

IFCD0110 CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

Módulo formativo: MF0950_2: Construcción de páginas web

Unidad formativa: UF1302 Creación de páginas web con lenguajes de marcas

Unidad 3. HTML5. Etiquetas básicas. Imágenes. Enlaces. Listas. Tablas.

FORMACIÓN PROFESIONAL PARA EL EMPLEO.

CIPP VIRGEN DE GRACIA



GOBIERNO
DE ESPAÑA

MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



1. Elementos de texto y párrafo. Elementos para marcar caracteres.	2
2. Etiquetas básicas de párrafo.	3
3. Elementos de marcado de texto simple.	6
4. Citas.	7
5. Otros elementos de marcado de caracteres.	8
6. Elemento address.	12
7. Elementos de marcado de correcciones.	12
8. Listas.	12
9. Elementos estructurales.	15
9.1. Div.	15
9.2. Span.	16
10. Elementos estructurales semánticos.	16
10.1. Header.	16
10.2. Footer.	17
10.3. Main.	17
10.4. Section.	17
10.5. Nav.	17
10.6. Article.	18
10.7. Figure.	18
10.8. Figcaption.	18
10.9. Aside.	18
11. Títulos en los elementos de sección.	19
13. Elementos para acciones interactivas.	20
13.1 Elementos para indicar secciones de tipo mostrar/ocultar.	20
13.2 Elementos de cuadro de diálogo.	20
14. Crear enlaces en las páginas web.	20
14.1 Enlaces globales.	21
14.2 Enlaces locales relativos.	21
15. Enlaces internos.	21
16. Atributos del elemento a.	22
17. Imágenes.	26
18. Inserción de imágenes.	27
19. Elemento picture.	28
20. Elemento source.	28
21. Uso de picture.	29
22. Mapas de imagen.	30
23. Tablas.	33
24. Tablas simples.	34
25. Agrupación de filas.	34
26. Agrupación de columnas.	34
27. Combinar celdas.	35
28. Formato CSS para tablas.	35
29. Bibliografía.	38

1. Elementos de texto y párrafo. Elementos para marcar caracteres.

Hoy en día en una página web, todo el texto está dentro de algún elemento (delimitados mediante etiquetas), en realidad hay tres contextos principales a tener en cuenta sobre el texto

- **Elementos capaces de contener varios párrafos.** La W3C llama a estos elementos *sectioning content*. Cada elemento de esta parte es lo que hoy en día se conoce como **capa** o **sección**. Si no indicamos capa alguna, entonces todos los párrafos están contenidos en la etiqueta **body**, lo que significa que el contenedor de los párrafos será el área que el navegador deja para el texto y que puede ser casi toda la pantalla.
- **Elementos que contienen párrafos individuales.** Aunque en HTML 5 se permite escribir texto que no esté dentro de un párrafo, no es una buena práctica, ya que no podríamos indicar más adelante el formato del párrafo (alineaciones, márgenes,...).

Lógicamente dentro de un párrafo, no podemos indicar otro párrafo; es decir no podemos meter una etiqueta **p**, dentro de una etiqueta **h1**.

Cuando un texto está dentro de un elemento de párrafo (como **p**), hará que el texto siguiente quede en otro párrafo, por lo tanto habrá un salto de párrafo entre ambos textos.

- **Elementos que contienen texto simple o conjuntos de caracteres.** La W3C llama a estos elementos *phrasing content*. Son elementos pensados para ser interiores a un párrafo. Un mismo párrafo puede contener muchos elementos de este tipo. El mismo texto puede estar contenido en varios elementos de este tipo a la vez.

Por ejemplo, podemos indicar que un texto es **strong** y **em** a la vez.

2. Etiquetas básicas de párrafo.

Desde hace años todo texto en un documento HTML debe ir dentro de una etiqueta que sirva para decir qué tipo de texto es. Y eso significa, que el texto debe de ir dentro de una etiqueta de párrafo. Se comentan las etiquetas disponibles para marcar párrafos.

PÁRRAFO SIMPLE

El elemento más sencillo lo marca la etiqueta **p**, que indica un elemento de párrafo normal. Todo texto dentro de una etiqueta **p**, queda marcado como texto dentro de un párrafo normal.

