Contenidos a Trabajar

1. Hipervínculos/Enlaces
2. Listas - Listas (ul – lo)
3. Tablas (table - tr - td - th)
4. Imágenes (formatos)
5. Iframe (incrustar Youtube o Google Maps)
6. Formularios: Form - input (text - email - password - submit - check - radio - textarea - button - label - Select / option)

Formateando el texto en HTML

En esta sección de la unidad trabajaremos en colocar **negritas**, *itálicas*, subrayados, subíndices y superíndices.

Además de todo lo relativo a la organización de los párrafos, uno de los aspectos primordiales del formateo de un texto es el de la propia letra. Resulta muy común y práctico presentar texto resaltado en negrita, itálica y otros. Paralelamente el uso de índices, subíndices resulta vital para la publicación de textos científicos.

Todo esto y mucho más es posible por medio del HTML a partir de multitud de etiquetas entre las cuales vamos a destacar algunas.

Pero antes de comenzar cabe hacer una reflexión sobre por qué son interesantes estas etiquetas y se siguen usando, a pesar que están entrando prácticamente en el terreno de CSS, ya que en la práctica están directamente formateando el aspecto de las fuentes. Son importantes porque las etiquetas en si no están para definir un estilo en concreto, sino una función de ciertas palabras dentro de un contenido. Por ejemplo, las negritas quieren decir que algo tiene más fuerza o importancia dentro de un texto y una itálica se puede usar para un texto que está citado o algún énfasis particular. En cuanto a subíndices y superíndices todavía es más claro, ya que éstos especifican cosas que tiene que ver con el contenido y no con la presentación.

En conclusión, el uso de estas etiquetas refuerza el concepto de web semántica y el sentido que le damos a nuestro HTML.

Negrita

Podemos escribir texto en negrita incluyéndose dentro de las etiquetas **** o ****.

Escribiendo un código de este tipo:

```
<b>Este texto está en negrita</b>
```

Obtenemos este resultado:

Este texto está en negrita

Este texto no

Nota: ¿Qué diferencia hay entre B y STRONG? Aunque las dos etiquetas hacen el mismo efecto, tienen una peculiaridad que las hace distintas. La etiqueta B indica negrita, mientras que la etiqueta STRONG indica que se debe escribir con fuerza.

Itálica

También en este caso existen dos posibilidades, **** o ****.

He aquí un ejemplo de texto en itálica:

```
<i>Este texto está en itálica</i>
```

Que da el siguiente efecto:

Este texto está en itálica

Este texto no

Subrayado

El HTML nos propone también para el subrayado el par de etiquetas: **<u></u>**(underlined). Sin embargo, el uso de subrayados ha de ser aplicado con mucha precaución dado que los enlaces hipertexto van, a no ser que se indique lo contrario, subrayados con lo que podemos confundir al lector y apartarlo del verdadero interés de nuestro texto.

Subíndices y superíndices

Este tipo de formato resulta de extrema utilidad para textos científicos. Las etiquetas empleadas son:

**** para los superíndices
<sub></sup> para los subíndices

Anidar etiquetas

Todas estas etiquetas y por supuesto el resto de las vistas y que veremos más adelante pueden ser anidadas unas dentro de otras de manera a conseguir resultados diferentes. Así, podemos sin ningún problema crear texto en negrita e itálica embebiendo una etiqueta dentro de la otra:

```
<b>Esto sólo está en negrita <i>y esto en negrita e itálica</i></b>
```

Esto nos arroja un texto como este:

Esto sólo está en negrita *y esto en negrita e itálica*

Enlaces en HTML y sus distintos tipos.

Un sitio web podría ser considerado como un conjunto de archivos, basado en páginas HTML e imágenes, que constituyen el contenido al que el usuario tiene acceso. Sin embargo, no podríamos hablar de navegación si estos archivos HTML no estuviesen debidamente conectados entre ellos y el exterior de nuestro sitio por medio de enlaces url´s.

La particularidad de HTML de relacionar contenidos de los archivos introduciendo referencias bajo forma de enlaces que permitan un acceso rápido a otra información deseada, fue una de las características que lo impulsó vertiginosamente.

De poco serviría tener páginas aisladas a las que el usuario no puede acceder o no vincular una información con otra.

Un enlace puede ser fácilmente detectado por el usuario en una página. Basta con deslizar el puntero del ratón sobre las imágenes o el texto y ver como cambia de su forma original transformándose por regla general en una mano con un dedo señalador. Adicionalmente, estos enlaces suelen ir, en el caso de los textos, coloreados y subrayados para que el usuario no tenga dificultad en reconocerlos.

Sintaxis de un enlace

Para colocar un enlace, nos serviremos de la etiqueta **<a>** y su cierre. Dentro de la etiqueta de apertura deberemos especificar asimismo el destino del enlace. Este destino será introducido bajo forma de atributo, el cual lleva por nombre "href".

La sintaxis general de un enlace es por tanto de la forma:

```
<a href="rutaDestino">Haz click aquí</a>
```

Siendo el "haz click aquí" un texto o una imagen. Es la parte de la página que se colocará activa y donde deberemos pulsar para acceder al enlace. Por su parte, "destino" será una página, y también (con algún cambio) puede ser un correo electrónico o un archivo.

Por ejemplo, un enlace a Google, tendría esta manera ***Ir a Google***

Ahora, si queremos que el contenido del enlace sea una imagen y no un texto, podremos colocar la correspondiente etiqueta IMG dentro de la etiqueta **<a>**.

<img src="logo.jpg">

Tipos de enlaces

Para estudiar en profundidad los enlaces tenemos que clasificarlos por su tipo, porque dependiendo de ese tipo algunas cosas cambiarán a la hora de construirlos.

En función del destino los enlaces pueden ser agrupados en:

Enlaces internos: los que se dirigen a otras partes dentro de la misma página.

Enlaces locales: los que se dirigen a otras páginas del mismo sitio web.

Enlaces remotos: los dirigidos hacia páginas de otros sitios web.

Enlaces de emails: para crear un mensaje de correo dirigido a una dirección.
Enlaces con archivos: para que los usuarios puedan hacer download de ficheros.

