

HTML.

- Hipertext Markup Language – Lenguaje de Marcado o Marcas o Etiquetas de Hipertexto.
- Es un lenguaje estándar a cargo del consorcio W3C (www.w3c.org), cuya función es actualizar y estandarizar todo tipo de tecnologías web.
- Página para aprender HTML:
 - w3schools.com
- Tim Berners Lee es el autor del HTML, así como de HTTP y URL.
- Características:
 - Lenguaje diseñado para crear páginas web.
 - HTML sólo define la estructura y contenido del documento web que luego será interpretado por un navegador.
 - Hipertexto:
 - Un documento web contiene hipertexto.
 - Hipertexto hace referencia a un documento con enlaces a otros documentos web u otro contenido (imágenes, videos, ...), al que se accede de forma no secuencial.
 - No es un lenguaje de programación.
 - Los documentos o páginas web se guardan con la extensión html o htm.
- Versiones:
 - HTML 1.0, 2.0, 3.0, 3.2, 4.0, 4.0.1, 5.0, 5.1, 5.2, 5.3.
 - XHTML 1.0, 1.1, 2.0 (HTML dinámico con XML).
- Documento DTD.
 - Declaración de tipo de documento.
 - Describe de la estructura y la sintaxis de un documento web o XML.
- Web Semántica:
 - HTML5 incorpora etiquetas semánticas que, además de informar al navegador, también lo hacen a los motores de búsqueda.
 - Las etiquetas semánticas describen cual es el significado del contenido.

HERRAMIENTAS DE EDICIÓN WEB.

1) Editores de texto plano.

- Editores de texto sin formato.
- Son archivos txt que se pueden guardar como html.
- Programas:
 - Bloc de notas de Windows
 - Editor Textedit de Apple.

2) Editores de texto con ventanas.

- Muchos son multiplataforma, es decir, tienen versiones para distintos sistemas operativos como Windows, Linux, iOS de Apple, Solaris, etc.
- Permiten escribir código en ventanas y previsualizar el resultado.

- Tiene ayudas para la edición como autocompletar código, indentado y sangrado de texto, indicación de errores, códigos de colores para los elementos del código, etc.
- Permiten editar código diverso: HTML, CSS, JavaScript, Java, C, PHP y otros lenguajes.
- Programas:
 - Atom, Notepad++, Visual Studio Code, Adobe Brackets, Sublime Text, JsBin, Aptana Studio, Kompozer y otros.

3) **Editores visuales WYSIWYG.**

- What you see is what you get - Lo que ves es lo que obtienes.
- Muchos son multiplataforma, es decir, tienen versiones para distintos sistemas operativos como Windows, Linux, iOS de Apple, Solaris, etc.
- No es necesario escribir código, se colocan los distintos elementos sobre una vista previa de la página y el programa genera el código HTML.
- También tienen ventanas de edición HTML.
- Programas:
 - Adobe Dreamweaver, FrontPage, Amaya, Kompozer, Bluegriffon, CoffeCupFree, Bluefish, NetBeans y otros.

4) **CMS.**

- Content Management System - Sistema gestor de contenido.
- Software que se ejecuta en el navegador y permite crear, administrar y modificar un sitio web y su contenido.
- Utilizan una interfaz visual.
- Programas:
 - Wordpress, Joomla, Drupal, Prestashop.

5) **Otras Herramientas.**

- Editores de imagen:
 - Permiten optimizar las imágenes para su uso en Internet
 - Para ello se usan programas de edición y retoque fotográfico.
 - Programas:
 - Photoshop, Gimp, Paint, Photoscape, Ribbet, Canvas y otros
- Editores de audio:
 - Editores de sonido y grabación de sonido digital.
 - Programas:
 - Audacity, OceanAudio, Free Audio Editor, Wavepad (Mac) y otros.
- Editores de video:
 - Herramientas para la creación y edición de video.
 - Permiten editar video eliminando secuencias no útiles y añadiendo títulos, introducciones o efectos especiales entre otras funciones.
 - Programas:
 - Moviemaker, Lighthworks, Blender, Clipchamp y otros.
 - También existen herramientas que permiten la conversión de video a distintos formatos como Freevideo Converter o Freemake.
 - Animaciones Flash:

- Tecnología para reproducir contenido multimedia usando Adobe Flash Player.
- Abandonado el soporte en 2021 por problemas de seguridad, por ser software propietario y por utilizar decodificación por hardware.
- Actualmente se usa HTML5 y CSS para reproducir contenido multimedia.
- Paletas de colores:
 - Selección de combinaciones de colores homogéneas y armoniosas para la web.
 - Programas y aplicaciones web:
 - Adobe Color CC, Color Hunt, Color Code, Sip (para Apple) y otros.

CODIFICACIÓN Y JUEGO DE CARACTERES.

Codificación de caracteres.

- Método permite convertir un carácter del lenguaje natural en un símbolo de otro sistema de representación.

Juego de caracteres:

- Conjunto de caracteres (letras, números y símbolos), representable en un sistema informático.

Juegos:

- **ANSI.**
 - American National Standard Institute.
 - Usa un byte (8 bits), para representar un carácter.
 - Permite representar 256 caracteres. (0→255).
- **ASCII.**
 - Usa un byte (8 bits) de la siguiente manera:
 - 7 bits para representar un carácter.
 - 1 bit adicional (bit de paridad), para detectar errores en la transmisión de datos.
 - Permite representar 128 caracteres. (0 →127).
 - Incluye caracteres latinos en inglés.
- **Latin-1 (ISO-8859-1).**
 - ISO - International Organization Standardization.
 - Juego de caracteres para idiomas europeos (occidentales).
 - Utiliza 1 byte para representar 1 carácter de la siguiente manera:
 - Los 128 primeros son los mismos caracteres que en ASCII.
 - Los 128 siguientes incluyen tildes, símbolos monetarios, etc.
 - No incluye el euro y representa los 256 primeros caracteres Unicode.
- **Latin-9 (ISO-8859-15).**
 - Revisión del Latin-1 (ISO-8859-1), que incorpora el símbolo del € y algunos caracteres franceses, estonios y fineses.
- **UTF-8.**
 - Unicode Transformation Format.

- Sistema de codificación Unicode.
- Usa símbolos de longitud variables representables con:
 - 1 byte, 2 bytes, 3 bytes o 4 bytes.
- Permite representar miles de caracteres de los diferentes idiomas pudiéndose crear sitios web multilingües.

Caracteres especiales.

- Caracteres que hay que codificar de forma especial para que no haya problemas de ambigüedad en HTML.
- Algunos caracteres a codificar son:
 - < Indica apertura de etiqueta, se codifica como < para ser un carácter más.
 - > Indica cierre de etiqueta, se codifica como > para ser un carácter más.
 - “ Indica valor de atributo, se codifica como " para ser un carácter más.
 - & Indica inicio de referencia de carácter, se codifica como & para ser un carácter más.
 - Ejemplo:
 - <header>
 - <head>

Otros caracteres especiales:

Se usan si no existe el carácter en el juego de caracteres (charset) que se está utilizando en la página web.

- **Letras acentuadas.**
 - Sintaxis:
 - &letraacute;
 - Ejemplo:
 - La letra á con tilde se escribiría como á
- **Letras acentuadas con tilde inversa.**
 - Sintaxis:
 - &letragrave;
 - Ejemplo:
 - La letra ò con tilde inversa se escribiría como ò
- **Letras con acento circunflejo.**
 - Sintaxis:
 - &letracirc;
 - Ejemplo:
 - La letra û se escribiría como û
- **Símbolo del euro.**
 - Se escribe como €
- **Espacio en blanco.**
 - Se escribe como

ETIQUETAS O ELEMENTOS HTML.

- Una etiqueta es una orden, instrucción o comando interpretado por un navegador.
- Se encierran entre los caracteres < y >
- La mayoría tienen apertura y cierre. <>.....</>
- Sintaxis:
 - <Etiqueta>Contenido</Etiqueta>
 - <Etiqueta Atributo = "valor">Contenido</Etiqueta>
 - <Etiqueta Atributo = "valor">
 - <Etiqueta Atributo = "valor"></Etiqueta>
 - <Etiqueta Atributo1 = "valor" Atributo2 = "valor" AtributoN = "valor" ></Etiqueta>
 - <Etiqueta Atributo1 = "valor" Atributo2 = "valor" AtributoN = "valor" >Contenido</Etiqueta>
 - <Etiqueta Atributos /> Etiqueta sin el par de cierre.
- Las etiquetas pueden anidarse cerrándose en sentido contrario a como se han abierto.
 - <a><c> Contenido </c>
 - <a>
 -
 - <c>
Contenido
</c>
 -
 -

Estructura de un documento o página web

- **<!Doctype>**
 - Directiva que indica a un navegador que versión HTML se está utilizando. (HTML y XHTML)
 - También indica al navegador que reglas debe usar para procesar la página.
 - No tiene etiqueta de cierre.
 - Para usar HTML 5 basta con incluir el valor html en la directiva:
 - <!Doctype html>
- **<html>...</html>**
 - Etiqueta principal o raíz.
 - Obligatoria.
 - Sólo una en todo el documento.
 - Todo el código de la página web se encierra entre ellas.
 - Puede incluir el código de país para indicar el lenguaje con el que se ha creado la página usando el atributo lang:
 - <html lang = "código de país">
 - <html lang = "es">
- **<head>...</head>**
 - Representa la cabecera del documento web.
 - Incluye metadatos e información para servidores, navegadores y buscadores.

- Incluye también enlaces a los archivos que deben ejecutarse dentro de la página como hojas de estilo CSS, archivos JavaScript, fuentes externas y otro tipo de archivos.
- Lo que se escribe en la cabecera no aparece en la página web.
- Incluye la etiqueta <title> que permite poner un título en la página.

- Sintaxis:

- <title>Título</title>

- **METAETIQUETAS.**

- Permiten especificar metadatos para un documento.

- Metadato:

- Metainformación.
- Datos que describen datos.

- Se incluyen en la cabecera <head> del documento Web.
- Aportan información técnica a los motores de búsqueda y mejoran el posicionamiento SEO (Search Engine Optimization – Optimización para Motores de Búsqueda).
- Es una forma de comunicarse con los motores de búsqueda e informales de cómo deben interpretar nuestra web o que deben mostrar en los **SERP** (resultados de una búsqueda).

- **SERP.**

- Search Engine Results Page - Página de resultados de un motor de búsqueda.

- Sintaxis:

```
<head>
  <meta atributo 1 = "valor" atributo 2 ="valor" .... atributo N = "valor"/>
</head>
```

- **Atributos.**

- **charset.**

- Especifica el conjunto de caracteres usado en el documento.

- Sintaxis:

- <meta charset =" juego de caracteres">

- Ejemplo:

- <meta charset = "utf-8">

- **name.**

- Especifica nombre del metadato a utilizar.
- Especifica la meta-etiqueta a usar.
- Atributo asociado al atributo content.

- Sintaxis:

- name = "valor"

- **content.**

- Especifica el valor o contenido de un metadato concreto.
- Atributo asociado a los atributos name y http-equiv.

- Sintaxis:

- content = "valor"

- **http-equiv.**

- Especifica órdenes o directivas para el procesamiento http en el servidor.
- Permite indicar al servidor algunas propiedades o comportamiento del documento web.
- Atributo asociado al atributo content.

- Sintaxis:

- http-equiv = "metadato".

- **Sintaxis combinada name y content o http-equiv y content.**
 - `<meta name="metadato" content="valor del metadato">`
 - `<meta http-equiv="metadato" content="valor del metadato">`
 - Ejemplos:
 - `<meta name="Author" content="Ana López García">`
 - `<meta http-equiv="refresh" content="30">`
- **Listado de metadatos para el atributo name.**
 - **application-name.**
 - Describe el nombre de la aplicación web que el documento representa.
 - Solo puede aparecer una vez.
 - **author.**
 - Nombre del autor del documento web.
 - Se pueden incluir varios autores, cada uno en una declaración meta distinta.
 - Se puede incluir el correo electrónico del autor.
 - **owner.**
 - Indica quien es el propietario.
 - **copyright.**
 - Especifica el nombre del propietario de los derechos de autor.
 - Indica el copyright de la página.
 - **description.**
 - Breve descripción de la página o su propósito.
 - Sólo puede haber una descripción en el documento web.
 - Consejo:
 - Entre 50 y 160 caracteres.
 - **keywords.**
 - Palabras clave en la página web.
 - Especifican un conjunto de identificadores separados por comas que son palabras clave que describen los contenidos del documento web.
 - Importante para el posicionamiento SEO.
 - **generator.**
 - Indicar el nombre programa o programas de diseño utilizados para generar el documento web.
 - Si se han usado múltiples programas hay que incluir múltiples declaraciones meta.
 - **robots.**
 - Especifica las órdenes o directivas para los robots de rastreo.
 - Indica a los robots de los buscadores que información debe o no, ser tomada en cuenta.
 - Indica cómo se deben rastrear los enlaces que el robot encuentre en el documento web.
 - Indica a los robots si se debe o no indexar la página.
 - En el atributo **name** se pueden especificar todos los robots con el valor "robots" o incluir rastreadores específicos sustituyendo "robots" por el nombre de éstos.
 - Ejemplos:
 - `name="robots"` (Todos los robots de todos los motores de búsqueda).
 - `name="Bingbot"` (Un robot concreto).
 - `name="Bingbot, Yandex, Baiduspider"` (Varios robots).
 - Directivas:

- **all.**
 - Sin restricciones de indexación o publicación.
 - Valor predeterminando.
 - Equivale a index y follow juntos.
- **Index / noindex.**
 - La página web debe / no debe ser indexada en la base de datos del motor de búsqueda.
 - Por lo tanto, la página se muestra/no se muestra en los resultados de una búsqueda.
- **follow / nofollow.**
 - Permite / Impide seguir los enlaces de una página web cuando ésta es rastreada por un robot.
- **none.**
 - Equivale a noindex y nofollow juntos.
- **noarchive.**
 - Se usa para que el rastreador no guarde la página en la cache de búsqueda.
 - Esta cache almacena el contenido que tenía una página la última vez de que se accedió a ella, así, la siguiente vez, el robot accederá más rápido a la página.
 - En caso de que la página haya sido modificada, es mejor que se acceda al servidor para rastrear la página en su último estado, que a la cache.
- **nosnippet.**
 - No se muestra un fragmento de la página o de la descripción en los resultados búsqueda.
- **noimageindex.**
 - No se indexan las imágenes de la página.
- **language.**
 - Especifica en que idioma está escrita la página web.
 - Se usa un código de país de 2 caracteres: es, pt, uk, en_UK en_US.
- **revisit-after.**
 - Indica al robot cada cuanto tiempo se actualiza la página y por tanto, cada cuanto deben revisarla.
 - Periodo temporal escrito en inglés y medido en días o meses (1 month, 15 days, ...).
- **viewport.**
 - Permite ajustar el contenido del documento web a las distintas dimensiones, resoluciones y densidades de pixeles de los dispositivos.
 - Indica al navegador como controlar la dimensión de la pantalla escalando el contenido para que no se salga de los márgenes del dispositivo.
 - Si no se usa, no se cargan los distintos diseños creados para las pantallas de los diferentes dispositivos.
- **Listado de metadatos del atributo http-equiv.**
 - **content-type.**
 - Especifica el tipo de contenido y juego de caracteres que usa en la página.
 - Alternativa al atributo charset = “juego de caracteres”.
 - Sintaxis:

- `<meta http-equiv = "content-type" content="tipo MIME; charset = juego de caracteres">`
 - **refresh.**
 - Se emplea para actualizar o redireccionar una página web.
 - Actualizar:
 - Permite indicar el tiempo de refresco que debe pasar entre cada actualización.
 - Sintaxis:
 - `<meta http-equiv = "refresh" content= "nº segundos">`
 - Redireccionar:
 - Tras pasar el tiempo especificado, el servidor envía directamente a otra página web.
 - Sintaxis:
 - `<meta http-equiv = "refresh" content= "nº segundos; URL del destino">`
 - Ejemplo:
 - `<meta http-equiv = "refresh" content= "30; URL = http://www.unsitio.es/inicio.html">` (Externa)
 - `<meta http-equiv = "refresh" content= "30; URL = redirecciones/inicio.html">` (Interna)
 - **expires.**
 - Especifica el día y hora exactos en el cual expira la página guardada en la cache del navegador, por lo que ésta debe descargarse de nuevo desde el servidor para poder visualizarse.
 - Sintaxis:
 - `<meta http-equiv = "expires" content= "día de la semana, dd mm aaaa hh:mm:ss GMT">`
 - `<meta http-equiv = "expires" content= "0">`
 - Ejemplo:
 - `<meta http-equiv = "expires" content= "Sun, 20 07 2020 12:00:00 GMT">`
 - **cache-control.**
 - Indica a los navegadores si se deben o no guardar en cache (Archivos temporales de Internet), las páginas y archivos web que se cargan.
 - Sintaxis:
 - `<meta http-equiv="Cache-control" content="valores">`
 - Ejemplo:
 - `<meta http-equiv="Cache-control" content="no-cache">`
 - Incluye las metaetiquetas mediante el elemento `<meta>`.
 - Sintaxis:
 - `<meta atributo = "valor"/>`
- **<body>...</body>**
 - Representa al cuerpo del documento web.
 - El contenido que se incluya en `<body>` será visible en la página al visualizarla en un navegador.
 - Incluye el código HTML necesario para crear una página web.