Normalmente los navegadores utilizan fuentes tipo *Times* de tamaño *11pt* para la letra de párrafo normal. Ejemplo de uso:

```
<body>
<p>
    Párrafo con un poco de texto
</p>
<p>
Este es un párrafo completo englosado dentro de una etiqueta <em>p</em>,  

aunque  

    el texto se escriba  

    con todos estos saltos de línea      y      espacios  

    se mostrará seguido,  

    sin saltos     de  

    línea interiores  

                                y entre cada palabra
</p>
</body>
```

Párrafo con un poco de texto

Este es un párrafo completo englosado dentro de una etiqueta *p*, aunque el texto se escriba con todos estos saltos de línea y espacios se mostrará seguido, sin saltos de línea interiores y entre cada palabra dejando un solo espacio

En la imagen se observa el modo en el que el navegador muestra el párrafo. Por defecto los párrafos tipo *p* dejan espacio arriba y abajo (normalmente media línea en cada dirección).

TÍTULOS

Hay una serie de siete etiquetas que comienzan con la letra *h* a la que le sigue un número del 1 al 7. Sirven para marcar párrafos de forma que se considerarán títulos del texto. De modo que el elemento **h1** marcará títulos de primer nivel, **h2** de segundo nivel,... y así hasta **h6** (sexto nivel).

Intenta reproducir la siguiente página:

Sistema Solar

Planetas

Mercurio

Mercurio es el planeta del Sistema Solar más próximo al Sol y el más pequeño (a excepción de los planetas enanos). Forma parte de los denominados planetas interiores o rocosos y carece de satélites

Venus

Venus es el segundo planeta del Sistema Solar en orden de distancia desde el Sol, y el tercero en cuanto a tamaño, de menor a mayor

...

Sol

El Sol es la estrella del sistema planetario en el que se encuentra la Tierra; por tanto, es la más cercana a la Tierra y el astro con mayor brillo aparente

Satélites

Luna

La Luna es el único satélite natural de la Tierra y el quinto satélite más grande del Sistema Solar.

Io

Fue descubierto por Galileo Galilei en 1610 y recibió inicialmente el nombre de Júpiter I como primer satélite de Júpiter.

...

Otros Sistemas

Fomalhaut

Estrella conocida desde la prehistoria ahora se le han descubierto planetas

Vega

Antigua estrella polar, muy venerada. Posee un disco de polvo que podría contener planetas o bien formarse pronto

SALTO DE LÍNEA

A veces es necesario dentro del texto de un determinado párrafo hacer un salto de línea. El elemento que lo realiza no tiene cierre y se llama **br**. Ejemplo:

```
<p>Primera línea <br>Segunda línea</p>
```

Primera línea
Segunda línea

LÍNEA HORIZONTAL

Otra posibilidad es hacer un salto pero dejando una línea horizontal en el hueco de las palabras. Esto lo hace la etiqueta **hr** (que tampoco tiene cierre):

```
<p>Primera línea <hr>Segunda línea</p>
```

Aunque los navegadores entienden este código. En realidad **hr** tiene que estar fuera de las etiquetas de párrafo, es decir lo correcto es:

```
<p>Primer párrafo</p>
<hr>
<p>Segundo párrafo</p>
```

Primer párrafo

Segundo párrafo

3. Elementos de marcado de texto simple.

TEXTO IMPORTANTE

En HTML 5 el texto importante se marca con el elemento **strong**. En general todos los navegadores marcan el texto **strong** en negrita (letra más gruesa), pero que se muestre en negrita es un acuerdo visual que se puede modificar con el lenguaje CSS. Es decir, no es lo mismo texto en negrita que texto importante.

Sin duda, **strong** es uno de los elementos HTML fundamentales e imprescindible en los documentos HTML.

TEXTO ENFATIZADO

El elemento **em** marca texto como enfatizado. El resultado visual con que le dan formato por defecto los navegadores es texto en cursiva.

La idea semántica es marcar con **em** texto con énfasis de segundo nivel (dejando a **strong** el primer nivel) que normalmente se dedica a palabras no aceptadas aun, títulos de trabajos, apodos, texto subjetivo, etc.