Sintaxis de los enlaces en la misma página

Crear un enlace que apunte al final de la página. Lo primero será colocar nuestro enlace origen. Este enlace de origen es el que el usuario podrá hacer clic.

Ir abajo

Como se puede ver, el contenido del enlace es el texto "Ir abajo" y el destino, #abajo, es un punto de la misma página que todavía no hemos definido. Ojo al símbolo "#": es él quien especifica al navegador que el enlace apunta a una sección en particular, a un punto interno dentro de la misma página.

Cómo construir enlaces en HTML cuyo destino sean otras páginas dentro del mismo sitio web.

Como hemos dicho, un sitio web está constituido de páginas interconexas, que se relacionan mediante enlaces de hipertexto. Para abordar el estudio dividimos la materia por los distintos tipos de enlaces que nos podemos encontrar, atendiendo al tipo de destino. En el capítulo anterior vimos cómo enlazar distintas secciones dentro de una misma página.

En este artículo nos pondremos con otros tipos de enlaces, a los que hemos llamado "Enlaces locales". Se trata de un tipo de enlace mucho más común en el día a día del desarrollo. De hecho, es el tipo de enlace que más se produce en lo general. Estos enlaces locales nos permiten relacionar distintos documentos HTML que componen un sitio web. Gracias a los enlaces locales podremos convertir varias páginas sueltas en un sitio web completo, compuesto de varios documentos.

Para crear este tipo de enlaces, hemos de usar la misma etiqueta A que ya conocemos, de la siguiente forma:

contenido

Rutas de los enlaces

Hacer un enlace en sí no es para nada complejo. No requiere muchas explicaciones con lo que ya conocemos del manual de HTML, sin embargo hay que abordar con detalle un tema importante: las rutas de los enlaces. Como rutas nos referimos al destino del enlace, o sea, lo que ponemos en el atributo "href" y es importante que nos paremos aquí porque nos puede dar algunos problemas al desarrollar, sobre todo para las personas que puedan tener menos experiencia en el trabajo con el ordenador.

Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios.

Estos directorios suelen contener diferentes secciones de la página, imágenes, scripts, estilos... Es por ello que en muchos casos no nos valdrá con especificar el nombre del archivo, sino que tendremos que especificar además el directorio en el que nuestro archivo.html está alojado.

Si la página de destino está en una carpeta o subdirectorio interior al directorio donde está el archivo de origen, hemos de marcar la ruta enumerando cada uno de los directorios por los que pasamos hasta llegar al archivo de destino, separándolos por el símbolo barra "/". Al final obviamente, escribimos el nombre del archivo destino.

Si la página destino se encuentra en un directorio padre (superior al de la página del enlace), hemos de escribir dos puntos y una barra "../" tantas veces como niveles subamos de carpetas hasta dar con el directorio donde está emplazado el archivo destino.

Imagina que tienes la siguiente estructura de carpetas y archivos. La que aparece en la siguiente imagen.

Por ejemplo:

Para hacer un enlace desde yyy.html con destino xxx.html cuando xxx.html está en una carpeta diferente:

Ir ahora a xxx.html

Enlazar con una página diferente, pero en una sección interna

Los enlaces locales pueden, a su vez, apuntar a la página y a una sección concreta. Este tipo de enlaces resultan ser un híbrido de interno y local.

La sintaxis es de este tipo:

contenido

Como para los enlaces internos, en este caso hemos de marcar la sección con un ancla:

Enlaces remotos

Son los enlaces que se dirigen hacia páginas que se encuentran fuera de nuestro sitio web, es decir, cualquier otro documento que no forma parte de nuestro sitio. Generalmente nuestro sitio web estará en un dominio determinado, tipo ***example.com***.

Los enlaces remotos son los que van a páginas que estarían en otro dominio diferente.

Este tipo de enlaces es muy común y no representa ninguna dificultad. Simplemente colocamos en el atributo ***href*** de nuestra etiqueta A la URL o dirección de la página con la que queremos enlazar. Será algo parecido a esto.

ir a google.com

Sólo cabe destacar que todos las direcciones web (URLs) empiezan por http://, o https:// en el caso que la página de destino se sirva mediante un servidor seguro. Este tipo de rutas que comienzan por "http" también se conocen como "rutas absolutas". Cuando enlazas con páginas que están en otros dominios necesitas usar necesariamente rutas absolutas.

Nota: La parte por la que inician las direcciones de sitios web (http://) nos indica que el protocolo por el que se accede es HTTP, el utilizado en la web. No debemos olvidarnos de colocarlas, porque si no lo hacemos, los enlaces serán tratados como enlaces locales a nuestro sitio.

Otra cosa interesante es que no tenemos que enlazar con una página web con el protocolo HTTP necesariamente. También podemos acceder a recursos a través de otros protocolos como el FTP. En tal caso, las direcciones de los recursos no comenzarán por http:// sino por ftp. En nuestra unidad de Infraestructura hablaremos más detalladamente de protocolos y cual es su uso y funcionamiento.

Enlaces a direcciones de correo

Los enlaces a direcciones de correo son aquellos que al seleccionarlos nos abre un nuevo mensaje de correo electrónico dirigido a una dirección de mail determinada. Estos enlaces son muy habituales en las páginas web y resultan la manera más rápida de ofrecer al visitante una vía para el contacto con el propietario de la página.

Para colocar un enlace dirigido hacia una dirección de correo colocamos mailto: en el atributo href del enlace, seguido de la dirección de correo a la que se debe dirigir el enlace.

juan@gmail.com

Importante: Si un usuario no tiene configurado un programa de correo por defecto en su ordenador no podrá enviar mensajes con esta metodología.

Además de la dirección de correo del destinatario, también podemos colocar en el enlace el asunto del mensaje. Esto se consigue colocando después de la dirección de correo un interrogante, la palabra subject, un signo igual (=) y el asunto en concreto.

abc@gmail.com

Podemos colocar otros atributos del mensaje con una sintaxis parecida. En este caso indicamos también que el correo debe ir con copia a roberto@gmail.com.

juan@gmail.com

Nota: El visitante de la página necesitará tener configurada una cuenta de correo electrónico en su sistema para enviar los mensajes. Lógicamente, si no tiene servicio de correo en el ordenador no se podrán enviar los mensajes y este sistema de contacto con el visitante no funcionará.

