6 ORIGEN Y EVOLUCIÓN DE HTML

Cuando hablamos del origen de **HTML**, nos remontamos al año 1980, momento en el que elfísico **Tim Bernés-Lee**, que trabajaba para **CERN** (Organización Europea para la Investigación Nuclear), propone un nuevo sistema de "**hipertexto**" para compartir diferentes documentos.

Tim Bernés-Lee.

Este sistema ya se había desarrollado con anterioridad, aunque en el ámbito de la informática. En ese caso, «**hipertexto**», tiene que ver con el acceso de los usuarios a información que relacionada con aquellos documentos electrónicos que están

visibles. Así, los "**hipertextos**" iniciales se asimilaban siempre a los **enlaces** a distintas páginas web.

Años posteriores, **Tim Berners-Lee** se une al ingeniero de sistemas **Robert Cailliau** y juntos ganan la propuesta

World – Wide Web (W3).

- En **1991** se presenta el primer documento con descripción ***HTML*** bajo el nombre de: ***HTML Tags*** (Etiquetas HTML).

- En **1993** aparece la primera propuesta de carácter oficial para **convertir HTML en un estándar**.

Aunque se dieron avances muy significativos (se definieron las etiquetas de imágenes, las tablas y los formularios), no se llegó a conseguir el estándar oficial.

- Ya en **1995**, es el organismo IETF el que se encarga de poner en marcha un grupo para trabajar con HTML y es cuando se consigue publicar, el 22 de septiembre de 1995, el estándar > **HTML 3.0**: primer estándar oficial de HTML.

- A partir del año **1996**, los diferentes estándares de HTML los publica otro organismo distinto denominado > W3C (World Wide Web Consortium). Aparece entonces la versión **HTML 3.2**, el 14 de enero de 1.997. Esta fue la primera recomendación de HTML que publicó W3C.

- Con la versión **HTML 4.0**, publicada el 24 de diciembre de **1999**, se consiguieron numerosos avances sobre las versiones anteriores, entre ellos la posibilidad de añadir pequeños programas (*scripts*) en las páginas web. Este hecho mejoraba la accesibilidad a las páginas que ya estaban diseñadas, facilitaba trabajar mediante la utilización de tablas más compleas y perfeccionaba los formularios.

- La publicación de **HTML 4.01**, se centró sobre todo en revisar publicaciones anteriores, pero no añadía novedades significativas. Detuvo un poco el desarrollo de HTML para centrarse más en el estándar XHTML.

- Sobre el año 2004, y debido al parón anterior, algunas empresas como Apple, Mozilla y Opera empiezan a mostrar su preocupación por la falta de interés del W3C hacia HTML y es entonces cuando comienzan a organizar una nueva asociación denominada **WHATWG** (*Web Hypertext Application Technology Working Group*).

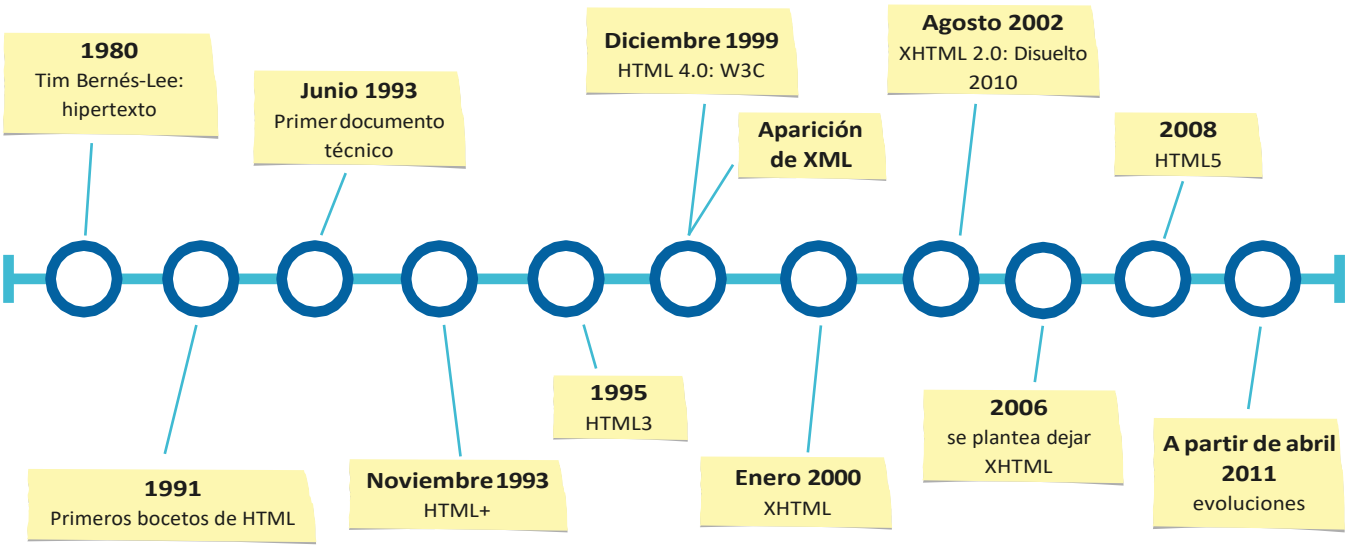
- El 22 de enero de **2008** se publica el primer borrador oficial del estándar **HTML5.**
- De forma paralela, se sigue avanzando con la **estandarización de XHTML**

(versión avanzada de **HTML** basada en XML), cuya primera versión había aparecido en enero del 2000.

- **XHTML 1.0** está basado en la adaptación de HTML 4.01 al lenguaje XML por lo que utiliza sus mismas etiquetas y muchas de sus características, aunque añade algunas nuevas.

- Las versiones **XHTML 1.1** y **XHTML 2.0** se publicaron en forma de borrador con la intención de modularizar **XHTML**.

- En 2010, se decidió no seguir evolucionando el lenguaje **XHTML**.



- **1980**
  Sistema de «hipertexto» para compartir documentos
- **1991**
  Primeros bocetos de HTML
- **Junio 1993**
  https://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt
- **Noviembre 1993**
  https://www.w3.org/MarkUp/HTMLPlus/htmlplus_1.html
- **1995**
  https://www.w3.org/MarkUp/html3/
- **Aparición de XML**
  Aparición de XML
- **Diciembre 1999**
  HTML 4.0:W3C
- **Enero 2000**
  https://www.w3.org/TR/2000/REC-xhtml1-20000126/
- **Agosto 2002**
  Borrador de la evolución
- **2006**
  Evolución de HTML
- **2008**
  HTML5
- **A partir de abril 2011**
  Evoluciones

Una vez conocido el origen y la evolución de este lenguaje de marcas, nos pararemos un poco a conocerlo más a fondo.

