante la  
copia del símbolo-->
```

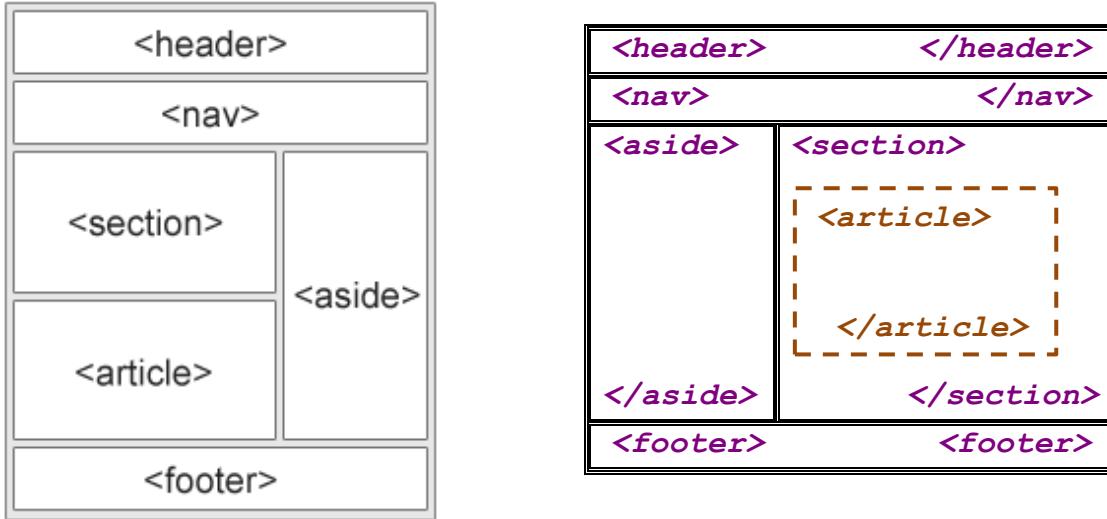
← ← ←



4. Layouts

El layout define la distribución de los elementos de la página por **zonas con nombre propio**.

En HTML5 existe una etiqueta para cada una de las zonas, como ya hemos visto, denominadas **etiquetas semánticas** ya que su propio nombre aclara su contenido.



- **<header>**: sección que define la **parte superior o cabecera** de la web. Se utiliza para dar la bienvenida, por ello es muy común incluir el logotipo y el título principal. Últimamente, también es común incluir el menú de navegación en el **<header>**.
- **<nav>**: contenedor para pestañas principales de **navegación**. Para formar la lista de opciones del menú de navegación, normalmente se usan las etiquetas **** o **** y **** dentro de **<nav>**.
- **<main>**: Representa el **contenido principal** del 'body' de un documento. El área principal consiste en el contenido que está directamente relacionado con el tema central de un documento. Este contenido debe ser único para ese documento. Es importante tener en cuenta que **solo debe hacer un elemento <main>** en un documento, y **no debe descender de un elemento <article>, <aside>, <footer>, <header> o <nav>**.
- **<section>**: para definir una sección de **contenido relacionado**. Un **<section>** puede contener varios elementos de tipo **<p>** o **<article>**. Por tanto, las etiquetas **<section>** definen las áreas más globales que va a tener la página.



- **<article>**: para definir un artículo o **elemento indivisible o independiente**, como un **post** de un foro. Se pueden definir tantos elementos de este tipo como información vaya a contener la página web. **Pueden estar subdivididas en más partes**: cabecera ('header'), cuerpo ('section') y pie de artículo ('footer').
- **<aside>**: para definir secciones aparte del contenido principal. Muestra **información** que puede estar **relacionada** con el contenido de la página web, pero **no es parte del contenido**. Pueden estar en los laterales.
- **<footer>**: es **el pie de la página**, y sirve para **cerrar la página**. Da información muy genérica. Zona de contacto y similar.
En páginas muy grandes, se podría justificar el uso de <footer> para cerrar cada una de las secciones. **Lo normal es tener solo uno**.
- **<div>**: es la etiqueta más usada en html para estructurar contenido en **bloque**. Se usa para **agrupar subconjuntos** de elementos que no tienen un nombre o etiqueta semántica propia.
Se puede usar dentro de otras etiquetas como <footer>, <section>... El objetivo real de la agrupación que hacemos con <div> es asignarle un estilo con CSS, ya que es más cómodo que asignar estilos elemento por elemento.





Se usará **CSS** para **posicionar** cada una de las secciones anteriores correctamente por la página creando así el layout.

Un **ejemplo** de estructura utilizando etiquetas semánticas podría ser el siguiente:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>

  </head>
  <body>
    <header>
      <h1></h1>
      <nav>
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </nav>
    </header>
    <section>
      <article></article>
      <article></article>
      <article></article>
    </section>
    <footer>

    </body>
  </html>
```

A continuación, se presenta un ejemplo de creación de un layout con estilos.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Pagina de prueba</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    * {
      box-sizing: border-box;
    }

    body {
      font-family: Arial, Helvetica, sans-serif;
    }

    header {
      background-color: #666;
```



```
padding: 30px;
text-align: center;
font-size: 35px;
color: white;
}

nav {
    float: left;
    width: 30%;
    height: 300px; /* only for demonstration, should be removed */
    background: #ccc;
    padding: 20px;
}

nav ul {
    list-style-type: none;
    padding: 0;
}

article {
    float: left;
    padding: 20px;
    width: 70%;
    background-color: #f1f1f1;
    height: 300px; /* only for demonstration, should be removed */
}

section:after {
    content: "";
    display: table;
    clear: both;
}
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}

</style>
</head>
<body>

<header>
    <p>Ciudades</p>
</header>
```



```
<section>
    <nav>
        <ul>
            <li><a href="#">Londres</a></li>
            <li><a href="#">Paris</a></li>
        </ul>
    </nav>

    <article>
        <p>Londres</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae ipsum dui. Nullam consectetur facilisis odio blandit pellentesque. Cras tincidunt tellus eu nibh viverra, vel lobortis dui imperdiet. Proin vehicula vulputate velit, ac accumsan leo lacinia ut. Donec vitae diam in nisi pellentesque dapibus venenatis ut ligula. Ut consequat, odio nec sodales volutpat, ligula nunc pellentesque elit</p>
    </article>
</section>

