Introducción a HTML 5

Contenido

1. Introducción

1.1 ¿Qué es HTML?

Es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a ***HyperText Markup Language***, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

- Permite describir el contenido de una página, incluyendo texto y otros elementos (imágenes, videos, pequeñas aplicaciones, etc.).
- Una página HTML consta de texto y marcas especiales que permiten indicar algún tratamiento especial (estructura, formato, hiperenlace, etc.)
- Las marcas se indican en formato <marca>…</marca>

Mejores referencias:

https://www.w3schools.com/html/

https://developer.mozilla.org/es/docs/Web/HTML

1.2 ¿Qué es CSS?

CSS **(Cascading Style Sheets)** es un lenguaje que nos permite controlar el aspecto de las páginas web escritas en HTML o en cualquier lenguaje de marcado basado en XML.

1.3 Entorno de trabajo

Editor: Visual Studio Code de Microsoft. Otros: SublimeText.

Extensiones:

- auto Close Tag (cierra automáticamente las etiquetas)
- auto Rename Tag (corrige errores al escribir mal una etiqueta)
- Beauty (tabula los ficheros para dar formato)
- IntelliSense for CSS (mejora el trabajo con CSS)

Además, se puede usar GitHub como repositorio.

Además, usaremos las herramientas de desarrollo de los navegadores. Para HTML, sobre todo la pestaña "***Elements***".

Se recomienda descargar CHEATSHEETS, que son documentos con las instrucciones más usadas. Ejemplo de estas son

- https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf
- https://websitesetup.org/css3-cheat-sheet/

Con estas hojas podremos buscar las etiquetas básicas de HTML y CSS hasta que las vayamos conociendo.

2. Documentos HTML

2.1 Evolución de HTML hasta HTML5

HTML 1.0 (1991) … HTML5 (2014)

HTML: Lenguaje original

Basado en SGML (*Standard Generalized Markup Language*)

Es casi un lenguaje XML, pero tiene elementos no compatibles con XML

Empieza a cobrar fuerza la idea de no mezclar marcado de estructura (tabla, sección, encabezado, etc.) con marcado de presentación (fuente, color, alineamiento, etc.)

HTML 4.01: Última versión publicada de HTML antes de HTML5.

Dos modos:

• Strict: no permite marcado de presentación.

• Transitional: permite marcado de presentación y elementos obsoletos (center, font, align, bgcolor, …).

Framesets:

• Primera incorporación al estándar de la etiqueta <frameset> (aunque ya se usaba comúnmente en Netscape)

XHTML: versión basada en XML de HTML 4.01

Compatibilidad con documentos XML.

XHTML es más estricto que HTML, exige que los documentos estén bien formados (requisito de XML).

HTML5: Evolución de XHTML y HTML 4.01. Es el estándar vigente a día de hoy.

• XHTML 2.0 ha sido abandonado

• Un paso más:

- Construye sobre 'strict'; acaba con 'frameset' y 'transitional‘
- "Estándar viviente": W3C publica.
- Añaden cosas y recomiendan no usar otras, pero quitan poco o nada.

- HTML5 vs. XHTML5

  - "Un estándar, dos formas de escribir”
  - La recomendación vigente es codificar en HTML5 (la transición hacia navegadores optimizados para XML puro no ha funcionado).

Para ver qué partes de tu navegador acepta HTML5 visita https://html5test.com/

Una vez realizado un documento HTML, podremos validarlo para comprobar si es correcto en **https://validator.w3.org/**

2.2 Etiquetas, atributos y comentarios

Etiquetas: <etiqueta> …contenido… </etiqueta>. Hay etiquetas sin contenido: <contenido> o <contenido />. HTML permita que las etiquetas sin emparejar no lleven cierre (/).

Atributos: Proporcionan información adicional de una etiqueta. Siempre van en la apertura de la etiqueta. Se representan como *clave=valor*. Ejemplo: id=”p1”, src=”foto.jpg”, type=”text”, href=”styles.css”, …

Buenas prácticas: etiquetas y atributos en minúsculas, cerrar todos los elementos con contenido, poner el valor de los atributos entre comillas, y tabular, tabular, tabular.

Comentarios: Texto que el navegador no va a mostrar. Sirven para aclarar el código.

<!-- comentario -- >

2.3 Estructura de una página web

Árbol DOM: Estructura de una página Web que tiene forma de árbol genealógico con un único nodo raíz.

```
<!DOCTYPE html>
<html lang=”es”>
<head>
      <meta charset=”UTF-8”>
      <meta name=”viewport” content=”width=device-width, initial-scale=1.0”>
```

```
    <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
    <link rel="stylesheet" href="styles.css">
    <title>Mi Web</title>
</head>
<body>
</body>
</html>
```

2.4 Cabecera de una página web

No representa contenido alguno, contiene metadatos que describen características de mi página web. Puede contener etiquetas*: title, style, meta, link, script, base*.

Etiqueta *meta*: define los metadatos del documento HTML. Los metadatos son datos (o información) sobre los datos. Siempre se define dentro de la etiqueta <head> del HTML.

Los navegadores utilizan los metadatos (cómo mostrar contenido o recargar la página) en los motores de búsqueda (palabras clave) y otros servicios web.

Posibles valores:

<meta **charset**="UTF-8"> → Usa el juego de caracteres UTF-8 para poder visualizar "ñ" y tildes. Hacer un ejemplo con y sin esta etiqueta en varios navegadores.

<meta **name="description"** content="……"/> → Para describir el contenido de mi página web. Los usan los buscadores como Google.

<meta **name="keywords"** content=" k1, k2, k3, ../> → Palabras clave de mi página web.

<meta **name= "author"** content="Pepito" /> → Especifica quién es el autor de la web.

<meta **name="viewport"** content="width=device-width, initial-scale=1.0"> → Permite ajustar mi web a la anchura de la pantalla del dispositivo donde se vea. Es una primera aproximación para el diseño responsive.

<meta **http-equiv**="**refresh**" content="*n*" /> → Hace que la web se refresque cada *n* segundos.

<meta **http-equiv**="**expires**" content="Sat, 31 Dec 2020 23:59:00 GMT"> → Indica que la web expirará (deja de estar almacenada en la caché) en la fecha GMT dada. Si se pone 0, el navegador tendrá que recargarla cada vez que se visite.

<meta **http-equiv**="**Cache-Control**" content="no-store" /> → No guarda la web en la caché del navegador. Esto hará que siempre se recargue de nuevo desde el servidor.

3. Añadiendo contenido. Etiquetas básicas

3.1 Etiquetas relacionadas con texto

Vamos a empezar a añadir elementos a nuestra página web que, es lo mismo que decir, que vamos a empezar a añadir etiquetas dentro de la etiqueta <body>.

Hay muchas etiquetas para añadir texto y nosotros vamos a presentarlas siguiendo la siguiente clasificación:

- Encabezados
- Etiquetas de formato
- Otras etiquetas interesantes

Encabezados

Son etiquetas que nos van a permitir añadir "títulos" o encabezados a distintas secciones de nuestra página. Estas etiquetas tienen el siguiente formato:

```
<hx>
    ...
    Contenido o texto
    ...
</hx>
```

x deberá ser sustituido por un número del 1 al 6, desde 1 que es el mayor tamaño hasta 6 que es el más pequeño. El texto se mostrará en negrita.

Veamos un ejemplo simple:

```
<h1>Hola</h1>
<h2>Hola</h2>
<h3>Hola</h3>
<h4>Hola</h4>
<h5>Hola</h5>
<h6>Hola</h6>
```

Etiquetas de formato

Hay multitud, todas le dan cierto estilo al texto que contienen y destacaremos las siguientes:

- **...** para poner un texto en negrita. También ****. La diferencia entre ambas es semántica, mientras que da fuerza a un texto, solo resalta visualmente. *
- **...** para poner un texto en cursiva.
- **<u>...</u>** para poner un texto subrayado.
- **~~...~~** para mostrar un texto tachado.
- **...** para poner un texto con énfasis (similar a cursiva). La diferencia entre y es semántica. da énfasis y solo resalta, aunque el resultado sea el mismo visualmente. *
- **^{...}** para poner un texto como superíndice de otro texto.