HTML es un lenguaje de marcas que nos permite desarrollar diferentes páginas web. Para ello, necesitamos:

Editor de textos ASCII, **Navegador web** **Editor HTML**

---

a) Un **editor de textos ASCII**, mediante el cual vamos a poder añadir el contenido que pretendemos mostrar.

---

b) Un **navegador web** con el que podemos visualiz

Todos aquellos ficheros que contengan documentos
o .***HTM***.

c) Otro **editor HTML**, aunque un poco más específico, puede ser **WYSIWYG** (*what you see is what you get*, que se traduce como: "Lo que ves es lo que obtienes").

De esta forma, podemos escribir diferentes documentos **HTML** y ver, simultáneamente, cómo quedaría el resultado final de la página web, tal y como se vería cuando la publicásemos en internet.

También existen otras herramientas de edición que podemos utilizar para llevar a cabo esta tarea de forma más profesional, como pueden ser, entre otras:

Notepad++, **Atom**, **Dreamweaver** y **Sublime**.

Adobe Dreamweaver.

Esta serie de editores utilizan diferentes menús e iconos en los que podemos añadir:

1. Algunas **ETIQUETAS** (directivas) de **HTML** sin que las tengamos que teclear.
2. REGLAS estilo **CSS**.
3. Diferentes **FUNCIONES** destinadas a la creación y mantenimiento de la **página web.**

EL SERVIDOR WEB

Cuando tengamos la página web lista para su posterior publicación en internet, vamos a necesitar un SERVIDOR en el que poder almacenarlas.

El servidor web es un software que se encuentra en el propio ordenador y que debe estar conectado siempre a internet.

¡RECUERDA!

Cuando pongamos las páginas en el servidor, se hacen accesibles a todos los usuarios que pertenezcan a la misma red.

Existen PROVEEDORES DE SERVICIOS de internet que ofrecen a sus clients

- sitios webs gratuitos para que pueden publicar sus
  - páginas web personales o corporativas y que, de esta forma, eviten instalar un servidor web propio.

6.1. Definiciones y estándares SGML

Veamos una serie de definiciones básicas para poder acercarnos un poco a estos lenguajes.

La definición que más se acerca a la hora de definir el **HTML** (*hyper text markup language*) es lenguaje de marcado de cifrado, utilizado para conseguir establecer un límite sobre los diferentes elementos del lenguaje.

Permite describir el contenido y la estructura de las páginas web y, a su vez, que sean interpretadas y/o visualizadas haciendo uso de los navegadores de internet: **Firefox**, **Internet Explorer**, **Chrome**, etcétera.

Gracias al lenguaje **HTML** podemos publicar una serie de documentos en línea con **encabezados**, **textos**, **listas**, tablas, **foros**, etc.; así como recuperar información novedosa mediante enlaces de hipertexto (con un solo clic).

También posibilita diseñar interfaces con formularios, que los usuarios puedan rellenar con sus datos. Sin embargo, **HTML** no permite modificar los datos que se introducen en los formularios. Si necesitamos hacerlo, debemos hacer uso de algunos lenguajes de programación específicos como son, entre otros, **PHP** y **JSP**.

Otra definición bastante importante es la de SGML

(**s***tandard* **g***eneralized* **m***arkup* **l***anguage*),

tecnología estándar que utilizamos para definir aquellos lenguajes de marcado generalizado que cuentan con la posibilidad de organizar, mediante etiquetas, los distintos documentos.

A partir de **SGML**, se creó posteriormente **HTML** con una versión más simplificada.

Este estándar está basado, principalmente, en dos postulados:

A. El marcado tiene que ser **declarativo**, es decir:

es conveniente que describa la estructura que tiene un documento junto con otros atributos.

B. El marcado tiene que ser **riguroso**, esto es:

permite que otros programas procesen los documentos.

A la hora de definir documentos, utilizamos **DTD** (*document type definition*), que ofrece la posibilidad al navegador de conocer tipo de documento que hay que interpretar.

Una **UDT** es un documento que está basado en SGML y que cuenta con reglas sintácticas para definir un tipo de documento determinado.

Posee elementos permitidos, junto con sus atributos, además de otras funciones sobre la anidación y los valores de estos. Si comprobamos un documento con su correspondiente UDT y la cumple, verifica- remos que es válido. En caso contrario, que es fallido.

A continuación, vamos a centrarnos en la definición de **XHTML** (**e***xtensible* **h***ypertext* **m***arkup* **l***anguage*).

En este caso, nos encontramos con un lenguaje que está considerado como

una variante, bastante más restrictiva y ordenada, de HTML y que se lleva a cabo

mediante la utilización de la propia sintaxis de XML (*extensible markup language*).

En sus comienzos, fue considerado como el sustituto de HTML.

No obstante, cuando se llevó a la práctica, dejó de serlo porque se dieron una serie de diferencias entre los distintos fabricantes.

Fue HTML el que siguió en un continuo crecimiento.

Aunque XML cuenta casi con los mismos elementos que HTML, debemos señalar que su sintaxis es diferente, ya que este último se basa en una composición de elementos más coherente que HTML. Para ello, trabaja separando el contenido de un documento de su formato, sin alterar su contenido.

6.2. VERSIONES HTML

Como consecuencia del crecimiento tan rápido de la web y de la evolución de las versiones HTML, aparece la necesidad de estandarizarlo para que reconocer el tipo de versión a utilizar.

HTML llegó a convertirse en estándar en 1995 y, con el paso del tiempo, ha entrado en un desarrollo constante. Hace algunos años, la versión de HTML recomendada por el W3C era **HTML 4.01.**

Cuando diseñamos una página web, es conveniente especificar la versión de HTML con la que vamos a trabajar, y lo haremos escribiendo en la primera línea de la etiqueta: <***!DOCTYPE***>. Gracias a esta información, el navegador puede interpretar de forma correcta. **HTML 4.01 cuenta con tres variantes** disponibles de **DTD**:

- **HTML 4.01 Strict** (*Strict DTD*).
- **HTML 4.01 Transitional** (*Transitional DTD*).
- **HTML 4.01 Frameset** (Frameset DTD).

- **HTML 4.01 (Strict DTD):** Es la más restrictiva de todas ya que no permite la utilización de etiquetas que estén anticuadas; solo hacer uso de las definidas en HTML 4.01.