- Atributos:
 - **bgcolor.**
 - Permite aplicar un color al fondo de la página.
 - Sintaxis:
 - `<body bgcolor="color">...</body>`
 - Ejemplo:
 - `<body bgcolor="#34fa1c">...</body>`
- **Comentarios.**
 - Son ignorados por los navegadores y sirven para documentar el código, facilitar su comprensión, añadir ideas, sugerencias, etc., ...
 - Sintaxis:
 - De una línea:
 - `<!-- Texto del Comentario -->`
 - De varias líneas:
 - `<!-- Texto del Comentario Línea 1`

Texto del Comentario Línea 2

Texto del Comentario Línea 3 -->

LISTADO DE ETIQUETAS HTML.

Tipos de Etiquetas o Elementos.

- **De bloque.**
 - La caja fuerza un salto de línea al final del contenido.
 - Respetan las propiedades width, height, márgenes y bordes.
 - Elementos:
 - address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul, dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.
- **En línea.**
 - Ocupan el espacio de su contenido.
 - La caja no fuerza un alto de línea a no ser que varios elementos de línea acaben ocupando toda la línea.
 - No respetan width ni height, ya que su anchura y altura dependen del tamaño en pixeles de su contenido.
 - No tienen márgenes externos inferior ni superior, sólo izquierdo y derecho.
 - Elementos:
 - a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Encabezados o títulos.

- **<h nº>...</h nº>**
 - Etiqueta de bloque.
 - Permite incluir títulos o encabezados.
 - Hay seis niveles de encabezado del documento, <h1> es el más importante, y <h6>, el menos importante.
 - Por defecto se muestran en negrita y con un salto de línea entre éste y el resto del contenido.
 - Sintaxis:
 - <h1..h6>Texto del Título</h1..h6>
 - Atributos:
 - align:
 - Permite alinear el encabezado.
 - Sintaxis:
 - align = "valor"
 - Valores:
 - left, center, right.

Párrafos.

- **<p>... </p>**
 - Sirve para crear un párrafo.
 - Etiqueta de bloque.
 - Distintos párrafos son separados automáticamente por un salto de línea.
 - Sintaxis:
 - <p>Texto del párrafo</p>
 - Atributos:
 - align:
 - Sirve para alinear un párrafo.
 - Sintaxis:
 - align = "valor"
 - Valores:
 - left, center, right, justify.

Salto de línea.

- **
**
 - Fuerza un salto de línea.
 - No tiene etiqueta de cierre.

Negrita.

- **..****
 - Texto en negrita
 - Sintaxis:
 - **Texto****

Cursiva.

- **<i>..**</i>**
 - Texto en cursiva.
 - Sintaxis:
 - **<i>Texto</i>****

Negrita. Etiqueta semántica.

- **...**
 - Énfasis para reforzar el significado del texto.
 - Es texto se muestra en negrita, pero en realidad sirve para enfatizar o destacar un contenido que es importante.
 - Refuerza la importancia de un texto y un navegador de voz lo entona de forma distinta al resto de texto.
 - Sintaxis:
 - **Texto**

Cursiva. Etiqueta semántica.

- **...**
 - Énfasis para reforzar el significado del texto.
 - Cursiva, pero en realidad sirve para enfatizar o destacar un contenido que es importante.
 - Refuerza la importancia de un texto y un navegador de voz la entona de forma distinta.
 - Sintaxis:
 - **Texto**

Subíndice.

- **_{...}**
 - Permite crear un subíndice.
 - Sintaxis:
 - **_{texto}**
 - Ejemplo:
 - **<p>H₂O</p>**

Superíndice.

- **^{...}**
 - Permite crear un superíndice.

- Sintaxis:
 - ^{texto}
- Ejemplo:
 - <p>X²</p>

Subrayado.

- <u>...</u>
 - Subrayar texto.
 - Sintaxis:
 - <u>texto</u>

Tachado.

- <s>...</s>
 - Muestra un texto tachado.
 - Sintaxis:
 - <s>texto</s>

Separador horizontal.

- <hr>
 - Añade una línea horizontal.
 - No tiene etiqueta de cierre.
 - **Formato de la barra.**
 - **width.**
 - Anchura o longitud de la línea.
 - Sintaxis:
 - width = “valor”
 - Ejemplo:
 - width = “500px”
 - **size.**
 - Altura o grosor de la línea.
 - Sintaxis:
 - size = “valor”
 - Ejemplo:
 - size = “50px”
 - **color.**
 - Color de la línea.
 - Sintaxis:
 - color = “valor”
 - Ejemplo:
 - color = “blue”

Texto preformateado.

- **<pre>...</pre>**
 - Muestra el texto tal como se ha escrito en el editor.
 - Se muestran en pantalla los saltos de línea, tabulaciones y espacios.
 - El texto aparece con una fuente monoespaciada.

CITAS.

- **Cita corta.**
 - **<q>...</q>**
 - No muestra un salto de línea, aparece en la misma línea que el párrafo.
 - No hay salto de tabulación o indentación a la derecha.
 - Texto de la cita aparece entre comillas.
 - Sintaxis:
 - **<q>Texto de la cita</q>**
 - Ejemplo:
 - Él dice: «Hoy es miércoles»
 - **<p>Él dice: <q>Hoy es miércoles</q></p>**
- **Cita Larga.**
 - El texto de la cita aparece indentado respecto a otros párrafos.
 - Sintaxis:
 - **<blockquote>Texto cita</blockquote>**
 - Ejemplo:
 - **<p>Él dice: <blockquote>Hoy es miércoles</blockquote></p>**
- **Cita de obras.**
 - Para citar libros, discos, películas, operas, ...
 - El texto de la cita aparece en cursiva.
 - Sintaxis:
 - **<cite>Texto cita</cite>**
 - Ejemplo:
 - **<p>Su disco: <cite>Canciones de ayer</cite></p>**

Inserción y eliminación de texto.

- **<ins>...</ins>**
 - Muestra texto nuevo añadido para sustituir o no a otro.
 - Para distinguirlo del resto de texto aparece subrayado.
 - Sintaxis:
 - **<ins>Texto nuevo</ins>**
 - Ejemplo:
 - **<p><ins>Este texto es nuevo</ins></p>**
- **...**

- Muestra texto antiguo que va a ser eliminado.
- Para distinguirlo del resto de texto aparece tachado.
- Sintaxis:
 - `Texto a eliminar`
- Ejemplo:
 - `<p>Este texto se va a eliminar</p>`

Formato de fuentes.

- **`...`**
 - Permite aplicar a los textos distintos tipos de fuentes, tamaños y colores.
 - Sintaxis:
 - `Texto`
 - Atributos:
 - **face**:
 - Para aplicar un tipo de fuente.
 - Sintaxis:
 - `texto`
 - Ejemplo:
 - ` texto`
 - **size**:
 - Tamaño de fuente.
 - Valores predeterminados entre 1 y 7.
 - Tamaños en orden creciente desde 1 (mínimo) a 7 (máximo).
 - Sintaxis:
 - `texto`
 - Ejemplo:
 - `texto`
 - **color**:
 - Aplicar color al texto.
 - Sintaxis:
 - `texto`
 - Forma de aplicar los colores:
 - **Nombre**:
 - Se escribe el nombre del color en inglés.
 - Permite aplicar 140 colores seguros.
 - Algunos colores:
 - red, pink, yellow, black, blue, maroon, gray...
 - Ejemplo:
 - `texto`
 - **Valor Hexadecimal**:
 - Ejemplo:

- `texto`

Modelo de color RGB (Red, Green, Blue).

Características:

- Modelo de color basado en la intensidad de los colores primarios de la luz.
- Se utiliza en TV y pantallas ordenadores, tabletas, móviles, etc.
- Es uno de los modelos utilizado en diseño Web.
- No se usa en el atributo color de la etiqueta ``, pero si con CSS.

Uso con valores decimales:

- Para cada color se utiliza un byte expresando su valor en decimal:
 - Dígitos 0 1 2 3 4 5 6 7 8 9.
- Los distintos valores decimales están comprendidos entre el 0 y el 255.
- Sintaxis:
 - `rgb (0-255,0-255,0-255)`
- Ejemplos:
 - `rgb (135, 206, 235)` Azul cielo.
 - `rgb (0,0,0)` Negro.
 - `rgb (255,255,255)` Blanco.
 - `rgb (127,127,127)` Gris medio.
 - `rgb (34,34,34)` Gris oscuro.
 - `rgb (0,255,0)` Verde.

Uso con nombres del color.

- Se escribe el nombre del color en inglés.
- Hay 140 valores de colores seguros, es decir, colores que pueden mostrar en cualquier dispositivo.
- Sintaxis:
 - `propiedad: nombre del color.`
- Ejemplo:
 - `color:green`

Uso con valores hexadecimales:

- Para cada color se utiliza 2 bytes expresando su valor en hexadecimal:
 - Dígitos y letras 0 1 2 3 4 5 6 7 8 9 a b c d e f.
- Sintaxis:
 - `#rrggbb` - Notación estándar.
 - `#rgb` - Notación abreviada si los valores hexadecimales para cada color son los mismos.
- Ejemplos:

- #87CEEB Azul cielo.
- #000000 Negro == #000 en notación abreviada.
- #ffffff Blanco == #fff en notación abreviada.
- #808080 Gris medio.
- #252525 Gris oscuro.
- #00ff00 == #0f0 Verde.

Listas ordenadas.

- **..**

- Etiqueta que sirve para crear una lista ordenada.
- Los distintos elementos que forman lista se crean con la etiqueta
- Sintaxis:

- Lista simple:

```
<ol>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento N</li>
</ol>
```

- Lista anidada:

```
<ol>
    <li>Elemento 1</li>
    <li>Elemento 2
        <ol>
            <li>Elemento 2.1</li>
            <li>Elemento 2.2</li>
        </ol>
    <li>Elemento N</li>
</ol>
```

- Atributos:

- **type:**

- Permite cambiar el sistema de numeración.

- Sintaxis:

- <ol type = "valor">

- Valores:

- 1:
 - Ordenar con números.
 - Valor por defecto.
 - a:
 - Ordena con letras minúsculas.
 - A:
 - Ordena con letras mayúsculas.

- Ordena con letras mayúsculas.
 - i:
 - Ordena con números romanos en minúscula.
 - I:
 - Ordena con números romanos en mayúscula.
- **start:**
 - Indica a partir de qué punto va a empezar a numerarse una lista ordenada.
 - Sintaxis:
 - `<ol start = "valor">`
 - Valores:
 - Número que será el primer valor en la lista, independientemente de que ésta se numere con letras, números o números romanos.

Listas no ordenadas.

- **`...`**
 - Etiqueta que sirve para crear una lista desordenada.
 - Los distintos elementos que forman lista se crean con la etiqueta `...`.
 - Sintaxis:
 - Lista simple:

```

<ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento N</li>
</ul>
          
```
 - Lista anidada:

```

<ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
        <ul>
            <li>Elemento 2.1</li>
            <li>Elemento 2.2</li>
        </ul>
    <li>Elemento N</li>
</ul>
          
```
 - Atributos:
 - **type:**
 - Permite cambiar el tipo de viñeta.
 - Sintaxis:
 - `<ul type = "valor">`
 - Valores:

- disc:
 - Circulo relleno.
 - Valor por defecto.
- circle:
 - Circulo no relleno.
- square:
 - símbolo de un cuadrado.

Listas de definición.

- **<dl>...</dl>**
 - Permite crear una lista de definición o glosario de términos.
 - Está asociada a las etiquetas <dt> y <dd> a las que incluye.
- **<dt>...</dt>**
 - Sirve para especificar el término a definir.
- **<dd>...</dd>**
 - Sirve para crear la definición del término.
- Sintaxis:

```

<dl>
  <dt>
    Término a definir.
  </dt>
  <dd>
    Definición.
  </dd>
</dl>

```

Listas de directorios y menús.

- Equivalen a listas no ordenadas y ya no se utilizan en HTML 5.
- Se usan las etiquetas <dir>...</dir> y <menu>...</menu>.

TABLAS.

Etiquetas:

- **<table>...</table>**
 - Etiqueta que permite crear una tabla para almacenar datos no para maquetar una página como se hacía antiguamente.
 - Para crear la tabla se usan las etiquetas <tr> y <td> dentro de etiqueta principal <table>.
- **<tr>...</tr>**
 - Permite crear las filas de una tabla.
 - Hay que usar una etiqueta por fila.
 - Dentro de la etiqueta <tr> se usan una o varias etiquetas <td> para indicar cuantas celdas contendrá la tabla.

- **<td>...</td>**
 - Sirve para crear las celdas e incluir su contenido.
 - Sintaxis:
 - `<td>Contenido de la celda</td>`
- **<th>...</th>**
 - Sirve para crear una celda de encabezado en una tabla.
 - Es opcional (la cabecera se puede crear de otras formas).
 - El contenido de la celda aparece por defecto en negrita y centrado.
 - Con ella, no se necesitaría usar en la primera fila (o las que se usen para la cabecera), las etiquetas `<td>` para crear las celdas y su contenido.
 - Sintaxis:
 - `<th>Contenido de la celda</th>`
- **<caption>...</caption>**
 - Permite incluir un título en una tabla.
 - Se pone inmediatamente después de la etiqueta `<table>`
 - Sintaxis:
 - `<caption>Título</caption>`
- **<colgroup>...</colgroup>**
 - Permite agrupar una o varias columnas de una tabla.
 - Las columnas que formaran el grupo se seleccionan de forma consecutiva de izquierda a derecha.
 - Si hay varios grupos creados con `<colgroup>` el primero tendrá las primeras columnas que formen parte del grupo. El siguiente grupo, las siguientes a partir de las que forman el grupo previo y así sucesivamente.
 - La etiqueta **<colgroup>** debe ser incluida al principio al definir la tabla antes de cualquier otra etiqueta excepto `<caption>`.
 - Columnas para un grupo.
 - Para indicar el número de columnas que formaran un grupo hay 2 **opciones**:
 - **span**.
 - Atributo que permite dentro de `<colspan>` indicar el número de columnas que formarán el grupo.
 - Sintaxis:
 - `span = "número entero"`
 - Ejemplo:
 - `<colgroup span = "2">` ----- Grupo formado por 2 columnas.
 - **<col>**
 - Esta etiqueta se puede incluir dentro de **<colgroup>** (hijo) y, usando el atributo `span`, se especifica el número de columnas que formarán el grupo.
 - Si no se incluye `span` `<col>` representa a una sola columna.
 - Esta etiqueta se usa para definir estilos o especificar un conjunto de atributos para todas las celdas de una o varias columnas.
 - También se puede usar sin `span` y sin `<colgroup>`, en este caso se incluye después de la etiqueta `<table>` y presentar un orden: la primera declaración afecta a la primera columna, la segunda declaración a la segunda columna, y así sucesivamente.

- Sintaxis:
 - <col>
 - <col span = "número entero">
- Ejemplos:

```
<table border = "1">
  <colgroup span="3"></colgroup> ---- Grupo creado con las 3 primeras columnas (1,2 y 3).
  <colgroup> ---- Nuevo grupo con las 3 columnas siguientes (4, 5 y 6)
    <col> --- Columna 4 sin ningún atributo o estilo especial.
    <col span="2" class = "c56" style = "background-color: rgb(155, 67, 12); border: 1px solid red;"> ---- Columnas 5 y 6 con diversos estilos CSS y con un identificador class.
  </colgroup>
  <tr>.....<td>...</td>.....</tr>.....</table> ---- Resto de etiquetas para definir la tabla.
```

Secciones de una tabla.

- Son opcionales.
- **<thead>...</thead>**
 - Encabezado de la tabla.
 - Es la primera fila e incluye las etiquetas <tr> y <td> o <th>.
- **<tbody>...</tbody>**
 - Todas las celdas con datos.
 - Se aplica al resto de filas, excepto a la última si hubiese un pie.
 - Incluye las etiquetas <tr> y <td> de todas las filas y celdas con datos.
- **<tfoot>...</tfoot>**
 - Pie de la tabla.
 - Suele ser la última fila, aunque puede haber un pie de más de una fila.
 - Incluye las etiquetas <tr> y <td> de la última fila.

Atributos de una tabla.