<footer>
    <p>Footer</p>
</footer>

</body>
</html>
```

Ciudades

[Londres](#)
[Paris](#)

Londres

orem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae ipsum dui. Nullam consectetur facilisis odio blandit pellentesque. Cras tincidunt tellus eu nibh viverra, vel lobortis dui imperdiet. Proin vehicula vulputate velit, ac accumsan leo lacinia ut. Donec vitae diam in nisi pellentesque dapibus venenatis ut ligula. Ut consequat, odio nec sodales volutpat, ligula nunc pellentesque elit

Footer



5. Formularios

Los formularios son **documentos interactivos** que se utilizan para **recoger información** en un sitio web. Esta **información es enviada al servidor donde es procesada**. Cada formulario contiene uno o varios tipos de controles, que permiten recolectar la información de varias formas diferentes, desde el ingreso de una palabra en un cuadro de texto, hasta la subida de un archivo. Se crean con la etiqueta `<form>`.

Se pueden ver todos los atributos de `<form>` en https://www.w3schools.com/tags/tag_form.asp. Los principales son los siguientes:

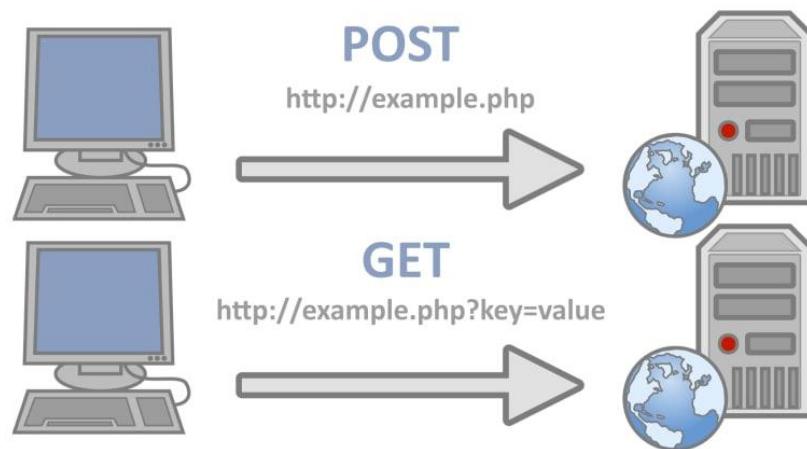
- A. **`action`**: permite especificar cuál será el **archivo que procese** el formulario. Es decir, especifica dónde enviar los datos del formulario cuando éste se envía.
- B. **`enctype`**: permite especificar **cómo están codificados** los datos del formulario.
- C. **`method`**: permite especificar el **método HTTP que se usará para enviar los datos** a los archivos especificados en el atributo `action`. Los **métodos disponibles** son:

⇒ **`get`**:

- Pasa los datos del formulario **a través de la URL** en pares **nombre/valor**.
- Es útil cuando queremos **permitir que el usuario cree un enlace de favoritos** a dichos datos.
- Cada navegador permite pasar una **cantidad limitada de datos por get**, por lo que hay que tener cuidado en este aspecto,
- La **información** pasada por get es **visible**, por lo que no se debe de pasar por `get` información sensible.

⇒ **`post`**:

- Pasa los datos del formulario como una **transacción**.
- **No permite añadir a favoritos los datos** ya que no se pasan a través de la URL.
- Es un método **más robusto y seguro** que `get`, además de no tener límite en la cantidad de datos que puede manejar.



D. ***target***: permite especificar **dónde se mostrará la respuesta recibida** después de enviar el formulario. Sus valores pueden ser:

- ⇒ ***_blank***: en una **nueva ventana**.
- ⇒ ***_self***: en la **misma ventana** del formulario o marco en caso de usar **frames**.
- ⇒ ***_parent***: en la **ventana padre** en caso de usar **frames**.
- ⇒ ***_top***: en la **ventana actual destruyendo los marcos** en caso de usar **frames**.

5.1 Elementos más comunes

Dentro de la estructura formada por **<form> </form>** podemos utilizar los siguientes elementos:

- 1) **<fieldset>** y **<legend>**. Estas etiquetas se usan para **agrupar elementos** de formulario relacionados y **proporcionar un título o leyenda descriptiva** para el grupo.

La etiqueta **<fieldset>** actúa como un contenedor que agrupa campos del formulario, **dibujando un cuadro** alrededor de estos.



Un **ejemplo** de la aplicación de dichos atributos es el siguiente:

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

<h1>The fieldset element</h1>

<form action="/proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">

<fieldset>
<legend>Personalia:</legend>
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname"><br><br>
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname"><br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>
<label for="birthday">Birthday:</label>
<input type="date" id="birthday" name="birthday"><br><br>
<input type="submit" value="Submit">
</fieldset>
</form>

</body>
</html>
```

The fieldset element

Personalia:

First name:

Last name:

Email:

Birthday: dd/mm/aaaa

En este ejemplo, la etiqueta **<fieldset>** envuelve el grupo de elementos de formulario relacionados con la información de contacto. La etiqueta **<legend>** se utiliza como una leyenda o título para el fieldset, proporcionando una descripción del grupo.

El uso de etiquetas **<fieldset>** y **<legend>** ayuda a **mejorar la accesibilidad y usabilidad de los formularios** al proporcionar una estructura clara y agrupación visual de campos relacionados. También mejora el estilo general y la organización del formulario.

- 2) **<input>**: permite **crear los objetos del formulario**. Estos se especificarán en su atributo **type**, cuyos valores se verán más adelante.

Todos los elementos **<input>** **llevarán un nombre especificado en el atributo name** para poder ser enviados. Este atributo lo veremos con más detalle más adelante.

- 3) **<label>**: permite **crear etiquetas** para poder describir la función de los elementos de entrada que formen los formularios.



Podemos vincular una etiqueta a un elemento HTML mediante el atributo **for** de **<label>**, especificando en dicho atributo el valor de **id** del elemento HTML.

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

The input element

First name:

Last name:

Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php".

4) **<select>**: lista de elementos seleccionables.

- Cada elemento de la lista se crea con la etiqueta **<option>** y se le asigna un valor a través del atributo **value**.
- Para marcar una opción por defecto se pondrá el atributo **selected="selected"**.
- Con el atributo **size** se podrá especificar el número de elementos que se muestran a la vez.
- A través del atributo **multiple="multiple"** se permite la selección de más de un elemento. Para ello habrá que mantener la tecla **Mayúsculas** presionada.

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

  <h1>The select element</h1>

  <form action="/prod_form.php" method="post" target="_blank"
        enctype="multipart/form-data">

    <label for="cars">Selecciona un coche:</label>
    <br />
    <select id="cars" name="cars" size="4" multiple="multiple">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
      <option value="opel">Opel</option>
      <option value="audi">Audi</option>
      <option value="seat">Seat</option>
      <option value="kia">Kia</option>
    </select>
    <br><br>
    <input type="submit" value="Submit">
  </form>

</body>
</html>
```

The select element

Selecciona un coche:

Volvo
Saab
Opel
Audi



- 5) **<textarea>**: campo de **entrada de texto multilínea**. A través de los atributos **rows** y **cols** se podrá especificar el **tamaño en filas y columnas**, respectivamente.

El texto dentro de la etiqueta **<textarea>** se mostrará inicialmente dentro del área de texto y los usuarios podrán editarlo y seleccionarlo.

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

<h1>The textarea element</h1>

<form action="/proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">
    <textarea name="descripcion_coche" rows="3" cols="20"
placeholder="describe tu coche" > </textarea>
    <br>
    <input type="submit" value="Submit">
</form>

</body>
</html>
```

The textarea element

describe tu coche

Submit

- 6) **<button>**: define un **botón**. La **acción** que realizará el botón se especificará en el atributo **onclick**.

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

<h2>The onclick Event</h2>

<p>Click to trigger a function that will output "Hello World":</p>
<button onclick="myFunction()">Click me</button>

<p id="demo"></p>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello World";
}
</script>

</body>
</html>
```

The onclick Event

Click to trigger a function that will output "Hello World":

Click me

También se puede utilizar el atributo **type** para definir el **tipo de botón**. Algunos valores comunes para el atributo **type** son:

- **button** (por defecto): un botón estándar.
- **submit**: un botón que envía un formulario.
- **reset**: un botón que reinicia los valores iniciales de los campos de un formulario.

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



7) **<output>**: se usa para **mostrar resultados de salida**, como los devueltos por un script o el resultado de un cálculo.

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

<h1>The output element</h1>

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
<input type="range" id="a" value="50">
+<input type="number" id="b" value="25">
=<output name="x" for="a b"></output>
</form>

</body>
</html>
```

The output element



5.2 El elemento <input>

La forma en que **<input>** funciona viene determinada por el valor del atributo **type**. El valor predeterminado de **<input>** es “**text**”, permitiendo el ingreso de una única línea de texto.