TEXTO RESALTADO

Se trata del tercer elemento de HTML5 para remarcar texto. Se hace a través el elemento **mark**. En este caso, su efecto visual, suele ser aplicar en el texto así marcado un fondo de color vivo que simula el uso de un subrayador. Ejemplo:

```
<p>Yo soy texto normal, <mark>y yo estoy remarcado</mark></p>
```

Yo soy texto normal, y yo estoy remarcado

Este elemento no es tan antiguo como los anteriores porque apareció con HTML5 (los navegadores antiguos no lo reconocen). La idea semántica es marcar texto que queremos realzar de forma muy llamativa indicando texto a recordar o sobre el que profundizar.

4. Citas.

En HTML5 existen tres formas de marcar citas:

- **q.** Usada para indicar citas cortas incluidas en un párrafo con otro tipo de texto. Indica un texto que procede de otra fuente. Remarca una cita, con la connotación de “texto entrecomillado”. De hecho, los navegadores, por defecto, entrecomillan su contenido. Ejemplo de uso:

```
Como dijo Julio César, <q  
cite="http://es.wikipedia.org/wiki/Alea_iacta_est">  
Alea Jacta es</q>
```

No es obligatorio utilizar el atributo **cite**. Este atributo permite indicar una fuente para consultar la cita original (los navegadores, sin embargo no hacen nada con el contenido de este enlace).

- **cite.** Este elemento era el que originalmente se usaba para citas literales. Actualmente se recomienda su uso para indicar títulos de trabajos y el nombre de sus autores. Eso incluye títulos de libros, memorias, ensayos, poemas, obras musicales, sitios web, entradas de blog o foros, comentarios, tweets, discursos, etc. Es decir cualquier trabajo creativo. Ejemplo:

```
<p>  
Edgar Codd publicó: <cite>Un modelo relacional de datos para grandes  
bancos de datos compartidos</cite> en 1970  
</p>
```

Los navegadores suelen mostrar el contenido de **cite** en cursiva.

- **blockquote.** Este es el único elemento de marcado de citas que trabaja a nivel de párrafo. Es decir, indicado párrafos completos que son citas procedentes de fuentes externas al documento. Se le puede indicar (al igual que se hace con el elemento **q**) el atributo **cite** para indicar un enlace a la fuente de la cita. Ejemplo:

```
<blockquote
cite="http://noticias.juridicas.com/base_datos/Admin/constitucion_tp.html#a1">
<strong>Artículo 1 de la constitución</strong><br>
España se constituye en un Estado social y democrático de Derecho,
que propugna como valores superiores de su ordenamiento jurídico
la libertad, la justicia, la igualdad y el pluralismo político.
</blockquote>
```

Puede utilizar el atributo **cite** (como se observa en el ejemplo), para indicar la dirección URL que se ha utilizado como fuente de la cita.

Visualmente los navegadores muestran el texto **blockquote** añadiendo una sangría a la derecha y a la izquierda. Ignoran, visualmente, el uso del atributo **cite**, pero otras herramientas sí usan este contenido.

Ejemplo 0_citas.html

5. Otros elementos de marcado de caracteres.

SUBÍNDICES

Permite que el texto aparezca por debajo de la línea base y en un tamaño más pequeño. Lo hace el elemento **sub**, ejemplo:

```
<p>La fórmula del agua es H<sub>2</sub>O</p>
```

SMALL

Marcado de letra pequeña. Se usa, por ejemplo en textos que marquen copyright, derechos de uso, comentarios, notas anexas y letra pequeña (tipo la de los contratos) en general. El efecto visual es que el texto sale con un tamaño más pequeño. Es posible incluso anidar un elemento **small** dentro de otro, de modo que cada vez sale más pequeño el texto. Ejemplo:

```
Texto normal <small>Pequeño
<small>Aún más pequeño</small></small>
```

Texto normal Pequeño Aún más pequeño

ELEMENTOS

Se trata de un antiguo elemento de las páginas web que permitía indicar un subrayado. En la versión de HTML 4 quedó obsoleto, no se recomendaba su uso ya que el subrayado se reserva para los enlaces. Con HTML 5 se ha redefinido su uso para marcar texto obsoleto que, visualmente, se muestra con efecto de tachado.

```
La capital de Alemania es <s>Bonn</s> Berlín
```