Más adelante en el tema de formularios, mediante los cuales seremos capaces de implementar una serie de campos donde solicitar información de todo tipo, que luego se pueda enviar por email. Es una manera más usable en lo que respecta al contacto con el usuario.

Enlaces con archivos

Este no es un tipo de enlace propiamente dicho, aunque es de uso muy habitual.

El mecanismo es el mismo que hemos conocido en los enlaces locales y los enlaces remotos, con la particularidad de estar dirigidos hacia un archivo que puede abrirse o descargarse.

Si queremos enlazar con un archivo fichero.zip que se encuentra en el mismo directorio que la página se escribiría un enlace así.

Descargá fichero.zip

En enlace de este tipo en un navegador descargará el fichero, haciendo la pregunta típica de "Qué queremos hacer con el archivo. Abrirlo o guardarlo en disco".

Si queremos enlazar hacia otro tipo de archivo como un PDF, podemos hacerlo de la misma manera ya que si el navegador reconoce el tipo de archivo, y será el responsable de abrirlo utilizando el conector o programa adecuado para ello. Así, si por ejemplo enlazamos con un PDF pondrá el programa Acrobat Reader en funcionamiento para mostrar los contenidos.

Este sería un ejemplo de enlace a un documento PDF.

Descarga el PDF aquí

Formatos de imágen/gráficos para páginas web

Las nociones básicas para el uso de archivos gráficos son sencillas, conocerlas, aunque sea ligeramente, nos ayudará a crear sitios agradables y rápidos. No cometer errores en el uso de las imágenes es fundamental, aunque no seas un diseñador , su buen uso también redundará en una carga del sitio más ágil y dinámica.

Tipos de archivos

En Internet se utilizan principalmente los tipos de archivos gráficos PNG, GIF y JPG, pensados especialmente para optimizar el tamaño que ocupan en disco, ya que los archivos pequeños se transmiten más rápidamente por la Red. Actualmente, también el uso de formatos SVG, se está incrementando debido a que se caracterizan por su pequeño tamaño, sencillez y nitidez.

Hay varios formatos de imágenes o gráficos vectoriales, el más usado de ellos es el SVG.

El formato de archivo GIF se usa para las imágenes que tengan dibujos, mientras que el formato JPG se usa para las fotografías. Los dos comprimen las imágenes para guardarlas. La forma de comprimir la imagen que utiliza cada formato es lo que los hace ideales para unos u otros propósitos.

Adicionalmente, se puede usar un tercer formato gráfico en las páginas web, el PNG. Este formato no tiene tanta aceptación como el GIF o JPG por varias razones, entre las que destacan el desconocimiento del formato por parte de los desarrolladores, que las herramientas habituales para tratar gráficos (como por ejemplo Photoshop) generalmente no lo soportaban y que los navegadores antiguos también tienen problemas para visualizarlas.

Sin embargo, el formato se comporta muy bien en cuanto a compresión y calidad del gráfico conseguido, por lo que resulta muy útil en infinidad de casos. Todos estos problemas han pasado y ya sólo Internet Explorer tiene algunos fallos cuando trata con PNG, pero la aceptación actual es más que suficiente para incorporarlo a nuestras posibilidades reales de trabajo con formatos y optimización de archivos.

Formato GIF

Además de ser un archivo ideal para las imágenes que estén dibujadas tiene muchas otras características que son importantes y útiles.

Compresión: Es muy buena para dibujos, como ya hemos dicho. Incluso puede ser interesante si la imagen es muy pequeña, aunque sea una foto.

Transparencia: es una utilidad para definir ciertas partes del dibujo como transparentes. De este modo podemos colocar las imágenes sobre distintos fondos sin que se vea el cuadrado donde está inscrito el dibujo, mostrando la silueta del dibujo en cuestión.

Para crear un gif transparente debemos utilizar un programa de diseño gráfico, con el podemos indicar qué colores del dibujo queremos que sean transparentes. Generalmente, definimos la transparencia cuando vamos a guardar el gráfico.

Colores: Con este formato gráfico podemos utilizar paletas, conjuntos, de 256 colores o menos. Este es un detalle muy importante, puesto que cuantos menos colores utilicemos en la imagen, por lo general, menos ocupará el archivo. En ocasiones, aunque utilicemos menos colores en un gráfico, este no pierde mucho en calidad, llegando a ser inapreciable a la vista.

En algunos programas podemos modificar la cantidad de colores al guardar el archivo, en otros lo hacemos mientras creamos el gráfico.

En esta imagen, tomada con distintas paletas de colores, se puede apreciar cómo con pocos colores se ve bien el gráfico y como pierde un poco a medida que le restamos colores.

Formato JPG

Compresión:
Cuenta con un algoritmo de compresión casi ideal. Este formato, es ideal para fotografías. Además, con JPG podemos definir la calidad de la imagen, con calidad baja el fichero ocupará menos, y viceversa.

Colores:
JPG trabaja siempre con 16 millones de colores, ideal para fotografías.

Optimizar ficheros:
Para que las imágenes ocupen lo menos posible y se transfieran rápidamente por la Red debemos aprender a optimizar los ficheros gráficos. Para ello debemos hacer lo siguiente:

Para los archivos GIF y PNG:
Reduciremos el número de colores de nuestra paleta. Esto se hace con nuestro editor gráfico, en muchos casos podremos hacerlo al guardar el archivo.

Para los archivos JPG:
Ajustaremos la calidad del archivo cuando lo estemos guardando. Este formato nos permite bajar mucho la calidad de la imagen sin que esta pierda mucho en su aspecto visual.

Es imprescindible disponer para optimizar la imagen de una herramienta buena que nos permita configurar estas características de la imagen con libertad y fácilmente. Hay muchas herramientas de uso libre que incorporan una opción de "Exportar para Web" con la que podemos definir los colores del gif, calidad del JPG y PNG u otras opciones en varias muestras a la vez. Así con todas las opciones disponibles podemos optimizar la imagen de una manera precisa con los resultados que deseamos.