- **_{...}** para poner un texto como subíndice de otro texto.
- **<mark>...</mark>** para poner un texto subrayado (nuevo en html5).
- **<q>...</q>** para mostrar una pequeña cita.
- **<cite>...</cite>** para mostrar el título de una referencia bibliográfica.
- **<time>...</time>** para mostrar horas y fechas. Se utiliza para traducir la hora a un formato legible por la máquina para que los navegadores puedan ofrecer agregar recordatorios de fecha a través del calendario del usuario, y los motores de búsqueda pueden producir resultados de búsqueda más inteligentes. Se puede usar con la propiedad ***datetime***. **Aquí**
- tienes más información.
- **<address>...</address>** para mostrar direcciones.
- **<blockquote>...</blockquote>** para poner citas largas especificando la fuente. Tiene varias propiedades que puedes ver **aquí**. Permite incluir un párrafo, un grupo de párrafos o un conjunto de muchos otros elementos incluyendo imágenes, tablas, entre otros. Esta es la principal diferencia entre este elemento y **<q>**, que está diseñado para citar únicamente líneas de texto. Visualmente, solo tabula el texto de la cita.

  Se suele usar con la propiedad ***cite=URL***, pero la URL especificada es usada por los motores de búsqueda para obtener más información sobre la cita.
- **<abbr title=…></abbr>** Permite poner abreviaturas de un término. Por ejemplo:

```
<abbr title="Organización mundial de la salud">OMS</abbr>
```

- **<code>** Permite mostrar el texto con formato de código en algún lenguaje de programación.
- **<pre> <pre>…</pre>** Texto pre formateado. Igual que pero se tienen en cuenta espacios en blanco y líneas en blanco. Esto es, se muestra el texto tal cual se escribe.
- **<samp> … </samp>** Define algún texto como muestra de salida de un programa de computadora en un documento.
- **<kbd>…</kbd>** Define el texto como entrada de teclado de un documento.

*** En teoría es mejor usar que <b> y que <i>, ya que se usa para accesibilidad y SEO, aunque es difícil de demostrar.**

Otras etiquetas interesantes

- **
** Salto de línea.
- ** ... ** Párrafo.
- **<hr/>** Separación de Tema. Dibuja distintos tipos de líneas.
- ** ** Espacio en blanco.

Etiqueta

Mecanismo más importante para agrupar diversos elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc.). No puede insertarse dentro de una etiqueta en-línea (strong, em, etc.) o de un bloque de texto. Si se puede insertar dentro de una tabla, de un bloque de cita <blockquote> o de otra división .

@font-face

Permite especificar fuentes online para visualizar en tus páginas web. Al permitir a los autores proporcionar sus propias fuentes, `@font-face` elimina la necesidad de depender del número limitado de fuentes de usuarios instaladas en las computadoras.

```
@font-face {
  font-family: <un-nombre-de-fuente-remota>;
  src: <origen> [,<origen>]*;
  [font-weight: <peso>];
  [font-style: <estilo>];
}
```

- **<un-nombre-de-fuente-remota>**

Especifica el nombre de una fuente que será utilizada como valor de ***font face*** por las propiedades de fuente.

- **<origen>**

Dirección URL para la ubicación remota del archivo de fuente ***url(…)***, o el nombre de una fuente **instalada** ya en la computadora del usuario en la forma ***local("Nombre de Fuente")***. Si esa fuente no es encontrada, se intentarán otros orígenes hasta encontrar una fuente.

Con ***url(…)*** también se pueden especificar rutas locales y además **la fuente no tiene por qué estar instalada.**

- **<peso>**

Un valor de peso/grosor de fuente.

- **<estilo>**

Un valor de estilo de fuente.

Ejemplo:

```
@font-face {
  font-family: MyHelvetica;
  src: local("Helvetica Neue Bold"), local("HelveticaNeue-Bold"),
url(MgOpenModernaBold.ttf);
  font-weight: bold;
}

/*Usando Google fonts:*/
@font-face {
  font-family: myGoogleFont;
  src: url(https://fonts.google.com/share?selection.family=Palette%20Mosaic)
format('woff2');
  font-weight: bold;
}

div {
  font-family: myGoogleFont;
```

```
}
```

Uno de los formatos de fuente que aceptan la mayoría de navegadores es “woff2”, para especificarlo puedes añadir `format('woff2')` cuando se especifica la url.

Especificar una fuente a usar a través de una url externa **no es muy seguro**. Además, se tarda más en cargar las fuentes.

3.2 Imágenes, enlaces y rutas

Tras añadir elementos de texto a nuestra página, al cuerpo de la misma, vamos a continuar mejorando y haciendo nuestras webs más bonitas e interactivas.

Para ello, en este apartado vamos a añadir dos elementos básicos en toda web:

- Las imágenes
- Los enlaces

Y, además, explicaremos qué es una ruta y las clases que hay ya que, estos dos elementos anteriores, van a utilizar rutas en sus atributos.

Imágenes simples

La inclusión de imágenes simples se viene haciendo desde las primeras versiones de HTML con la etiqueta sin contenido ****

Los atributos más comunes que le podemos poner a esta etiqueta son:

- **src:** que indica la ruta en la que se encuentra el archivo de la imagen.
- **alt:** un texto alternativo para describir la imagen en caso de que no se cargue o para dispositivos especiales para usuarios con discapacidad visual, por ejemplo.
- **width:** para especificar la anchura de la imagen (px, %, etc..). Si no se escogerá la anchura propia de la misma.
- **height:** para especificar la altura de la imagen (px, %, etc..). Si no se escogerá la altura propia de la misma.
- **title:** texto que se muestra en forma de “tip” (cuadrito amarillo que aparece cuando se sitúa el ratón encima de la imagen).

Es importante destacar que la imagen es un elemento en línea que se pone, si cabe, inmediatamente después de los elementos previamente añadidos.

Algunos ejemplos de utilización de estos atributos podrían ser:

```
<!-- Si no pongo nada se respetan las dimensiones originales de la imagen-->
<!-- Ruta relativa (suele ser la mejor opción)-->
<img alt="Logo de Empresa" src="img/empresa-logo.jpg">
```

```
    <!-- Ruta absoluta -->
    <img alt="Logo de Empresa" src="home/empresa/images/empresa-logo.jpg">

    <!-- Ruta URL -->
    <img alt="Logo de Empresa" src="https://www.empresa.com/wp-content/uploads/empresa-
logo.jpg">

    <!-- Si modifico una dimensión se respetan las proporciones-->
    <img height="50px" alt="Logo de empresa" src="img/empresa-logo.jpg">
    <img width="50px" alt="Logo de empresa" src="img/empresa-logo.jpg">

    <!-- Si pongo ambas dimensiones puedo modificar las proporciones-->
    <img height="300px" width="100px" alt="Logo de empresa" src="img/empresa-logo.jpg">
```

Rutas

El concepto de ruta es un concepto muy importante ya que se utiliza es muchos temas relacionados con la informática y en concreto, en la creación de páginas WEB, se utiliza para referenciar archivos, recursos y/o partes de alguna web. De manera general podemos distinguir:

- **Relativas:** Toman como base el directorio en el que se encuentra nuestro fichero. Son las recomendadas.

```
    <img alt="Logo de Empresa" src="img/logo.jpg">
```

- **Absolutas:** Toman como base el directorio raíz de mi equipo (/ en Linux y c:\ en Windows). Cuidado, sólo funcionarán en tu mismo equipo.

```
    <img alt="Logo de Empresa"
src="/home/pekechis/public_html/empresa_iniciacion_html_css/Parte_III_Aniadiendo_contenido_
Etiquetas_Basicas/examples/img/empresa-logo.jpg">
```

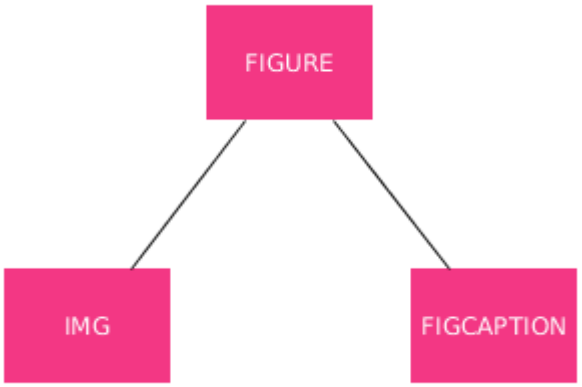
- **Url:** La dirección de Internet de un recurso (fichero, imagen etc..). Puede desaparecer y entonces dejará de mostrarse en nuestra web.

```
<img alt="Logo de Empresa" src="https://www.empresa.com/wp-content/uploads/empresa-
logo.jpg">
```

Figuras

Una novedad en HTML5 es la construcción de etiquetas alrededor de una imagen que nos va a permitir mostrar una imagen con un texto asociado.