PARA + INFO

Si queremos utilizar esta versión, escribimos en la primera línea la siguiente instrucción:

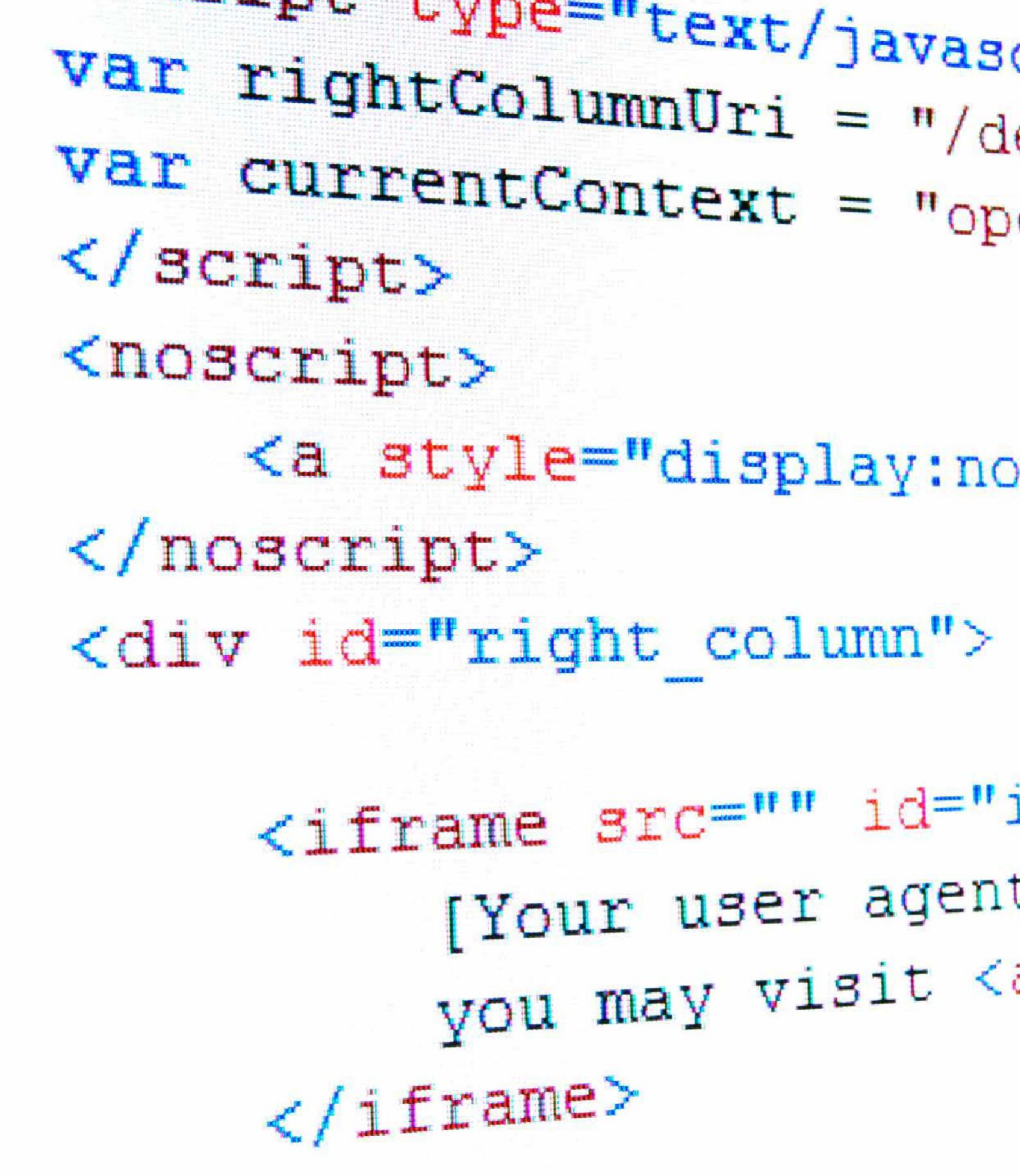
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

http://www.w3.org/TR/html4/strict.dtd>

- **HTML 4.01 Transitional** (*Transitional **DTD***).
- **HTML 4.01 Frameset** (Frameset **DTD**).

La diferencia entre **HTML 4.01** y **HTML5** es que en HTML5, solamente hay que que declarar:

```
<!DOCTYPE html>
```



7

ESTRUCTURA DE UN DOCUMENTO HTML

7.1. IDENTIFICACIÓN SGML

Antes de comenzar a utilizar cualquier lenguaje de marcas es conveniente familiarizarnos con una serie de normas básicas que debemos tener en cuenta como:

• Etiquetas

Las etiquetas, también conocidas como marcas, definen una serie de elementos que forman el léxico del lenguaje HTML. Se encuentran entre los signos de *menor que* (<) y *mayor que* (>). Podemos diferenciar entre dos tipos de etiquetas: **cerradas** y **abiertas**.

– Etiquetas cerradas

Constan de una para la apertura (indica el comienzo de la etiqueta)

y otra para el cierre (

indica que hemos terminado de trabajar con ella y lleva el símbolo «/» antes del nombre).

Por ejemplo:

<*p*>

– Etiquetas abiertas

Cuentan con una única palabra reservada para indicar el inicio y el fin a la vez.

Por ejemplo:

<hr>

En HTML5 NO ES NECESARIO escribir las etiquetas de cierre gracias a los navegadores actuales.

• Atributos

Los **atributos**, al igual que las etiquetas, se pueden definir tanto en mayúsculas como en minúsculas, a pesar de que los *valores* que les asignen sí guardarán diferenciación.

Por ello, es recomendable utilizar siempre **minúsculas** y, de esta forma, evitar confusiones.

Las etiquetas pueden contener atributos si necesitan realizar alguna > configuración sobre alguna característica determinada.

Estos atributos se definen a continuación de la palabra reservada en la etiqueta de apertura separada por un espacio en blanco y antes del signo de cierre.

Asignaremos el valor correspondiente al atributo a través del signo "=".

PARA + INFO

Cada comando cuenta con una serie de atributos con sus correspondientes valores; por ejemplo:

<*p ALIGN = "left"***>**

Definición de un párrafo y alineación del texto a la izquierda.

Es recomandable poner el valor entre dobles comillas para que sea más legible.

Comentarios

Los comentarios son líneas que definen, de cara al usuario, lo que vamos realizando en cada momento, aunque no son interpretadas por el navegador.

De esta forma, si alguien necesita trabajar con el código, es capaz de interpretarlo con un simple vistazo, gracias a los comentarios que aparecen en él.

Los comentarios van escritos entre los símbolos: <!--y-->.

Por ejemplo:

<!- - Esto es un comentario - ->

Estructura básica

Como ya hemos indicado, las etiquetas que utilicemos en HTML siempre van a ir entre los símbolos "**<**" y "**>**". Y, cada vez que tengamos que cerrar una etiqueta, pondremos el nombre correspondiente comenzando con el símbolo "**/**".

Todas las etiquetas afectan al código que se encuentre delimitado entre la apertura y el cierre de la etiqueta.

```
<html>          Inicio del documento
    <head>      Comienzo de la cabecera
    <title>     Inicio del título del documento
    Título
    </title>    Fin del título del documento
    ...


    </head>     Fin de cabecera
    <body>      Inicio del cuerpo
    ...
    </body>     Fin de cuerpo
</html>         Fin del documento
```

HTML5, mediante unas etiquetas nuevas, añade una serie de características y elementos cuya función es facilitar la tarea a los autores de la aplicación web.