Qué es el formato PNG

El formato PNG (Portable Network Graphics) es un formato de archivos de gráficos de mapa de bits (una trama). Fue desarrollado en 1995 como una alternativa gratuita al formato GIF, cuyos derechos pertenecen a Unisys (propietario del algoritmo de compresión LZW) y a quien los editores de software deben pagar regalías por usar este formato. Por lo tanto, PGN es un acrónimo recursivo de PNG No es GIF.

Características del formato PNG

El formato PNG permite almacenar imágenes en blanco y negro (una profundidad de color de 16 bits por píxel) y en color real (una profundidad de color de 48 bits por píxel), así como también imágenes indexadas, utilizando una paleta de 256 colores. Además, soporta la transparencia de canal alfa, es decir, la posibilidad de definir 256 niveles de transparencia, mientras que el formato GIF permite que se defina como transparente solo un color de la paleta. También posee una función de entrelazado que permite mostrar la imagen de forma gradual. La compresión que ofrece este formato es (compresión sin pérdida) de 5 a 25 % mejor que la compresión GIF. Por último, PNG almacena información gama de la imagen, que posibilita una corrección de gama y permite que sea independiente del dispositivo de visualización. Los mecanismos de corrección de errores también están almacenados en el archivo para garantizar la integridad.

Características del formato SVG

SVG es la abreviatura en inglés de Scalable Vector Graphics (Gráficos Vectoriales Escalables). Es un estándar usado para la creación y representación de gráficos e imágenes vectoriales en las páginas web y aplicaciones de internet, aunque también se pueden emplear en la computadora offline.

SVG es un lenguaje gráfico que emplea el formato XML.

Las imágenes en este formato a diferencia de las tradicionales se pueden editar usando editores de texto plano como *Visual Studio Code* o hasta con el Bloc de Notas de Windows.

El formato SVG es recomendado por el W3C y es compatible por la mayoría de los navegadores web modernos.

Las imágenes SVG se caracterizan por su pequeño tamaño y por ser imágenes vectoriales.

Son usadas en las páginas cuando se necesita mostrar o representar elementos sencillos que no necesitan de alta resolución como gráficos, diagramas, mapas, logotipos, iconos, texto, etc.

En este artículo compartimos de forma sencilla, de forma tal que pueda ser entendido por todos, como crear las imágenes SVG, como insertarlas en las páginas web o usarlas offline en la computadora.

Atributos básicos para imágenes en HTML.

La etiqueta que utilizaremos para insertar una imagen es IMG (image). Esta etiqueta no posee su cierre correspondiente y en ella hemos de especificar obligatoriamente el paradero de nuestro archivo gráfico mediante el atributo src (source).

La sintaxis queda entonces de la siguiente forma:

<img src="logo.png">

Para expresar el camino, lo haremos de la misma forma que vimos para los enlaces. Las reglas siguen siendo las mismas, lo único que cambia es que, en lugar de una página destino, el destino es un archivo gráfico. En el código anterior estamos enlazando con un archivo con extensión .png, pero podrá ser otro tipo de archivo como .gif o .jpg.

Aparte de este atributo, indispensable obviamente para la visualización de la imagen, la etiqueta IMG nos propone otra serie de atributos de mayor o menor utilidad, que listamos a continuación:

Atributo **alt**

Dentro de las comillas de este atributo colocaremos una brevísima descripción de la imagen. Esta etiqueta no es indispensable pero presenta varias utilidades. La sintaxis te quedaría de esta manera:

<img src="logo.png" alt="Descripción del logo o imagen">

Primeramente, sirve para el posicionamiento de la página en buscadores.

De los atributos alt el buscador puede extraer palabras clave y le ayuda a entender qué función o contenido tiene la imagen, y por lo tanto la página.

Otra utilidad importante la encontramos en determinadas aplicaciones, usadas por personas con discapacidad. Navegadores para ciegos, por ejemplo, no muestran las imágenes y por tanto los alt ofrecen la posibilidad de leerlas. Nunca está de más pensar en crear páginas accesibles.

Listas en HTML

Las posibilidades que nos ofrece el HTML en cuestión de tratamiento de texto son realmente notables. No se limitan a lo visto hasta ahora, sino que van más lejos todavía. Varios ejemplos de ello son las listas, que sirven para enumerar y definir elementos, los textos preformateados y las cabeceras o títulos.

Las listas originalmente están pensadas para citar, numerar y definir cosas a través de características, o al menos así lo hacemos en la escritura de textos. Sin embargo, las listas finalmente se utilizan para mucho más que enumerar una serie de puntos, en realidad son un recurso muy interesante para poder maquetar elementos diversos, como barras de navegación, pestañas etc.

Pero esto lo veremos más adelante, cuando apliquemos estilos CSS a las listas.

Podemos distinguir tres tipos de listas HTML:

- Listas desordenadas
- Listas ordenadas
- Listas de definición

Las veremos detenidamente una a una.

Listas desordenadas

Están delimitadas por las etiquetas **
** y su cierre (unordered list).

Cada uno de los elementos de la lista es citado por medio de una etiqueta **
**.

Un ejemplo es:

```
<p>Lenguajes de programación</p>
<ul>
    <li>PHP</li>
    <li>JAVA</li>
    <li>Python</li>
</ul>
```

Lenguajes de programación

- PHP
- JAVA
- Python

Listas ordenadas

Las listas ordenadas sirven también para presentar información, en diversos elementos o ítems, con la particularidad que éstos estarán precedidos de un número o una letra para enumerarlos, siempre por un orden.

Para realizar las listas ordenadas usaremos las etiquetas OL (ordered list) y su cierre. Cada elemento será igualmente indicado por la etiqueta LI, que ya vimos en las listas desordenadas.

Pongamos un ejemplo:

```
<p>Reglas de convivencia</p>
<ol>
    <li>No hacer ruidos molestos</li>
    <li>Ser amable con los vecinos</li>
</ol>
```

Reglas de convivencia

1. No hacer ruidos molestos
2. Ser amable con los vecinos

Listas de definición

Las listas de definición sirven para hacer un conjunto de elementos con pares concepto-descripción.