En este caso el árbol DOM será el siguiente:



Donde:

- `<figure>` es la etiqueta padre.
- `<img ../>` es una etiqueta de imagen que hemos visto anteriormente.
- `<figcaption>` es una etiqueta que contendrá el texto asociado a la imagen.

Un ejemplo:

```
<figure>
        <img alt="Logo de empresa" src="img/empresa-logo.jpg">
        <figcaption>Logotipo de empresa con la etiqueta figure</figcaption>
</figure>
```

Gráficos vectoriales

Se puede usar para dibujar gráficos con varias formas y colores a través de secuencias de comandos, generalmente JavaScript. Los gráficos vectoriales son escalables, fáciles de crear y editar. También es compatible con la interactividad y la animación. Tener un tamaño de archivo más pequeño hace que la transferencia y carga de gráficos sea mucho más rápida en la web. Esa es la razón por la que muchas personas prefieren utilizar gráficos vectoriales.

Ejemplo:

```
<svg id = "svgelem" height = "200">
   <circle id = "redcircle" cx = "50" cy = "50" r = "50" fill = "red" />
</svg>
```

Más info **aquí**.

Existen muchos generadores de svg online. Te animo a que explores y realices algunos gráficos svg.

Enlaces

Los enlaces, que se representan mediante el uso de la etiqueta <a> es una de las construcciones más importantes en HTML. Esta etiqueta puede tener varios atributos, de los cuales los más importante son:

- **href:** que es la dirección de Internet de destino (ya sea otra página web, una imagen, un fichero o lo que sea). Si se empieza la cadena con ***"mailto:---"***, se entenderá que se desea abrir la url en un cliente de correo electrónico con esa dirección como destinatario.

- **target:** que indica dónde voy a abrir ese enlace. Si no pongo nada se abrirá en la misma pantalla y si le doy el valor **target="_blank"** se abrirá en una nueva ventana de mi navegador.

- **rel:** El atributo ***rel*** especifica la relación que hay entre el documento actual y el documento al cual se hace referencia en el enlace. **Aquí** tienes algunos de sus posibles valores. Para añadir más seguridad a nuestros enlaces, se han añadido un par de opciones al atributo *rel*. Estas opciones son: **noopener** y **noreferrer**. Cuando se añade la etiqueta HTML *rel = "noopener"* se evita que las nuevas pestañas aprovechen la función de JavaScript ***window.opener*** para que puedan hacer referencia a la ventana padre y acceder a sus componentes. Al mismo tiempo el atributo *rel = "noreferrer"* evita pasar la información del remitente a la nueva pestaña. Se suelen usar juntas:

  rel ="noopener noreferrer"

Varios ejemplos de enlaces:

```
<p><a href="http://www.empresa.net">Esto abre un enlace en la propia página</a></p>

 <p><a href="http://www.empresa.net" target="_blank">Esto abre un enlace en una pestaña
nueva</a></p>

<!-- Haciendo que una imagen sea enlace. Anidando etiquetas -->
<a href="http://www.empresa.net"><img width="100px" alt="Logo de empresa"
src="img/empresa-logo.jpg"></a>
```

Enlaces dentro de la misma página

Puedo enlazar enlaces dentro de la misma página con construcciones como la siguiente:

```
    ...
    <a href="#contacto">Contacto</a>
    ...
    <section id="contacto">
        .....
    </section>
    ....
```

3.3 Práctica: Imágenes y enlaces

Realizar una web que muestre tres imágenes centradas de HTML5, CSS3 y Chrome. Estas imágenes abrirán una nueva pestaña. Las imágenes estarán guardadas de forma local. Las dos primeras imágenes estarán en una misma línea (párrafo, por ejemplo) y la de Chrome en otra distinta.

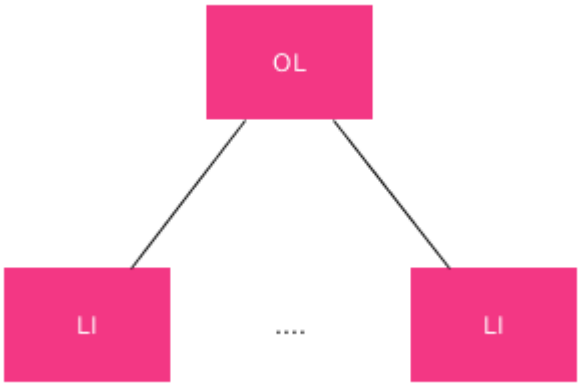
3.4 Listas

Las listas son una de las construcciones más usadas a la hora de elaborar textos, no sólo en HTML. No obstante, en HTML están también presentes y pueden ser de 3 tipos:

- **Listas Numeradas:** que son aquellas que expresan un orden entre los diferentes elementos de la lista. Este orden podrá ser numérico, alfabético etc..
- **Listas no numeradas:** que simplemente muestra los elementos de la lista uno tras otro.
- **Listas de definición:** que muestran diversos términos junto con su definición.

A continuación, vamos a ver el árbol DOM para cada una de estas variedades:

Lista Numeradas



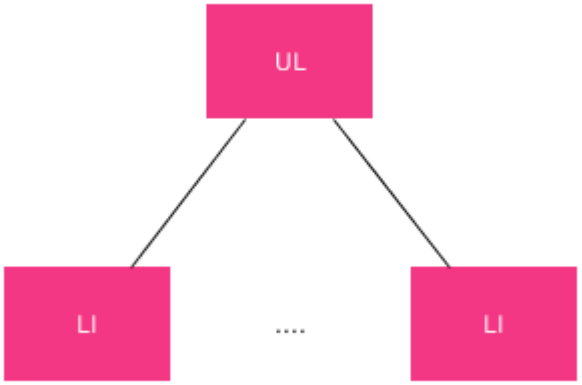
Un ejemplo:

```
<!-- Listas numeradas -->
<h3>Lista Numerada - Ordered List (ol)</h3>
<ol>
    <li>Primer elemento</li>
```

```
        <li>Segundo elemento</li>
        <li>Tercer Elemento</li>
        <li>Cuarto elemento</li>
    </ol>
```

ol sería la etiqueta padre y cada uno de los elementos de la lista iría en una etiqueta **li**.

Listas no Numeradas



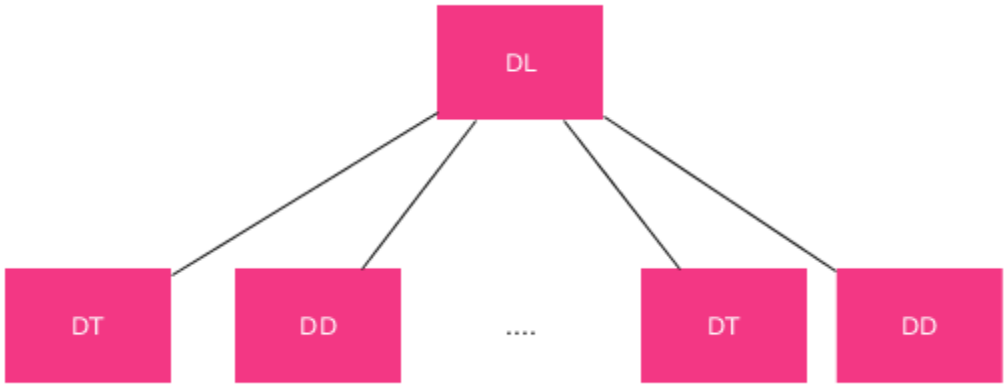
Un ejemplo:

```
    <!-- Listas no numeradas-->
    <h3>Lista NO Numerada - Unordered List (ul)</h3>
    <ul>
        <li>Primer elemento</li>
        <li>Segundo elemento</li>
        <li>Tercer Elemento</li>
        <li>Cuarto elemento</li>
    </ul>
```

ul sería la etiqueta padre y cada uno de los elementos de la lista iría en una etiqueta **li**.

Listas de definición

Se usan para poner glosarios de términos.