La capital de Alemania es ~~Bonn~~ Berlín

ELEMENTO B

Se trata del elemento histórico (de las primeras versiones de HTML) utilizado para marcar texto en negrita. En HTML 4 quedó como obsoleto. En HTML 5 se volvió a permitir pero solo si necesitamos un nivel de marcado de texto más allá del uso de **strong**, **em** o **mark**. En definitiva, elemento de uso muy poco aconsejable que se mantienen solo por la gran cantidad de creadores que lo siguen utilizando.

ELEMENTO I

Tiene el mismo problema que el anterior. Es el elemento histórico (de las primeras versiones de HTML) utilizado para marcar texto en cursiva. En HTML 4 quedó como obsoleto. En HTML 5 se puede utilizar para marcar texto tipo alternativo si no encaja con otros elementos de marcado de caracteres.

Al igual que ocurre con el elemento *b*, su uso no es aconsejable.

ABREVIATURAS CON ABBR

Marca texto que indica una abreviatura o un acrónimo. Su uso requiere utilizar el atributo **title** para indicar el significado de la abreviatura. En general, los navegadores no muestran ningún resultado, pero sí un cartel con el contenido de **title** cuando se arrima el ratón. Ejemplo:

```
<p>Se ha coronado a
<abbr title="Su Alteza Real">SAR</abbr> el Rey Felipe VI </p>
```

En HTML 5 se eliminó el uso del elemento **acronym** que estaba pensado para acrónimos. Ahora tanto acrónimos como abreviaturas se deben utilizar con **abbr**.

DEFINICIONES

El elemento **dfn** se usa de forma similar al anterior. Sirve para indicar definiciones para el término o palabras que encuadra.

```
<dfn title="usar dos conceptos de significado opuesto en una sola expresión">oxímoron</dfn>
```

Los navegadores suelen mostrar el contenido del elemento en cursiva y, como siempre, en un cartelito muestran el contenido del atributo **title**:

oxímoron

usar dos conceptos de significado opuesto en una sola expresión

ESCRITURA DE CÓDIGO O TEXTO INFORMÁTICO

```
<p>Este es un ejemplo de código en JavaScript:</p>
<code>
    window.onload=function(){
        alert("Hola");
    }
</code>
```

Este es un ejemplo de código en JavaScript:

```
window.onload=function(){ alert("Hola"); }
```

```
<p>Este es un ejemplo de código en JavaScript:</p>
<pre><code>
    window.onload=function(){
        alert("Hola");
    }
</code></pre>
```

Este es un ejemplo de código en JavaScript:

```
window.onload=function(){
    alert("Hola");
}
```

Con intenciones similares al elemento anterior, tenemos el elemento **samp**. Se utiliza cuando queremos en un documento indicar ejemplos de resultados de operaciones en un ordenador. Ejemplo:

```
<p>Al ejecutar el comando obtuve el mensaje
<samp>Comando o nombre de archivo incorrecto </samp>
</p>
```

Al ejecutar el comando obtuve el mensaje Comando o nombre de archivo incorrecto

TECLAS

El elemento **kbd** sirve para indicar el nombre de una tecla o combinación de teclas (también comandos de voz o entradas de menú). Se puede indicar el atributo **title** para señalar el significado de la tecla.

Ejemplo:

```
<p>Para finalizar puse <kbd>Ctrl+C</kbd></p>
```

VARIABLES

El elemento **var** sirve para indicar texto que se refiere a nombres de variables de un programa informático. Sólo tiene sentido su uso para documentos sobre programación.

FECHAS Y HORAS

El elemento **time** está presente desde HTML5 (en Internet Explorer desde la versión 9) y permite indicar una fecha y una hora (por ejemplo 10:00). Atributos:

- **datetime**. Hora en formato estándar (yyyy-mm-dd o yyyy-mm-ddThh:mm:ssTZ) que representa la hora exacta que muestra el texto.

Ejemplo 1_fechas.html

ELEMENTOS ESPECIALES

Ejemplo 2_textoReves.html

wbr. Contiene una palabra a la que estaremos indicando que puede ser partida por un guión por parte del navegador.