Es decir, se especificarán varios términos por su nombre y se escribirá una definición para cada uno.

Cada elemento es presentado junto con su definición, uno detrás de otro. Para realizar una lista de definición, la etiqueta principal es DL y su cierre (definition list). La etiquetas del elemento y su definición son DT (definition term) y DD (definition definition) respectivamente.

Aquí un código de ejemplo y su resultado:

```
<p>Alimentos Lácteos</p>
<dl>
    <dt>Leche</dt>
        <dd>descremada</dd>
        <dd>entera</dd>
    <dt>Queso</dt>
        <dd>Fontina</dd>
        <dd>Muzzarella</dd>
</dl>
```

Alimentos Lácteos

Leche
    descremada
    entera
Queso
    Fontina
    Muzzarella

Veremos que cada línea DD se ubica más hacia la izquierda. Este tipo de etiquetas son usadas a menudo con el propósito de crear textos más o menos desplazados hacia la izquierda.

Anidando listas

Podemos anidar estas etiquetas obteniendo listas de la manera deseada, como por ejemplo:

```
<p>Ciudades del mundo</p>
<ul>
    <li>Argentina</li>
 <ol>
    <li>Buenos Aires</li>
    <li>Bariloche</li>
 </ol>
    <li>Uruguay</li>
 <ol>
    <li>Montevideo</li>
    <li>Punta del Este</li>
 </ol>
</ul>
```

Ciudades del mundo

- Argentina
    1. Buenos Aires
    2. Bariloche
- Uruguay
    1. Montevideo
    2. Punta del Este

Tablas en HTML

Una tabla es un conjunto de celdas dentro de las cuales podemos incorporar contenidos.

HTML dispone de una gran variedad de etiquetas para crear tablas, con sus atributos, de las cuales veremos una introducción en este artículo. En general, se utilizan para representar información tabulada, en filas y columnas. Esto es una realidad en los últimos años, desde que las tablas se han descartado para fines relacionados con la maquetación.

Nota: Durante un tiempo, gran parte de los diseñadores de páginas basaron su maquetación en este tipo de artilugios.

En efecto, una tabla nos permite organizar y distribuir los espacios de la manera más adecuada. Nos puede ayudar a generar texto en columnas como los periódicos, prefijar los tamaños ocupados por distintas secciones de la página o poner de una manera sencilla un pie de foto a una imagen.

Hablar hoy de tablas como solución para maquetación ha pasado a la historia. Las webs actuales han acabado con técnicas que incrementan el tamaño del código fuente de las páginas web, mezclando presentación y contenido. Actualmente toda la maquetación de una página se organiza con CSS, lo que nos da un mayor control de todos los elementos de la página y la posibilidad de separar todos los estilos para definir el aspecto de una web en un fichero aparte del HTML.

Por ello, en el momento actual las tablas se utilizan mucho menos que en el pasado y realmente la recomendación es usarlas solo en los casos en los que necesitemos incluir en una página información tabulada, es decir, dispuesta en filas y columnas. Todo uso basado en tablas para procurar colocar elementos en determinadas posiciones de la página sería incorrecto en las técnicas actuales de diseño de páginas web.

Como veremos, existen diversas etiquetas que se deben utilizar en una forma determinada para la creación de tablas. Por ello, puede que en un principio nos resulte un poco complicado trabajar con estas estructuras. Si deseamos mostrar datos de una manera sencilla de leer, dispuestos en filas y columnas, tarde o temprano observaremos que las tablas son una rápida solución y apreciamos las posibilidades nos ofrecen, por otro lado, las tablas han dejado de utilizarse debido a su falta de correlación con el desarrollo Responsive, entre otras características.

Etiquetas básicas para tablas en HTML

Para empezar, nada más sencillo que por el principio: las tablas son definidas por las etiquetas **<table></table>** y su cierre.

Dentro de estas dos etiquetas colocaremos todas las otras etiquetas de las tablas, hasta llegar a las celdas. Dentro de las celdas ya es permitido colocar textos e imágenes que darán el contenido a la tabla.

Las tablas son descritas por líneas de arriba a abajo (y luego por columnas de izquierda a derecha).

Cada una de estas líneas, llamada fila, es definida por otra etiqueta y su cierre: **<tr></tr>**

Asimismo, dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otra etiqueta: **<td></td>**. Dentro de ésta y su cierre será donde coloquemos nuestro contenido, el contenido de cada celda.

Ejemplo de estructura de tabla:

```
<table>
    <tr>
        <td>Celda 1, linea 1</td>
        <td> Celda 2, linea 1</td>
    </tr>
    <tr>
        <td> Celda 1, linea 2</td>
        <td> Celda 2, linea 2</td>
    </tr>
</table>
```

El resultado:

Celda 1, linea 1	Celda 2, linea 1
Celda 1, linea 2	Celda 2, linea 2

También es parte de una tabla la etiqueta **<th></th>** (Table Header), que sirve para crear una celda cuyo contenido posea un título o cabecera de la tabla.

Ejemplo:

```
<table>
    <tr>
        <th>Celda 1, linea 1</th>
        <th> Celda 2, linea 1</th>
    </tr>
    <tr>
        <td> Celda 1, linea 2</td>
        <td> Celda 2, linea 2</td>
    </tr>
</table>
```

Atributos para tablas, filas y celdas

A partir de esta idea simple y sencilla, las tablas adquieren otra magnitud cuando les incorporamos toda una lista de atributos aplicados sobre cada tipo de etiquetas que las componen.

cellspacing: es el espacio entre celdas de la tabla.
cellpadding: es el espacio entre el borde de la celda y su contenido.
border: es el número de píxeles que tendrá el borde de la tabla.

Podemos usar prácticamente cualquier tipo de etiqueta dentro de la etiqueta **<td></td>** para, de esta forma, escribir su contenido.

Las etiquetas situadas en el interior de la celda no modifican el resto del documento.