Un ejemplo:

```
<!-- Listas de definición-->

<h3> Lista de descripción - Description List (dl)</h3>
<dl>
    <dt>Primer término</dt>
    <dd>Definición del primer término</dd>
    <dt>Segundo término</dt>
    <dd>Definición del segundo término</dd>
    <dt>Tercer término</dt>
    <dd>Definición del tercer término</dd>
    <dt>Cuarto término</dt>
    <dd>Definición del cuarto término</dd>
</dl>
```

dl sería la etiqueta padre y cada término se define mostrando consecutivamente las etiquetas **dt**, que se corresponde con el término que vamos a definir, y **dd** que es la definición del término anterior.

3.5 Práctica: Listas

Hacer una web que visualice una lista numerada otra no numerada con tres niveles de anidación y donde podamos definir el tipo de numeración y la forma de la viñeta de cada uno de los tipos de listas que hemos visto.

3.6 Tablas

Otras de las construcciones básicas en HTML son las tablas.

Además de para la presentación de elementos, se usaron históricamente para dar estructura a las páginas. No obstante, por motivos que explicaremos más adelante, ya no se maqueta con tablas.

A pesar de eso siguen siendo un elemento importante y a continuación vamos a presentar varias formas de hacer tablas.

Tablas Simples

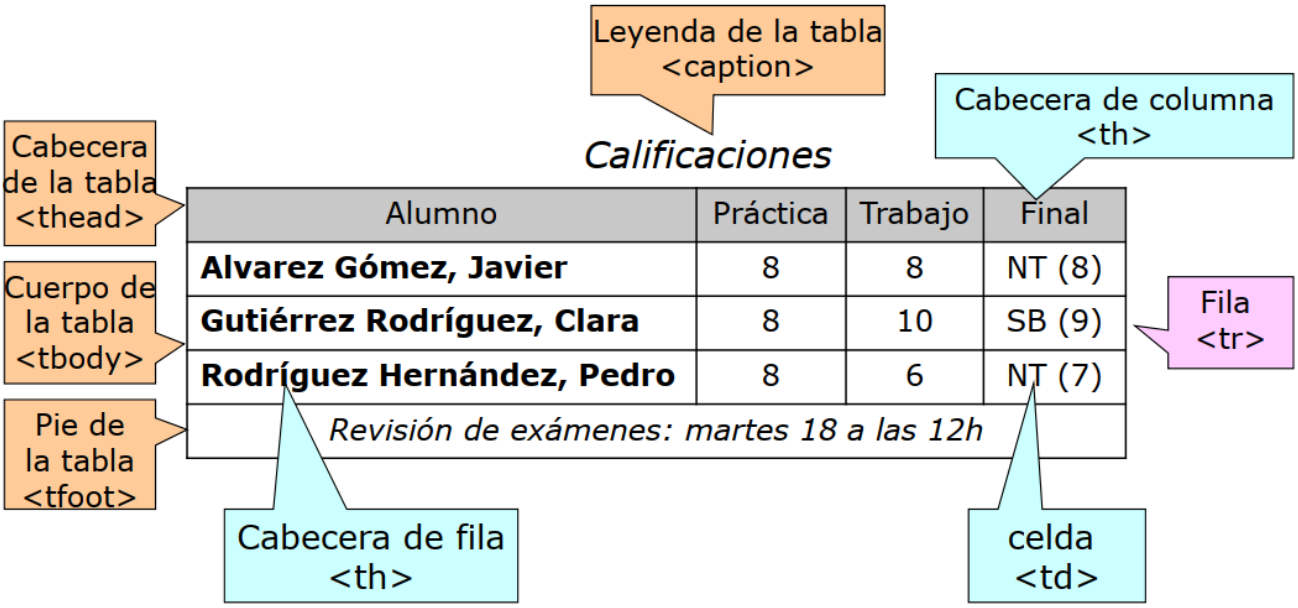
La estructura del árbol DOM más simple para una tabla sería similar a la siguiente:

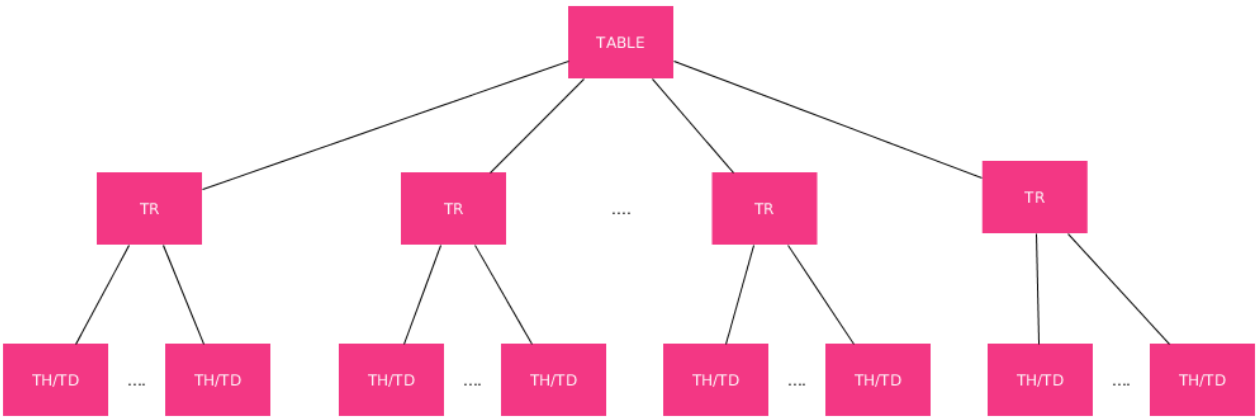
- Una etiqueta **<table>** que contiene toda la tabla.
- Como hijos directos, tantas etiquetas ***<tr>*** como filas queramos que tenga nuestra tabla.
- Dentro de cada fila, tantas celdas ***<th>*** o ***<td>*** como queramos que tengan nuestras filas. La diferencia entre la primera y la segunda es que en la primera nos pondrá el texto de la celda centrado y en negrita. <th> se suele usar para la cabecera.

NOTA: Si no coinciden el número de celdas en todas las filas veremos que suceden cosas "extrañas".

NOTA: La anchura de las celdas de una misma columna será la anchura de la más ancha de la columna.

NOTA: La altura de las celdas de una misma fila será la altura de la más alta de la fila.





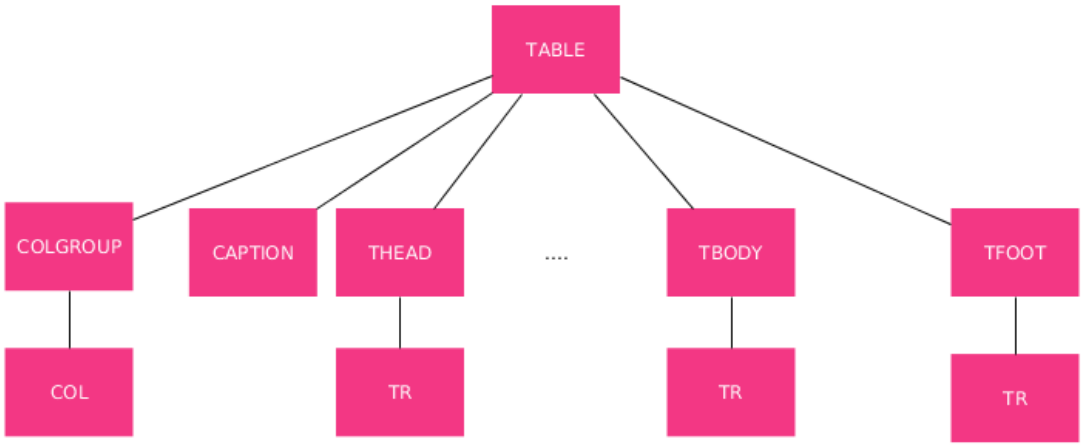
Un ejemplo sería:

```
<table>
    <tr>
        <th>Nombre</th>
        <th>Apellidos</th>
        <th>Dirección</th>
    </tr>
    <tr>
        <td>Pepe</td>
        <td>Pérez</td>
        <td>Aquí</td>
    </tr>
    <tr>
        <td>Manuel</td>
        <td>López</td>
        <td>Allí</td>
    </tr>
    <tr>
        <td>María</td>
        <td>Fernández</td>
        <td>Mas allá</td>
    </tr>
    <tr>
        <td>Sara</td>
        <td>Gallardo</td>
        <td>Mas aquí</td>
    </tr>

</table>
```

Tablas Completas

Existe además una forma más precisa y completa de construir tablas. Una forma que puede contener (aunque no es obligatorio) otras etiquetas dentro de la etiqueta raíz **<table>**. Estas nuevas etiquetas pueden ser:

- **<colgroup>** que nos permite agrupar las columnas para darles estilos. Cada uno de esos grupos lo definiremos usando una etiqueta **col** con un atributo **span** para definir el número de columnas de cada grupo.
- **<caption>** para añadir un título o leyenda a la tabla en la parte superior.
- **<thead>** que contendrá la fila (usualmente) o filas que sean la cabecera de una tabla.
- **<tbody>** que es donde pondremos las filas que son el contenido propiamente dicho de la tabla, el cuerpo.
- **<tfoot>** que es donde pondremos las filas que son el pie de nuestra tabla.