Todos los elementos **<input>** llevarán un nombre especificado en el atributo **name** para poder ser enviados.

A continuación, se recogen los **valores más utilizados para el atributo type**. La lista completa se puede encontrar en https://www.w3schools.com/tags/tag_input.asp

- **text** (**<input type="text">**): cuadro de texto de una sola línea.

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

<form action="proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">

<fieldset style="width: 400px;">
<legend> Datos personales </legend>
<label for="nombre_text">Nombre:</label>
<input type="text" id="nombre_text" name="nombre_text"
placeholder="introduce tu nombre">
<br /><br />
<label for="apellidos_text">Nombre:</label>
<input type="text" id="apellidos_text" name="apellidos_text"
placeholder="introduce tus apellidos">
</fieldset>

</form>

</body>
</html>
```

Datos personales

Nombre:

Nombre:



- **button** (`<input type="button">`): define un **botón**. La acción que realizará el botón se especificará en el atributo **onclick**.

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

<h1>Show a Push Button</h1>

<p>The button below activates a JavaScript when it is clicked.</p>
<form>
  <input type="button" value="Click me" onclick="msg()">
</form>

<script>
function msg() {
  alert("Hello world!");
}
</script>

</body>
</html>
```

Show a Push Button

The button below activates a JavaScript when it is clicked.

Un **botón** creado mediante la etiqueta `<input type="button">` sólo puede contener **texto**, mientras que un botón creado mediante la etiqueta `<button>` permite incluir elementos html como imágenes. El contenido del botón `<input type="button">` se define mediante el atributo **value**, por lo que sólo puede contener texto.

- **checkbox** (`<input type="checkbox">`): crea un elemento de **selección múltiple**.

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

<form action="proc_form.php" method="post" target="_blank" enctype="multipart/form-data">
  <fieldset style="width: 400px;">
    <legend> Intereses </legend>
    <input type="checkbox" id="lectura" name="lectura" value="Lectura">
    <label for="lectura">Lectura</label>
    <input type="checkbox" id="cine" name="cine" value="cine">
    <label for="cine">Cine</label>
    <input type="checkbox" id="deporte" name="deporte" value="Deporte">
    <label for="deporte">Deporte</label>
  </fieldset>
</form>
</body>
</html>
```

Intereses

Lectura Cine Deporte

- **radio** (`<input type="radio">`): **grupo de elementos de elección única**. Poner el mismo nombre en el atributo **name** en cada uno de los elementos `<input>`



de este tipo, creará el grupo de **radiobuttons** de forma que cada **elección será exclusiva**.

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

<h1>Display Radio Buttons</h1>

<form action="/action_page.php">
    <p>Please select your favorite Web language:</p>
    <input type="radio" id="html" name="fav_language" value="HTML">
    <label for="html">HTML</label><br>
    <input type="radio" id="css" name="fav_language" value="CSS">
    <label for="css">CSS</label><br>
    <input type="radio" id="javascript" name="fav_language" value="JavaScript">
    <label for="javascript">JavaScript</label>

    <br>

    <p>Please select your age:</p>
    <input type="radio" id="age1" name="age" value="30">
    <label for="age1">0 - 30</label><br>
    <input type="radio" id="age2" name="age" value="60">
    <label for="age2">31 - 60</label><br>
    <input type="radio" id="age3" name="age" value="100">
    <label for="age3">61 - 100</label><br><br>
    <input type="submit" value="Submit">
</form>

</body>
</html>
```

Display Radio Buttons

Please select your favorite Web language:

- HTML
- CSS
- JavaScript

Please select your age:

- 0 - 30
- 31 - 60
- 61 - 100

- **file** (`<input type="file">`): selector para **subir archivos**.
- **submit** (`<input type="submit">`): botón para **enviar los datos** del formulario al archivo manejador especificado en el atributo **action** de la etiqueta `<form>`.
- **reset** (`<input type="reset">`): botón que pone todos los campos del formulario a sus **valores por defecto**.

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

<h1>File, submit y reset</h1>

<form action="proc_form.php" method="post" target="_blank"
      enctype="multipart/form-data">

    <label for="archivo">Seleccionar archivo:</label>
    <input type="file" id="archivo" name="archivo">
    <br /><br />
    <input type="submit" value="Enviar Datos">
    <input type="reset" value="Borrar Datos">

</form>

</body>
</html>
```

File, submit y reset

Seleccionar archivo: Ninguno ...hivo selec.



- **datalist**: desplegable de opciones predefinida para un elemento `<input>`. Para hacer esta asociación entre los dos elementos el valor del atributo `list` de `<input>` debe ser igual al `id` del `datalist`.

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

<h1>The datalist element</h1>

<form action="/action_page.php" method="get">
    <label for="browser">Selecciona un navegador de la lista:</label>
    <input list="browsers" name="browser" id="browser">
    <datalist id="browsers">
        <option value="Edge">
        <option value="Firefox">
        <option value="Chrome">
        <option value="Opera">
        <option value="Safari">
    </datalist>
    <input type="submit">
</form>

</body>
</html>
```

The datalist element

Selecciona un navegador de la lista:

Enviar

- Edge
- Firefox
- Chrome
- Opera
- Safari

- **password** (`<input type="password">`): campo para **introducir contraseñas**.
- **email** (`<input type="email">`): **introducción de emails**. Suele incorporar **validación por parte del navegador**.
- **number** (`<input type="number">`): campo para introducir **valores numéricos**.

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

<h1>Email, password y number </h1>

<form action="proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">
    <fieldset style="width: 450px;">
        <legend>Personal Data </legend>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email"><br><br>
        <label for="pwd">Password:</label>
        <input type="password" id="pwd" name="pwd" minlength="8"><br><br>
        <label for="age">Age:</label>
        <input type="number" id="age" name="age" min="1" max="150"><br><br>
        <input type="submit">
    </fieldset>
</form>

</body>
</html>
```

Email, password y number

Personal Data

Email:

Password:

Age:

Enviar

- **color** (`<input type="color">`): crea un elemento de **selección de color**.



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

<h1>Show a Color Picker</h1>

<form action="/action_page.php">
    <label for="favcolor">Select your favorite color:</label>
    <input type="color" id="favcolor" name="favcolor" value="#ff0000"><br>
<br>
    <input type="submit">
</form>

</body>
</html>
```

Show a Color Picker

Select your favorite color:

Enviar

R 255 G 0 B 0

- **date** (`<input type="date">`): elemento para introducción de fechas.

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

<h1>Show a Date Control</h1>

<form action="proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">
    <label for="birthday">Cumpleaños:</label>
    <input type="date" id="birthday" name="birthday">
    <input type="submit">
</form>

</body>
</html>
```

Show a Date Control

Cumpleaños: dd/mm/aaaa □ Enviar

octubre de 2022 ↑ ↓

L	M	X	J	V	S	D
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Borrar Hoy

- **datetime-local** (`<input type="datetime-local">`): elemento para introducir fecha y hora.

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

<h1>Datetime-local</h1>

<form action="proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">
    <label for="tiempo_local">Introduce fecha y hora:</label>
    <input type="datetime-local" id="tiempo_local" name="tiempo_local">
    <input type="submit">
</form>

</body>
</html>
```

Datetime-local

Introduce fecha y hora: dd/mm/aaaa -- : -- Enviar

octubre de 2022 ↑ ↓

20	16
21	17
22	18
23	19
24	20
31	01
Borrar	Hoy

- **range** (`<input type="range">`): barra de desplazamiento para seleccionar un valor entre un rango.