Las etiquetas de fuera de la celda no son tenidas en cuenta por ésta. Podemos especificar el formato de nuestras celdas a partir de etiquetas introducidas en su interior o mediante atributos colocados dentro de la etiqueta de celda **<td></td>** o bien, dentro de la etiqueta **<tr></tr>**, si deseamos que el atributo sea válido para toda la línea.

La forma más útil y actual de dar forma a las celdas es a partir de CSS que veremos más adelante.

Etiqueta Iframe

Los frames (en inglés = marcos) son herramientas que han pasado a ser soportada por todos los navegadores y formar parte de las especificaciones de HTML, para luego retirarse de nuevo del estándar en HTML5.

No obstante, los frames han permanecido en uso y dentro del estándar con la etiqueta IFRAME que vamos a ver en este artículo, que todavía hoy tiene mucha utilidad.

En concreto iframe sirve para crear un espacio dentro de la página donde se puede incrustar otra web. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia página, incluidas por los atributos width y height en la propia etiqueta **<iframe></iframe>**.

El iframe tiene asociada una página web, que se carga en el espacio y operará de manera totalmente independiente. Esa página web tendrá sus propios contenidos y estilos. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al iframe.

<iframe></iframe> se utiliza en muchos contextos. Dentro de un iframe podemos mostrar contenidos de otras páginas, como si estuvieran en la nuestra, por lo que sirven para ejemplos como:

- Visualizar contenidos de terceros, como bloques de noticias o novedades que ofrecen en

otras webs.

- Interfaces de usuario, en el que ciertas actividades se realizan de forma autónoma y el procesamiento está en otra página web.

- Incrustar videos desde YouTube
- Incrustar mapas de Google Maps
- banners de publicidad desde otro sitio Construcción de la etiqueta iframe

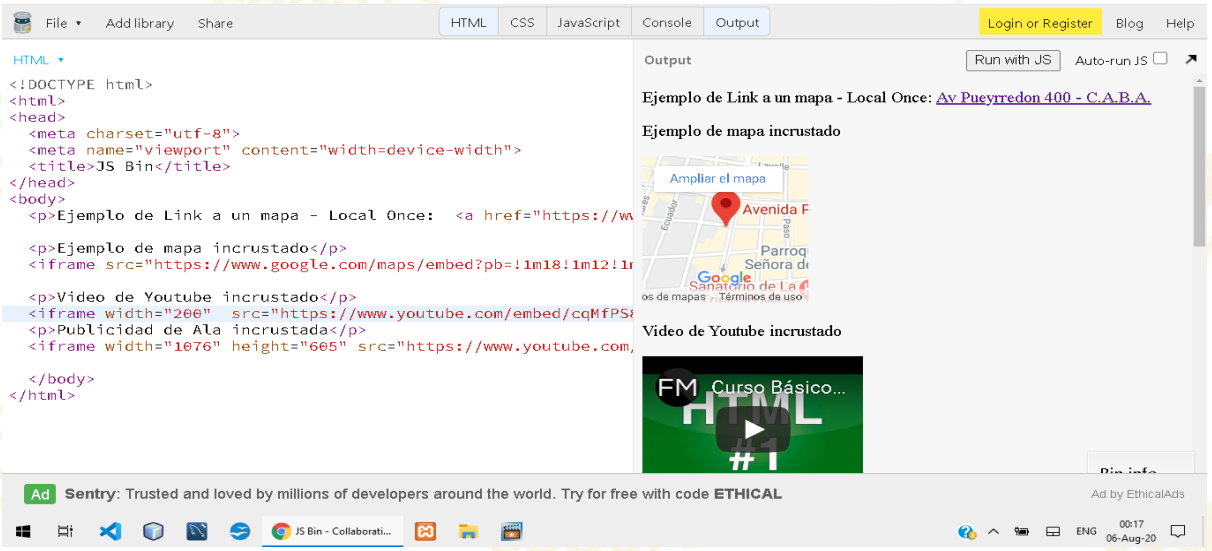
Como decimos, el iframe se coloca directamente en el código HTML, en el lugar donde queremos que aparezca.

Se coloca con un código como este:

<iframe src="pagina_fuente.html" width=290 height=250>Texto para cuando el navegador no conoce la etiqueta iframe</iframe>

Los atributos principales de iframe son la página web que se va a mostrar en el espacio y el ancho y alto del recuadro que reservemos para el frame flotante.

La etiqueta iframe tiene su correspondiente etiqueta de cierre. Todo el texto que coloquemos entre la etiqueta de inicio y la de cierre es texto alternativo, que sólo se mostrará en caso que el navegador del visitante no acepte la etiqueta iframe.



Formularios HTML

Hasta ahora hemos visto la forma en la que el HTML gestiona y muestra la información, esencialmente mediante texto, imágenes y enlaces. Nos queda por ver de qué forma podemos intercambiar información con el usuario. Desde luego, este nuevo aspecto resulta primordial para gran cantidad de acciones que se pueden llevar a cabo mediante el Web:

Comprar un artículo, rellenar una encuesta, enviar un comentario al autor.

Hemos visto anteriormente que podíamos, mediante los enlaces a direcciones de email, contactar directamente con un correo electrónico. Sin embargo, esta opción puede resultar en algunos casos poco versátil, si lo que deseamos es que el navegante nos envíe una información bien precisa y además requiere que el visitante tenga instalado en su ordenador algún correo electrónico en un programa como Outlook Express. Es por ello que el HTML propone otra solución mucho más amplia:

Los formularios.

Los formularios son cajas de texto y botones que podemos encontrar en muchas páginas web y se utilizan para realizar búsquedas o bien para introducir datos personales o claves de acceso. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Qué se puede hacer con un formulario

Usando HTML podemos enviar el contenido del formulario a un correo electrónico, es decir, construir un formulario con diversos campos y, a la hora pulsar el botón de enviar, generar una ventana de redacción de un email con los datos que el usuario haya escrito en cada uno de esos campos.

A menudo desearemos hacer cosas más complejas con los formularios y para ello tendremos que procesar el formulario mediante un programa. Entonces, tendremos que emplear otros lenguajes más sofisticados que el propio HTML.