Un ejemplo sería:

```
<table>
        <colgroup>
            <col span="3" style="background-color: grey">
            <col style="background-color: yellow">
            <col style="background-color: green">
        </colgroup>
        <caption>Alumnos matriculados</caption>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Apellidos</th>
                <th>Dirección</th>
                <th>Teléfono</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <tr>

                <td>Pepe</td>
                <td>Pérez</td>
                <td>Aquí</td>
                <td>11111111</td>
                <td>yosoy@pepe.es</td>
            </tr>
            <tr>
                <td>Manuel</td>
                <td>López</td>
                <td>Allí</td>
                <td>222222222</td>
                <td>yosoy@manuel.es</td>
            </tr>
            <tr>
                <td>María</td>
                <td>Fernández</td>
                <td>Mas allá</td>
                <td>33333333</td>
                <td>yosoy@maria.es</td>
            </tr>
            <tr>
```

```
                <td>Sara</td>
                <td>Gallardo</td>
                <td>Mas aquí</td>
                <td>44444444</td>
                <td>yosoy@sara.es</td>
            </tr>
        </tbody>
        <tfoot>
            <tr>
                <td>Pie de la tabla donde pongo más texto para que se vea como crecen</td>
            </tr>
        </tfoot>
    </table>
```

Tablas Complejas

Con todo lo explicado anteriormente podemos ver que las tablas que conseguimos son relativamente sencillas. En la vida real, nos encontraremos con estructuras tabulares más complejas. Éstas también se pueden construir en HTML utilizando los siguientes atributos en las etiquetas, es decir, en las celdas.

- **rowspan:** me va a permitir que una celda ocupe más de una fila.
- **colspan:** me va a permitir que una celda ocupe más de una columna.

NOTA: Antes de escribir el código HTML de una tabla compleja es recomendable estudiar su estructura previamente.

Un ejemplo sería:

```
<table>
        <caption>HORARIO DE CLASE CURSO 20--/20--</caption>

        <thead>
            <tr>
                <th>HORAS</th>
                <th>Lunes</th>
                <th>Martes</th>
                <th>Miércoles</th>
                <th>Jueves</th>
                <th>Viernes</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>8:00</td>
                <td rowspan="2">Matemáticas</td>
                <td>Lengua</td>
                <td>Inglés</td>
                <td colspan="2">Ciencias</td>
            </tr>
            <tr>
                <td>9:00</td>
                <td>Historia</td>
                <td>Ciencias</td>
                <td>Matemáticas</td>
```

```
            <td>Lengua</td>
        </tr>
        <tr>
            <td>10:00</td>
            <th colspan="5">RECREO</th>
        </tr>
        <tr>
            <td>10:30</td>
            <td>Inglés</td>
            <td rowspan="2">Ciencias</td>
            <td>Matemáticas</td>
            <td>Lengua</td>
            <td>Historia</td>
        </tr>
        <tr>

            <td>11:30</td>
            <td>Historia</td>

            <td>Ciencias</td>
            <td>Matemáticas</td>
            <td>Inglés</td>
        </tr>

    </tbody>
</table>
```

3.7 Bordes en las tablas

En el apartado anterior, apartado en el que hemos presentado las tablas, no hemos hablado para nada de los bordes que normalmente pueden presentar estas estructuras.

De hecho, habéis podido ver que, por defecto, los navegadores no les ponen borde a las tablas.

Para ponerles bordes nos tenemos que adelantar un poquito en el temario y hablar de estilos, de CSS.

No es totalmente correcto lo que vamos a hacer aquí, pero nos va a servir para entender los tipos de bordes más usados (hay muchos más).

En principio, lo que vamos a hacer es lo siguiente:

```
<head>
    ...
    <!--Etiqueta para definir estilos-->
    <!-- Aquí definiremos los bordes-->
    <style>
        ...
        /*Estilos para los bordes*/
        ...
    </style>

</head>
```

Y vamos a presentar tres tipos de bordes, que son los más usados.

- **Bordes Simple**
- **Bordes sin colapsar**
- **Bordes inferior o superior**

Bordes simples

Pondremos dentro la etiqueta **<style>** lo siguiente:

```
table {
  border-collapse: collapse;
}

td,th {

  border: 1px solid black;
}
```

Bordes sin colapsar

Añade borde al borde la tabla y a cada celda, además. Pondremos dentro la etiqueta **<style>** lo siguiente:

```
table,td,th {

  border: 1px solid black;
}
```

Bordes Inferior/Superior

Pondremos dentro la etiqueta **<style>** lo siguiente:

```
table {
  border-collapse: collapse;
}

td,th {
  border-bottom: 1px solid black;
}
```

En el ejemplo anterior solo aparecerá el borde inferior o superior de cada celda. Si lo quisiéramos superior debemos cambiar *border-bottom* por *border-top*.

3.8 Práctica: Tablas.

Realiza una tabla con tu horario de clases. Añade bordes y colores tanto de fondo como al texto.

4. Formularios HTML

Los formularios son otros de los elementos que se han asociado desde siempre a las páginas webs. Los hemos usado tanto que es difícil calcular cuántos formularios habremos rellenado a lo largo de nuestra vida como usuario de la web.

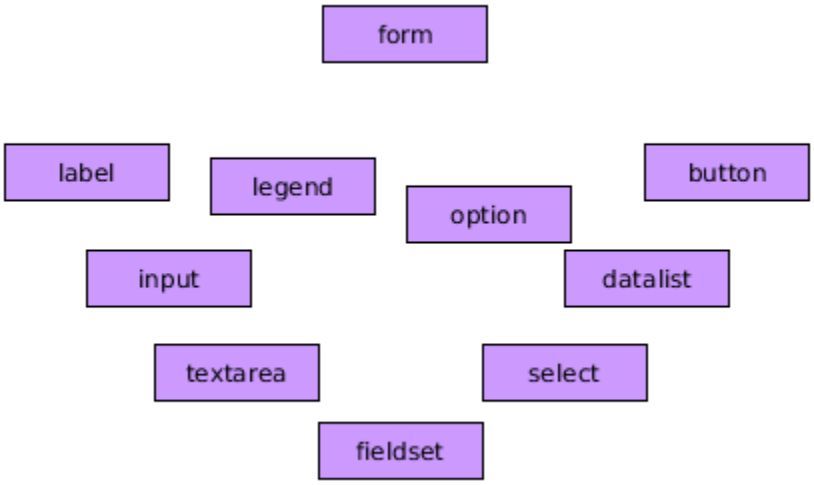
Formularios de registro, usuario y contraseña, solicitudes etc., son contextos en los que los podemos encontrar.

De un modo más formal podemos decir lo siguiente:

- Los formularios son elementos, que, como los enlaces, permiten una interacción del usuario con la página web.
- Su tarea principal es recoger información. El usuario, por tanto, debe introducir esa información en los campos del formulario.
- Una vez se ha recogido esa información el formulario la enviará al servidor, por correo o donde decida el programador para ser tratada, mostrada y/o almacenada.

Estructura del formulario

En un formulario nos podemos encontrar con muchos elementos.



El más importante de todo es el elemento padre, la etiqueta **<form>** que va a contener en su interior todos los elementos que conformen nuestros formularios.

Esta etiqueta puede tener varios atributos de entre los que destacamos:

- **action** que indica el destino de nuestros datos. Normalmente será una URL o dirección de Internet. Existen más opciones.
- **method** que indica cómo se va a pasar la información al destino (servidor). Puede ser por GET (se ve la información en la barra del navegador) y por POST (no se ve y es la opción por defecto). Las diferencias más importantes son:

  GET:
  • Limitado a 512 bytes.