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

<h1>Range</h1>

<form action="proc_form.php" method="post" target="_blank"
enctype="multipart/form-data">

<fieldset style="width: 450px;">
<label for="puntos"> Puntos (entre 0 y 10):</label>
<input type="range" id="puntos" name="puntos" min="0" max="10">

</form>

</body>
</html>
```

Range

Puntos (entre 0 y 10):

- **search** (`<input type="search">`): campo de **búsqueda**.

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

<h1>Display a Search Field</h1>

<form action="/action_page.php">
<label for="gsearch">Search Google:</label>
<input type="search" id="gsearch" name="gsearch">
<input type="submit">
</form>

</body>
</html>
```

Display a Search Field

Search Google: Enviar

- **tel** (`<input type="tel">`): campo de entrada de **números de teléfono**.

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

<h1>Display a Telephone Input Field</h1>

<form action="/action_page.php">
<label for="phone">Introduce tu numero de telefono:</label>
<br><br>
<input type="tel" id="phone" name="phone" placeholder="123-45-678"
pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required><br><br>
<small>Formato: 123-45-678</small><br><br>
<input type="submit">
</form>

</body>
</html>
```

Display a Telephone Input Field

Introduce tu numero de telefono:

Formato: 123-45-678

- **url** (`<input type="url">`): campo para **introducir URLs**. Suele incorporar **validación por parte del navegador**.



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

<h1>URL Input Field</h1>

<form action="/action_page.php">
  <fieldset style="width: 450px;">
    <label for="homepage">Añade tu pagina de inicio:</label>
    <input type="url" id="homepage" name="homepage"><br><br>
    <input type="submit">
  </fieldset>
</form>

</body>
</html>
```

URL Input Field

Añade tu pagina de inicio:

- **time** (`<input type="time">`): campo para introducir la **hora**.

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

<h1>Time Input Control</h1>

<form action="/action_page.php">
  <label for="appt">Select una hora:</label>
  <input type="time" id="appt" name="appt">
  <input type="submit">
</form>

</body>
</html>
```

Time Input Control

Select una hora:

20	34
21	35
22	36
23	37
00	38
01	39
02	40

- **week** (`<input type="week">`): permite seleccionar **semana y año**.

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

<h1>Week Input Control</h1>

<form action="/action_page.php">
  <label for="week">Select una semana:</label>
  <input type="week" id="week" name="week">
  <input type="submit">
</form>

</body>
</html>
```

Week Input Control

Select una semana:

octubre de 2022 ▾ ↑ ↓

Semana	L	M	X	J	V	S	D
39	26	27	28	29	30	1	2
40	3	4	5	6	7	8	9
41	10	11	12	13	14	15	16
42	17	18	19	20	21	22	23
43	24	25	26	27	28	29	30
44	31	1	2	3	4	5	6

- **month** (`<input type="month">`): permite seleccionar **mes y año**.



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

<h1>Month Input Control</h1>

<form action="/action_page.php">
  <label for="bdaymonth">Cumpleaños (mes y año):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">
  <input type="submit">
</form>

</body>
</html>
```

Month Input Control

Cumpleaños (mes y año): Enviar

2022

ene.	feb.	mar.	abr.
may.	jun.	jul.	ago.
sep.	oct	nov.	dic.

Borrar Este mes

Además de **type**, hay más **atributos aplicables** a las etiquetas **<input>**:

- **checked**: especifica que un **checkbox** o **radiobutton** tiene que estar **preseleccionado**.
- **disabled**: especifica que un **campo** tiene que estar **deshabilitado**.
- **min**: especifica un **valor mínimo** para un campo.
- **max**: especifica un **valor máximo** para un campo.
- **maxlength**: especifica un **número máximo de caracteres** en campos de texto.
- **pattern**: especifica una **expresión regular** para validar la entrada.
- **readonly**: especifica que un **campo** es de **solo lectura**.
- **required**: especifica que un **campo** es **obligatorio**.
- **size**: especifica el **ancho en caracteres** de un campo.
- **step**: especifica el **incremento de un campo numérico**.
- **value**: **valor por defecto** para un campo.
- **placeholder**: da una **pista de la información que debe contener** un campo **<input>**.
- **autofocus**: **asigan el foco** en el campo marcado automáticamente al cargar la página.
- **autocomplete**: indica si se debe usar o no la opción de autocompletar.

5.3 Atributo pattern y expresiones regulares

El atributo **pattern** nos permite **definir nuestras propias reglas para validar el valor de entrada** de los campos usando **expresiones regulares** o **regexp** (contracción de las palabras inglesas *regular expression*).



Veamos un **resumen** de algunos de los caracteres que nos ayudan a construir expresiones regulares:

Clases de caracteres	
.	Cualquier carácter excepto salto de línea
[abc]	Cualquiera de los caracteres entre corchetes
[^abc]	Que NO sea cualquiera de los caracteres entre corchetes
[a-g]	Cualquier carácter entre a y g (en minúscula)
(a b)	a o b

ANCLAS

^abc	Comienzo de una línea
abc\$	Final de una línea

CARACTERES ESPECÍFICOS

\w \d \s	Palabra, dígito, espacio en blanco
\W \D \S	Que NO sea palabra, dígito o espacio en blanco
\t \n \r	Tabulador, salto de línea, retorno de carro

CUANTIFICADORES

a* a+ a?	0 o más veces, 1 o más veces, 0 o 1 vez
a{5} a{2,}	Solo 5, 2 o más
a{1,3}	Entre 1 y 3

En la siguiente página web podemos ver una referencia de patrones y expresiones regulares. Además, permite la validación de expresiones regulares: [regexr.com](https://www.regexr.com)

A continuación, se pueden ver algunos **ejemplos** con las expresiones regulares comúnmente utilizadas en atributos HTML:

- 1) **Validación de correo electrónico.** Este patrón comprueba que el valor de entrada sea una dirección de correo electrónico válida.

```
<input type="email" pattern="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" required>
```



- 2) Validación de URL.** Este patrón asegura que el valor de entrada comience con «http://» o «https://».

```
<input type="url" pattern="https?://.+>
```

- 3) Entrada numérica.** Este patrón permite solo valores numéricos en el campo.

```
<input type="number" pattern="[0-9]+>
```

- 4) Patrón personalizado.** Este patrón especifica un formato personalizado donde la entrada debe comenzar con tres letras seguidas de tres dígitos.

```
<input type="text" pattern="[A-Za-z]{3}\d{3}>
```

- 5) Validación de código postal.** Este patrón asegura que el valor de entrada sea un código **postal** válido de 5 dígitos.

```
<input type="text" pattern="[0-9]{5}>
```

- 6) Validación de fecha (formato AAAA-MM-DD):** Este patrón valida que el valor de entrada sea una fecha en el formato AAAA-MM-DD. No valida la lógica de los valores de fecha, solo el formato de la cadena.

```
<input type="text" pattern="\d{4}-\d{2}-\d{2}>
```

- 7) Validación de nombre de usuario (solo caracteres alfanuméricos):** Este patrón permite solo caracteres alfanuméricos para el nombre de usuario.

```
<input type="text" pattern="[A-Za-z0-9]+>
```



8) En este ejemplo se muestra un formulario que solicita los siguientes datos:

- Nombre de 40 caracteres como máximo y sólo permite letras y números.
- Código postal que admite 5 números.
- Edad (solo dos dígitos numéricos).

Además, todos los campos son obligatorios y disponen de placeholder. Además, el foco se encuentra en el primer campo.