HTML5 se basa, principalmente, en una estructuración avanzada que se encarga de definir los contenidos agrupándolos en distintas etiquetas. Estas tienen un nombre asignado según la tarea que se va a realizar:

- **<header>**: define un conjunto de ayudas introductorias en una página.

  Podemos encontrar elementos como:

  - El menú de navegación principal con los enlaces a las secciones de la web.
  - La marca o logotipo de la web.
  - Una pequeña descripción (en ocasiones).
  - El buscador de la página (en ocasiones).

- **<nav>**: define los enlaces de navegación.
- **<article>:** crea algún artículo que se haya publicado.
- **<section>**: parte correspondiente a algún artículo.
- **<aside>:** define contenido lateral de la página.
- **<footer>**: define el pie de página.
- **<dialog>**: define distintos diálogos o comentarios.

Además de estas etiquetas, HTML5 también cuenta con componentes como <***div***> y <***span***>, utilizados para poder agrupar los diferentes elementos "hijos" haciendo uso de atributos como: *class*, *id* o *tittle*. De esta manera, se pretende usar una misma semántica con un estilo común.

7.2. CABECERA

La cabecera del programa se encuentra siempre entre:

<*head***>**

...

<*/head***>**

Lo que se haya, a su vez, dentro de un elemento superior como es <*html*>.

Dentro de la cabecera es donde vamos a definir los elementos generales, como el título de la página:

- **<*title*>**: es el título que va a aparecer en el navegador web, en la barra superior.

- **<*meta*>**: encargada de indicar el contenido de nuestras palabras junto con los términos clave. Esta directiva suele llevar dos atributos (***name*** *y* ***content***) que hacen referencia al nombre de la página y a sus principales contenidos.

<meta name = "description" content = "Página prin- cipal ILERNA con alumnos y profesores">

<meta name = "keywords" content = "Nombre y apellidos de alumnos matriculados">

PARA + INFO

Otro uso diferente de la etiqueta <*meta*> es el "refresco automático": transcurrido un tiempo estimado, la misma página pasa a actualizarse. Es bastante recomendable para aquellas en las que el contenido se modifica con mucha frecuencia.

<*meta http-equiv = "refresh" content = "10"; url = "http://www.google.es"***>**

Con esta instrucción estamos indicando que, pasados 10 segundos, se va a acceder a la página web de Google.

- <***base***>: podemos indicar una URL base de algún documento, sonido, gráfico, etc., que hagan referencia a una determinada página web. ÇCuando creamos una página web, es conveniente que, inicialmente, realicemos una planificación de su diseño para, después, ordenar la información y los recursos que se van a ofrecer.

Para llevar a cabo esta tarea, es recomen- dable que hagamos uso de una estructura de directorios.

- <***style***>: gracias a esta etiqueta podemos definir los estilos de nuestro HTML:

```
<html>
<head>
<style>
h1 {color:red;}
p {color:blue;}
</style>
</head>
<body>

<h1>Titulo</h1>
<p>Párrafo</p>

</body>
</html>
```

Etiquetas que definen metadatos son <*link*> y <*script*>.

7.3. CUERPO DEL DOCUMENTO

El cuerpo del programa se encuentra siempre entre:

<body>

...

</body>

Siempre está situado detrás de la cabecera <*head*>.

Va a contener todo el cuerpo correspodiente a una determinada página web junto con los elementos propios de la página:

- gráficos,
- textos,
- imágenes, etcétera.

En él, tenemos la posibilidad de definir una serie de acciones necesarias para **eventos** más concretos, entre las que podemos destacar:

Onload ***Onunloaunload*** ***Ononline*** ***Onoffline*** ***Onafterprint***: ***Onbeforeprint***:

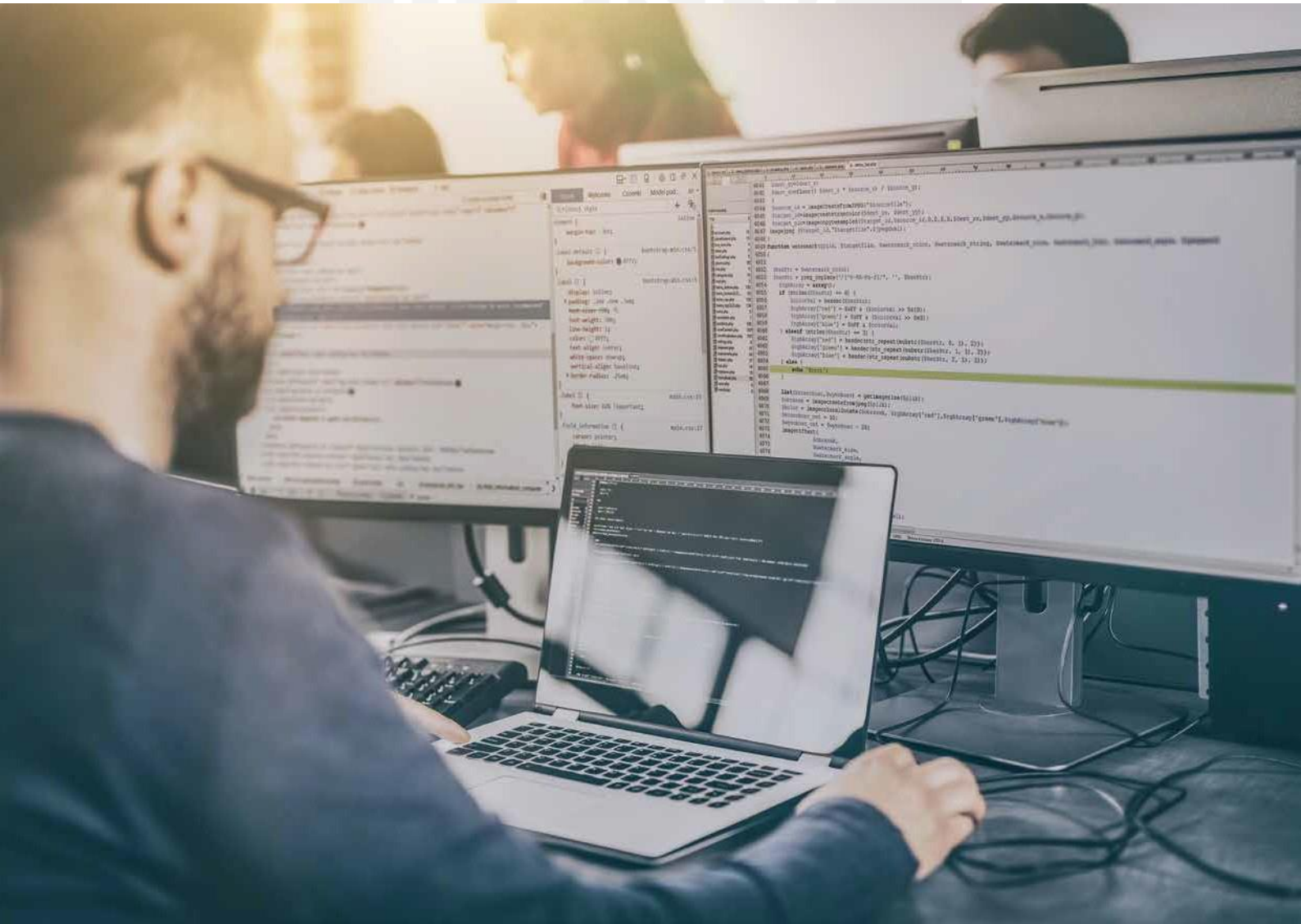
- ***Onload***: ejecuta un evento inmediatamente después de cargar la página.
- ***Onunload***: ejecuta un evento cuando un usuario descar- ga un documento.
- ***Ononline***: ejecuta un evento cuando el navegador se abre y tiene salida a internet.
- ***Onoffline***: ejecuta un evento cuando el navegador se abre y trabaja sin conexión.
- ***Onafterprint***: ejecuta un evento cuando la página ha co- menzado a imprimir o si el diálogo con la impresora ha sido cerrado.
- ***Onbeforeprint***: ejecuta un evento cuando la página está a punto de ser impresa.