- Los datos enviados se añaden al final de la URL de la página y por tanto se ven en la barra del navegador.
- Se suele usar cuando se envía información que no modifica el servidor (por ejemplo, términos para una búsqueda).
- Si no se especifica, los navegadores suelen hacer GET.

POST:
- Puede enviar más información.
- Permite enviar ficheros adjuntos.
- Los datos enviados no se ven en la barra del navegador.
- Se suele usar cuando se envía información que puede modificar el servidor.

- **enctype:** Tipo de codificación al enviar el formulario al servidor:

  - "application/x-www-form-urlencoded" o "multipart/form-data"
  - Sólo se indica cuando se adjuntan archivos.

A continuación, vamos a ver algunos de los elementos más frecuentes y a describir su estructura y funcionamiento.

Labels e Input

Normalmente para la recogida de información los formularios usando etiquetas **<input>**. Pero, para poder saber qué campos estamos rellenando, una buena práctica es poner una etiqueta (label) delante de cada input para dar nombre y asociar ambas.

Un ejemplo sería:

```
<label for="nombre">Nombre:</label>
<input type="text" name="nombre" />
```

Aquí estamos asociando la etiqueta al campo usando los atributos ***for*** y ***name***.

El campo que hemos usado para la recogida de información es un input. Existen muchos tipos que se verán más adelante. Algunos de sus atributos para los inputs son:

- **type**: indica el tipo de control de datos que se ha elegido. Valores que puede incluir: text, password, button, checkbox, reset, radio, hidden, file, image, submit (html5): color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week.
- **src**: para el control que permite crear botones con imágenes, indica el URL del archivo.
- **step**: (html5) permite establecer la cantidad de valores posibles dentro de un rango. Ej: `<input type="number" name="num" step="5"> (Valores de 5 en 5)`
- **value**: indica el valor inicial del control.
- **autofocus**: (html5) fuerza el foco (la posición del cursor) dentro de un campo, una vez que se carga la página. En Javascript conseguimos lo mismo con la función focus().
- **required**: (html5) atributo que se aplica a un campo de entrada y obliga al usuario a escribir un valor en dicho campo. anteriormente se conseguía esto mediante javascript analizando la longitud del valor del campo en cuestión mediante campo.value.length.
- **placeholder**:(html5) muestra un texto por defecto en el input hasta que éste tiene el foco, en ese momento el texto desaparece para que el usuario introduzca el texto que desea.

- **autocomplete**: (html5) activa (on) o desactiva (off) el auto completado de la entrada de datos en base a los textos introducidos anteriormente por el usuario. Puede usarse como atributo de un campo de entrada o del formulario completo.
- **alt**: se utiliza cuando el tipo es una imagen y no se puede cargar en la interfaz del usuario, se muestra este texto alternativo.
- **checked**: para indicar si el elemento está seleccionado por defecto en los controles checkbox y radiobutton.
- **disabled**: es un valor booleano que indica que el elemento está desactivado, con lo cual no admite entrada de datos. Y su valor no se envía al Servidor junto al resto de datos.
- **readonly**: valor booleano que indica que el valor del campo no se puede modificar, solo lectura.
- **form→enctype**: (html5) indica el modo de envío de los datos: aplication/x-www-form-urlencoded; multipart/form-data; text/plain
- **form→action**: (html5) es la dirección url que se utiliza al activar el elemento.
- **form→novalidate**: (html5) es un valor booleano que indica que el formulario no se valida antes de enviarse.
- **form→target**:(html5) ventana del destino del formulario. _blank, _self, _parent, _top, framename.
- **height**: (html5) altura del elemento tipo image.
- **width**: (html5) anchura del elemento tipo image.
- **size**: Tamaño inicial del control. En text/password indica la cantidad de caracteres que se pueden introducir en el campo, para el resto significa su tamaño en pixels.
- **max**: (html5) valor máximo que se puede entrar en el elemento de datos.
- **min**: (html5) valor mínimo que se puede entrar en el elemento de datos.
- **maxlength**: longitud máxima de caracteres para los campos text/password.
- **list**: (html5) es un identificador de una lista que se define con datalist.
- **multiple**: (html5) valor booleano que indica que se permite la selección de valores múltiples.
- **name**: Nombre que identifica al campo de datos. Importante para su identificación por el Servidor.
- **selected**: indica la opción seleccionada de una lista, por ejemplo**.**
- **pattern**: (html5) Expresión regular que se utiliza para validar entradas de datos. este atributo permite hacer validaciones muy complejas y utilizado correctamente puede ahorrarse muchas líneas de código.
- **tabindex** para asegurarnos que podemos navegar por los elementos usando el tabulador podemos establecer un orden para cada uno.
- Y muchos más…

Barra de progreso (progress bar)

Con este componente podemos indicar el nivel de progreso de una tarea. Suele llevar asociada una etiqueta o <label> y su uso se complementa con JavaScript. Ejemplo:

```
<label for="file">Downloading progress:</label>
<progress id="file" value="32" max="100"> 32% </progress>
```

La etiqueta ***<progress>*** no es adecuada para representar un indicador (por ejemplo, uso de espacio en disco o relevancia del resultado de una consulta). Para representar un indicador, es mejor utilizar la etiqueta ***<meter>*** en su lugar.

Estado: Determinado de la barra de progreso

Para que la barra de progreso se encuentre en el estado "**determinado**", debe de estar presente el atributo value.

```
<progress value="0.5"></progress>
```

Estado: Indeterminado de la barra de progreso

Para que la barra de progreso se encuentre en el estado "**Indeterminado**", no debe de estar presente el atributo value.

El porqué de este estado; imaginemos que tenemos una barra de progreso en nuestra página y perdemos la conexión con el host; nuestro código JavaScript debería de detectar este posible estado y borrar el atributo *value* para indicarle al usuario que hubo un problema con la conexión.

Para dar ***estilo*** a la barra de progreso se deben usar en CSS dos *pseudoclases* (*se verá más adelante*):

- **progress-bar:** es la pseudoclase que se puede usar para diseñar el contenedor de la barra de progreso.
- **progress-value:** es la pseudoclase para diseñar el valor de la barra de progreso. El color de fondo, por ejemplo.

```
progress {
    display: block;
    -webkit-appearance: none; /*Necesario para que no se apliquen los estilos del
navegador*/
}

progress::-webkit-progress-bar { /*Chrome*/
    background: black;
    border-radius: 50px;
    padding: 2px;
}
progress::-moz-progress-bar { /*Firefox*/
    background: black;
    border-radius: 50px;
    padding: 2px;
}

progress::-webkit-progress-value { /*Chrome*/
    border-radius: 50px;
    background: orange;
}
```

Listas desplegables

Son elementos que también hemos visto muchas veces. Al hacer *click* sobre ellos, nos muestran varias opciones de las que podremos escoger una o varias.

Para crear listas desplegables usaremos las etiquetas **<select>** como etiqueta padre y una etiqueta **<option>** para cada una de las opciones que tengamos en la lista desplegable. Si queremos agrupar esas opciones las meteremos a su vez dentro de una etiqueta **<optgroup>**.

Un ejemplo sería:

```
<select name="provincia">
    <optgroup label="Aragon">
        <option value="Z">Zaragoza</option>
        <option>Huesca</option>
        <option selected>Teruel</option>
    </optgroup>

</select>
```

Fijaros los nuevos atributos que se han añadido (**selected**).

Áreas de Texto

Son campos de formulario para meter textos largos. Tienen dos atributos interesantes.

- **rows:** para indicar el número de filas que tiene el campo.
- **cols:** para indicar el número de columnas que tiene el campo.

Un ejemplo sería:

```
<textarea cols="10" rows="10">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Cumque voluptate corrupti ut earum, aliquam sint amet repellendus, vitae placeat repellat quis, ipsa qui. Velit placeat reiciendis tempore autem nostrum eaque?
</textarea>
```

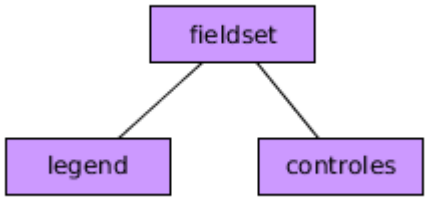
Botones

El atributo más importante que tienen es ***type*** que puede tomar tres valores:

- **button:** se comporta como un botón que puede estar presionado o no. Nada especial.
- **reset:** al hacer click borra todos los datos introducidos previamente en el formulario y pone los valores por defecto.
- **submit:** al hacer click envía los datos introducidos al destino que hemos puesto en el ***action*** del formulario.