```
<h2>Formulario de Contacto</h2>

<form>

<input name="nombre" type="text" placeholder="Nombre" pattern="[a-zA-Z]{1,40}" required autofocus/>

<input name="cp" type="text" placeholder="Código postal"
pattern="[0-9]{5}" required />

<input name="edad" type="text" pattern="[0-9]{2}"
placeholder="Edad" required />

<button id="enviar" name="enviar" type="submit">ENVIAR</button>

</form>
```

Formulario de Contacto

Nombre	Código postal	Edad	ENVIAR
--------	---------------	------	--------



6. Tablas

Las tablas permiten **mostrar la información de manera más organizada**. Para la creación de una tabla, en primer lugar, se definen las características de la tabla y, a continuación, fila a fila, se indica el valor de cada uno de los elementos de cada columna, es decir, el contenido de cada celda.

- Las tablas se definen con la etiqueta **<table>** y sus **elementos** con las siguientes etiquetas:
 - **<th>**: cabecera.
 - **<tr>**: fila.
 - **<td>**: celda.
 - **<caption>**: permite poner un **título** a la tabla. Si se usa, **debe ir justo después de la etiqueta <table>**. Para poner el título de la tabla **debajo** de la misma, se usará el **<caption align="bottom">**.
 - **<thead>**: se utiliza para agrupar el contenido del encabezado en una tabla HTML. La etiqueta **<thead>** se utiliza junto con los elementos **<tbody>** y **<tfoot>** para especificar cada parte de una tabla (encabezado, cuerpo, pie).

Los navegadores pueden utilizar estos elementos para permitir el desplazamiento del cuerpo de la tabla independientemente del encabezado y el pie de página. Además, al imprimir una tabla grande que abarca varias páginas, estos elementos pueden permitir que el encabezado y el pie de la tabla se impriman en la parte superior e inferior de cada página.



```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>

<h1>The thead, tbody, and tfoot elements</h1>

<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
</table>
```

The thead, tbody, and tfoot elements

Month	Savings
January	\$100
February	\$80
Sum	\$180

- Se usarán los atributos **colspan** y **rowspan** para, columnas y filas que ocupan más de una posición, respectivamente.
- Si no se especifica, la tabla aparecerá sin bordes. Para ello se usarán **estilos (CSS)**, preferiblemente.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    table, th, td, tr { border: 2px solid black; }
  </style>
</head>

<body>
<table>
  <tr>
    <th colspan="2">Tabla Personas</th>
```



```
</tr>
<tr>
    <th>Nombre</th>
    <th>Edad</th>
</tr>
<tr>
    <td>Pepe</td>
    <td>26</td>
</tr>
<tr>
    <td>Ana</td>
    <td>41</td>
</tr>

</table>
</body>
</html>
```

Aqui el titulo	
Tabla Personas	
Nombre	Edad
Pepe	26
Ana	41

6.1 Atributos border, align y bgcolor

Para dar estilos a los elementos utilizaremos siempre CSS y no atributos HTML. No obstante, se van a mencionar estos atributos porque todavía se encuentran presentes en multitud de proyectos online.

Verás que en muchos códigos se le añade un borde a la tabla usando el atributo **border**. Sin embargo, **este atributo ya no es aceptado en HTML5**. A pesar de ello, los navegadores reconocen este atributo por estar definido en versiones antiguas de HTML.

```
<table border="2">
```

Para modificar la **posición** de la tabla también se utilizaba el atributo **align**.

```
<table align="center">
```

También podemos asignar **fondos** a las celdas y filas de una tabla con el atributo **bgcolor**.

```
<td bgcolor="#77DDEE">
```



7. Marcos.

Aunque HTML5 deja obsoleto el uso de los marcos, a continuación se describe brevemente su funcionamiento, puesto que, en algún caso, pueden encontrarse sitios web así implementados, sobre los que se haga necesario definir mejoras o en los que deban llevarse a cabo mantenimientos.

Los marcos consisten en ventanas independientes incorporadas dentro de la página general; de esta forma, cada página quedará dividida en varias subpáginas, lo que permite realizar un diseño mucho más organizado y limpio a la vista. Este tipo de documentos, en lugar de tener las partes habituales head y body, **incorpora un frameset en lugar del body**.

Para llevar a cabo la construcción de estos marcos, se utiliza la etiqueta **<frameset>** y, a continuación, se indica la disposición del número de nuevas ventanas; en concreto, en primer lugar, se indica el número de columnas, así como su disposición dentro de la página principal. La etiqueta **<frame>** se utiliza para definir las características que va a tomar cada uno de los marcos, de forma individual.

La sintaxis de los marcos para la creación de columnas, es la siguiente:

```
<frameset cols=anchoColumna1, anchoColumna2, ...anchoColumnaN>
  <frame ...>
  <frame ...>
</frameset>
```

De esta forma quedan definidas las columnas, que darán lugar a los frames que componen la página web.

```
<frameset cols=10%,20%,70%>
  <frame src="1">
  <frame src="2">
  <frame src="3">
</frameset>
```

Además de esta distribución por columnas, también se crearán filas, obteniendo con la combinación de ambas, así cualquier tipo de diseño de marcos que se le ocurra al diseñador. La sintaxis utilizada es similar a la anterior, modificando el atributo 'cols' por 'rows' en la etiqueta **<frameset>**.

```
<frameset cols=30%,70%>
  <frame src="1">
  <frameset rows=40%,*>
    <frame src="2">
    <frame src="3">
  </frameset>
</frameset>
```



A continuación, se presenta un **ejemplo** simple.

Código de la página principal:

```
<html>
<head>
<title>Diseño con frames 2</title>
</head>
<frameset rows="100,* ,100">
    <frame src="pagina1.html" />
    <frameset cols="15%,*,15%">
        <frame src="pagina4.html" />
        <frame src="pagina2.html" />
        <frame src="pagina5.html" />
    </frameset>
    <frame src="pagina3.html" />
</frameset>
</html>
```

Código de la página 2:

```
<html>
<head>
<title>página2</title>
</head>
<body bgcolor="silver">
Página 2: esta va a ser el contenido.
</body>
</html>
```

Código de la página 1:

```
<html>
<head>
<title>página1</title>
</head>
<body bgcolor="aqua">
Página 1: esta va a ser la cabecera.
</body>
</html>
```

Código de la página 3:

```
<html>
<head>
<title>página3</title>
</head>
<body bgcolor="yellow">
Página 3: esta va a ser el pie de página.
</body>
</html>
```

Código de la página 4:

```
<html>
<head>
<title>página4</title>
</head>
<body bgcolor="lime">
Página 4: esta va a ser el menú lateral izquierdo.
</body>
</html>
```

Código de la página 5:

```
<html>
<head>
<title>página5</title>
</head>
<body bgcolor="fuchsia">
Página 5: esta va a ser la columna lateral derecha.
</body>
</html>
```

Resultado:

Página 1: esta va a ser la cabecera.

Página 4: esta va a ser el menú lateral izquierdo.

Página 2: esta va a ser el contenido.

Página 5: esta va a ser la columna lateral derecha.

Página 3: esta va a ser el pie de página.



8. Configuración metaetiqueta viewport, web responsive

La declaración **viewport** nos permite definir los parámetros de visualización de una página web en los diferentes dispositivos. Consiste en una metaetiqueta mediante la que se establece si se puede hacer zoom en una página, el **zoom inicial** o la **anchura de la pantalla** del dispositivo.

Cuando trabajamos con una **web responsive** es necesario **definir un viewport adecuado**, de lo contrario es muy probable que la página no lea correctamente los media queries y que se vea en formato muy reducido, siendo necesario hacer zoom para ver el contenido.