Además, existe un conjunto de atributos (opcionales) de <***body***> que permiten realizar diferentes configuraciones sobre la apariencia de un documento.

Hoy en día están obsoletos, ya que HTML5 no los soporta.

Algunos de ellos son:

- *bgcolor*,
- *text*,
- *link*
- *y vlink*.




pleasure seeker
1080
pleasure and don't cost much. | great pleasure a film that has given great pleasure to millions
2 ▶ ENJOYABLE EXPERIENCE ◀ [C,U] an activity or experience that you enjoy very much: the simple pleasures of life | it is a pleasure to do sth It's been a great pleasure to meet you.
3 take pleasure in to be pleased about and proud of something: He took great pleasure in his grandchildren's achievements.
4 take pleasure in doing sth to enjoy doing something bad to other people: Charlie seems to take pleasure in bullying the younger kids.
5 (it's) my pleasure spoken used when someone has thanked you for doing something and you want to say that you were glad to do it
6 with pleasure spoken used to say politely that you are happy to do something that someone has just asked you to do: "Will you come?" "With pleasure madam."
7 have the pleasure of (doing) sth a) formal to enjoy the experience of something: I don't think I've had the pleasure of meeting your wife. b) humorous used when you do not think something is enjoyable: We had the pleasure of Rob's company last week.
(her Majesty's pleasure BrE law if someone is
promise that you will do something
sth They have pledged to fight any changes in
tion laws. pledge that The UK government has
that the wishes of the minority community in
Ireland will be respected. pledge support
solidarity etc (=promise to give your s
2 to make someone give a firm promise
were pledged in advance to vote for the
pledge yourself to sth The Republicans
selves to a tough stand against crime.
thing with someone as a PLEDGE (4)
become a member of a university FRATERN
ORITY in the US
pleis·to·cene /ˈplaɪstəsiːn/ adj belonging to the
in the Earth's history that started about a m
ago and lasted about 800,000 years, when
Earth was covered with ice
ple·na·ry /ˈpliːnəri/ adj [only before noun] (
plenary meeting is one that is attended by
has a right to attend: The conference ended with
debate. 2 plenary power or authority is
has no limit: The envoy was given plenary
tiate with the rebels. —plenary n (C)
plen·i·po·ten·tia·ry /ˌplenɪpəˈtenʃəri, ˌplen
mal or technical someone who has full
action or make decisions, especially as a repre
of their government in a foreign country—plen
tiary adj: plenipotentiary powers
tude /ˈplenɪtjuːd/ -tuːd/ n literary 1
2 a plenitude of sunshine
plod /plɒd/ adv/prep
and great
children were
ded up the hill
steadily, especially
difficult: I'll just plod
plod·der /ˈplɒdə/ plɒdə/
plodder.
plonk¹ /plɒŋk/plɑːŋk, plɔːŋ
something down somewher
lessly: plonk sth in/on
where in my room. 2 p
down heavily and then

8 TEXTO

8.1. BLOQUES DE TEXTO

• Saltos de línea

- **br** Representa un salto de línea.
- **Hr** Representa un salto de párrafo.

• Párrafos

- <*pre*>: devuelve una copia exacta del texto respetando los espacios en blanco, tabulaciones y retorno de carro, es decir, tal y como se ha escrito. </**pre**>
- <*p*>: escribe un texto en forma de párrafo.</*p*>

  <*blockquote*>: permite visualizar una cita con el margen izquierdo mayor, produciéndose un efecto de una sangría </*blockquote*>.

8.2. FORMATO Y ELEMENTOS

• Colores

Podemos representar los colores mediante el símbolo de "#" seguido de tres pares de dígitos hexadecimales. El rango de colores indica la intensidad de los primarios (rojo, verde y azul), que en dígitos hexadecimales son (**#RRVVAA**).

Los diferentes pares de cifras hexadecimales oscilan desde **00** hasta **FF**, proporcionando un rango que va desde(**0** hasta 255)valores diferentes.

También tenemos otra forma de expresar colores en HTML haciendo uso de una notación hexadecimal más corta, que utiliza un dígito para cada color (**#RGB**). En este caso, el rango de valores va a oscilar entre 0 y 15 o, si lo preferimos, podemos indicar el color de forma directa: *red*, *green*, *blue*.

• Cabeceras

Existen seis tipos diferentes de cabeceras (elementos de encabezado):

<h1> </h1>

<h2> </h2>

<h3> </h3>

<h4> </h4>

<h5> </h5>

<h6> </h6>

El texto que se escriba en *h1* va a ser el del título de mayor tamaño, hasta el de *h6*, que va a ser el menor.

Contenido de ejemplo	Visualización
Texto de Prueba	Texto de Prueba
Texto de Prueba	Texto de Prueba
Texto de Prueba	Texto de Prueba
Texto de Prueba	Texto de Prueba
Texto de Prueba	Texto de Prueba
Texto de Prueba	Texto de Prueba

Semántica en textos

Es una serie de elementos HTML que ofrece un significado a una parte del contenido de un texto,

haciendo de la negrita, el subrayado o la cursive

(<**b**> <**/b**>, <u> </u> y <*i*> <*/i*>).