- **border.**
 - Permite poner un borde a una tabla.
 - Es obligatorio para ver las líneas de división de la tabla.
 - Sintaxis:
 - <table border = "número">
 - Ejemplo:
 - <table border = "1">
- **cellspacing:**
 - Define la distancia o espaciado entre celdas.
 - Unidad de medida es un valor numérico único.
 - El valor cero elimina la distancia y simula un borde de trazo simple.
 - Sintaxis:
 - <table cellspacing = "número">
 - Ejemplo:

- <table cellpadding = "0">
- **width:**
 - Define anchura de la tabla o de las celdas.
 - Unidad de medida en píxeles o porcentajes.
 - Sintaxis:
 - <table width = "número unidad de medida">
 - <td width = "número unidad de medida">
 - Ejemplos:
 - <table width = "400px">
 - <td width = "50px">
- **height:**
 - Define altura de la tabla o de las celdas.
 - Unidad de medida en píxeles o porcentajes.
 - Sintaxis:
 - <table height = "número unidad de medida">
 - <td height = "número unidad de medida">
 - Ejemplos:
 - <table height = "400px">
 - <td height = "50px">
- **align:**
 - Si se aplica a <table>, alinea la tabla respecto de la ventana del navegador.
 - Si se aplica a una celda, alinea su contenido.
 - Sintaxis:
 - <table align = "valor">
 - <td align = "valor">
 - Valores:
 - left, center, right.
- **bgcolor:**
 - Color de fondo de la tabla, celda o fila según se aplique a <table>, <td> o <tr>.
 - Colores aplicados mediante valor hexadecimal o nombre del color.
 - Sintaxis:
 - <table bgcolor = "color">
- **colspan:**
 - Permite unir un número determinado de celdas en horizontal.
 - Sintaxis:
 - <td colspan = "número de celdas a combinar">contenido de celda</td>
 - Ejemplo:
 - <td colspan = "2">Celdas 1 y 2 unidas en horizontal</td>
- **rowspan:**
 - Permite unir un número determinado de celdas en vertical.
 - Sintaxis:

- `<td rowspan = "número de celdas a combinar">contenido de celda</td>`
- Ejemplo:
 - `<td rowspan = "2">Celdas 1 y 2 unidas en vertical</td>`

CONTENEDORES.

- Un contenedor sirve para agrupar contenidos relacionados.

Genéricos.

- **`...`**
 - Contenedor genérico para líneas de texto.
 - Permite crear un contenedor para poder seleccionar contenido dentro de elementos de texto como párrafos u otros, y así poder actuar sobre él, por ejemplo, aplicándole un formato.
 - Sintaxis:
 - `texto`
- **`<div>...</div>`**
 - Contenedor de bloque sin significado semántico.
 - Define una división.
 - Permite agrupar todo tipo de contenidos: elementos de bloque, línea y otras divisiones, para posteriormente con CSS aplicarles un estilo común.
 - Son invisibles a no ser que se les dé un formato, bordes o colores de fondo, por ejemplo.
 - Las divisiones pueden anidarse, solaparse, sobreponerse y ocultarse.
 - Atributos más comunes:
 - `id`, `class`, `title`, `width`, `height` y `style`.

Semánticos.

- Se ha incluido en HTML 5 etiquetas semánticas que aportan significado a determinadas secciones de un documento web.
- Así los buscadores interpretan mejor las páginas web y se fijan en la información más relevante.
- Navegadores de voz pueden ajustarse para ir directos al contenido o sección saltándose la información menos relevante.
- Útiles para accesibilidad.
- Se pueden anidar entre ellos y con los genéricos.
- Tipos:
 - **`<header>...</header>`**
 - Va incluida en `<body>`.
 - Se usa para crear una cabecera o zona de encabezados.
 - Uso:
 - Incluir logotipos, carritos de la compra, enlaces de idioma, fecha y hora, botones de registro o identificación, títulos, barras de navegación, ...

- **<footer>...</footer>**
 - Se usa para crear una zona de pies de página.
 - Uso:
 - Ayuda, Copyright, Condiciones de uso, enlaces a redes sociales, Política de privacidad, política de cookies, formas de contacto, mapas del sitio, ...
- **<nav>...</nav>**
 - Permite representar una lista o zona de navegación con enlaces.
 - Uso:
 - Crear la barra de navegación principal del sitio.
 - Crear enlaces ente páginas del sitio.
 - Se puede incluir una barra de navegación principal en <header>.
- **<aside>...</aside>**
 - Permite crear zonas laterales con contenido secundario o vinculado al contenido principal.
 - Uso:
 - Barras de navegación lateral.
 - Zona de widgets.
 - Glosarios.
 - Comentarios.
 - Entradas de un blog.
 - Notas.
 - Listas de tweets.
 - Enlaces a otros blogs.
 - Presentación del autor.
- **<main>...</main>**
 - Representa al contenido principal de la página.
 - Este debe ser único y no estar repetido en la página.
 - Uno solo por página.
 - Uso:
 - Incluir el contenido principal.
 - Otros contenedores como <article>, <aside>, <section>, <nav>,
- **<section>...</section>**
 - Permite agrupar elementos que comparten una misma temática.
 - Podría agrupar en una misma sección un contenido estructurado con su cabecera, pie, sección principal, etc.
 - Uso:
 - Agrupar contenido temático.
- **<article>...</article>**
 - Permite insertar contenido independiente o autónomo.
 - Podría agrupar en una misma sección de artículo un contenido estructurado con su cabecera, pie, sección principal, etc.

- Uso:
 - Contenido textual o multimedia de la información.
 - Artículo de blogs o actualidad.
 - Mensajes de foros.

ATRIBUTOS GLOBALES.

Los **atributos globales** son atributos comunes a todos los elementos HTML.

- **id**
 - Permite asignar un identificador para un elemento o etiqueta.
 - Debe ser único en el documento o página web.
 - Se usa para referirse al elemento o etiqueta desde vínculos, scripts de JavaScript, definiciones de estilo de CSS u otros programas.
- **class**
 - Permite asignar uno o varios nombres de clase a distintos elementos o etiquetas.
 - Se utiliza para que otros programas como CSS y JavaScript seleccionen y accedan a elementos de la página web.
- **title:**
 - Muestra un texto descriptivo al pasar por encima de un enlace, imagen, reproductor de audio u otro objeto.
 - Ideal para accesibilidad, en especial para los navegadores y lectores de voz.
 - Sintaxis:
 - title=" texto descriptivo".
- **style**
 - Atributo que permite incluir en el código HTML propiedades de CSS.
 - Aunque se recomienda que los estilos se definan en un archivo o archivos separados, empleados desde este atributo tienen el propósito de permitir un estilo rápido, por ejemplo, con fines de prueba.
 - El valor del atributo es una propiedad de CSS.
 - Sintaxis:
 - Una sola propiedad:
 - <etiqueta style = "propiedad: valor">
 - Varias propiedades:
 - <etiqueta style = "propiedad 1: valor ; propiedad 2: valor;...; propiedad N: valor">

Algunas propiedades de CSS que se pueden incluir en el atributo style.

Sintaxis genérica para las declaraciones de estilo en CSS:

- propiedad: valor

Tipo de fuente:

- font-family: tipo de fuente 1, tipo de fuente 2,
- Ejemplo:
 - `<p style="font-family: comic sans ms, arial, serif; font-size:1em">`

Tamaño de Fuente:

- font-size: número unidad de medida.
- Unidades de medida:
 - pt, px, mm, cm, in, em, %...
 - 1pt = 0,35 mm.
 - 12pt = 16 px.
- Ejemplo:
 - `<p style="font-size: 20pt">`
 - `<p style="font-size: 30px">`

Alineación de texto:

- text-align: valor.
- Valores:
 - center.
 - left.
 - right.
 - Justify.
- Ejemplo:
 - `<p style="text-align: center">`

Negrita:

- Font-weight: bold
- Ejemplo:
 - `<p style="font-weight: bold">`

Cursiva:

- Font-style: italic
- Ejemplo:
 - `<p style="font-style: italic">`

Subrayado:

- text-decoration: underline
- Ejemplo:
 - `<p style=" text-decoration: underline">`

Suprayado:

- text-decoration: overline
- Ejemplo:
 - `<p style=" text-decoration: overline">`

Tachado:

- text-decoration: line-through
- Ejemplo:
 - `<p style=" text-decoration: line-through">`

Versales o versalitas:

- Font-variant: small-caps
- Ejemplo:
 - `<p style=" font-variant: small-caps">`

Espacio entre caracteres.

- Permite aumentar o reducir el espacio entre caracteres.
- letter-spacing: número unidad medida.
 - Si el número es positivo, aumenta el espacio entre caracteres.
 - Si el número es negativo, se reduce el espacio entre caracteres.
- Ejemplo:
 - `<p style=" letter-spacing: 5px">`

Superíndice:

- Eleva el índice por encima de la línea base de escritura.
- Vertical-align: super
- Ejemplo:
 - `<p style=" vertical-align: super">`

Subíndice:

- Desplaza el índice por debajo de la línea base de escritura.
- Vertical-align: sub
- Ejemplo:
 - `<p style=" vertical-align: sub">`

Color de texto:

- color: color
- Ejemplo:
 - `<p style=" color: blue">`
 - `<p style=" color: #76a4c1">`

- `<p style=" color: rgb(34,56,128)">`

Color de fondo:

- background-color: color
- Ejemplo:
 - `<p style=" background-color: blue">`
 - `<p style=" background-color: #76a4c1">`
 - `<p style=" background-color: rgb(34,56,128)">`

Anchura:

- width: número unidad de medida.
- Unidades de medida:
 - pt, px, mm, cm, em, %, ...
- Ejemplo:
 - `<p style="width: 300px">`
 - `<p style="width: 30%">`

Altura:

- height: número unidad de medida.
- Unidades de medida:
 - pt, px, mm, cm, em, %, ...
- Ejemplo:
 - `<p style="height: 300px">`
 - `<p style="height: 30%">`

Márgenes Externos:

Hay varios atributos:

- margin: 1 valor o 2 valores o 3 valores o 4 valores o auto.
 - 1 valor: 4 márgenes iguales (auto o valor numérico).
 - 2 valores: 1º valor márgenes verticales (superior e inferior), 2º valor márgenes horizontales (izquierdo y derecho).
 - 3 valores: 1º margen superior, 2º número: márgenes izquierdo y derecho, 3º margen inferior.
 - 4 valores: 1º margen superior, 2º margen derecho, 3º margen inferior y 4º margen izquierdo.
 - **auto**: El valor de los márgenes por los 4 lados se reparte equitativamente según el espacio disponible en el contenedor.
- margin-top: valor
- margin-left: valor
- margin-right: valor

- margin-bottom: valor
- Ejemplos:
 - <p style=" margin-left: 300px">
 - <p style=" margin: 45px>
 - <p style=" margin: auto>
 - <p style=" margin: 45px 30px>
 - <p style=" margin: 45px 30px 20px>
 - <p style=" margin: 45px 30px 20px 50px>

Márgenes Internos:

Hay varios atributos:

- padding: 1 valor o 2 valores o 3 valores o 4 valores.
 - 1 valor: 4 márgenes iguales (auto o valor numérico).
 - 2 valores: 1º valor márgenes verticales (superior e inferior), 2º valor márgenes horizontales (izquierdo y derecho).
 - 3 valores: 1º margen superior, 2º número: márgenes izquierdo y derecho, 3º margen inferior.
 - 4 valores: 1º margen superior, 2º margen derecho, 3º margen inferior y 4º margen izquierdo.
- padding-top: valor
- padding-left: valor
- padding-right: valor
- padding-bottom: valor
- Ejemplo:
 - <p style=" padding-left: 300px">
 - <p style=" padding: 45px>
 - <p style=" padding: auto>
 - <p style=" padding: 45px 30px>
 - <p style=" padding: 45px 30px 20px>
 - <p style=" padding: 45px 30px 20px 50px>

Bordes:

- Sintaxis:
 - border: grosor estilo color
- Valores:
 - **grosor:**
 - Se especifica con un número y una unidad de medida.
 - Es obligatorio.
 - **estilo:**
 - Es obligatorio.
 - Existen varios estilos, siendo los más comunes:

- solid: borde de trazo simple continuo.
- double: borde de trazo doble continuo.
- dotted: borde discontinuo de puntos.
- dashed: borde discontinuo de guiones.
- **color:**
 - Aplica un color al borde especificado con los formatos de color habituales.
 - No es obligatorio, siendo el color por defecto el negro.
- Ejemplo:
 - `<p style=" border: 5px double red">`

ENLACES

Concepto.

- Un enlace es la esencia de HTML y de las páginas web, ya que permite navegar por los sitios web.
- Puede ser un texto o imagen que se encuentra en una página web y en la que, al pulsar, se es dirigido a otro sitio o contenido diferente.
- También se denominan links, vínculos o hipervínculos.

Destino de un enlace.

- Un sitio externo.
- Otra página del sitio web actual.
- A un ancla, es decir, a un lugar determinado de la página actual.
- Una dirección de correo electrónico.
- Un archivo para descargar.
- Un servidor ftp para descargar un archivo.
- Un teléfono o Skype.
- Destino vacío.

Etiqueta.

- `<a>Contenido del enlace`
- El contenido del enlace puede ser un texto, imagen, etc., y es el lugar en el que se pulsa.
- Elemento de línea.

Atributos:

- **href:**
 - Define la dirección del destino del enlace, que puede ser:
 - URL (absoluta o relativa).
 - Ancla.
 - Dirección de correo electrónico.
 - Un archivo.

- Un teléfono.
- Sintaxis:
 - `Texto del enlace`
- Ejemplo:
 - `El Corte Inglés`
- **target:**
 - Especifica donde se abrirá el enlace.
 - Sintaxis:
 - `<a target = "valores">`
 - Valores:
 - _blank:
 - El enlace se abre en una nueva ventana o pestaña.
 - _self:
 - El enlace se abre en la ventana actual.
 - No es necesario especificarlo ya que es el valor por defecto.
 - _top:
 - En ventanas con marcos, el enlace se abre en la ventana de nivel superior y se eliminan los marcos.
 - _parent:
 - En ventanas con marcos, el enlace se abre en la ventana anterior o inmediatamente superior.
 - Ejemplo:
 - `El Corte Inglés`
- **title:**
 - Muestra texto descriptivo al pasar por el enlace.
 - Sintaxis:
 - `title=" texto descriptivo".`
- **type:**
 - Valor informativo, no funcional.
 - Especifica tipo de contenido o medio enlazado en forma de tipos MIME para la URL enlazada.
 - **MIME** (Multipurpose Internet Mail Extensions)
 - Especifica los tipos de medios o contenido de Internet.
 - Buena práctica incluirlo porque provee información del tipo de medio.
 - Sintaxis:
 - `tipo/árbol.subtipo + sufijo; parámetros` (En color rojo valores opcionales)
 - Ejemplos:
 - `application/rss+xml`
 - `image/jpg`
 - `video/mp4`
 - `text/html`

- Sintaxis:
 - type = "tipo MIME"
- Ejemplo:
 - type = "video/avi"
- **rel:**
 - Especifica la relación entre el documento de destino con el documento de origen o del enlace.
 - Sintaxis:
 - rel = "valores"
 - Valores:
 - author: Enlace al autor del documento o artículo actual.
 - Help: vínculo a una página de ayuda.
 - License: Descripción de una licencia de copyright que cubre el contenido principal del documento actual.
 - stylesheet: Enlace a una hoja de estilo de CSS que incluye los estilos para el documento actual.
 - follow/nofollow: El contenido del enlace (no) es seguido a la hora de posicionarlo en el buscador.
 - index/noindex: El enlace es o no indexado o publicado en la base de datos del buscador.
 - alternate: Representación alternativa del documento actual.
 - next: En una serie de documentos web es el siguiente.
 - prev: En una serie de documentos web es el anterior.
 - Ejemplos:
 - Ir a procesadores
 -
 - Ir a discos duros
 - Ver acuerdo de licencia
 - Ver página en inglés
 - Ver página en inglés
- **hreflang:**
 - Especifica el lenguaje que utiliza el documento enlazado.
 - Sintaxis:
 - hreflang = "Código de idioma">
 - Ejemplo:
 - hreflang = "en">

DIRECCIONES ABSOLUTAS, RELATIVAS Y RUTAS DE ACCESO.

Ruta de acceso.

Camino que hay que seguir para llegar a una determinada posición en un disco (local o remoto) y así utilizar un recurso.

Rutas de acceso locales.

- Windows:
 - Las unidades de disco físicas se nombran con letras, como por ejemplo, C:, D:, E:, etc.
 - Las unidades lógicas o directorios raíz con la barra \ más el nombre de la unidad física:
 - C:\ o D:\.
 - Es a partir del directorio raíz donde empiezan las rutas de acceso, separándose cada carpeta o directorio que hay que abrir con una barra \.
 - Ejemplos de rutas de acceso:
 - D:\Cursos\Ejercicios\HTML\html.docx
 - C:\Program Files\Microsoft SQL Server\130\Tools\Binn\Resources\es-ES\SqlLocalDB.rll.mui
- Linux:
 - Concepto idéntico a Windows, sólo que el directorio raíz se nombra únicamente con la barra /.
 - Ejemplos de rutas de acceso:
 - /Cursos/Ejercicios/HTML/html.docx
 - /home/usuario1/fotos/paisaje.jpg

Rutas de acceso remotas.