La etiqueta **viewport** se define en el elemento `<head>` del documento HTML y puede contener diferentes atributos para ajustar el comportamiento de la página en dispositivos móviles.

```
<meta name="viewport" content="...>
```

Las **propiedades** más comunes utilizadas en el contenido de la metaetiqueta **viewport** son:

- 1) **“width”**. Especifica el **ancho de la pantalla o del viewport**. Por ejemplo, **“width=device-width”** establece el ancho inicial del viewport para que coincida con el ancho del dispositivo.
- 2) **“initial-scale”**. Define el **nivel de escala inicial** del **viewport**. Por ejemplo, **“initial-scale=1.0”** establece que la página se mostrará inicialmente sin ninguna escala.
- 3) **“minimum-scale”** y **“maximum-scale”**. Estos atributos permiten establecer los **límites mínimos y máximos de escala** del viewport, controlando si el usuario puede hacer zoom en la página.
- 4) **“user-scalable”**: Este atributo permite **habilitar o deshabilitar la capacidad del usuario para hacer zoom** en la página. Si se establece en “yes”, el usuario podrá hacer zoom; si se establece en “no”, se deshabilitará el zoom.



5) **Height**. Altura virtual de la pantalla o altura del *viewport*.

Un **ejemplo** de cómo se vería la metaetiqueta “**viewport**” en el código HTML es el siguiente:

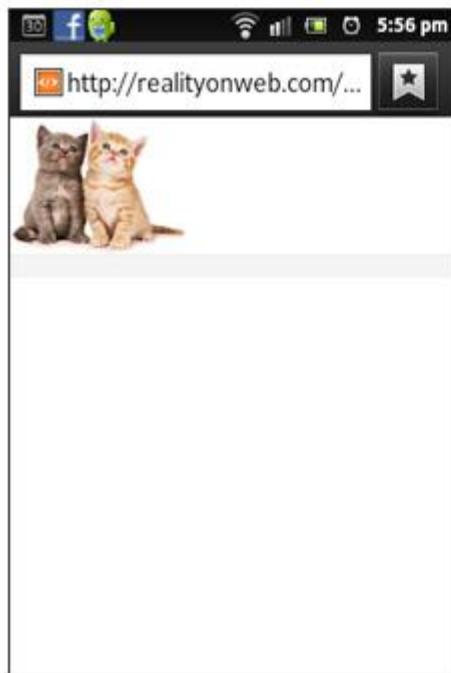
```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-
  scale=1.0">
  <title>Mi página web</title>
</head>
<body>
  <!-- Contenido de La página -->
</body>
</html>
```

Un usuario puede hacer zoom en la página web, con el gesto típico en la pantalla táctil.

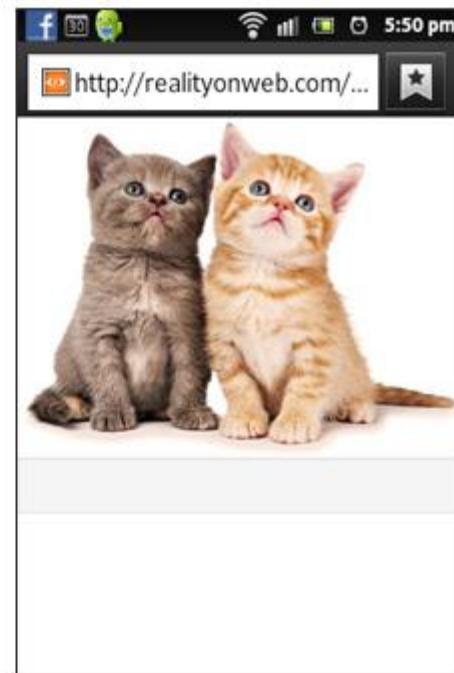
Al hacer zoom, realmente no estaría cambiando el viewport, sino la escala con la que se visualiza el documento.



Without Viewport



With Viewport





8.1 Ejemplo de la etiqueta **viewport** que no permite escalar la página

Lo más habitual es definir la anchura con el valor “**device-width**”, que es una medida que hace referencia a la anchura de la pantalla del dispositivo, y no con una medida fija. Por lo tanto, con **width=device-width** conseguimos que el **viewport sea igual a la anchura real de la pantalla** del dispositivo, de modo que no se tratará de emular una pantalla mayor de lo que realmente es y veremos los píxeles reales.

Con **initial-scale=1** conseguimos **definir la escala inicial** del documento para evitar transformaciones. **Ejemplo:**

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

También podemos hacerlo con **user-scalable=no** o con **maximum-scale con un valor igual al de initial-scale** conseguimos que el usuario **no pueda hacer zoom** en la página, con lo que siempre se mantendrán las medidas que nosotros hemos definido al construir la web. **Ejemplo:**

```
<meta name="viewport" content="user-scalable=no, width=device-width,  
initial-scale=1">
```

Desactivar el zoom con **user-scalable=no** nos facilita **tener siempre la escala a como nosotros hayamos definido** en la etiqueta **meta viewport**, lo que puede simplificarnos la vida a la hora de definir cómo se debe ver una web. Sin embargo, estaremos **limitando la posibilidad de que el usuario haga zoom** puntualmente para agrandar o empequeñecer alguna cosa. Por todo ello, cabe **sopesar** bien qué es lo que queremos permitir y si realmente definiendo un **maximum-scale** y **minimun-scale**, estamos acotando bien el uso de nuestra web.





8.2 Ejemplo etiqueta *viewport* que permite escalar la página

Para no limitar al usuario el uso del zoom en nuestra página web y darle la posibilidad de agrandar y empequeñecer cualquier parte que necesite ver mejor lo ideal es **no definir ni el *maximum-scale* ni el *user-scalable=no***. Ejemplo:

```
<meta name="viewport" content="width=device-width,initial-scale=1.0">
```

Por lo tanto, es interesante que el usuario pueda **hacer zoom** en la página web (en un smartphone será con los dedos sobre la pantalla táctil). En este caso, **el usuario no está cambiando los valores de *viewport*, sino la escala** con la que se visualiza el documento.



9. Test de verificación HTML y herramientas útiles.

Hay una gran cantidad de **herramientas muy útiles** para desarrolladores de forma online o que se instalan en los navegadores en forma de extensiones o plugins y que nos permiten **funcionalidades extra**. A continuación veremos algunas de ellas.

9.1 Validador HTML, W3C

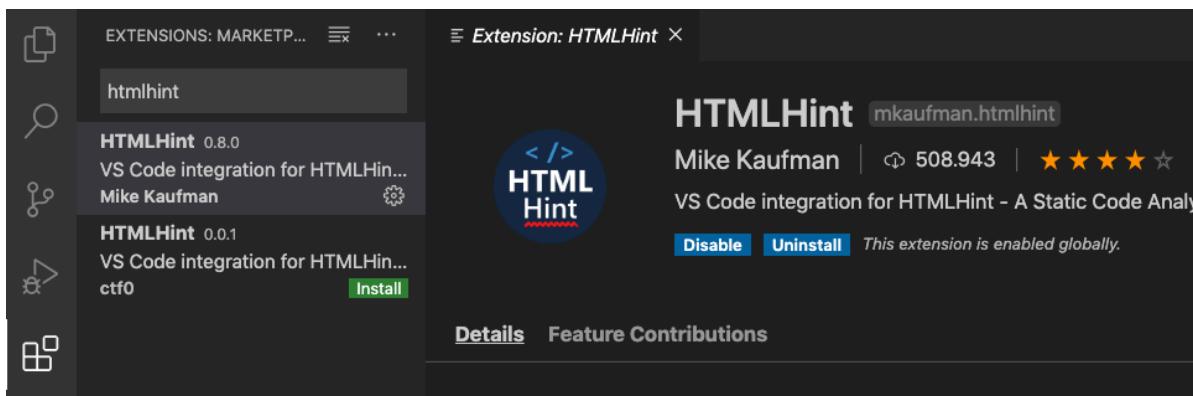
El **W3C** nos ofrece varias herramientas e información muy valiosa para la elaboración de nuestros desarrollos. Entre las herramientas destacadas se encuentra el **validador HTML** (validator.w3.org) que nos permite encontrar errores en nuestros ficheros. Este validador HTML nos permite validar nuestras páginas seleccionando los ficheros HTML a validar o eligiendo directamente la url publicada.