Existen algunas etiquetas, bastante parecidas entre sí, que tienen como fin ofrecer un estilo diferente para alguna letra especial de un texto determinado.

Algunos tipos de los elementos semánticos más utilizados para los distintos textos los vemos en el cuadro que se detalla a continuación.

9 HIPERVÍNCULOS

Los hipervínculos son elementos del lenguaje HTML que permiten acceder a otro recurso; es decir, enlaces que redirigen a otro sitio web, a un fichero, a una imagen, etc.

PARA + INFO

La sintaxis que debemos utilizar a la hora de incluir un hipervínculo es:

<a>

El texto que se encuentre en esa directiva se puede convertir en un hipervínculo.

Si hacemos clic con el ratón, nos debe llevar al sitio referenciado.

En el caso en el que el sitio web esté referenciado por un texto, debe aparecer subrayado y de otro color.

El atributo *href* es el que nos ofrece la posibilidad de crear un hiperenlace. Debemos indicar la URL a la que queremos acceder al hacer clic en el hiperenlace. **A continuación, si en el elemento <a> no indicamos el atributo href, entonces representa un marcador de posición que va a ser a la que referencie otro hipervínculo en su atributo href**.

Disponemos de otro atributo, *target* (opcional), que hace referencia al destino de la información disponible en la dirección a la que nos lleva.

Anclas y vínculos internos

Los vínculos internos permiten acceder a un sitio concreto dentro de una página web. Si queremos hacer uso de los vínculos internos, antes debemos establecer un ancla: el punto fijo de posición al que accederemos tras un vínculo interno.

9.1. Relativos y absolutos

- **Absolutos**

Aquellos que enlazan con páginas, cuya dirección absoluta se indica en el atributo *href* del comando a. Suelen ser páginas web externas a nuestro proyecto.

¡RECUERDA!

La sintaxis que debemos utilizar a la hora de incluir un hipervínculo es:

**<a> **

de tal forma que el texto que se encuentre en esa directiva se puede convertir en un hipervínculo.

Las direcciones absolutas empiezan a direccionarse desde el comienzo de la ruta que indicamos.

- **Relativos**

Aquellos enlaces cuya dirección relativa se indica en el atributo *href* del comando *a*. Suelen ser enlaces a páginas internas del mismo proyecto.

```
<a href = "./pagina2/pagina2.html">
```

Las direcciones relativas empiezan a direccionarse a partir del directorio actual.

10 IMÁGENES

Las imágenes pueden estar en diferentes formatos tales como los mapas de bits (archivos PNG, GIF, JPEG), documentos vectoriales de una página (archivos PDF, XML), mapas y gráficos de bits animados, etc.

No obstante, existen archivos de otros tipos que no se consideran imágenes, como los archivos PDF de varias páginas, los interactivos, los documentos HTML y los que no tienen formato, los archivos SVG...

Gracias al elemento <***img***> podemos representar una determinada imagen, ayudados por su atributo (obligatorio) ***src***. En este caso, indicamos:

- La dirección válida en la que está la imagen que queremos visualizar.
- Una ruta relativa, si es que la imagen está en alguna parte local.
- Una URL si se refiere a una imagen externa que se encuentra almacenada en una página web diferente.

El atributo ***alt*** nos permite que indiquemos un texto alternativo capaz de representar el contenido de una imagen. Podemos utilizar esta forma en el caso en el que el navegador no pueda visualizar o descargar las distintas imágenes.

Por otro lado, las etiquetas <***figure***> y <***figcaption***> son novedosas para HTML5 y nos ofrecen la posibilidad de agrupar una imagen junto con su información o leyenda.

10.1. Incorporación de imágenes

Una vez llegado el momento de añadir imágenes a una página web, debemos contar con que los navegadores trabajan con ficheros en formatos JPEG o GIF, ya que son los más recomendables.

Si queremos que una imagen se muestre en una web, tenemos que, en primer lugar, declarar una etiqueta <***img***> que no necesita etiqueta de cierre. Por ejemplo:

```
<img src=" imagen.jpg" alt="Texto">
```

Aparte de estos atributos *src* y *alt*, también contamos con algunos más que mostramos en la siguiente tabla.

Etiqueta	Atributo	Valor	Significado
Img	arc	URL	Indica la URL de la imagen
	alt	Texto	Define un texto alternativo por si no se encontrara la imagen deseada
	align	Top, middle, bottom, left, right, center	Alinea la imagen respecto al texto, tanto en sentido horizontal como vertical
	border	Número	Pone un borde o marco a la imagen. Se expresa en píxeles
	height	Número %	Especifica la altura de la imagen. Se expresa en píxeles o porcentaje
	width	Número %	Especifica la anchura de la imagen. Se expresa en píxeles o porcentaje
	hspace	Número	Especifica en píxeles la separación horizontal entre el texto y la imagen
	vspace	Número	Especifica en píxeles la separación vertical entre el texto y la imagen

10.2. USO DE MAPAS SENSIBLES

Los mapas de imagen posibilitan la definición de distintas zonas accesibles dentro de una imagen, que pueden dirigir a una URL distinta.

De tal forma que el usuario puede hacer clic sobre ellas.

Estas zonas se van creando mediante

- rectángulos,
- círculos
- y polígonos.

A la hora de crear un mapa de imagen:

1. En primer lugar, insertamos la imagen original mediante la etiqueta <***img***>.

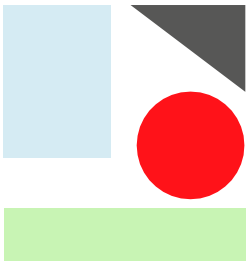
2. A continuación, utilizamos la etiqueta <***map***> para definir las diferentes zonas o regiones de la imagen. Cada una de ellas, se creará a partir de la etiqueta <***area***>.

Veamos un ejemplo práctico de los mapas de bits:

<map>	**Mapa de imagen**
Atributos comunes	**Básicos, i18n y eventos**
Atributos específicos	• **name = "texto": nombre con el que se identifica de forma única el mapa definido (es obligatorio indicar un nombre único)**
Tipo de elemento	**Bloque y en línea**
Descripción	**Se utiliza para definir mapas de imagen**

<area>	**Área de un mapa de imagen**
Atributos comunes	**Básicos, i18n, eventos y foco**
Atributos específicos	• *href* **= "***url***": URL a la que se accede al clicar sobre el área.** • *Nohref* **= "***nohref***": se utiliza para las fórmulas que no son seleccionables.** • *Shape* **= "***default*/*rect*/*circle*/*poly***": indica el tipo de área que se define (toda la imagen, rectangular, circular o poligonal).** • *Coords* **= "***lista de números***": consiste en una lista de números, separados por comas, que representan las coordenadas del área. Rectangular = X1, Y1, X2, Y2... XnYn (coordenadas de los vértices del polígono). Si las últimas coordenadas son diferentes de las primeras, el polígono se cierra de manera automática uniendo ambos vértices.**
Tipo de elemento	**Bloque y en línea**
Descripción	**Se emplea para definir mapas de imagen**