- Son rutas al contenido de un servidor o equipo remoto.
- Para encontrar un recurso se emplea la denominada URL.

URL.

- **Uniform Resource Locator** - Localizador Uniforme de Recursos.
- Sistema de localización o direccionamiento de los documentos web.
- Permite a un navegador encontrar una dirección o sitio en Internet, ya que es la dirección de un recurso en la red.

Estructura de una URL.

- Protocolo://tipo de servidor.nombre del dominio.dominio/ruta de acceso/archivo.html
- Ejemplo:
 - <https://www.elcorteingles.es/moda/hombre/accesorios-de-hombre/gafas-de-sol/gafas.html>

Elementos de una URL.

Protocolo:

- Conjunto de reglas que gobiernan la comunicación entre dispositivos que están conectados a una red.
- Incluyen instrucciones que permiten a los dispositivos identificarse y conectarse entre sí, además de aplicar reglas de formateo, para que los mensajes viajen de la forma adecuada de principio a fin.
- Dichas reglas de formateo determinan si los datos son recibidos correctamente o si son rechazados o ha habido algún tipo de problema en la transferencia de la información.

Algunos protocolos.

- **TCP/IP.**
 - **TCP.**
 - Protocolo de Control de Transmisión.
 - Garantiza que los datos se transmiten de manera adecuada a través de Internet.
 - Su función principal es asegurar que los datos a destino.
 - **IP.**
 - Protocolo de Internet.
 - Determina cómo se transmiten los datos a través de la red.
 - Queda definido por dos características:
 - **Direccionamiento:**
 - Asegura que cualquier dispositivo conectado a una determinada red cuente con una dirección de IP única. Así se puede conocer el origen y el destino de los datos.
 - **Enrutamiento:**
 - Establece el camino para el tráfico de datos basándose en la dirección IP.
 - Este trabajo lo realizan los routers.
 - Dirección IPv4:
 - protocolo://0-255.0-255.0-255.0-255
 - https://198.129.6.125
- **http.**
 - Hypertext Transfer Protocol - Protocolo de transferencia de hipertexto.
 - Es utilizado por navegadores web para solicitar archivos HTML a los servidores.
 - Permite solicitar y transferir páginas web entre un servidor y un equipo cliente.
- **https.**
 - Hypertext Transfer Protocol Secure - Protocolo seguro de transferencia de hipertexto.
 - Protocolo de transferencia de hipertexto en modo seguro.
 - Los datos viajan cifrados a través de un canal seguro que conecta en exclusiva al cliente con el servidor.
- **ftp.**
 - File Transfer Protocol - Protocolo de transferencia de archivos.
 - Utilizado para compartir archivos entre dos ordenadores.
- **mailto, smtp, pop3 o imap.**
 - Son protocolos de correo electrónico con distintas características.

- mailto:nombre usuario@servidor
- **telnet.**
 - Protocolo de conexión remota a un ordenador.
- **file.**
 - Protocolo de archivo local.
 - Permite crear un enlace a un archivo local.
- **IRC.**
 - Internet Relay Chat.
 - Protocolo de comunicación en tiempo real basado en texto, que permite debates o charlas entre dos o más personas.
- **DNS.**
 - Servidor de nombres de dominio.
 - Se encarga de asignar y administrar los nombres de dominio.

Tipo de servidores más comunes.

- **www** - servidor web.
- **ftp** - servidor ftp.

Nombre del dominio.

- Nombre que identifica a la organización, institución, empresa, persona, marca comercial, etc. que posee el dominio.
- Es en realidad un nombre fácil de recordar asociado a una dirección IP física de Internet.

Dominio.

- Nombre único que identifica a una subárea de Internet.
- Tipos:
 - Genéricos:
 - Son de propósito general.
 - Originales: com, net, edu, org, mil, gob y nato.
 - Posteriores: biz, info, museum, kids, jobs, travel, xxx, ...
 - Territoriales:
 - Identifican un país, región o continente.
 - Originales: es, pt, br, fr, uk, ...
 - Posteriores: asia, gal, lat, ...

Página index.html

- Página de inicio o entrada a un sitio web.
- Otros nombres: index.php, default.asp, etc.

Dirección absoluta:

- Aquella que contiene toda la ruta de acceso hasta un archivo, incluida la carpeta o directorio raíz, si es una ruta local, o el protocolo http/https, si es una ruta externa.
- Ejemplo ruta externa:
 - A una página especificada:
 - <https://www.elmundo.es/madrid/2021/05/07/6094f9bd21efa031238b457d.html>
 - Si no se accede a la página inicial de un sitio, sino a otra en cualquier posición de la ruta, hay que escribir el nombre del archivo.
 - Si se omite el nombre del archivo en la ruta se entiende que es index.html.
 - <https://www.elmundo.es/>
- Ejemplo ruta local:
 - C:\Windows\explorer.exe
 - D:\Cursos\Documentos\curso.html

Dirección relativa:

- Aquella que expresa a partir de punto o ubicación determinada de una ruta de acceso.
- Se usan dentro de un mismo sitio web.
- Relativa porque dependen de la posición que ocupa, dentro de la estructura de archivos, el archivo que contiene la referencia relativa.

ENLACES INTERNOS Y USO DE DIRECCIONES RELATIVAS.

- Enlaces a páginas o archivos que están alojados en el mismo sitio web, tanto a nivel remoto, como a nivel local.
- Uso de direcciones relativas.
- Opciones:
 - **Ruta de acceso al mismo nivel.**
 - Enlace entre páginas que están en la misma carpeta.
 - Sintaxis:
 -
 - Ejemplo:
 - **Ruta de acceso a nivel inferior.**
 - Enlace a una página que está en una carpeta ubicada en un nivel inferior de la estructura del sitio.
 - Si la ruta incluye varias carpetas unas dentro de otras, se especifican todas en el mismo orden en el que están organizadas.
 - Sintaxis:
 -
 -
 - Ejemplo:
 -

- ``
-
- **Ruta de acceso a nivel superior.**
 - Enlace a una página que está en una carpeta ubicada en nivel superior de la estructura del sitio.
 - Para acceder a la carpeta donde está la página a enlazar hay que subir de nivel o salir de la carpeta donde está la página que llevará el enlace.
 - Notación para subir de nivel:
 - `../`
 - Si hay que subir varios niveles, es decir, salir de varias carpetas, se incluye la notación `"../"` tantas veces como carpetas se vaya subiendo.
 - Sintaxis:
 - ``
 - ``
 - Ejemplo:
 - ``
 - ``
- **Ruta de acceso entre carpetas de distintos niveles.**
 - Hay que salir de unos niveles o carpetas para entrar en otros niveles o carpetas.
 - En la ruta se especifica primero la notación para subir de nivel y a continuación, el nombre de la carpeta a la que hay que acceder.
 - Puede ser necesario salir de una o varias carpetas para entrar después en una o varias carpetas.
 - Sintaxis:
 - ``
 - ``
 - ``
 - ``
 - Ejemplos:
 - ``
 - ``
 - ``
 - ``

ENLACES A CORREO ELECTRÓNICO.

- Se incluyen en el atributo `"href"` la/s dirección/es de correo del destinatario/s incluyendo el protocolo `"mailto"`.
- Incluir direcciones principales:

- Varias direcciones se separan con una coma o un punto y coma.
- Sintaxis:
 - `texto del enlace`
 - `texto del enlace`
- Ejemplos:
 - `Correo a un Destinatario`
 - `Correo a Varios Destinatarios`
- Incluir otros elementos de la cabecera y cuerpo del mensaje.
 - Tras la dirección/es principales se pone una interrogación de cierre (?) y después los elementos a añadir.
 - **Elementos:**
 - Dirección con copia:
 - Se envía una copia a otro destinatario.
 - Sintaxis:
 - `?cc=dirección de copia.`
 - Ejemplo:
 - `Correo con Copia`
 - Dirección con copia oculta:
 - Se envía una copia oculta a otro destinatario.
 - Sintaxis:
 - `?bcc=dirección de copia.`
 - Ejemplo:
 - `Correo con Copia Oculta`
 - Asunto del mensaje:
 - Sintaxis:
 - `?subject = texto del asunto.`
 - Ejemplo:
 - `Correo con Asunto`
 - Texto en el mensaje:
 - Para incluir indicaciones o instrucciones para el usuario que va a enviar el mensaje.
 - Se pueden usar caracteres de salto como:
 - `%20 →` Añadir espacio en blanco.
 - `%0D%0A →` Dar un salto de línea.
 - Sintaxis:
 - `?body = texto del mensaje`

- Ejemplo:
 - ``
- Incluir varios elementos en la cabecera.
 - Cada elemento se encadena al siguiente con un ampersand (&) sin incluir espacios en blanco.
 - Ejemplo:
 - `Correo con Copia y Asunto`

ENLACES PARA DESCARGA DE ARCHIVOS.

- Se emplea para descargar archivos que no sean páginas web como, por ejemplo, documentos, imágenes, instaladores, pdf's, etc.
- En el atributo href se incluye el nombre del archivo y su extensión.
- Debe estar en la misma carpeta que el documento web, si no, hay que escribir la ruta de acceso al archivo.
- Debe existir en el equipo un programa que pueda abrir el archivo descargado.
- Sintaxis:
 - `Texto del enlace`
 - `Texto del enlace`
- Ejemplo:
 - `Descargar presentación`

ENLACES A UN TELÉFONO.

- Necesario tener instalada una aplicación que permita hacer llamadas telefónicas como "Tu teléfono" de Microsoft, VoiceCheap, PC-telephone, VoipBuster u otros.
- Sintaxis:
 - `Texto del enlace`
- Ejemplo:
 - `Llámanos`

ENLACE A SKYPE.

- Necesario tener instalada la aplicación Skype.
- Sintaxis:
 - `Texto del enlace`
- Ejemplo:
 - `Llámanos`

ENLACE VACÍO O SIN DESTINO.

- Enlace que no abre ningún destino.

- Sintaxis:
 - `Texto del enlace`

ENLACES A LA PROPIA PÁGINA O ANCLAS.

- Enlace a otra parte del mismo documento web o a un punto concreto de otra página.
- Enlace se hace a un ancla o punto de fijación.
- Pasos para su creación:

1. Declarar el ancla.

- Colocar el cursor en el punto de la página donde se va a crear el ancla.
- Sintaxis:
 - ``
- Ejemplo:
 - ``

2. Crear el enlace hacía el ancla.

- En la misma página:
 - Se crea un enlace con el #nombre del ancla en otro sitio distinto a donde está el ancla.
 - Sintaxis:
 - `Texto del enlace`
 - Ejemplo:
 - `Ir al final de la página`
- A otra página:
 - Se crea un enlace con el #nombre del ancla, la página donde está el ancla y la ruta de acceso si la hubiera.
 - Crear el ancla en la página de destino y en enlace al ancla en la página de origen.
 - Sintaxis:
 - `Texto del enlace`
 - Ejemplo:
 - `Ir al final de la página`
 - `Ir al final de la página`
 - `Ir al final de la página`

ENLACE A UN FAVICON.

- Concepto.
 - Un favicon, o icono para favoritos, es un pequeño icono que aparece, según navegador, en:
 - La barra de título o pestañas del navegador.
 - El menú de favoritos/marcadores.
 - Facilitan a los usuarios la identificación de una página web cuando hay muchas pestañas abiertas en el navegador.

- Inicialmente deben llevar en el nombre del archivo la extensión .ico (formato gráfico de los iconos), no obstante, algunos navegadores soportan favicons en otros formatos de imagen (png, jpg, ...).
- Normalmente el nombre del archivo suele ser favicon para reconocerlo, pero también es válido cualquier otro.
- Creación.
 - Se puede crear un icono desde cero o a partir de un fichero de imagen (jpg, gif, png, etc.), usando un programa de dibujo.
 - El icono debería tener 16 × 16 píxeles (tamaño más común), y guardarse con el nombre favicon.ico.
 - Se recomienda guardarlo en la raíz del sitio.
- Sintaxis:
 - <link rel="shortcut icon" href = "ruta de acceso/archivo.extensión" type="image/ico">
 - <link rel="shortcut icon" href = "ruta de acceso/archivo.extensión" type="image/x-icon">
 - La ruta de acceso puede ser absoluta o relativa, interna o externa.
- Ejemplo:
 - <link rel="shortcut icon" href = "favicon.ico" type="image/ico">
 - <link rel="shortcut icon" href = "imagenes/favicon.png" type="image/ico">

Cambio de colores a los enlaces.

- Colores por defecto:
 - Vínculo no visitado: azul
 - Vínculo visitado: violeta o morado.
 - Vínculo activo: rojo.
- Cambio de colores:
 - Incluir atributos correspondientes en <body>.
 - Sintaxis:
 - Vínculo no visitado: link = "color".
 - Vínculo visitado: vlink = "color".
 - Vínculo activo: alink = "color".
 - Ejemplo:
 - <body link="Green" vlink="#87ac7d" alink="pink">

MULTIMEDIA

- Las imágenes, audios y videos tienen un propietario y, por tanto, éste unos derechos.
- **Licencias Creative Commons.**
- Muy utilizada para compartir imágenes, audios, videos y otros documentos.
- Licencias de derecho de autor abierta y/o libre para obras culturales, científicas y educativas.
- Permiten que un autor otorgue permiso para compartir y usar su trabajo bajo los términos y condiciones de su elección.

- El autor no pierde los derechos sobre su obra.
- Limitaciones:
 - **Atribución / Attribution (BY).**
 - Mención expresa del autor en los términos que éste especifique al copiar, distribuir, exhibir o modificar su obra.
 - **Compartir igual /Share Alike (SA).**
 - Permite usar una obra para crear otra, pero se tendrá que publicar con la misma licencia que la original.
 - **No comercial / Non-Commercial (NC).**
 - La obra no puede ser usada con fines comerciales.
 - Se puede distribuir, copiar, exhibir o modificar, pero no obtener un beneficio económico con ello.
 - **No derivada / No Derivate Works (ND).**
 - Se puede distribuir, copiar, exhibir y obtener un beneficio económico con ello, pero no modificar la obra original.
 - No se puede crear una obra derivada de la original.
- **Tipo de licencias Creative Commons.**
 - **CC0** (Creative Commons Zero)
 - Licencia de dominio público.
 - Se puede usar una obra sin restricciones, sin permiso y sin atribución.
 - **CC BY** - Sólo atribución.
 - **CC BY-SA** - Atribución + compartir igual.
 - **CC BY-ND** - Atribución + No derivada.
 - **CC BY-NC** - Atribución + No comercial.
 - **CC BY-NC-SA** - Atribución + No comercial + Compartir Igual.
 - **CC BY-NC-ND** - Atribución + No comercial + No derivada.
- **Sitios web con bancos de imágenes Creative Commons.**
 - Morguefile.
 - Freepik.
 - Pixabay.
 - Unsplash.
 - Flickr.
 - Gratisography.
 - <https://search.creativecommons.org/>

IMÁGENES

Tipos o formatos de imagen para Internet.

1. GIF.

- Graphics Interchange Format.
- Permite codificar imágenes con un máximo de 256 colores.

- Formato de compresión rápido y eficaz.
- Compresión sin pérdida de datos y calidad.
- Peso reducido.
- Función de entrelazado:
 - Previsualización rápida de la imagen que progresivamente se va haciendo más nítida.
- Fácilmente editables y modificables.
- Permiten crear animaciones: Gif's animados.
- Uso para logotipos, botones, líneas, barras, símbolos, etc.
- Sustituido poco a poco por png.

2. JPG o JPEG.

- Joint Photographic Expert Group.
- Uso universal.
- Permite codificar imagen de hasta 16,7 millones de colores.
- Compresión con pérdida de calidad.
- Pérdida de calidad acumulativa.
- Tasa de compresión: a mayor tasa, mayor pérdida.
- Soporte para todos los modelos de color: RGB, CMYK, Escala de grises, Lab, HSB,...
- Función de entrelazado.
- No permite animaciones.
- No permite transparencia.
- Adecuado para fotografía.
- Para una página web comprimir a 300 o 400 KB.
- Resolución adecuada para pantalla 72 píxeles a 150 píxeles.

3. PNG.

- Portable Network Graphic.
- Concebido para el uso de imágenes en páginas web.
- Admite transparencia.
- Admite compresión sin pérdida de datos o calidad. (Entre un 5% y 25%).
- Función de entrelazado.
- Formato abierto no patentado.
- No admite animaciones.

4. WEBP.

- Formato desarrollado por Google.
- Admite compresión con y sin pérdida.
- Admite compresión de hasta un 39% con pérdida de calidad imperceptible.
- Admite transparencia.
- Permite animaciones.
- A igualdad de calidad de imagen que jpg y png pesan menos.

Optimización de imágenes para Internet.

- **Programas.**
 - Adobe Photoshop y Lightroom (P)
 - Gimp (G)
 - JPG_Mini (P)
 - ImageOptim(G)
 - Riot(G)
- **Servicios Online**
 - Tinypng
 - Compressor.io
 - Optimizilla
 - Imageoptimizer,
 - Resize Photos

Etiquetas Para Imágenes.