Si los documentos se validan correctamente y no disponen de ningún error, se puede incluir en la página web desarrollada el icono que certifica que el código cumple con el estándar.



9.2 HTMLHint, extensión para Visual Studio Code

Las páginas web se pueden validar en Visual Studio Code mediante la **extensión HTMLHint**. Para ello, accede a la pestaña “Extensions” de Visual Studio Code e instala la extensión tal y como puedes ver en la siguiente captura:



Una vez realizada la instalación la herramienta **mostrará los errores de validación en el código de cada documento HTML**, tal y como se puede ver en la siguiente captura:



The screenshot shows the Visual Studio Code interface. In the center, there is an editor window titled "bad.html" containing the following code:

```
<!--<!DOCTYPE html>-->
<ul>
    <li>
        <a href='http://www.github.com'>github</a>
    </li>
</ul>
```

Below the editor, a status bar shows "Ln 4, Col 47 (36 selected) Tab Size: 4 UTF-8 CRLF HTML".

To the left of the editor is the "EXPLORER" sidebar, which lists "OPEN EDITORS" (bad.html), "HELLO" (bad.html, index.html), ".vscode", and other icons.

At the bottom, a "PROBLEMS" panel displays four errors for "bad.html":

- Doctype must be declared first. (1, 1)
- The value of attribute [href] must be in double quotes. (4, 5)
- Tag must be paired, missing: [], start tag match failed [] on line 3. (5, 2)

There is also one error for "index.html":

- 1

Para ver los errores tan solo tenemos que ir al menú ver/problemas.

9.3 Firefox developer edition

Se trata de un navegador **creado específicamente para desarrolladores web** que ofrece herramientas muy interesantes como por ejemplo: panel de tipografías, editor de formas, inspector, consola o depurador, entre otras. mozilla.org/es-ES/firefox/developer/.



The screenshot shows the Firefox Developer Edition browser window. The address bar shows "127.0.0.1:8080". The main content area displays a grid of colored boxes. The "Inspector" tool is open, showing the DOM structure on the left and a "Grid" overlay on the right. The "Grid" settings include "Display line numbers" and "Extend lines infinitely".

9.4 Web developer

Consiste en una **extensión disponible para Chrome, Firefox o IE** que nos ofrece un control exhaustivo de varias partes de la web, tales como el CSS, los formularios o las

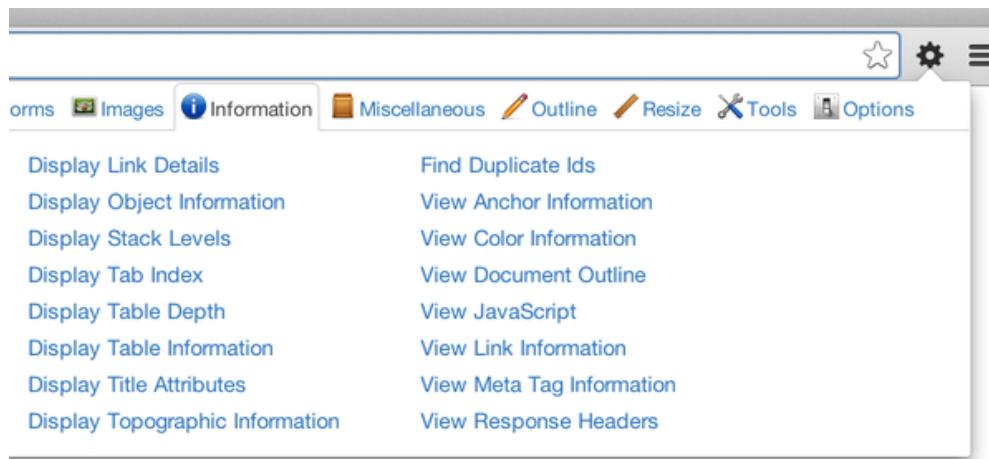


Web Developer

Ofrecido por: chrispederick.com



imágenes. Con Web Developer instalado podemos, por ejemplo, alterar casi cualquier elemento final de la web en tiempo real y así aplicarlo a los archivos finales.



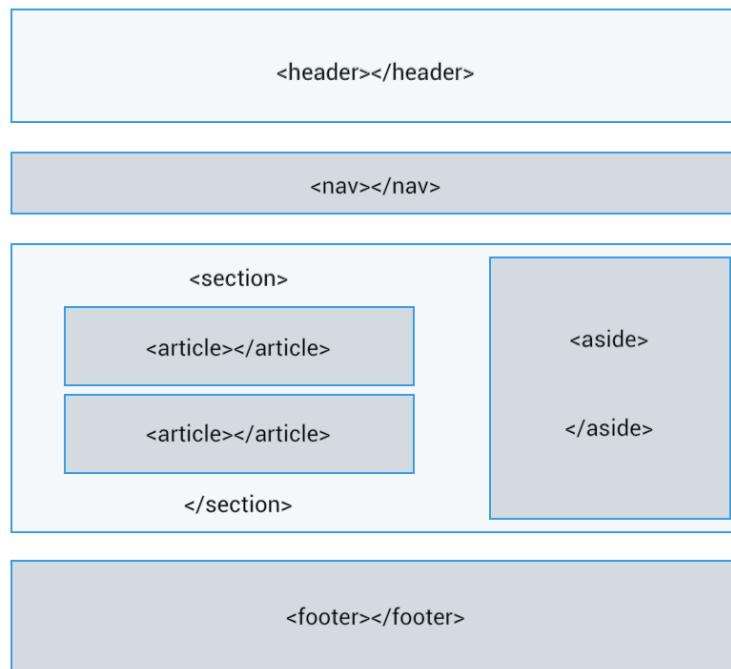
9.5 CodePen

CodePen es una comunidad en línea para **probar y mostrar fragmentos de código HTML, CSS y JavaScript**. Funciona como un editor de código en línea y un entorno de aprendizaje de código abierto, donde los desarrolladores pueden crear y probar fragmentos de código, llamados «bolígrafos».



10. Resumen de etiquetas HTML.

10.1 Elementos semánticos



10.2 Elementos raíz

Elemento	Descripción
<!DOCTYPE html>	Indica que el documento está bajo el estándar de HTML5
<html>	Representa la raíz de un documento HTML o XHTML.

10.3 Metadatos

Elemento	Descripción
<head>	Colección de metadatos sobre el documento, incluyendo enlaces a scripts y hojas de estilo.
<title>	Título del documento. Se muestra en la barra superior del navegador o en las pestañas de página.
<link>	Usada para enlazar JavaScript y CSS externos con el documento HTML actual.



<meta>	Define los metadatos que no pueden ser definidos usando otro elemento HTML.
<style>	Etiqueta de estilo usada para escribir CSS en línea.

10.4 Scripting

Elemento	Descripción
<script>	Define ya sea un script interno o un enlace hacia un script externo. El lenguaje de programación es JavaScript
<noscript>	Define un contenido alternativo a mostrar cuando el navegador no soporta scripting.