6. Elemento address.

El elemento **address** permite indicar una dirección, especialmente indicada para mostrar información sobre el autor o los créditos del documento (con información de los autores por ejemplo) al estilo de los pies de artículos periodísticos.

Aunque, de forma clásica, se le considera un elemento de marcado de párrafos, actualmente se le ha denominado como un elemento de marcado de secciones

Ejemplo 3_address.html

7. Elementos de marcado de correcciones.

Permiten marcar texto para indicar como texto que se está revisando y que falta decidir su contenido final. Hay dos elementos de este tipo **ins** y **del**.

- **ins.** Indica un texto añadido al documento. Puede utilizar dos atributos:
- **del.** Indica un texto que se desea retirar del documento.

Ambos elementos pueden utilizar dos atributos:

- **cite.** Con un enlace a una dirección de Internet que proveerá de más información sobre la corrección.
- **datetime.** Indica la fecha de la modificación con cuatro cifras para el año, dos para el mes y dos para el día.

Ejemplo 4_correcciones.html

8. Listas.

Las listas permiten crear párrafos agrupados y alineados mediante símbolos como viñetas o números para facilitar la lectura y organización de las ideas del documento.

Podemos encontrar:

- Listas con viñetas
- Listas numéricas
- Listas anidadas
- Mezclas

- Listas de términos

Ejemplo 5_listasviñetas.html

Ejemplo 6_listasnumericas.html

Ejemplo 7_listasanidadadas.html

Ejemplo 8_listas.html

Ejemplo 9_listasterminos.html

Crea una página HTML que produzca el siguiente resultado:

El proceso de **maquetado** de una página Web™ consta de:

1. Reunión con el cliente
2. Boceto HTML
3. Posicionado de elementos
4. Alojamiento y pago

El pseudocódigo podría ser

```
while not fin:  
    revisar  
    comprobar con el cliente  
end while
```

Crea una página HTML que produzca el siguiente resultado:

Aircraft Types

- i. General Aviation
 - a. Single Aviation Aircraft
 - b. Dual-Egine Aircraft
- ii. Commercial Aviation
 - Dual Engine
 - Tri-Engine

- A
 - B
 - 1. C
 - 2. D
 - E
 - F
 - G
- B

Crea una página HTML que produzca el siguiente resultado:

Internet

Contenido

1. Principales protocolos
2. Formas de conectarse

1. Principales protocolos

- Transmisión de ficheros:
 - FTP
- Correo electrónico:
 - IMAP
 - POP
 - SMTP

2. Formas de conectarse

Red Telefónica Comutada (RTC)

La línea telefónica de toda la vida. Para acceder a Internet es necesario un módem.

Red Digital de Servicios Integrados (RDSI)

Una línea telefónica especial. Para acceder a Internet es necesario un módem RDSI.

Línea de Abonado Digital Asimétrica (ADSL)

Se basa en la conversión de una línea RTC en una línea de alta velocidad. Para acceder a Internet es necesario un módem ADSL.

Fibra Óptica

Una línea de fibra óptica. Normalmente la fibra óptica no llega hasta el usuario final, por lo que el término más apropiado es Fibra híbrida coaxial.

Crear una página HTML que produzca el siguiente resultado:

Lista sin elemento gráfico:

- Elemento 1
- Elemento 2
- Elemento 3

Lista con círculos:

- Elemento 1
- Elemento 2
- Elemento 3

Lista con cuadrados:

- Elemento 1
- Elemento 2
- Elemento 3

Lista con números romanos:

- i. Elemento 1
- ii. Elemento 2
- iii. Elemento 3

Lista con letras griegas:

- α. Elemento 1
- β. Elemento 2
- γ. Elemento 3

9. Elementos estructurales.

9.1. Div.

Se trata de un elemento clásico, ya estaba presente en las primeras versiones de HTML, que se utiliza como contenedor de otros elementos a fin de poder manipularlos todos a la vez. Dentro de **div** se pueden colocar todo tipo de etiquetas (tablas, párrafos, imágenes,...).