- Permite insertar imágenes en un documento web.

<figcaption>...</figcaption>

- Para incluir una leyenda debajo de una imagen.
- Sintaxis:
 - <figcaption>Texto de la leyenda</figcaption>

<figure>...</figure>

- Permite para asociar una leyenda a su imagen.
- Para ello ambas etiquetas y <figcaption> van incluidas dentro de <figure>.
- Sintaxis:

```
<figure>
  <img src = "imagen. extensión">
  <figcaption>Texto de la leyenda</figcaption>
</figure>
```

•

Imagen como enlace:

- Se inserta la etiqueta dentro de las etiquetas <a>...
- Sintaxis:
 - <a>
- Ejemplo:

```
<a href = "https://www.elcorteingles.es/moda/hombre/accesorios-de-hombre/gafas-de-sol/">
  <img src = "gafas.jpg">
```


Atributos:

- **src**
 - Obligatorio.
 - Sirve para especificar el archivo de imagen.
 - Sintaxis:
 - Local:
 - ``
 - Externa:
 - ``
 - Ejemplos:
 - Locales:
 - ``
 - ``
 - ``
 - Externo:
 - ``
- **width**
 - Permite especificar la anchura de la imagen en píxeles.
 - Ejemplo:
 - ``
- **height**
 - Permite especificar la altura de la imagen en píxeles.
 - Ejemplos:
 - ``
 - ``
- **alt:**
 - Sirve para incluir un texto alternativo cuya función es informativa si no se carga la página.
 - Importante de cara a la accesibilidad si no se puede ver la imagen por discapacidad.
 - Importante también para el posicionamiento.
 - Sintaxis:
 - `alt = "texto descriptivo"`
 - Ejemplo:
 - ``
- **title:**
 - Muestra texto descriptivo al pasar por la imagen.
 - Sintaxis:
 - `title = "texto descriptivo".`

MAPA DE IMÁGENES.

- Permiten crear un conjunto de enlaces dentro de una imagen.
- Permiten enlazar diferentes zonas de una imagen a distintos destinos.

<map>...</map>

- Define un mapa de imágenes.
- Atributo:
 - **name**
 - Se utiliza para asignar un nombre para el mapa.
 - Obligatorio.
 - Sintaxis:
 - <map name = "nombre del mapa"></map>
 - Ejemplo:
 - <map name = "patatas"></map>

<area>

- Permite definir las áreas enlazables.
- No tiene etiqueta de cierre y se incluye dentro de la etiqueta <map>
- Sintaxis:
 - <area shape = "forma" coords = "coordenadas" href ="enlace" alt ="Texto alternativo."
- Atributos:
 - **coords**:
 - Especifica en píxeles los puntos a partir de los cuales se genera el área enlazable.
 - **shape**:
 - Define la forma que tendrá el área enlazable.
 - Tipos de áreas:
 - **circle**:
 - Define un área circular.
 - Las coordenadas necesarias son:
 - x = valor según el eje horizontal en píxeles.
 - y = valor según el eje vertical en píxeles.
 - Valor para el radio.
 - Sintaxis:
 - <area shape = "circle" coords ="x,y,radio" href ="destino">
 - Ejemplo:
 - <area shape = "circle" coords ="451,332,50" href ="inicio.html">
 - **rect**:
 - Define áreas rectangulares.
 - Se utilizan las coordenadas x1 e y1 para la esquina superior izquierda, y x2 e y2 para la esquina inferior derecha del cuadrado o rectángulo que formará el área enlazable.

- Sintaxis:
 - `<area shape = "rect" coords ="x1,y1,x2,y2" href="destino">`
- Ejemplo:
 - `<area shape = "rect" coords ="471,15,661,95" href="inicio.html">`
- **poly:**
 - Define areas poligonales.
 - Se utilizan las coordenadas x1, y1, x2, y2, ...xn, yn para cada vértice el polígono.
 - Es indiferente qué coordenadas se definen como iniciales.
 - Para cerrar el polígono, las coordenadas primera y última deben coincidir.
 - Sintaxis:
 - `<area shape = "poly" coords =" x1, y1, x2, y2,...xn, yn" href="destino">`
 - Ejemplo:
 - `<area shape = "poly" coords ="471,15,78,90,661,95,67,34,123,45,471,15" href="inicio.html">`
- **default:**
 - Para enlazar el resto de una imagen no definido mediante áreas enlazables.
 - Hay incluir después del resto de áreas enlazables para que no las anule.
 - No necesita coordenadas.
 - Sintaxis:
 - `<area shape ="default" href= "destino">>`
 - Ejemplo:
 - `<area shape ="default" href= "inicio.html">>`

Asociar mapa de imagen a una imagen:

- Se usa el atributo usemap en la etiqueta con valor igual que el valor asignado en el atributo name de <map>.
- Sintaxis:
 - ``
- Ejemplo:
 - ``

AUDIO.

Formatos de archivos de audio:

- **OGG**
 - Fundación Xiph.org.
 - Libre sin patentes.
 - Alternativa a mp3.
 - Utiliza el códec Vorbis.
- **MP3**
 - MPEG 1 y MPEG2 Audio Layer 3.
 - Compresión alta con pérdida de calidad aceptable para el oído.
 - Tecnología propietaria con patente y licencia.
- **ACC**

- Advanced Audio Coding.
- Compresión con pérdida.
- Mejor rendimiento que mp3 y ogg.
- Soportado por Apple (iTunes, iPod).
- **WAV**
 - Wave Form Audio File Format.
 - Formato de Microsoft e IBM.
 - Sin compresión calidad alta.
 - Tamaño archivo alto.
- **FLAC**
 - Free LossLess Audi Códec
 - Libre.
 - Sin pérdida.
 - Excelente calidad.
 - Mayor tamaño de archivo.

Etiquetas HTML para sonidos.

- **<audio>...</audio>**
 - Etiqueta para reproducir sonidos en documentos web.
 - Permite reproducir cualquier archivo de audio en una página web como canciones, discursos, sonidos, etc.
 - Si el audio no se escucha se puede incluir un texto alternativo:
 - `<audio></audio>`
 - `<p>Texto Alternativo</p>`
- **<source>:**
 - Permite resolver el problema de los distintos formatos de audio que uno navegadores reproducen y otro no.
 - Se especifica varios archivos de audio con distinto formato para que el navegador reproduzca el que soporte.
 - Los archivos de audio se leen en orden, si uno no es soportado, se pasa al siguiente, y si es soportado, se reproduce sin tener en cuenta el resto.
 - Sintaxis:

```

<audio>
  <source src = "archivo de audio.extension1">
  <source src = "archivo de audio.extension2">
  <source src = "archivo de audio.extensionN">
</audio>

```
 - Atributos:
 - **src**:
 - Permite especificar el archivo de audio.
 - Obligatorio.
 - Rutas pueden ser absolutas o relativas.
 - Sintaxis:

- Local:
 - `<audio src = "ruta de acceso/nombre archivo.extensión"></audio>`
- Externa:
 - `<audio src = "http://nombre dominio/ ruta de acceso/nombre archivo.extensión"></audio>`
- Ejemplos:
 - Locales:
 - `<audio src = "playa.mp3"> </audio>`
 - `<audio src = "sonidos/pajaro.ogg"> </audio>`
 - `<audio src = "../we will rock you.flac"> </audio>`
 - Externo:
 - `<audio src = "http://musica.com/rock/we will rock you.flac"> </audio>`
- **controls**:
 - Para Incluir los controles del reproductor de audio (Pausa, Play, Stop, etc.).
 - Ejemplo:
 - `<audio src = "playa.mp3" controls> </audio>`
- **autoplay**:
 - El audio se reproduce automáticamente.
 - Incluir también el reproductor permite controlar el audio y así no molestar.
 - Ejemplo:
 - `<audio src = "playa.mp3" controls autoplay> </audio>`
- **loop**:
 - El audio se repite automáticamente.
- **muted**:
 - Establece audio sin sonido.
- **preload**:
 - Se usa cuando muchos archivos pueden ralentizar la carga de una página.
 - Sintaxis:
 - `preload = "valores"`
 - Valores:
 - **none**:
 - Sin precarga, se reproduce al pulsar Play.
 - **auto**:
 - Precarga o almacenamiento temporal sin haber pulsado Play.
 - Ejemplos:
 - `<audio src = "playa.mp3" controls loop preload = "auto"> </audio>`

VIDEO.

Formatos o tipos de video para Internet.

- Existen múltiples formatos de video como:
 - AVI, MPG, WMV, MOV, H.264, MKV (Matrioska), DIVx, XVID, FLV (Flash Player), ...

- Más comunes:
 - **OGV** (También OGG).
 - Formato libre.
 - Usa el códec Theora.
 - **H.264** (Códec)
 - También se le conoce como MPEG-4 Parte 10.
 - Se puede incorporar a múltiples formatos de contenedor para MP4, MOV, FLV, 3GP, ...
 - Permite compresión digital y video de alta resolución según normas MPEG-4.
 - No es libre ni gratuito.
 - Estándar no oficial de video en la web.
 - **WEBM**
 - Es de Google.
 - Usa el códec VP8 para video y Vorbis para audio.
 - Libre y gratuito.
- **Códec:**
 - Compresor/descompresor de video digital o audio.
 - Su uso puede implicar pérdida de calidad y datos.
 - Ejemplos:
 - XVID, Theora, RealVideo, MPEG-1, MPEG-2 y MPEG-4, H.264, ON2,

Etiquetas HTML para Video.

- **<video>...</video>**
 - Etiqueta para reproducir videos en documentos web.
 - Permite reproducir cualquier archivo de video en una página web siempre que esté instalado el códec adecuado en el sistema operativo.
 - Si se incluye un video usando la etiqueta de <audio>, no se visualiza el video, pero si se oye el sonido.
 - Si el video no se escucha se puede incluir un texto alternativo:
 - <video><p>Texto Alternativo</p></video>
- **<source>:**
 - Permite resolver el problema de los distintos formatos de video que uno navegadores reproducen y otro no.
 - Se especifica varios archivos de video con distinto formato para que el navegador reproduzca el que soporte.
 - Los archivos de video se leen en orden, si uno no es soportado, se pasa al siguiente, y si es soportado, se reproduce sin tener en cuenta el resto.
 - Sintaxis:

```
<video>
  <source src = "archivo de video.extension1">
  <source src = "archivo de video.extension2">
  <source src = "archivo de video.extensionN">
</video>
```
 - Atributos:
 - **src:**
 - Permite especificar el archivo de video.

- Obligatorio.
- Rutas pueden ser absolutas o relativas.
- Sintaxis:
 - Local:
 - `<video src = "ruta de acceso/nombre archivo.extensión"></video>`
 - Externa:
 - `<video src = "http://nombre dominio/ ruta de acceso/nombre archivo.extensión"></video>`
 - Ejemplos:
 - Locales:
 - `<video src = "playa.mp4"> </video>`
 - `<video src = "sonidos/pajaro.avi"> </video>`
 - `<video src = "../we will rock you.webm"> </video>`
 - Externo:
 - `<video src ="http://musica.com/rock/we will rock you.mkv"> </video>`
- **controls**:
 - Para Incluir los controles del reproductor de video (Pausa, Play, Stop, etc.).
 - Ejemplo:
 - `<video src = "playa.mp4" controls> </video>`
- **title**:
 - Muestra texto descriptivo al pasar por el video.
 - Sintaxis:
 - `title = "texto descriptivo"`.
- **width**:
 - Controla la anchura de video en pixeles.
 - Sintaxis:
 - `width = "número px"`
- **height**:
 - Controla la altura de video en pixeles.
 - Sintaxis:
 - `height = "número px"`
 - Si no se especifican width y height, el video se muestra con sus dimensiones originales.
- **poster**:
 - Especifica una imagen a mostrar mientras se está cargando el video, si no se pone, se ve el primer fotograma.
 - Sintaxis:
 - `poster= "ruta de acceso/nombre archivo. extensión"`
 - Ejemplo:

- `<video src = "playa.mp4" poster="fotos/playa de San Juan.jpg" controls> </video>`
 - **autoplay:**
 - El video se reproduce automáticamente.
 - Incluir también el reproductor permite controlar el audio y así no molestar.
 - Ejemplo:
 - `<video src = "playa.mp3" controls autoplay> </video>`
 - **loop:**
 - El video se repite automáticamente.
 - **preload:**
 - Se usa cuando muchos archivos pueden ralentizar la carga de una página.
 - Sintaxis:
 - `preload = "valores"`
 - Valores:
 - **none:**
 - Sin precarga, se reproduce al pulsar Play.
 - **auto:**
 - Precarga o almacenamiento temporal sin haber pulsado Play.
 - Ejemplos:
 - `<video src = "playa.mp4" controls loop preload = "auto"> </video>`
- **Tiempo.**
 - **Inicio de video.**
 - Se puede iniciar el video en su primer segundo o especificar otro momento inicial.
 - Sintaxis:
 - `src = "ruta al archivo/nombre del archivo.extensión#t=tiempo en segundos"`
 - Ejemplo:
 - `src = "matrix.mp4#t=600" // La película comenzará en el minuto 10.`
 - **Intervalo de visualización.**
 - Se puede visualizar un intervalo temporal concreto medido en segundos de un video.
 - Sintaxis:
 - `src = "ruta al archivo/nombre del archivo.extensión#t=tiempo en segundos inicial, tiempo en segundos final"`
 - Ejemplo:
 - `src = "matrix.mp4#t=600, 720" // La película comenzará en el minuto 10 y acabará en el 12.`
- **<track>**
 - Permite incluir subtítulos, títulos, descripciones, capítulos o metadatos para elementos multimedia.
 - Los subtítulos se incluyen en un archivo aparte.
 - Se incluye dentro de `<video>`.
 - Extensiones de archivos con subtítulos:
 - `srt, sub, sbv, mpsub, lrc o cap.`
 - Los subtítulos se crean con programas de edición de subtítulos como:
 - `Subtitle Edit, We Transcriber u otros.`
 - Atributos:
 - **kind:**
 - Permite especificar el tipo de información a mostrar en el video.
 - Sintaxis:

- kind = "valores"
- Valores:
 - **subtitles:**
 - Una transcripción o traducción del audio, pensado para aquellos casos en los que el audio se escucha adecuadamente pero no es comprendido, como cuando el usuario lo comprende el lenguaje hablado en el recurso. Los navegadores deberían mostrar esta información como texto superpuesto en el video. Este es el valor por defecto.
 - **captions:**
 - Información relevante de audio para ayudar a los usuarios que no pueden escuchar apropiadamente el recurso, como cuando el usuario es sordo, cuando el recurso se encuentra silenciado o cuando el ruido del ambiente es elevado. Los navegadores deberían mostrar esta información como texto superpuesto en el video.
 - **descriptions:**
 - Una descripción que explica al video (o a una parte de él), pensada para ayudar a los usuarios que no pueden verlo apropiadamente, como cuando el usuario es ciego, cuando está conduciendo o cuando el brillo de su pantalla tiene baja intensidad. Los navegadores deberían usar sintetizadores de voz para reproducir el contenido textual.
 - **chapters:**
 - Títulos de los capítulos, pensados para permitir a los usuarios navegar el recurso de medios. Los navegadores pueden proveerlos como una lista interactiva.
 - **metadata:**
 - Pensado únicamente para su uso con programas, no son mostrados por el navegador.
- **src.**
 - Permite especificar el archivo de subtítulos y su ruta de acceso.
 - Las rutas pueden ser absolutas o relativas.
 - Sintaxis:
 - Local:

```
<video src = "ruta de acceso/nombre archivo.extensión">
  <track kind= "valor" src = "ruta de acceso/nombre
    archivo.extensión">
</video>
```
 - Externa:

```
<video src = "http://nombre dominio/ ruta de
acceso/nombre archivo.extensión">
  <track kind="valor" label= "Texto" src = "ruta de
    acceso/nombre archivo.extensión" srclang
    ="Código de idioma">
</video>
```
 - Ejemplos:
 - Locales:
 - <video src = "playa.mp4">

```
<track kind="subtitles" label= "Subtítulos en
español" src = "sonidos mar.srt" srclang ="es">
</video>
```

- ```
<video src = "sonidos/pajaro.avi">
<track kind="caption" label= "Subtítulos en
español" src = "pajaritos.srt" srclang ="es">
</video>
```

- Externo:

- ```
<video src = "http://musica.com/rock/we will rock
you.mkv">
<track kind="subtitles" label= "Subtítulos en
español" src = "Queen.srt" srclang ="es">
</video>
```

- **srclang.**

- Permite especificar el idioma de los subtítulos.
- Atributo obligatorio.
- Sintaxis:
 - srclang = "código de idioma"
- Ejemplo:

```
<video src = "Titanic.mp4">
<track kind="subtitles" label= "Subtítulos en español" src = "subtitulos
de titanic.srt" srclang ="es">
</video>
```