Mediante este círculo, el triángulo y los dos rectángulos, podemos acceder a cuatro zonas diferentes de la imagen con el siguiente código HTML:



```
<img src="imagen.gif" usemap="#mapa_zonas" />
  <map name="mapa_zonas">
      <area shape="rect" coords="20,25,84,113" href="rectangulo.html" />
      <area shape="polygon" coords="90,25,162,26,163,96,89,25,90,24"
      href="triangulo.html"
      <area shape="circle" coords="130,114,29" href="circulo.html" />
      <area shape="rect" coords="19,156,170,211"
      href="mailto:rectangulo@direccion.com" />
      <area shape="default" nohref="nohref" />
  </map>
```


NEWS
Search
Home
World
Business
Finance
Social
Education
Europe
Asia
South America
Africa
Australia
Top 1
o visit once in your life
:o visit once in your life.
All-inclusive resorts are one way to. Like most travelers, I like to stretch my ensure you stick to a planned vacation budget travel budget. I look for discount flights and the best deals on hotels but sometimes it's hard to have it all when you're traveling on a set budget.The Derest is the worlds largest hot desert, coverins
Related Article
Long flight tips for travellers
Long tlight tip1 for travellers
Top 5 places to dino
Tho most stunnlng naturo
Fantastic nightlife in Asia
Ame7.ing cu llura l p erfotmancoa


Don't forget! to check
25
My Sites
Templates
Explore
Subscriptions
Support
<WEB DESIGN>
Start Now!


WEB DESIGN
*Start up
CODE023
DLTOLP

11 TABLAS

Son elementos de **HTML** que nos ofrecen la posibilidad de representar datos de una o varias dimensiones (matriz) con la información distribuida en **filas** y **columnas**.

Uno de los objetivos que pretende HTML5 es conseguir separar el contenido que se pretende mostrar en una página web de la forma en la que lo ha presentado, mediante hojas de estilo **CSS**.

A la hora de definir una tabla, debemos hacerlo haciendo uso de la siguiente directiva:

<**table**>

...

</**table**>

Hasta el momento, y para versiones anteriores a HTML5, se han podido utilizar distintos atri- butos como:

align, ***bgcolor***, ***cellspacing***, ***cellpadding***, ***width***, ***rules*** y ***summary***.

Sin embargo, desde la aparición de HTML5,, solo soporta el atributo ***border*** dentro de la etiqueta <*table*>,

que nos ofrece la información de si las celdas de la tabla deben tener fronteras o no. Según cual sea la respuesta, va a devolver 0 o 1.

Si tenemos una tabla, el primer elemento que podemos representar es el que aparece dentro de la etiqueta:

<**caption**>

Representa el título de la tabla que lo contiene.

11.1. Filas, columnas y celdas

Las tablas se componen de un número de filas y columnas que se pueden representar a través de la directiva:

<tr> ...</tr>

Después, cada fila cuenta con una serie de elementos:

<td> ...</td>

Nos permite ir generando las diferentes columnas con su valor asignado dentro de una determinada fila.

<th> ...</th>

Realiza columnas de cabecera y contiene un texto centrado y en negrita.

Cada dato de una fila y una columna determinada contiene un dato al que denominamos *valor*.

11.2. Combinación de celdas

Es posible combinar un conjunto de celdas de una determinada tabla, hecho que va a afectar a su definición:

<***colgroup***>: representa las columnas de una tabla.

<***col***>: indica el número de columnas que tiene una tabla.

Disponemos de un conjunto de elementos que nos permiten referirnos, en lenguaje HTML, a las diferentes partes de una tabla, como:

<***thead***>: cabecera

<***tbody***>: bloques de filas

<***tfoot***>: pie


SIDEBAR
SOCIAL MEDIA
Your Logo
HOME
NEWS
GALLERY
CONTACT
SIDEBAR
SOCIAL MEDIA
LOGO
SIDEBAR
TEXT

12

LISTAS

12.1. Listas numeradas

Ofrecen la posibilidad de representar los diferentes elementos de una lista enumerándolos en función del lugar que ocupen.

En las listas numeradas, utilizamos las directivas:

```
<ol>
   <li>Apartado 1</li>
   <li>Apartado 2</li>
   ...
</ol>
```

Por ser una lista numerada, se muestra de la siguiente forma:

1. *Apartado 1*
2. *Apartado 2*
3. …

La directiva <**ol**> tiene disponibles los siguientes parámetros:

- ***Start*** = numero: permite seleccionar el primer número de la lista.
- ***Type*** = tipo: nos permite elegir el tipo de numeración.

En HTML5, podemos diferenciar entre los siguientes

tipos:

- **1.** Expresa números desde 1, 2, 3, etc.
- **a.** Expresa letras minúsculas a partir de a, b, c, etc.
- **A.** Expresa letras mayúsculas a partir de A, B, C, etc.
- **i.** Expresa números romanos desde i, ii, iii, etc.
- **I.** Expresa números romanos en mayúscula desde I, II, III, etc.

12.2. Listas no numeradas

Ofrecen la posibilidad de representar los diferentes elementos independientemente del lugar en el que se vayan a almacenar. Lo que sí llevan es una especie de «viñeta» para marcar la lista.

En las listas no numeradas, utilizamos las directivas:

```
<ul>
   <li>Apartado 1</li>
   <li>Apartado 2</li>
   ...
</ul>
```

Por ser una lista numerada, se muestra de la siguiente forma:

- Apartado 1
- Apartado 2
- ...

La directiva <***ul***> tiene disponibles los siguientes parámetros:

- ***Type*** = tipo: nos permite seleccionar el tipo viñetas.

En HTML5, podemos diferenciar entre los siguientes tipos:

- ***Disk***: expresa una viñeta en forma de disco.
- ***Circle***: expresa una viñeta en forma de círculo.
- ***Square***: expresa una viñeta en forma de cuadrado.

12.3. Listas de definición o glosario

Ofrecen la posibilidad de representar los diferentes elementos de un diccionario a partir del término y su definición.

En las listas de definición, utilizamos las directivas.