En este caso, la solución más sencilla es utilizar los programas prediseñados que nos ofrecen un gran número de servidores de alojamiento y que nos permiten almacenar y procesar los datos en forma de archivos u otros formatos. Si tenemos una web alojada en un servidor que no nos propone este tipo de ventajas, siempre es posible recurrir a servidores de terceros que ofrecen este u otro tipo de servicios gratuitos para webs. Por supuesto, existe otra alternativa que es la de aprender lenguajes como JavaScript o PHP que nos permitirán, entre otras cosas, el tratamiento de formularios.

Así pues, en resumen, con HTML podremos construir los formularios, con diversos tipos de campos, como cajas de texto, botones de radio, cajas de selección, menús desplegables, etc. Sin embargo, debe quedar claro que desde HTML no se puede procesar la información, sino que deberemos contar con las herramientas de programación de Back-End que también veremos en este curso. Entonces, si deseamos que el formulario se envíe automáticamente o se procese en el servidor para generar otro tipo de respuesta, necesitaremos lenguajes de programación. Por el momento, estamos viendo Front-End por lo que nos limitaremos a explicar la creación de formularios y estudiaremos próximamente como procesar esa información.

Cómo hacer un formulario en HTML

Los formularios son definidos por medio de las etiquetas **<form></form>** y su cierre. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta **<form></form>** debemos especificar algunos atributos:

action: define el tipo de acción a llevar a cabo con el formulario.

Si lo que queremos es que el formulario sea procesado por un programa, hemos de especificar la dirección del archivo que contiene dicho programa. La etiqueta quedaría en este caso de la siguiente forma:

<form action="dirección del archivo" ...>

La forma en la que se expresa la localización del archivo que contiene el programa es la misma que la vista para los enlaces.

method: Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar esta atributo son post y get. A efectos prácticos y, hasta tanto no veamos Back-End utilizaremos el valor post.

enctype: Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario.

Elementos de Formularios.

Campos de texto

El lenguaje HTML nos propone una gran diversidad de alternativas a la hora de crear nuestros formularios, es decir, una gran variedad de elementos para diferentes propósitos. Estas van desde la clásica caja de texto, hasta la lista de opciones en un menú desplegable, pasando por las cajas de validación, etc.

Etiqueta **<input/>** para texto corto

Las cajas de texto son colocadas por medio de la etiqueta **<input/>**. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: *type* y *name*.
La etiqueta tendrá la siguiente forma:

<input type="text" name="nombre">
De este modo expresamos nuestro deseo de crear una caja de texto cuyo contenido será llamado "nombre" (por ejemplo, en el caso de la etiqueta anterior, pero podemos poner distintos nombres a cada uno de los campos de texto que habrán en los formularios).

El nombre del elemento del formulario es de gran importancia para poder identificarlo en nuestro programa de procesamiento. Por otra parte, es importante indicar el atributo type, ya que, como veremos, existen otras modalidades de elementos de formulario que usan esta misma etiqueta **<input/>**.

El empleo de estas cajas está fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta.

Además de estos dos atributos, esenciales para el correcto funcionamiento de nuestra etiqueta, existen otra serie de atributos que pueden resultar de utilidad pero que no son obligatorios:

size: define el tamaño de la caja de texto, en número de caracteres visibles. Si al escribir el usuario llega al final de la caja, el texto que escriba a continuación también cabrá dentro del campo pero irá desfilando, a medida que se escribe, haciendo desaparecer la parte de texto que queda a la izquierda.

maxlength: indica el tamaño máximo del texto, en número de caracteres, que puede ser escrito en el campo. En caso que el campo de texto tenga definido el atributo maxlength, el navegador no permitirá escribir más caracteres en ese campo que los que hayamos indicado.

Nota: Es importante no confundir el atributo **maxlength** con el atributo **size**. Mientras size define el tamaño visible de la caja de texto, maxlength indica el tamaño máximo real del texto que se puede escribir. Podemos tener una caja de texto con un tamaño aparente (size) que es menor que el tamaño máximo (maxlength). Lo que ocurrirá en este caso es que, al escribir, si sobrepasamos el espacio marcado por size, el texto irá desfilando dentro de la caja hasta que lleguemos a su tamaño máximo definido por maxlength, momento en el cual nos será imposible continuar escribiendo.

value: en algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo value.

Etiqueta **<input/>**, modalidad de texto oculto

Hay determinados casos en los que podemos desear esconder el texto escrito en el campo **<input/>**, por medio asteriscos, de manera que aporte una cierta confidencialidad. Este tipo de campos son análogos a los de texto, con una sola diferencia: reemplazamos el atributo type="text" por type="password":

<input type="password" name="nombre">

En este caso, al escribir dentro del campo, en lugar de texto se desplegarán asteriscos.

Etiqueta **<textarea />**

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta:

<textarea><textarea />

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre, teléfono, edad o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. en los que existe la posibilidad que el visitante desee rellenar varias líneas.

Dentro de la etiqueta textarea deberemos indicar, como para el caso visto anteriormente, el atributo name para asociar el contenido a un nombre que será asemejado a una variable en los programas de proceso. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

rows: define el número de líneas del campo de texto.
cols: define el número de columnas del campo de texto.

La etiqueta queda por tanto de esta forma:

<textarea name="comentario" rows="10" cols="40"></textarea>

Asimismo, es posible predefinir el contenido del campo. Para ello, no usaremos el atributo value, sino que escribiremos dentro de la etiqueta el contenido que deseamos atribuirle:

<textarea name="comentario" rows="10" cols="40">Escribe tu comentario....</textarea>

Otros elementos de formulario

Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta **<select></select>**.

Podemos ver, a partir de estas directivas, la forma más típica y sencilla de esta etiqueta:

```
<select name="estacion">
    <option>Primavera</option>
    <option>Verano</option>
    <option>Otoño</option>
    <option>Invierno</option>
</select>
```

Esta estructura puede verse modificada principalmente a partir de otros dos atributos:

size: Indica el número de valores mostrados a la vez en la lista. Lo típico es que no se incluya ningún valor en el atributo size, en ese caso tendremos un campo de opciones desplegable, pero si indicamos size aparecerá un campo donde veremos las opciones definidas por size y el resto podrán ser vistos por medio de la barra lateral de desplazamiento.

multiple: Permite la selección de más varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas ctrl o mayúsculas (la flecha hacia arriba, también llamada shift). Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual. Simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

La etiqueta **<option></option>** puede asimismo ser matizada por medio de otros atributos.

selected: del mismo modo que multiple, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta está elegida por defecto.