- **label.**

- Permite incluir un texto informativo en el video.
- Sintaxis:
 - label ="texto"

Marcos con iFrame (Inline Frame)

- Para insertar o incrustar un objeto dentro de un documento web.
- Los contenidos incrustados pueden ser:
 - Videos, pdf's, otras páginas web, imágenes, cualquier archivo en general.
- Etiqueta:
 - **<iFrame>**Texto si no se reconoce la etiqueta**</iFrame>**
- Atributos:
 - **src:**
 - Permite especificar el objeto o documento a insertar y su ruta de acceso.
 - Rutas pueden ser absolutas o relativas.
 - Sintaxis:
 - Local:
 - ```
<iFrame src = "ruta de acceso/nombre archivo.extensión">
```
        - ```
</ iFrame >
```
 - Externa:
 - ```
< iFrame src = "http://nombre dominio/ ruta de acceso/nombre
archivo.extensión"></ iFrame >
```
  - Ejemplos:

- Locales:
    - < iFrame src ="playa.html"> </iFrame >
    - < iFrame src ="sonidos/pajaro.jpg"> </ iFrame >
    - < iFrame src ="../we will rock you.pdf"> </ iFrame >
  - Externa:
    - < iFrame src ="http://musica.com/rock/we will rock you.mkv"> </ iFrame >
- **width:**
  - Controla la anchura del objeto o documento en pixeles.
  - Sintaxis:
    - width = "número px"
- **height:**
  - Controla la altura de del objeto o documento en pixeles.
  - Sintaxis:
    - height = "número px"
  - Si no se especifican width y height, del objeto o documento se muestra con sus dimensiones originales.
- **scrolling:**
  - Para incluir barras de desplazamiento:
  - Sintaxis:
    - scrolling = "valores"
  - Valores:
    - **Auto:**
      - Aparecen si son necesarias.
      - Valor por defecto si éste no se incluye.
    - **No:**
      - No aparecen nunca.
    - **Yes:**
      - Aparecen siempre.
- **frameborder:**
  - Permite mostrar el contenido con o sin marco.
  - Sintaxis:
    - frameborder = "valor"
  - Valores:
    - **1:**
      - Aparece un marco.
      - Valor por defecto.
    - **0:**
      - No aparece el marco.
- **allowfullscreen:**
  - Permite activar o desactivar la opción de visualización en pantalla completa.
  - Sintaxis:
    - allowfullscreen
      - Si se especifica el atributo, se permite la visualización en pantalla completa.
      - Si no se incluye se desactiva esta opción.

- **allow.**
  - Permite activar una directiva de políticas que indican que puede o no hacer iFrame.
  - Sintaxis:
    - allow = "valor 1, valor 2, ..., valor N".
    - **Valores:**
      - **accelerometer.**
        - Controla el acceso a los sensores del acelerómetro.
      - **ambient-light-sensor.**
        - Controla el acceso a los sensores de luz ambiental.
      - **autoplay:**
        - Controla el acceso a la reproducción automática a través de play() y autoplay.
      - **camera.**
        - Controla el acceso a los dispositivos de entrada de video.
      - **encrypted-media.**
        - Controla si requestMediaKeySystemAccess() está permitida.
      - **fullscreen.**
        - Controla si requestFullscreen() está permitida.
      - **geolocation.**
        - Controla el acceso a la interfaz de geolocalización.
      - **gyroscope.**
        - Controla el acceso a los sensores del giroscopio.
      - **magnetometer.**
        - Controla el acceso a los sensores del magnetómetro.
      - **microphone.**
        - Controla el acceso a los dispositivos de entrada de audio.
      - **midi.**
        - Controla el acceso al método requestMIDIAccess().
      - **payment.**
        - Controla el acceso a la interfaz PaymentRequest.
      - **picture-in-picture.**
        - Controla el acceso a *Picture in Picture*.
      - **speaker.**
        - Controla el acceso a los dispositivos de salida de audio.
      - **sync-xhr.**
        - Controla si transferencias XMLHttpRequest sincrónicas están permitidas.
      - **usb.**
        - Controla el acceso a dispositivos USB.
      - **vr.**
        - Controla el acceso a dispositivos de realidad virtual.
  - Ejemplo:
    - allow = "accelerometer, picture in picture, autoplay, geolocation, gyroscope"

## FORMULARIOS.

- Elemento en una página web que permite:
  - Recopilar información o datos del usuario.
  - Registrarse o identificarse en un sitio.
  - Recibir comentarios, sugerencias u opiniones de los usuarios.
- **<form>...</form>**



- Sirve para crear un formulario.
- Engloba todos los elementos, campos o controles de un formulario, ya que es un contenedor de otros elementos.
- No muestra nada en pantalla si está vacía.
- No se muestra el formulario si no se cierra la etiqueta (</form>).
- Atributos.
  - **action.**
    - Obligatorio.
    - Indica ubicación el programa para procesar la información enviada.
    - Especifica el destino del formulario.
    - Destinos:
      - Programa de procesamiento alojado en un servidor:
        - action = "[http://nombre\\_del\\_servidor/ruta\\_de\\_acceso/nombre\\_del\\_archivo\\_de\\_procesamiento.extensión](http://nombre_del_servidor/ruta_de_acceso/nombre_del_archivo_de_procesamiento.extensión)"
        - Ejemplo:
          - action = "<http://www.misitio.com/programas/procesamiento.php>"
      - Dirección de correo electrónico:
        - Para recuperar simplemente los datos.
          - action="[mailto:direccion\\_de\\_correo\\_electrónico](mailto:direccion_de_correo_electrónico)"
        - Ejemplo:
          - action = "<mailto:fulanito@gmail.com>"
      - Procesamiento de datos de forma interna.
        - Desde el lado cliente con programas como JavaScript.
        - action = " " (atributo action vacío)
        - No se usan los atributos enctype ni method.
  - **enctype.**
    - Especifica tipo de codificación.
    - Especifica en que formato informático (tipo MIME) se transmitirán los datos.
    - Sintaxis:
      - enctype="valores"
    - Valores:
      - application/x-www-form-urlencoded
        - Valor por defecto.
      - multipart/form-data
        - Para enviar archivos.
      - text/plain
        - Envío del formulario a una dirección de correo electrónico.
  - **method.**
    - Especifica la forma de envío del formulario.
    - Sintaxis:
      - method = "valores"
    - Valores o métodos:
      - **get.**
        - Datos son visibles en la URL.
        - Envía los datos usando la URL, es decir, se pasan variables (nombre+valor) concatenados con el operador &, la URL y el archivo de procesamiento unidos por una interrogación (?).
        - Ejemplo:
          - <http://www.misitio.com/programas/procesamiento.php?nombre=Pepe&password=1234.....>
        - Datos enviados en caracteres ASCII.

- Datos no pueden exceder de 100 caracteres (capacidad limitada).
- Seguridad comprometida porque se ven los datos en la URL.
- **post.**
  - Datos no son visibles en la URL.
  - Los datos se envían con el cuerpo del formulario.
  - Útil para el envío de datos que sean sensibles.
  - Datos no se codifican en ASCII.
  - Capacidad de caracteres ilimitada.
  - Permite enviar archivos al servidor.
  - Uso típico en un formulario de registro.
- **name**
  - Permite asignar un nombre al formulario.
  - En desuso para dar nombre a un formulario, en su lugar, mejor usar el atributo *id*.
  - Nombre útil para distinguir varios formularios o para referenciarlos con otros programas como JavaScript.
  - Sintaxis:
    - name = "nombre"
- **accept-charset.**
  - Especifica tipo de codificación de caracteres.
  - Se usa poco.
  - Sintaxis:
    - accept-charset = "tipo de codificación"
    - tipo de codificación: utf-8, ISO-8859, ...
- **novalidate.**
  - valor booleano que indica al navegador no validar los datos del formulario durante el envío.
  - Sintaxis:
    - novalidate = "novalidate"
    - novalidate
- **target.**
  - Especifica donde deben cargarse los resultados del procesamiento de datos.
  - Sintaxis
    - target = "valores"
  - Valores:
    - \_blank
    - \_parent
    - \_self
    - \_top
- Ejemplo de <form> con atributos.
  - <form action = " " method="post" target = "\_blank" name="formulario\_prueba">
  - <form action = "<http://www.misitio.com/programas/procesamiento.php>" enctype = "application/x-www-form-urlencoded" method="post" target = "\_blank" name = "datos\_clientes">
  - <form action = "mailto:juan@yahoo.es" enctype= "text/plain" method="get" name="opiniones">

## CAMPOS O CONTROLES DE UN FORMULARIO.

- Para incluir los elementos de un formulario se usa la etiqueta <input> dentro de <form>.

```

<form>
 <input.....>
 <input.....>
 <input.....>
</form>

```
- **<input> o <input/>**
  - Permite definir los controles de entrada de datos de un formulario como:

- Campos o cuadros de texto.
- Casillas de verificación.
- Casillas o botones de opción. (RadioButton)
- Archivos.
- Imágenes.
- Cuadros de texto enmascarado para password.
- Cuadro para texto oculto.
- Botones de acción o comando (Enviar y Reestablecer).
- Atributos:
  - **name, id, style, class.**
  - **type.**
    - Define el tipo de campo o control.
    - Tipos:
      - text, password, checkbox, radio, submit, reset, file, hidden, image, button.
  - **disabled.**
    - Deshabilita un campo para que no pueda ser utilizado.
    - Sintaxis:
      - disabled
      - disabled = "disabled"
  - Otros:
    - **size, readonly, value, ....**
- Sintaxis genérica:
  - `<input type = "tipo" id = "identificador" size = "tamaño" name = "nombre" value = "valor">`

## TIPOS DE CAMPOS O CONTROLES BASADOS EN LA ETIQUETA <INPUT>.

- **Campos de texto.**
  - Permiten insertar un texto alfanumérico o numérico en formularios.
  - Sintaxis:
    - type = "text"
  - Atributos.
    - **name.**
    - **form.**
      - Indica a qué formulario pertenece un campo, si hay varios campos tipo text con el mismo nombre.
      - Sintaxis:
        - form = "nombre del formulario incluido en el atributo id de la etiqueta <form>"
    - **size.**
      - Especifica la anchura del campo de texto.
      - Define el tamaño en caracteres del campo, no el total de caracteres a introducir.
      - Tamaño especificado con un numero entero.
      - Valor por defecto es de 20 caracteres.
      - Sintaxis:
        - size = "número"
    - **maxlength.**
      - Máximo número de caracteres a introducir en el campo.
      - Valor es un número entero.
      - Sintaxis:
        - maxlength = "numero"
    - **min o minlength según versión.**

- Mínimo número de caracteres a introducir en el campo.
- Valor es un número entero.
- Sintaxis:
  - minlength = "numero"
- **value.**
  - Indica el valor campo y se utiliza para informar sobre qué es lo que se pide.
  - Sintaxis:
    - value = "texto"
- **placeholder.**
  - Permite incluir una sugerencia acerca del valor a introducir.
  - Aparece atenuado (gris).
  - Desaparece al situar el cursor o el foco en el campo.
  - Sintaxis:
    - placeholder = "texto"
  - Se pueden incluir también dibujos y, en general, cualquier carácter Unicode.
  - Sintaxis:
    - placeholder = "&#x valor Unicode"
  - Ejemplo:
    - placeholder = "&#x2315"
  - Sitios con tablas de caracteres Unicode:
    - <https://unicode-table.com/es/#basic-latin>
    - <https://www.mclibre.org/consultar/htmlcss/html/html-unicode-simbolos.html>
- **readonly.**
  - Campo de sólo lectura.
  - No se puede editar o modificar el valor que por defecto tenga el campo.
  - Sintaxis:
    - readonly = "readonly"
    - readonly
- **autofocus.**
  - Sitúa el foco en un campo al cargar la página.
  - Sintaxis:
    - autofocus = "autofocus"
    - autofocus
- **required.**
  - Convierte un campo en obligatorio.
  - Sintaxis:
    - required = "required"
    - required
- **pattern.**
  - Permite crear un patrón que especifica que contenido puede incluirse en el campo.
  - Expresión de validación de datos en los campos.
  - Expresiones:
    - **[ ]**:
      - Especifica que valores son permitidos.
      - Ejemplos:
        - [A-Z] Permitidas todas las letras mayúsculas.
        - [A-L] Permitidas todas las letras mayúsculas entre la A y la L.
        - [0-9] Sólo números.
        - [A-Za-z] todas las letras mayúsculas y minúsculas.

- **{}**:
    - Especifica el total de caracteres o dígitos a incluir.
    - Sintaxis:
      - {número}
      - {número inicial, número final}
    - Ejemplo:
      - [0-9] {9} 823542352 --- correcto ya que el número total son 9 dígitos.
      - [0-9] {3,9} 98375 --- correcto ya que el número mínimo son 3 dígitos.
      - [a-z] {5,20} --- “ana” no es válido y “antonio” **si, por tener 7.**
    - Ejemplo con pattern:
      - <input type = “text” pattern = “[A-Za-z] {5,20}”>
      - <input type = “text” pattern = “[0-9] {10,20}”>
  - **list.**
    - Permite asociar un control a un <datalist>
    - Sintaxis:
      - list = “nombre del datalist aplicado con el atributo id”
  - **autocomplete.**
    - Permite completar datos si previamente ya se han escrito en otra ocasión.
    - Sintaxis:
      - autocomplete = “on”
      - autocomplete
- **Campos de contraseña:**
  - Campo de texto donde los caracteres escritos se sustituyen en la visualización por otros para que no sean legibles.
  - Se usan puntos o asteriscos.
  - Protege la visualización no los datos, que se transmiten sin cifrar.
  - Sintaxis:
    - type = “password”
  - Atributos.
    - **name, size, maxlength, minlength, id, disabled, autofocus, pattern, placeholder, readonly y form.**
- **Campo de texto oculto:**
  - Almacenan datos no visibles para los usuarios.
  - A veces es necesario enviar información anexa a los formularios que no necesita ser introducida por el usuario.
  - Uso por programadores, sobre todo con JavaScript.
  - Usos:
    - Tiempo que tarda el usuario en rellenar y enviar el formulario.
    - Almacenar datos de formularios anteriores.
    - Almacenar datos que proviene de otro script.
    - Almacenar datos en bruto.
  - Sintaxis:
    - type = “hidden”
  - Atributos:
    - **id, name, form, autofocus, disabled.**
    - **value.**
      - atributo debe ir relleno ya que no lo introduce el usuario.
- **Ficheros.**
  - Obligatorio en el atributo enctype el valor = “multipart/form-data”

- Permite adjuntar y transferir un archivo al servidor.
- Sintaxis:
  - type="file"
- Atributos:
  - **id, value, name, autofocus, disabled, required, form.**
  - **multiple.**
    - Subir varios archivos.
    - Uso de corchetes en el valor del atributo name.
    - Ejemplo:
      - <input type = "file" name = "archivos[]" multiple>
  - **accept.**
    - Permite especificar restricciones para los archivos que se pueden subir.
    - Sintaxis:
      - accept = "archivo o archivos permitidos"
      - Más de un archivo se separa con comas.
      - Se usan tipos MIME para especificar los archivos permitidos.
      - También se pueden incluir extensiones de archivo.
      - Se pueden usar comodines para especificar los archivos.
    - Ejemplos:
      - <input type = "file" accept = " image/jpg, image/png, video/\*, audio/\*, .pdf, .docx">

#### Casillas de verificación.

- Permiten seleccionar varias opciones.
- Tienen 2 valores: seleccionado / no seleccionado.
- La etiqueta que acompaña a la casilla se añade a continuación.
  - <input type = "checkbox" .....>Texto asociado a la casilla.
  - <input type = "checkbox" .....>Inglés
- Sintaxis:
  - type = checkbox
- Atributos:
  - **name** (obligatorio), **autofocus, id, required, form, disabled.**
  - **value.**
    - Indica al formulario que valor enviar si la casilla esta activada.
    - Para el procesado final hay que indicar un valor a cada una de la casilla.
  - **checked.**
    - Permite preseleccionar una casilla.
- Ejemplos:
  - <input type = "checkbox" name = "i" value = "Idioma inglés" checked>Inglés
  - <input type = "checkbox" name = "f" value = "Idioma francés">Francés
  - <input type = "checkbox" name = "a" value = "Idioma alemán">Alemán

#### Botones de opción

- Representan opciones excluyentes.
- Para que sean excluyentes hay que agruparlas.
- Sintaxis:
  - type = "radio"
- Atributos:
  - **id, autofocus, checked, disables, required, form**
  - **value.**
    - Indica al formulario que valor enviar si la casilla esta activada.
    - Para el procesado final hay que indicar un valor a cada una de la casilla.

- **name.**
  - Es obligatorio.
  - Para agrupar las casillas el atributo name debe tener en todos los botones de opción el mismo valor.
- Ejemplo:
  - `<input type = "radio" name = "estado" value = "Soltero" checked> Soltero`
  - `<input type = "radio" name = "estado" value = "Casado"> Casado`
  - `<input type = "radio" name = "estado" value = "Separado"> Separado`

## Botones.

- Según el tipo, se dibuja un botón diferente.
- Los botones típicos son los de envío y anulación.
- Además de los de envío o anulación, se pueden crear otros tipos de botones de acción `<Button>`
- Se puede incluir una imagen como botón.
- Tipos:
  - **Botón de envío:**
    - Sirve para enviar el formulario al destino especificado en el atributo "action".
    - Si no se especifica el texto del botón, lo escribirá el navegador.
    - Sintaxis:
      - `type="submit"`
    - Atributos:
      - **id, autofocus, form, name, disabled.**
      - **value.**
        - Especifica el texto o etiqueta del botón.
  - **Botón de envío con imagen.**
    - Permite utilizar una imagen como botón de envío en un formulario.
    - Las coordenadas x e y donde se ha hecho clic, también se envían con el formulario, recibiendo el mismo nombre que el control con .x e .y agregadas.
    - Atención a la accesibilidad.
    - Sintaxis:
      - `type: "image"`.
    - Atributos:
      - **name.**
      - **src**
        - Obligatorio.
        - Sirve para especificar el archivo de imagen.
        - Sintaxis:
          - Local:
            - `<input type="image" src = "ruta de acceso/nombre archivo.extensión">`
          - Externa:
            - `< input type="image" src = http://nombre dominio/ruta de acceso/nombre archivo.extensión>`
    - Ejemplos:
      - Locales:
        - `< input type="image" src = "playa.jpg">`
        - `< input type="image" src = "imagenes/casa.png">`
        - `< input type="image" src = "../receta.webp">`

- Externo:
    - `<input type="image" src = "http://flickr.com/angel/hotel.jpg">`
- **width**
  - Permite especificar la anchura de la imagen en píxeles.
  - Ejemplo:
    - `<input type="image" src = "coche.jpg" width = "200px" >`
- **height**
  - Permite especificar la altura de la imagen en píxeles.
  - Ejemplo:
    - `<input type="image" src = "coche.jpg" height = "150px" >`
  - Ejemplo combinando width y height:
    - `<input type="image" src = "playa.jpg" width = "200px" height = "150px">`
- **alt:**
  - Sirve para incluir un texto alternativo cuya función es informativa si no se carga la página.
  - Importante de cara a la accesibilidad si no se puede ver la imagen por discapacidad.
  - Importante también para el posicionamiento.
  - Sintaxis:
    - `alt = "texto descriptivo"`
  - Ejemplo:
    - `<input type="image" src = "coche.jpg" width = "200px" alt = "Coche de la marca Seat participando en el Paris-Dakar" >`
- **Botón de anulación o reset.**
  - Sirve para restaurar todos los campos del formulario a sus valores iniciales (vacíos o por defecto).
  - Si no se especifica el texto del botón, lo escribirá el navegador.
  - Sintaxis:
    - `type="reset"`
  - Atributos:
    - **id, autofocus, form, name, disabled.**
    - **value.**
      - Especifica el texto o etiqueta del botón.
- **Botón de acción o button.**
  - Representa un botón sin una acción predefinida asociada, lo que significa que, por defecto, el botón no producirá ningún efecto cuando se pulse.
  - Normalmente son utilizados con programas del lado cliente como JavaScript, que son ejecutados cuando el usuario los presiona.
  - Sintaxis:
    - `type="button"`
  - Atributos:



- **id, autofocus, form, name, disabled.**
- **value.**
  - Especifica el texto o etiqueta del botón.

## Etiquetas para los Campos.

### <label>...</label>

- Para incluir textos asociados a los campos que permitan identificar los distintos controles se usan etiquetas.
- Muy útiles en accesibilidad.
- Atributos:
  - **for.**
    - Identificador para la etiqueta.
    - Identificador que coincide con el valor del atributo id del control con el que está asociado.
    - Sintaxis:
      - for = "texto"
- **Formas de asociar una etiqueta a un control:**
  - **Opción A:**
    - Insertando el texto de la etiqueta y el control dentro de <label>.
    - Sintaxis:
      - <label> nombre de la etiqueta <input .....></label>
  - **Opción B:**
    - Haciendo que coincidan los valores del atributo "id" del control con el atributo "for" de la etiqueta.
    - Sintaxis:
      - <label for ="texto">Texto de la etiqueta</label>
      - <input type..... Id="texto">

## Área de Texto.

### <textarea>...</textarea>

- Permite crear cuadros de texto multilínea para que el usuario pueda editar múltiples líneas de texto.
- Saltos de línea pulsando la tecla de Intro.
- Utilidad:
  - Editar mensajes, comentarios, sugerencias, contenido de archivos, listas, reseñas, artículos, etc.
- Atributos:
  - **name, placeholder, readonly, maxlength, minlength, required, autofocus, disabled, form.**
  - **cols.**
    - Especifica el ancho del área en número de caracteres.
    - Sintaxis:
      - cols="número"
      - Valor por defecto son 20 caracteres.
  - **rows.**
    - Número de líneas visibles para el área de texto.
    - Si se escriben más de las especificadas, aparecerá una barra de desplazamiento.
      - Sintaxis:
        - rows="número"
      - Valor por defecto son 2 líneas.
  - **wrap.**
    - Especifica cómo se van a gestionar los saltos automáticos de línea cuando se envíe el formulario.

- Sintaxis:
  - wrap = "valores"
- Valores:
  - **hard:**
    - Se inserta un carácter de salto de línea junto con el texto.
    - Si se utiliza, es obligatorio usar el atributo cols.
  - **soft:**
    - No se inserta ningún carácter de salto de línea.
- Un área de texto puede redimensionarse con la propiedad de CSS resize:
  - **resize.**
    - Permite cambiar el tamaño de un elemento.
    - Sintaxis:
      - resize: valores.
    - Valores:
      - **none:**
        - Impide cambiar las dimensiones de un elemento.
      - **both:**
        - Permite redimensionar un elemento en horizontal y vertical.
        - Valor por defecto.
      - **vertical:**
        - Permite redimensionar un elemento en vertical.
      - **horizontal:**
        - Permite redimensionar un elemento en horizontal.

### Organizar campos de un formulario.

#### <fieldset>...</fieldset>

- Permite diferenciar un formulario de otro y/o agrupar de campos de formularios.
- Un conjunto de campos puede agruparse para organizar el formulario de forma lógica, sobre todo si son largos y complejos.
- El agrupamiento puede tener o no un nombre.
- Los campos agrupados aparecen rodeados por un borde.
- Sintaxis:
  - <fieldset>...</fieldset>
  - Dentro debe incluirse el formulario a agrupar.
- Ejemplo:

```

<fieldset>
<form>
 <fieldset>

 <input....>
 </fieldset>
 <fieldset>
 <input....>
 </fieldset>

</form>
</fieldset>

```

- Atributos:
  - **name, disabled, form.**

### **<legend>...</legend>**

- Etiqueta usada para especificar un nombre para el agrupamiento.
- Por defecto aparece a la izquierda, pero su alineación se puede cambiar con CSS.
- La etiqueta se sitúa después de <fieldset>
- Sintaxis:
  - <legend>Nombre</legend>
- Ejemplo:

```

<fieldset>
 <legend>Formulario 1</legend>
</fieldset>
<form>
 <fieldset>
 <legend> Nombre agrupamiento 1</legend>
 <input....>
 </fieldset>
 <fieldset>
 <legend>Nombre agrupamiento 2</legend>
 <input....>
 </fieldset>
</form>
</fieldset>
</fieldset>

```

### **Listas desplegables.**

- Lista que proporciona al usuario distintas opciones para elegir.
- **<select>...</select>**
  - Etiqueta para crear una lista.
- **<option>...</option>**
  - Permite especificar los elementos de la lista.
  - Para cada elemento una etiqueta <option>.
- Sintaxis:
 

```

<select>
 <option>opción 1</option>
 <option>opción 2</option>
 <option>opción N</option>
</select>

```
- **<optgroup>...</optgroup>**
  - Permite crear grupos de opciones.
  - Los nombres de los grupos no son seleccionables.

- Sintaxis:

```
<select>
<optgroup label = "nombre del grupo1">
 <option>opción 1</option>
 <option>opción 2</option>
</optgroup>
<optgroup label = "nombre del grupo2">
 <option>opción 1</option>
 <option label = "opción2"></option>
 <option>opción 3</option>
</optgroup>
</select>
```

- Atributos:

- De **<select>**

- **autofocus, disabled, required, form, name.**

- **size.**

- Especifica el número de elementos visibles en el menú.
      - Por defecto es igual a uno.
      - Sintaxis:
        - `<select size ="número">`

- **multiple.**

- Permite seleccionar varios elementos a la vez en una lista.
      - Por defecto solo se puede seleccionar uno.
      - Sintaxis:
        - `<select multiple>`
        - `<select multiple="multiple">`
      - Formas de selección.
        - Control + clic sobre cada opción a seleccionar.
        - Mayúsculas + clic sobre las opciones extremas.
      - Para enviar correctamente al servidor las varias opciones seleccionadas, es necesario usar el atributo *name* con un nombre más unos corchetes.
        - Ejemplo:
          - `<select name ="varios[]" multiple>`

- De **<option>**

- **disabled, value.**

- **selected:**

- Permite que un elemento de la lista aparezca preseleccionado.
      - Por defecto aparece le primero preseleccionado.
      - Sintaxis:
        - `<option selected> opción 5</option>`

- **label.**

- Sirve para indicar el texto de la opción.
      - Sintaxis:
        - `<option label = "Texto opción">`
      - Ejemplo:
        - `<option label = "opción1"> </option>`
        - También vale esta otra forma de asignar el texto a una opción:
          - `<option>opción 1 </option>`

- De **<optgroup>**

- **disabled, label.**

## Botones de acción.

### <button>...</button>

- Permiten ejecutar una acción al hacer clic en el botón.
- La acción la define el desarrollador usando un lenguaje de programación como JavaScript.
- Se pueden usar como botones de envío o anulación.
- Sintaxis:
  - <button>Texto del botón</button>
- Atributos:
  - Sólo para el tipo **submit**:
    - **formaction, formenctype, formtarget, formmethod, formnovalidate.**
  - Comunes a **submit, reset y button**:
    - **autofocus, disabled, name, value, form, type.**
    - **type**
      - Permite especificar el tipo de botón
      - Sintaxis:
        - type="valores"
      - Valores:
        - **submit, reset.**
        - **button.**
          - Sirve para crear un botón de acción.
          - Necesita un programa con el código de la acción que va a realizar.
    - Sólo para el tipo **button**.
      - **onClick.**
        - Evento de ratón que, al producirse, desencadena una acción.
        - Sintaxis:
          - onClick= "nombre de programa/código a ejecutar"
        - Ejemplos:
          - <button type = "button" onClick="abrir.js">
    - **Desactivar un botón tras su primer uso.**
      - Al usar el manejador de eventos *onClick* para llamar o ejecutar una o varias funciones puede este ser desactivado usando el atributo *disabled* como valor en dicho manejador.
      - El botón puede estar creado con las etiquetas <button> o <input>.
      - Sintaxis:
        - disabled= true.
        - this.disabled = true.
        - < ... onclick="función1 a ejecutar (), función2 a ejecutar (), ..., funciónN a ejecutar (), this.disabled = true">
      - Ejemplo:
        - <input type ="button" value ="Calcular" onclick="calcularTotal(), this.disabled = true">
        - <button onclick="calcularTotal(), this.disabled = true">Calcular</button>
    - **Añadir una imagen en la etiqueta de un botón.**
      - Hay que usar la etiqueta <img> anidada a la etiqueta <button>.
      - Sintaxis:

```

<button>


```

</button>

- Ejemplo:

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

```
<img src = "guitarra.png" alt = "Imagen de una guitarra" width = "20px" height
= "20px">
```

```
</button>
```

## NUEVOS ELEMENTOS PARA FORMULARIO EN HTML5.

### Colores.

- Campo para la entrada de un color.
- Sintaxis:
  - type = "color"
- Seleccionar el color:
  - Escribiéndolo en formato hexadecimal, rgb y hsl.
  - Seleccionándolo en una paleta de colores.
- Atributos.
  - autofocus, disabled, form, name, autocomplete.
  - value.
    - Permite definir un color inicial.
    - Color por defecto el negro.
    - Seleccionar otro color:
      - Escribiéndolo en formato hexadecimal.

### Cuadro de búsqueda.

- Representa un campo de texto de una línea para realizar búsquedas.
- Sintaxis:
  - type = "search"
- Atributos.
  - size, maxlength, min o minlength, autofocus, disabled, form, name, pattern, placeholder, readonly, required, value, autocomplete.
- Se puede incluir un icono dentro del cuadro con o sin un texto usando el atributo placeholder. Para ello hay que especificar su valor Unicode o hexadecimal.
  - Web para iconos:
    - <https://www.mclibre.org/consultar/htmlcss/html/html-unicode-simbolos.html>
- Valores hexadecimales para una lupa:
  - &#128270, &#x1f50d, &#xfe0e

### Fechas y horas.

#### Fechas.

- Para fechas se utiliza el valor date en el atributo "type".
- Sintaxis:
  - type = "date"
- Formato de fecha:
  - dd/mm/aaaa
  - Otros formatos deben crearse con lenguajes de programación.
- Selección de la fecha:
  - Escribiéndola.

- Seleccionándola de un calendario.
- Atributos:
  - **autofocus, disabled, form, name, readonly, value, required, autocomplete.**
  - **max, min y step.**
    - Sirven para restringir las fechas a introducir.
    - Pueden usarse juntos o por separado.
    - **min.**
      - Especifica una fecha mínima o inicial.
    - **max.**
      - Especifica una fecha máxima o final.
    - **step.**
      - Indica la distancia entre la fecha seleccionable y la siguiente.
  - Sintaxis:
    - min = "fecha inicial" max = "fecha final" step = "numero".
  - Ejemplo:
    - min = "2018-02-20" max = "2018-04-24" step = "7".

## Horas.

- Para las horas se usa el valor time.
- Sintaxis:
  - type="time"
- Formato de hora:
  - hh:mm:ss
- Atributos:
  - **autofocus, disabled, form, name, readonly, value, required, autocomplete.**
  - **max, min y step.**
    - Sirven para restringir las horas a introducir.
    - **min.**
      - Especifica una fecha mínima o inicial.
    - **max.**
      - Especifica una fecha máxima o final.
    - **step.**
      - Indica la distancia en segundos entre 2 valores permitidos.
    - Sintaxis:
      - min = "hora inicial" max = "hora final" step = "número de segundos"
    - Ejemplo:
      - min = "18:00" max = "22:00" step = "3600"

## Fecha y Hora.

- Permite introducir en un campo una fecha y una hora a la vez.
- Sintaxis:
  - type="datetime-local"
- Formato de fecha y hora:
  - dd/mm/aaaa
  - hh:mm
  - Otros formatos deben crearse con lenguajes de programación.
- Selección de la fecha y hora:
  - Escribiéndolas.
  - Seleccionándola de un calendario.
- Atributos:
  - **autofocus, disabled, form, name, readonly, value, required, autocomplete.**
  - **max, min y step.**

- Sirven para restringir las fechas y las horas a introducir.
- **min.**
  - Especifica una fecha/hora mínima o inicial.
- **max.**
  - Especifica una fecha/hora máxima o final.
- **step.**
  - Indica la distancia entre una fecha seleccionable y la siguiente o, si es para las horas, los segundos entre 2 valores permitidos.
  - Con un valor bajo en step los saltos afectan a la hora.
  - Con un valor muy alto en step los saltos afectan a la fecha.
- Sintaxis:
  - min="fecha inicialThora inicial" max=" fecha incialThora inicial " step="número de segundos"
- Ejemplo:
  - min="2015-02-20T10:00" max="2015-04-24T10:00" step="3600"

### Meses.

- Se puede introducir una fecha especificando el mes y el año.
- El mes se puede introducir con texto.
- Sintaxis:
  - type="month"
- Formato de fecha:
  - mm/aaaa
  - Otros formatos deben crearse con lenguajes de programación.
- Selección de la fecha:
  - Escribiéndola.
  - Seleccionándola de un calendario.
- Atributos:
  - autofocus, disabled, form, name, readonly, value, required, autocomplete.
  - max, min y step.
  - Sirven para restringir las fechas a introducir.
    - **min.**
      - Especifica una fecha mínima o inicial.
    - **max.**
      - Especifica una fecha máxima o final.
    - **step.**
      - Indica la distancia entre la fecha seleccionable y la siguiente de mes en mes.
  - Sintaxis:
    - min = "fecha inicial" max ="fecha final" step= "numero".
  - Ejemplo:
    - min ="2018-02" max ="2018-04" step ="7".

### Semanas.

- Representa un campo para la entrada de una semana.
- Selección de la semana:
  - Escribiéndola
  - Seleccionándola de un calendario.
- Sintaxis:
  - type = "week"
- Formato para atributos:
  - "aaaa-wnúmero"
    - w: week
    - número: Número entre 1 y 52 o 53.



- Atributos:
  - **list, autofocus, disabled, form, name, readonly, required, value, autocomplete.**
  - **max, min y step.**
    - Permiten acotar la semana.
    - **min.**
      - Especifica una fecha mínima o inicial.
    - **max.**
      - Especifica una fecha máxima o final.
    - **step.**
      - Indica la distancia entre la fecha seleccionable y la siguiente de semana en semana.
    - Ejemplo:
      - max = "2021-w30" min = "2021-w10" step = "3".

## Teléfonos.

- Campo para introducir un número de teléfono.
- No valida el numero introducido, para hacerlo hay que usar el atributo "pattern" o expresiones de validación con JavaScript.
- Sintaxis:
  - type="tel"
- Atributos:
  - **list, size, maxlength, min o minlength, autofocus, disabled, form, name, pattern, placeholder, readonly, required, value, autocomplete.**
- Ejemplo de validación de un campo para teléfonos con un patrón:
  - <input type="tel" pattern = "[0-9]{9}">

## Correo electrónico.

- Campo para introducir una dirección de correo electrónico.
- No valida la dirección introducida a no ser que incluya una arroba (@), o se use el atributo "pattern" o expresiones de validación con JavaScript.
- Sintaxis:
  - type="email".
- Atributos:
  - **list, size, maxlength, min o minlength, autofocus, disabled, form, name, pattern, placeholder, readonly, required, value, autocomplete.**
- Ejemplo de validación de un campo para correo electrónico con un patrón:
  - <input type="email" pattern = "[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}\$">

## Direcciones de Internet.

- Cuadro de texto que permite introducir una dirección web.
- Se realizado una validación automática del campo, por si la URL enviada no es correcta.
- La validación por defecto detecta la falta de protocolos o los espacios en blanco como fallos en la dirección.
- Sintaxis:
  - type = "url"
- Atributos:
  - **list, size, maxlength, min o minlength, autofocus, disabled, form, name, pattern, placeholder, readonly, required, value, autocomplete.**
- Ejemplo de validación de un campo para direcciones de Internet con un patrón:

- `<input type="url" pattern = "(ftp|http|https):/(w+:{0,1}w*@)?(S+)(:[0-9]+)?(\/|\/([\w#!:.?+=&%@!-\/]))?">`

## Números.

- Campo para la entrada de números.
- Aparecen botones de incremento y decremento para aumentar o reducir la cifra de unida en unidad.
- Permite incluir un intervalo entre un valor máximo y otro mínimo, así como establecer los saltos entre valores.
- Validación:
  - Uso del punto (.) para posiciones decimales.
  - Se puede usar el signo menos (-) para números negativos.
  - Se puede usar la letra E para notación científica.
- Sintaxis:
  - `type = "number"`
- Atributos:
  - **list, autofocus, disabled, form, name, readonly, required, value.**
  - **max, min y step.**
    - Sirven para especificar rangos o intervalos numéricos y el valor de los incrementos o decrementos.
    - **min.**
      - Especifica un valor mínimo o inicial.
    - **max.**
      - Especifica un valor máximo o final.
    - **step.**
      - Indica los saltos numéricos.
    - Sintaxis:
      - `min = "valor mínimo" max = "valor máximo" step = "saltos numéricos"`
    - Ejemplo:
      - `min = "5" max = "100" step = "5"`

## Lista de datos.

### `<datalist>...</datalist>`

- Añadida a un campo de texto muestra una lista de sugerencia para introducir en el campo.
- Las opciones se crean con la etiqueta `<option>`.
- Para enlazar el `<datalist>` a un control, el valor del atributo `"id"` del `<datalist>` debe coincidir con el valor del atributo `"list"` del control.
- Atributos:
  - Solo admite atributos globales como `id`, `name`, `class`, `style`, etc.
- Sintaxis:

```
<input type = "text" list = "nombre">
<datalist id = "nombre">
 <option>opción 1</option>
 <option>opción 2</option>
 <option>opción N</option>
</datalist>
```
- Ejemplo:

```
<input type = "text" list = "coches">
<datalist id = "coches">
```

```
<option>Ford Fiesta</option>
<option>Dacia Sandero</option>
<option>Seat Alhambra</option>
</datalist>
```

## Medidas.

### <meter>...</meter>

- Representa una medida dentro de un rango conocido.
- Es un control que se emplea sólo para visualizar datos, no permite interacción.
- Sintaxis:
  - <meter atributos>...</meter>
- Atributos:
  - **min.**
    - Mínimo valor de un rango.
    - Valor por defecto el cero.
    - Sintaxis:
      - min = "número"
  - **max.**
    - Máximo valor de un rango.
    - Valor por defecto el uno.
    - Sintaxis:
      - max = "número"
  - **low.**
    - Número que define el área de valores bajos.
    - Todos los valores que estén entre "low" y "min" se considerarán los valores bajos de la medida.
    - El valor de "low" debe estar entre "min" y "max" y ser menor que "high".
    - Sintaxis:
      - low = "número"
  - **high.**
    - Número que define el área de valores altos.
    - Todos los valores que estén entre "high" y "max" se considerarán los valores altos de la medida.
    - El valor de "high" debe estar entre "min" y "max" y ser mayor que "low".
    - Sintaxis:
      - high = "número"
  - **optimum.**
    - Número que se considera optimo dentro de la medida.
    - El valor de "optimum" debe estar entre "min" y "max".
    - Sintaxis:
      - optimum = "número"
  - **value.**
    - Número que indica el nivel actual de la medida.
    - El valor de "value" debe estar entre "min" y "max".
    - Sintaxis:
      - value = "número"
  - Ejemplos:
    - <label>Tráfico generado:<meter min="100" max="5000" value="2500"></meter></label>

## Cursor deslizante.

- Representa un deslizador que permite seleccionar un valor acotado entre un máximo y un mínimo.
- Validación:
  - Uso del punto (.) para posiciones decimales.
  - Se puede usar el signo menos (-) para números negativos.
  - Se puede usar la letra E para notación científica.
- Sintaxis:
  - type = "range"
- Atributos:
  - max, min, step, autofocus, disabled, form, list, name, value, autocomplete.
- Ejemplo:
  - <input type = "range" min = "0" max "10" step = "2">
- Escala de medida:
  - Si se quiere hacer el control deslizante con marcas que representen los valores: máximo, mínimo e intermedios, hay que crear un <datalist> con los valores a utilizar y asociarlo, con el atributo "list", al control de tipo range.
  - Ejemplo:

```
<p><input type="range" list="marcas" ></p>

<datalist id="marcas">
 <option value="1">
 <option value="2">
 <option value="3">
 <option value="4">
 <option value="5">
</datalist>
```
  - Para crear dichas marcas o etiquetas con texto en la escala de medida y poderlas usar en los navegadores que no soporten éstas, hay que utilizar propiedades CSS.
  - Para mostrar el deslizador en vertical hay que usar la propiedad {transform: rotate(grados)} de CSS.

## VALIDACIÓN DE FORMULARIOS EN HTML.

- Es necesario validar los datos introducidos por un usuario en un formulario para asegurarnos, por ejemplo, que tengan el formato correcto, estén incluidos dentro de un rango válido de valores o no contengan código malicioso.
- La validación se puede realizar en el lado cliente o en el lado servidor:
  - En el lado cliente se utiliza JavaScript, reglas CSS y atributos HTML.
  - En el lado servidor se utilizan lenguajes como PHP, Python, etc.
- **Datos a validar:**
  - El formato de campos como los nombres, apellidos, contraseñas, cuentas de correo electrónico, teléfonos, códigos postales, etc., es decir, datos personales.
  - Campos obligatorios.
  - Tipo de datos, es decir, si un campo debe sólo recibir cadenas de caracteres, números o ambas cosas como, por ejemplo, un DNI.
  - Asegurarse de que el dato introducido sea válido como, por ejemplo, una fecha, una ciudad, etc.
- **Validación lado cliente usando HTML.**
  - Para ello se utilizan los siguientes atributos en los campos de los formularios:
    - **required.**
      - Para hacer que los campos sean obligatorios.
      - Ejemplo:
        - <input type="date" required> / Campo de tipo fecha obligatorio.
    - **minlength y maxlength.**

- Restringir la longitud de los datos de tipo alfanumérico.
- Ejemplo:
  - `<input type="text" minlength="10" maxlength="100" required>` / Campo de texto obligatorio en el que hay que escribir un mínimo de 10 caracteres y un máximo de 100.
- **min y max.**
  - Restringir el rango de valores permitido en campos de tipo numérico o de fecha y hora.
  - Ejemplo:
    - `<input type="number" min="2" max="8">` / Los valores permitidos para este campo de tipo numérico están comprendidos entre el valor 2 y el valor 8, ambos inclusive.
    - `<input type="date" min="2022-06-21" max="2022-09-21">` / Las fechas permitidas son sólo del verano del año 2022.
- **step.**
  - Para campos numéricos establece el salto de números permitido. Por defecto, 1.
  - Para campos fecha establece el salto de días permitido. Por defecto, 1.
  - Para campos hora establece el salto en segundos permitido. Por defecto 1.
  - Ejemplo:
    - `<input type="number" step="5">` / Los valores permitidos para este campo de tipo numérico se muestran de 5 en 5, o sólo está permitido introducir números múltiplos de 5.
    - `<input type="date" step="7">` / Las fechas permitidas se muestran de 7 en 7 días, por lo que sólo se pueden seleccionar o escribir fechas de semana en semana.
    - `<input type="time" step="900">` / Las horas permitidas se muestran de 15 en 15 minutos, por lo que sólo se pueden escribir horas en ese rango de valores.
- **type.**
  - Restringir el tipo de datos a utilizar.
  - Ejemplo:
    - `<input type="text">` / El campo admite cadenas de caracteres alfanuméricos.
    - `<input type="email">` / El campo solo admite una cadena de caracteres con el formato de una cuenta de correo electrónico.
- **pattern.**
  - Permite especificar un patrón que los datos introducidos deben cumplir para ser válidos.
  - Ejemplo:
    - `<input type="text" pattern="[A-Z]{1}[A-Za-z]{0,19}" >` / En este campo de texto, se pueden introducir entre 1 y 20 letras, siendo la primera obligatoriamente mayúsculas y el resto, mayúsculas o minúsculas indistintamente.
    - `<input type="text" pattern="[0-9]{2}[A-Z]{2}" >` / Este campo sólo permite 4 caracteres. Los 2 primeros deben ser números y, los 2 últimos, letras mayúsculas.

## PSEUDOCASES DE VALIDACIÓN.

- Se puede hacer que un campo tenga un estilo, aspecto o formato cuando el dato introducido en éste sea válido y otro diferente, cuando no.
- Para ellos se usan las siguientes pseudoclases CSS.
  - **:valid.**
    - El campo presentará un estilo cuando el dato introducido sea válido.
    - Sintaxis:
      - Elemento de formulario: `valid{estilos CSS;}`
    - Ejemplo:
      - `input:valid{color:red;}` / El texto se verá de color rojo cuando se introduzca correctamente.

- **:invalid.**
  - El campo presentará un estilo cuando el dato introducido sea inválido, es decir, no cumpla con las reglas de validación establecidas.
  - Sintaxis:
    - Elemento de formulario:invalid{estilos CSS;}
  - Ejemplo:
    - `textarea:invalid{background-color:#ddd;}` / El color del fondo del área de texto se verá gris claro si el dato introducido es incorrecto.
- **:in-range.**
  - El campo presentará un estilo cuando el dato introducido esté comprendido dentro del rango de valores incluidos con los atributos *min* y *max*.
  - Sintaxis:
    - Elemento de formulario:in-range{estilos CSS;}
  - Ejemplo:
    - `input:in-range{color:white;background-color:black;}` / El texto se verá de color blanco y el fondo negro cuando se introduzca un valor numérico dentro del rango de valores establecido entre un mínimo y máximo.
- **:out-of-range.**
  - El campo presentará un estilo cuando el dato introducido sea incorrecto por estar fuera del rango de valores incluidos con los atributos *min* y *max*.
  - Sintaxis:
    - Elemento de formulario:out-of-range{estilos CSS;}
  - Ejemplo:
    - `input:out-of-range{color:blue;background-color:red;}` / El texto se verá de color azul y el fondo rojo cuando se introduzca un valor numérico fuera del rango de valores establecido entre un mínimo y máximo.
- **:required.**
  - El campo presentará un estilo cuando el dato introducido sea obligatorio por haber incluido en él atributo *required*.
  - Sintaxis:
    - Elemento de formulario:required{estilos CSS;}
  - Ejemplo:
    - `input:required{border:1px solid red;}` / Todos los campos obligatorios mostrarán un borde rojo.
- **:optional.**
  - El campo presentará un estilo cuando el dato introducido no sea obligatorio, es decir, no incluyen atributo *required*.
  - Sintaxis:
    - Elemento de formulario:optional{estilos CSS;}
  - Ejemplo:
    - `input:optional{border:1px solid blue;}` / Todos los campos opcionales mostrarán un borde azul.
- **label::after.**
  - El pseudo estilo `::after` (no es una pseudo clase de validación), permite añadir una imagen o un texto después de un elemento.
  - Utilizado con el elemento *label*, también se puede añadir texto, o una imagen del tamaño adecuado, a continuación de un campo, si se ha incluido una etiqueta tras él.
  - Útil para añadir un texto que indique si, en un campo, se han introducido datos correctos o no.
  - Para ello, a continuación de la pseudo clase de validación se incluye el pseudo estilo *after* separados ambos por un signo más (+).
  - Para añadir el texto o imagen se usa la propiedad *content*.

- Sintaxis:
  - Elemento de formulario:pseudo clase de validación + label::after{content="Texto para la etiqueta";}
- Ejemplo:
  - input:out-of-range + label::after{content: "Fecha fuera de rango.";}
  - input:in-range + label::after{content: "Números en el rango admitido.";} / Menos común el uso cuando el dato introducido es correcto.
- **label::before.**
  - El pseudo estilo ::before (no es una pseudo clase de validación), permite añadir una imagen o un texto antes de un elemento.
  - Utilizado con el elemento *label*, también se puede añadir texto, o una imagen del tamaño adecuado, delante de un campo, si se ha incluido una etiqueta tras él.
  - Útil para añadir un texto que indique si, en un campo, se han introducido datos correctos o no.
  - Para ello, a continuación de la pseudo clase de validación se incluye el pseudo estilo *after* separados ambos por un signo más (+).
  - Para añadir el texto o imagen se usa la propiedad *content*.
  - No es tan común utilizar esta pseudo estilo ya que, delante de los campos se suelen incluir las etiquetas que identifican a éstos.
  - Sintaxis:
    - Elemento de formulario:pseudo clase de validación + label::before{content="Texto para la etiqueta";}
  - Ejemplo:
    - input:out-of-range + label::before{content: "Fecha fuera de rango.";}
    - input:in-range + label::before{content: "Números en el rango admitido.";} / Menos común el uso cuando el dato introducido es correcto.

## FORMULARIOS Y ACCESIBILIDAD.

- Para garantizar la independencia de dispositivo en el ámbito de la accesibilidad se puede acceder a los campos de un formulario usando el teclado, en concreto, usando la tecla del tabulador y atajos de teclado.
- Posibilidad uso del teclado independientemente del ratón para acceder al sitio web.
- **Tabulador.**
  - Permite desplazarse por algunos elementos web como controles de formulario y enlaces.
  - Dentro de un formulario, cada toque a la tecla permite ir saltando de campo en campo en el orden de creación de estos y así, ir activándolos para poder utilizarlos.
  - El orden de activación se puede modificar para ir avanzando por los campos según nos interese.
    - **tabindex.**
      - Atributo global que permite establecer el orden de activación de un control al pulsar el tabulador.
      - Sintaxis:
        - tabindex = "valor"
        - Valor:
          - Numero entero que indica el orden de activación, así, un 1 indica que será el primero en activarse al pulsar el tabulador y un 8, que ese elemento se activará en octava posición.
      - Ejemplo:
        - <input type ="text " tabindex="1">
        - <input type ="number" tabindex="2">

- **Atajos de teclado.**

- Se puede acceder directamente a un elemento concreto de un formulario o un enlace usando atajos de teclado o teclas de acceso rápido.
- Existen combinaciones por defecto definidas por el navegador.
  - **accesskey.**
    - Atributo global que permite definir una tecla de acceso directo a dichos elementos combinándola con otras definidas por el navegador.
    - Sintaxis:
      - `accesskey = "tecla"`
      - Tecla:
        - Tecla de acceso que debe tener sentido en el idioma actual en el que se esté mostrando el contenido, a no ser que, por defecto se use para otra función.
        - Para usar la tecla definida o asignada hay que consultar con cuál hay que combinarla en cada navegador:
          - Firefox utiliza Alt + Mayus + tecla definida.
          - Chrome o Edge utilizan Alt + tecla definida.
    - Ejemplo:
      - `<input type = "text " acceskey = "ñ">` / Alt + ñ accederían directamente al elemento.