Agrupando atributos

Hay situaciones en las que los campos de los formularios por significado, por importancia o por cualquier razón, queremos que se muestren agrupados. Para estos casos, tenemos la etiqueta **<fieldset>** que la usaremos usando el siguiente esquema:



Siendo:

- **<legend>** una etiqueta que contiene el nombre del grupo y que se mostrará en la parte de arriba.
- **controles** todos los campos que queramos meter.

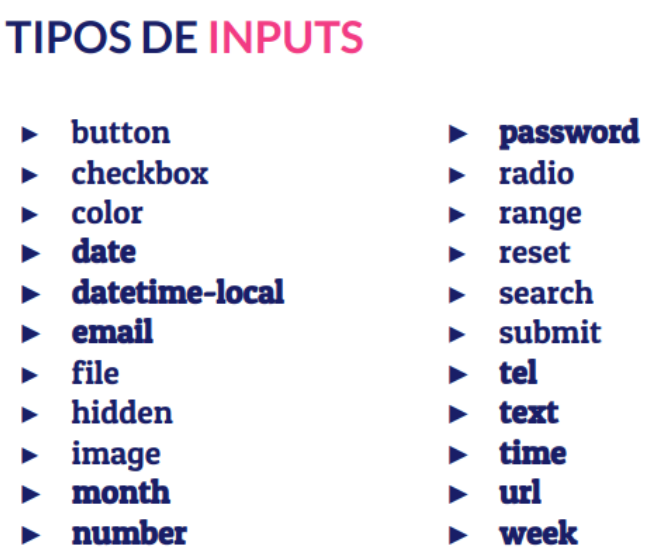
NOTA: la etiqueta tiene un borde por defecto

Un ejemplo sería:

```
<fieldset>
    <legend>Datos Personales</legend>
    <p>
        <label for="nombre">Nombre:</label>
        <input type="text" name="nombre" placeholder="Inserta Nombre" required />
    </p>
    <p>
        <label for="apellidos">Apellidos:</label>
        <input type="text" name="apellidos" value="Pérez" readonly />
    </p>

    <p>
        <label for="provincia">Provincia</label>
        <select name="provincia">
            <optgroup label="Aragon">
                <option value="Z">Zaragoza</option>
                <option>Huesca</option>
                <option selected>Teruel</option>
            </optgroup>

        </select>
    </p>

    <p>
        <input type="submit" value="Enviar" />
    </p>
</fieldset>
```

4.1 Tipos de inputs en formularios

La lista de tipos (***type***) de Inputs que podemos tener es muy larga. La podemos ver en la siguiente imagen:



Algunos de ellos interesantes son:

- **password** → oculta los caracteres que se escriben.
- **hidden** → campo oculto que no se muestra al usuario. Se usa para enviar datos en un formulario predefinidos por el desarrollador.
- **file** → permite agregar un botón para seleccionar uno o varios ficheros (propiedad *multiple*) que se enviarán con los demás datos del formulario.
- **image** → permite seleccionar una imagen como botón de *Submit*.
- **Algunos otros:** *keygen, output*.

Tienen especial relevancia los siguientes:

- Radio Groups
- Checkboxes
- Datalist
- Range

Radio Group

Es una agrupación de inputs que presenta opciones que queremos que sean excluyentes:

Un ejemplo sería:

```
<label for="genero">Sexo</label>
<input type="radio" name="genero" value="masculino" />Hombre<br />
<input type="radio" name="genero" value="fememino" checked />Mujer<br />
```

Para que sean excluyentes deben de tener el mismo valor para el atributo ***name*** y el ***type="radio"***

CheckBoxes

Es una agrupación de opciones que presenta opciones de las cuáles podemos elegir una o varias.

Un ejemplo sería:

```
<label for="dispositivos">Dispositivos electrónicos</label><br />
<input type="checkbox" name="dispositivos" value="pc" />PC<br />
<input type="checkbox" name="dispositivos" value="table" />Tableta<br />
<input type="checkbox" name="dispositivos" value="movil" />Móvil
```

Fijaros que para agruparlos deben de tener el mismo valor para **name** y el ***type="checkbox"***

DataList

Es una nueva forma en HTML5 de hacer una lista desplegable de valores posibles para un input.

Un ejemplo sería:

```
<input list="editor">

<datalist id="editor">
    <option value="Atom">
    <option value="NotePad++">
    <option value="VsCode">
    <option value="Sublime">
    <option value="Brackets">
</datalist>
```

Con el atributo **list** indicamos la lista de opciones que vamos a tener y en la etiqueta metemos las que queramos.

Es importante destacar que podríamos meter otras opciones en el Input, pero al usar esta estructura se nos ayuda a rellenarlo y a elegir la opción correcta.

Range

Define un control para ingresar un número cuyo valor exacto no es importante (como un control deslizante).

El rango predeterminado es de 0 a 100. Sin embargo, se pueden establecer restricciones sobre qué números se aceptan con los atributos a continuación:

- max: especifica el valor máximo permitido
- min: especifica el valor mínimo permitido
- step: especifica los intervalos de números legales
- value: especifica el valor predeterminado

Podremos modificar su estilo de diversas formas. **Aquí** tienes un ejemplo de las modificaciones que podemos hacer.

4.2 Práctica: Formularios

1.- Crear un formulario para registrar un nuevo cliente en un sitio web, solicitando datos personales necesarios, así como la creación de una clave de acceso (password).

2.- Crear un formulario para entrega de prácticas en una asignatura:

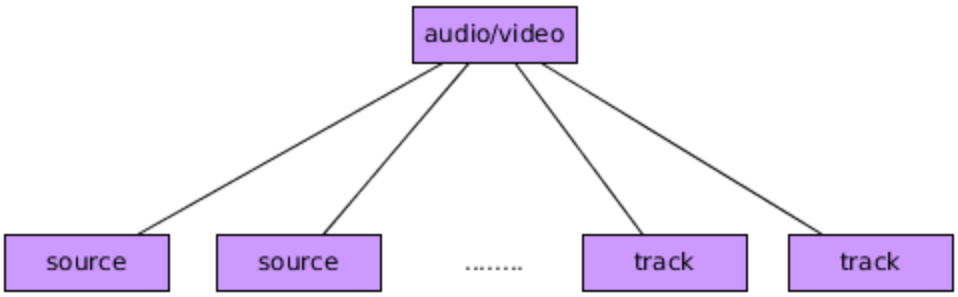
- Investiga cómo se puede comprobar el tipo de fichero

3.- Crea un formulario para realizar una encuesta

4.3 Etiquetas multimedia

Una de las novedades más esperadas en HTML5 es la aparición de etiquetas específicas para multimedia.

Las más importantes son las etiquetas **<audio>** y **<video>** y para representarlas vamos optar por usar, de manera general estructuras de árbol similares a las siguientes (hay otros árboles válidos, pero este es más fácil y general):



- La etiqueta raíz será **<audio>** o **<video>** según lo que queramos mostrar en nuestro navegador.
- Dentro de esta etiqueta raíz tendremos tantas etiquetas **<source>** como formatos distintos de ese mismo elemento ofrezcamos. Es importante asegurarnos de que el formato del fichero multimedia es soportado por los navegadores. Si ofertamos varias fuentes, el navegador decidirá cuál reproduce.
- Opcionalmente podremos añadir descripción de audio, subtítulos o similares usando una o varias etiquetas **<track>**. Buscar más información sobre los valores de esta etiqueta.

Las etiquetas **<audio>** y **<video>** pueden tener los siguientes atributos.

- **controls** para mostrar los controles gráficos de reproducción (play, stop, pause, volumen).
- **autoplay** para que se empiece a reproducir al cargar la página de manera automática.
- **loop** si queremos que se reproduzca en bucle de manera infinita.
- **muted** si queremos que se reproduzca en silencio.

Adicionalmente la etiqueta **<video>** puede tener más atributos:

- **width** y **height** para indicar la anchura y altura del vídeo.
- **poster** para mostrar una imagen mientras el vídeo se carga.