Así, si cambiamos la línea del código anterior:

<option>Otoño</option>

por:

<option selected>Otoño</option>

El resultado será que Otoño se despliegue como la opción inicial.

value: Define el valor de la opción que será enviado al programa si el usuario elige esa opción. Este atributo puede resultar muy útil si el formulario es enviado a un programa para su procesamiento, puesto que a cada opción se le puede asociar un número o letra, lo cual es más fácilmente manipulable que una palabra o texto. podríamos así escribir líneas del tipo:

<option value="1">Primavera</option>

De este modo, si el usuario elige primavera, lo que le llegara al programa es una variable llamada estación que tendrá con valor 1 (estacion=1).

Botones de radio

Existe otra alternativa para plantear una elección, en este caso, obligamos al internauta a elegir únicamente una de las opciones que se le proponen.

La etiqueta empleada en este caso es INPUT en la cual tendremos el atributo type ha de tomar el valor radio. Veamos un ejemplo:

```
<input type="radio" name="estacion" value="1">Primavera
<input type="radio" name="estacion" value="2">Verano
<input type="radio" name="estacion" value="3">Otoño
<input type="radio" name="estacion" value="4">Invierno
```

Nota: Hay que fijarse que la etiqueta **<input type="radio" />** sólo coloca la casilla pinchable en la página. Los textos que aparecen al lado, así como los saltos de línea los colocamos con el correspondiente texto en el código de la página y las etiquetas HTML que necesitemos.

Como puede verse, a cada una de las opciones se le atribuye una etiqueta input dentro de la cual asignamos el mismo nombre (name) para todas las opciones y un valor (value) distinto. Si el usuario elige supuestamente Otoño, recibiremos en nuestro correo: estacion=3.

Cabe señalar que es posible preseleccionar por defecto una de las opciones. Esto puede ser conseguido por medio del atributo checked:

<input type="radio" name="estacion" value="2" checked>Verano

Cajas de validación

Este tipo de elementos pueden ser activados o desactivados por el visitante por un simple clic sobre la caja en cuestión. La sintaxis utilizada es muy similar a las vistas anteriormente:

<input type="checkbox" name="jamon">Me gusta el jamón.

La única diferencia fundamental es el valor adoptado por el atributo type.

Del mismo modo que para los botones de radio, podemos activar la caja por medio del atributo checked.

El tipo de información que llegara a nuestro correo (o al programa) será del tipo: jamon=on (u off dependiendo si ha sido activada o no).

Envío, borrado y demás en formularios HTML

Siguiendo con la explicación de todo lo relativo a formularios que estamos ofreciendo en el Manual de HTML, ha llegado el momento de explicar cómo podemos hacer un botón para provocar el envío del formulario, entre otras cosas.

Como podremos imaginarnos, en formularios no solamente habrá elementos o campos donde solicitar información del usuario, sino también habrá que implementar otra serie de funciones.

Concretamente, han de permitirnos su envío mediante un botón. También puede resultar práctico poder proponer un botón de borrado o bien acompañar el formulario de datos ocultos que puedan ayudarnos en su procesamiento.

En este capítulo, para terminar la saga de formularios, daremos a conocer los medios de instalar todas estas funciones y acabaremos mostrando un ejemplo de formulario completo.

Botón de envío de formulario (botón de submit)

Para dar por finalizado el proceso de relleno del formulario y hacerlo llegar a su gestor, el navegante ha de enviarlo por medio de un botón previsto a tal efecto. La construcción un botón se realiza con las etiquetas **<input/>** ya vistas:

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

Como puede verse, tan solo hemos de especificar que se trata de un botón de envío (type="submit") y hemos de definir el mensaje que queremos que aparezca escrito en el botón por medio del atributo value. Este tipo de campos, **<input/>**, para envío de formularios, a menudo se conocen simplemente como "botones de submit".

Una vez enviada la información, ello es necesario realizar algo de programación, aparte del propio formulario en HTML, en un lenguaje avanzado, que sea del lado del servidor, como PHP, ASP, etc, para su procesamiento.

Botón de borrado (botón de reset)

Este botón nos permitirá borrar el formulario por completo, en el caso de que el usuario desee rehacerlo desde el principio. Su estructura sintáctica es análoga a la anterior:

<input type="reset" value="Borrar">

A diferencia del botón de envío, indispensable en cualquier formulario, el botón de borrado resulta meramente optativo y no es utilizado frecuentemente. Hay que tener cuidado de no ponerlo muy cerca del botón de envío y de distinguir claramente el uno del otro, para que ningún usuario borre el contenido del formulario que acaba de rellenar por error.

Datos ocultos (campos hidden)

En algunos casos, aparte de los propios datos rellenados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario.

Este tipo de datos, que no se muestran en la página pero sí pueden ser detectados solicitando el código fuente, no son frecuentemente utilizados por páginas construidas en HTML, son más bien usados por páginas que emplean tecnologías de servidor.

He aquí un ejemplo:

<input type=hidden name="clave" value="1234">

Esta etiqueta, incluída dentro de nuestro formulario, enviará un dato adicional al programa encargado de la gestión del formulario. podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo (el usuario actual, por ejemplo).

Botones normales

Dentro de los formularios también podemos colocar botones normales, pulsables como cualquier otro botón. Igual que ocurre con los campos hidden, estos botones por sí solos no tienen mucha utilidad pero podemos necesitarlos para realizar acciones en el futuro. Su sintaxis es la siguiente.

<input type=button value="Texto escrito en el botón">

El uso más frecuente de un botón es en la programación en el cliente. Utilizando lenguajes como Javascript podemos definir acciones a tomar cuando un visitante pulse el botón de una página web, esta acción genera lo que llamamos un “evento click” sobre el botón presionado.