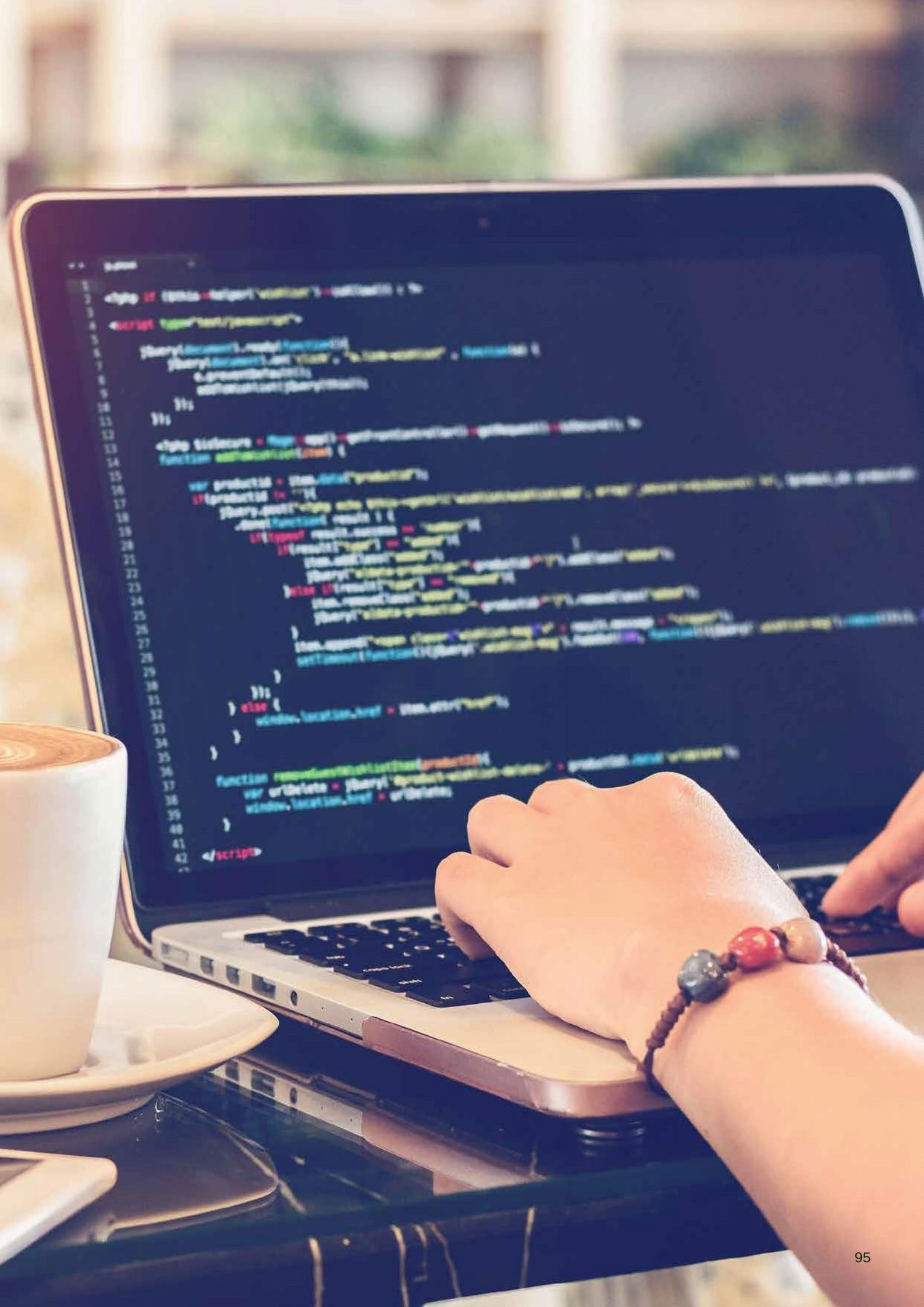
```
<dl>
   <dt>Coche</dt>
      <dd>Vehículo de cuatro ruedas</dd>
   <dt>Moto</dt>
      <dd>Vehículo de tres ruedas</dd>
   ...
</dl>
```

Coch
e

Vehículo de cuatro ruedas

Vehículo de cuatro ruedas

Moto



13 FORMULARIOS

Utilizaremos los formularios cuando necesitamos recoger la información específica de un objeto en cuestión.

Los formularios están compuestos, sobre todo, por diferentes elementos como: cajas de texto (*textBox*), casillas de verificación (*checkbox*), casillas de opción (*radio button*), listas desplegables y subformularios.

Toda la información que recogen debe tratarse por archivos implementados por el propio desarrollador. De esta forma, la almacenaremos en bases de datos diseñadas previamente.

Esta información también puede ser enviada o procesada mediante correo electrónico o servidores web, a través de un botón de envío (*submit*).

13.1. Propiedades de los formularios

Un formulario se define mediante el comando:

<form>

...

</form>

Por tanto, dentro de este bloque, debemos implementar todos los elementos necesarios que hemos enumerado. La marca *form* consta de varios atributos bastante importantes:

- ***action***: indica el lugar al que se envían los datos.
- ***method***: indica le método de transferencia de los datos en el servidor web. Los valores que puede tomar este atributo pueden ser:
  - ***post***, para el envío de los datos al usuario de forma codificada.
  - ***get***, para el envío de los datos a una dirección web.
- ***target***: se utiliza para indicar o especificar dónde vamos a mostrar la respuesta del formulario. Los diferentes valores que puede tomar este atributo son:
  - *_blank*: se muestra en una ventana nueva.
  - *_self*: se muestra en la misma ventana del formulario.
  - *_parent*: se muestra en la ventana padre, es decir, la que precede al formulario.
- ***name***: identifica, mediante un nombre, al formulario. De esta forma, podemos llamarlo desde otro archivo. Recomendamos que el nombre de formulario sea único para facilitar su tratamiento.

13.2. Elementos de los formularios

Dentro de un formulario, podemos tener distintas directivas que van a componer todo el diseño. Para comenzar, podemos agrupar un conjunto de elementos bajo un nombre, mediante el siguiente comando.

<fieldset> ... </fieldset>

A continuación, y a través de las etiquetas:

- ***<input> ... </input>***,

Vamos a poder crear varios tipos de elementos dentro de un formulario, dependiendo del valor que asignemos al atributo *type*. Lo primero que recomendamos en esta marca es especificar el valor del atributo *name* para identificar el elemento.

Seguidamente, elegiremos, mediante el atributo *type* el tipo de elemento que deseamos tener en el formulario.

- <***textarea***> ... </***textarea***>

En este caso, definiremos un campo de texto de grandes dimensiones para que el usuario tenga la posibilidad de escribir sin ningún tipo de limitación. Este comando se utiliza en muchos formularios cuando escribimos en un campo de observaciones.

Los atributos de esta marca serían los siguientes:

- ***Name***: identifica el nombre del área de texto.
- ***Cols***: número de caracteres que puede contener cada línea.
- ***Rows***: número de líneas del área de texto.
- ***Readonly***: para impedir que el usuario pueda editar este campo.

```
<select>
   <options>
   </options>
</select>
```

Este bloque de comandos se utiliza para definir una **lista desplegable** con opciones. Tendremos tantas marcas *options* como opciones deseemos tener en nuestra lista.