10.5 Secciones

Elemento	Descripción
<body>	Representa el contenido principal de un documento HTML. Solo hay un elemento <body> en un documento.
<section>	Define una sección en un documento.
<nav>	Define una sección que solamente contiene enlaces de navegación
<article>	Define contenido autónomo que podría existir independientemente del resto del contenido.
<aside>	Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si es removido, el contenido restante seguirá teniendo sentido
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>	Los elementos de cabecera implementan seis niveles de cabeceras de documentos; <h1> es la de mayor y <h6> es la de menor importancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.
<header>	Define la cabecera de una página o sección. Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.
<footer>	Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.
<address>	Define una sección que contiene información de contacto.



<main>	Define el contenido principal o importante en el documento. Solamente existe un elemento <main> en el documento.
---------------------	--

10.6 Contenido

Elemento	Descripción
<p>	Define una parte que debe mostrarse como un párrafo.
<hr>	Representa un quiebre temático entre párrafos de una sección o artículo o cualquier contenido.
<pre>	Indica que su contenido está preformatado y que este formato debe ser preservado.
<blockquote>	Representa un contenido citado desde otra fuente.
	Define una lista ordenada de artículos.
	Define una lista de artículos sin orden.
	Define un artículo de una lista enumerada.
<dl>	Define una lista de definiciones, es decir, una lista de términos y sus definiciones asociadas.
<dt>	Representa un término definido por el siguiente <dd> .
<dd>	Representa la definición de los términos listados antes que él.
<figure>	Representa una figura ilustrada como parte del documento.
<figcaption>	Representa la leyenda de una figura.
<div>	Representa un contenedor genérico sin ningún significado especial.

10.7 Etiquetas de texto

Elemento	Descripción
<a>	Representa un <i>hiperenlace</i> .
	Representa un texto <i>enfatizado</i> .
	Representa un texto importante.
<small>	Representa un comentario aparte, es decir, textos de políticas de responsabilidad o una nota de derechos de



	autoría, que no son esenciales para la comprensión del documento.
<s>	Representa contenido que no es exacto, tiene el estilo tachado.
<cite>	Representa el título de una obra.
<q>	Representa una cita textual entre comillas.
<dfn>	Sirve para marcar el término que se quiere definir.
<abbr>	Representa una abreviación o un acrónimo; mediante el atributo title se puede describir la abreviatura. El texto es usualmente representado como tooltip cuando se pasa el puntero sobre el elemento.
<time>	Representa un valor de fecha y hora.
<code>	Representa un código de programación.
<var>	Representa a una variable, es decir, una expresión matemática o una variable de un programa o similar.
<samp>	Representa la salida de un programa.
<kbd>	Representa el texto que debe introducir o la tecla que debe pulsar el usuario.
<sub><sup>	Representan un subíndice y un superíndice respectivamente.
<i>	Muestra el texto marcado con un estilo en cursiva o italica.
	Muestra el texto marcado con un estilo en negrita.
<u>	Muestra el texto subrayado.
<mark>	Representa un texto marcado o resaltado como referencia o anotación, debido a su relevancia o importancia.
	Representa texto en línea. Sirve para aplicar estilo al texto o agrupar elementos en línea.

	Representa un salto de línea.
<wbr>	Representa una oportunidad de salto de línea, es decir, un punto sugerido donde el texto puede ser dividido para mejorar su legibilidad.

10.8 Formulario

Elemento	Descripción
<form>	Define un formulario.



<fieldset>	Permite organizar en grupos los campos de un formulario.
<legend>	Representa el título de un <fieldset> .
<label>	Representa el título de un control de formulario.
<input>	Se usa para crear controles interactivos que reciben datos del usuario.
<button>	Representa un botón.
<option>	Representa una opción en un elemento <select> o <datalist> .
<select>	Representa un control que permite la selección entre un conjunto de opciones <option> .
<optgroup>	Representa un conjunto de opciones, agrupadas lógicamente.
<datalist>	Representa un control que permite la selección entre un conjunto de opciones <option> .
<textarea>	Representa un control de edición de texto multi-línea.
<output>	Representa el resultado de un cálculo.

Atributo	Ejemplo
placeholder	placeholder=>Indica tu nombre»
required	required=>true»
pattern	pattern=>[a-z]{1,5}»
min	min=>1"
max	max=>100"
step	step=>2" (saltos en un rango de números: 0, 2, 4...)
disabled	disabled=>true»
autofocus	autofocus=>true»
autocomplete	autocomplete=>true»

10.9 Tablas de contenido

Elemento	Descripción
<table>	Representa datos con más de una dimensión.
<caption>	Representa el título de una tabla.
<tr>	Representa una fila de celdas en una tabla.



<code><td></code>	Representa una celda de datos en una tabla.
<code><th></code>	Representa una celda encabezado en una tabla.
<code><colgroup></code>	Representa un conjunto de una o más columnas de una tabla.
<code><col></code>	Representa una columna de una tabla.
<code><tbody></code>	Representa el bloque de filas que describen los datos concretos de una tabla.
<code><thead></code>	Representa el bloque de filas que describen las etiquetas de columna de una tabla.
<code><tfoot></code>	Representa los bloques de filas que describen los resúmenes de columna de una tabla.

10.10 Contenido incrustado

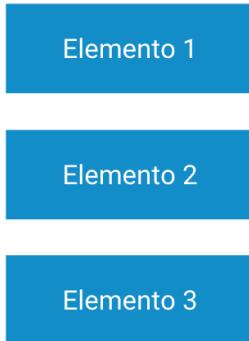
Elemento	Descripción
<code></code>	Representa una imagen.
<code><iframe></code>	Representa un contexto anidado de navegación, es decir, un documento HTML embebido.
<code><embed></code>	Representa un punto de integración para una aplicación o contenido interactivo externo que por lo general no es HTML.
<code><object></code>	Representa un recurso externo, que será tratado como una imagen, un sub-dокументo HTML o un recurso externo a ser procesado por un plugin.
<code><param></code>	Define parámetros para el uso por los plugins invocados por los elementos <code><object></code> .
<code><video></code>	Representa un vídeo, y sus archivos de audio y canciones asociadas, con la interfaz necesaria para reproducirlos.
<code><audio></code>	Representa un sonido o <i>stream de audio</i> .
<code><source></code>	Permite a autores especificar recursos multimedia alternativos para los elementos multimedia como <code><video></code> o <code><audio></code> .
<code><track></code>	Permite a autores especificar una pista de texto temporizado para elementos multimedia como <code><video></code> o <code><audio></code> .
<code><canvas></code>	Representa un área de mapa de bits en el que se pueden utilizar scripts para renderizar gráficos.
<code><map></code>	En conjunto con <code><area></code> , define un mapa de imagen.
<code><area></code>	En conjunto con <code><map></code> , define un mapa de imagen.
<code><svg></code>	Define una imagen vectorial embebida.



<math>

Define una fórmula matemática.

10.11 Elementos en bloque o block



Etiquetas HTML en bloque:

<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>-<h6>, <header>, <hr>, , <main>, <nav>, <noscript>, , <p>, <pre>, <section>, <table>, <tfoot>, , <video>

10.12 Elementos en línea o inline



Etiquetas HTML en línea:

<a>, <abbr>, <acronym>, , <bdo>, <big>,
, <button>, <cite>, <code>, <dfn>, , <i>, , <input>, <kbd>, <label>, <map>, <object>, <output>, <q>, <samp>, <script>, <select>, <small>, , , <sub>, <sup>, <textarea>, <time>, <tt>, <var>