Etiqueta <source>

Tiene dos atributos fundamentales:

- **src** para indicar dónde está el fichero multimedia (una ruta).
- **type** para indicar el tipo de fichero, por ejemplo: _type="audio/mp3"_ o _type="video/webm"_.

Etiqueta <track>

Tiene 5 atributos fundamentales:

- **src** para indicar dónde está el fichero de descripción (una ruta).
- **kind** para indicar si es subtítulo, descripción etc... Tiene diversos valores que se pueden consultar en la documentación.
- **scrlang** que indica el código del idioma (es, en, fr…) de la descripción. El navegador elegirá uno u otro dependiendo de la configuración el usuario.
- **label** la etiqueta que describe el idioma (español, inglés etc..).
- **type** el tipo de fichero. Hay diversos, aunque lo más normal es *type="text/vtt"*.

Un ejemplo de todo junto:

```
<audio controls>
    <source src="multimedia_files/Frase_de_Neil_Armstrong.mp3" type="audio/mp3" />
    <source src="multimedia_files/Frase_de_Neil_Armstrong.ogg" type="audio/ogg" />
    Tu navegador no soporta esta etiqueta (audio)
    <!--Tu servidor web tiene que estar debidamente configurado-->
    <track src="multimedia_files/Frase_de_Neil_Armstrong_es.vtt" kind="subtitles"
srclang="es" label="Español" type="text/vtt" />
    <track src="multimedia_files/Frase_de_Neil_Armstrong_en.vtt" kind="subtitles"
srclang="en" label="English" type="text/vtt" />
</audio>

    <!-- Etiqueta de vídeo-->
<video width="320" height="240" controls>
    <source src="multimedia_files/Zeppelin_Explodes.webm" type="video/webm">
    Tu navegador no soporta esta etiqueta (video)
</video>
```

5. Etiquetas semánticas

Las etiquetas semánticas son nuevas etiquetas, que aparecen con la versión HTML5, y que sirven para que desarrolladores, navegadores y buscadores entiendan mejor qué tipo de contenido tienen ciertas secciones o partes de nuestra web.

Las etiquetas semánticas que existen son las siguientes (su significado es auto explicativo si traduces del inglés):

- **header** → Cabecera del sitio, define la parte superior de la página web. Suele mostrar algún elemento representativo que caracterice a la marca del sitio web. En la mayor parte de los casos, se sitúa el menú de navegación *nav.*
- **footer** → Equivale al pie de página, pero puede utilizarse de forma general para todo el sitio web, y también para definir el pie de un elemento tipo *article.*
- **main** → Representa el contenido principal del *body* de un documento. El área principal consiste en el contenido que está directamente relacionado con el tema central de un documento. Este contenido debe ser único para ese documento. Es importante tener en cuenta que solo debe haber un elemento *main* en un documento, y no debe descender de un elemento *article, aside, footer, header o nav*.
- **section** → Se usa para crear diferentes secciones dentro de una misma página web. Estas secciones pueden contener, a su vez, varios elementos de tipo *article.*
- **article** → Se suele usar para describir las unidades de contenido. Pueden estar subdivididas en: (cabecera) header, cuerpo (section) y pie de artículo (footer).
- **aside** → Define la barra lateral de una web, aunque puede contener cualquier tipo de contenido, normalmente se usa para mostrar elementos que suponen un enlace hacia otros sitios web, redes sociales, etc.
- **details** → Específica detalles al usuario que pueden ser mostrados u ocultados bajo demanda. Se suele usar para crear widgets interactivos. Por defecto estará cerrado. Se suele usar con *summary.* ***Details*** acepta un parámetro booleano para mostrar o no el contenido.
- **summary** → encabezado visible para el elemento *details*. Se puede hacer clic en el encabezado para ver / ocultar los detalles.

```
<details>

  <summary>Epcot Center</summary>

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

</details>
```

- **dialog** → muestra un pequeño diálogo o subventana. Acepta un valor booleano (***open***) para indicar si el *dialog* está activo y el usuario puede interaccionar con él.
- **nav** → crea un menú de navegación, que en la con las diferentes partes del sitio web. Normalmente se sitúa justo debajo del encabezado o a la izquierda de la página.

Cuando usamos alguna de estas etiquetas sucede lo siguiente:

- A nivel visual se comportan como una etiqueta .
- Aportan significado, podemos conocer qué es lo que hay dentro (no como un div). Esto ayuda a buscadores y desarrolladores tal y como se dijo antes.
- Se utilizan para maquetar, como los divs.
- Tienen reglas de anidación y éstas son todas distintas. Por ejemplo: *Sólo puede haber un main y no puede ser hijo de article, aside, footer, header o nav*

6. Otras etiquetas

No las vamos a ver en detalle, pero pondremos un ejemplo para cada una de ellas.

- **<iframe>** nos permite mostrar páginas web dentro de otras páginas.

```
<!-- Etiqueta iFrame -->
<iframe src="http://www.as.com" height="300px" width="600px"></iframe>
```

- **<canvas>** nos permite dibujar, animar y un mundo infinito de posibilidades. Es la base de todos los juegos HTML5, desde el más sencillo al más complejo. Más info **aquí**.

- ** ** es una etiqueta para separar las secciones de los documentos y que se utiliza para maquetar. Es el elemento en bloque por excelencia. Ya veremos luego lo que eso significa.

```
<div>
  Some text inside a div
</div>
```

- **** es un elemento en línea que no aporta nada visualmente pero que a nivel lógico me sirve para, por ejemplo, poder distinguir ciertas partes dentro de un texto. Ya veremos luego lo que significa *"en línea"*.

```
<span>span</span>
```

- **<script>** nos sirve para introducir código escrito en otros lenguajes, típicamente en Javascript.

```
<script>
    var nombre = prompt("Hola como te llamas?");
    alert("Encantado " + nombre);
</script>
```

- **<object>** sirve para mostrar documentos externos o de terceros en la pantalla del navegador.

```
<!-- Adding external objects with object -->
    <!-- docs, videos, svg, depende de plugins de terceros-->
    <object type="application/pdf" data="https://www.um.es/atica/documentos/html.pdf"
width="300" height="200">
        No se ha podido abrir el documento.
    </object>

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

Algunos atributos de ***object:***

- **data="URL"** → Los datos que utiliza el objeto.

- **type=”tipo-mime”** → Tipo de contenido de los datos. El navegador decidirá el plugin o acción que corresponda en función del tipo:
- **classid, codebase, codetype** → Información específica que depende del tipo de objeto.

Algunos de los recursos que podemos añadir mediante ***<object>*** son: vídeos, archivos de sonido, applets de Java, archivos pdf, controles ActiveX, …

Si el navegador no sabe abrir el contenido, solicitará al usuario que descargue un plugin para ejectuar.

Cualquier texto que no sea una etiqueta se mostrará si el navegador no es capaz de reproducir el contenido.

Incluir un vídeo de YouTube:

```
<object width="640" height="360" type="application/x-shockwave-flash"
data="http://www.youtube.com/v/Fgh2dFngFsg">
    <param name="movie" value="http://www.youtube.com/v/Fgh2dFngFsg" />
    <param name="wmode" value="transparent" />
</object>
```

7. Accesibilidad en HTML

Por ley tienen que cumplirlo. Una web accesible tiene presente que puede ser visitada por todo tipo de aplicaciones y usuarios, incluso por aquellos con alguna discapacidad.

Algunas reglas generales:

- Usar **etiquetas semánticas** en vez de que no aportan significado.
- Usa el atributo **for** para las etiquetas en los formularios (los pone en relación con el campo correcto).
- Añade atributos **alt** a las imágenes.
- Especifica el **idioma** de la página.
- Utiliza las **cabeceras** (h1, h2, h3, …) correctamente para que los lectores “*conozcan*” la estructura de tu página.
- …

Como novedad en HTML5, tenemos los atributos **ARIA** (Accesible Rich Internet Application), que son atributos especiales que se añaden a las etiquetas HTML para que éstas sean **entendidas mejor** por los lectores que usan aquellas personas con alguna discapacidad. Hay muchos, algunos de los más importantes:

- Role
- Aria-owns
- Aria-hidden
- Aria-live
- Aria-label
- Aria-cheched
- Aria-disabled

- Aria-valuenow, aria-valuemax, aria-valuemin
- …..

Ayudan mucho a personas con problemas de visión a seguir nuestra web. Investiga su significado y pruébalos.