Desde hace años se ha convertido en el elemento utilizado, combinado con CSS, para crear **capas**. Las capas son elementos visuales que se pueden posicionar a voluntad en la página web. De esta forma, a través de las capas se consiguen maquetaciones muy artísticas.

Por defecto, los navegadores no dan formato al contenido de un elemento *div*, con lo que visualmente no se diferencia del resto de elementos, salvo por el hecho de que un elemento *div* tiene un display de tipo *block*. Es decir, el contenido de estos elementos, por defecto, se muestra de forma separada al resto.

Ejemplo 10_div.html

9.2. Span.

Es muy similar al anterior, pero como los navegadores no agregan nada (ni siquiera espacios) a esta etiqueta cuando se muestra por pantalla, en lugar de para definir capas (aunque se podría) se usa para marcar contenido interior a un párrafo, a fin de que a ese contenido se le pueda dar un formato especial mediante CSS. Es decir, con **div** marcamos capas (contenedores grandes) y con **span** marcamos bloques pequeños de texto.

Ejemplo 11_span.html

10. Elementos estructurales semánticos.

La idea en HTML5 (a diferencia de las versiones anteriores) es que los elementos HTML aporten valor semántico al contenido. Es decir, se trata de que el código HTML indique qué tipo de contenido es el que está dentro de cada elemento. Eso vale para casi todos los elementos ya conocidos como **p** (párrafo), **abbr** (abreviatura), **strong** (remarcado fuerte), etc.

Sin embargo los estándares anteriores a la versión 5 tenían una carencia muy importante. El único elemento para marcar contenedores grandes era **div**. Y este elemento no aporta ninguna semántica a su contenido.

En HTML 5 hay una serie muy importante de elementos que no dan ningún formato al texto, salvo el hecho de que aparezcan con un display de bloque, pero permiten darle un significado. Será CSS el encargado de que esos elementos aparezcan con un formato apropiado a su significado en la página final. La idea final es que estos elementos sustituyan el sobreuso que, tradicionalmente, se hacía de la etiqueta **div**.

No es fácil esta idea, puesto que existen numerosas plantillas y patrones de trabajo profesionales (como ocurría por ejemplo con **Bootstrap**) que se basan en *div* para trabajar con ellas. La mayoría de plantillas actuales (incluida la propia Bootstrap) se han adaptado a esta idea.

10.1. Header.

Permite marcar contenido como cabecera del documento. Hay que señalar que este elemento no tiene nada que ver con el elemento **head**. Header marca una serie párrafos para indicar que pertenecen a la cabecera de la página.

La cabecera de la página suele contener el logotipo de la página, el menú, el título principal, etc.

Realmente una página puede tener varios elementos **header**. Si está al nivel de la etiqueta **body** indica que su contenido es la cabecera de la página completa. Pero dentro de un elemento de sección, por ejemplo, **article**, indicaría que su contenido es la cabecera de ese artículo.

No se debe indicar un elemento header dentro de otro header ni dentro de un elemento footer o address.

10.2. Footer.

Similar al anterior, pero sirve para marcar el pie de una página, sección, artículo etc. En el caso de que un elemento footer se coloque al nivel del elemento **body**, servirá para indicar el pie general de la página que suele contener información sobre el autor, copyright, términos de uso de la página, contacto, etc.

Al igual que ocurría con header, no se puede colocar un footer dentro de otro ni puede estar dentro de header o address.

10.3. Main.

Es incluso más moderno que los anteriores. Permite indicar una sección (que podrá incluir artículos, cabeceras, títulos, párrafos, secciones de imagen,...) de forma que se entenderá que esta es la sección principal del documento.

Solo puede haber un elemento main en una página web. Se usa para diferenciar de las secciones de menú, publicidad o cualquier tipo de información en la página considerada como secundaria.

main no puede colocarse dentro de secciones de tipo **article**, **footer**, **header**, **aside** o **nav**. Sí se admite dentro de **section**, para marcar elementos principales dentro de una sección.

10.4. Section.

Es un elemento que permite dividir en diferentes partes o secciones un documento.

Ejemplo 12_section.html

El atributo **id** no es obligatorio, pero a veces se usa con la finalidad de identificar de modo único a cada sección, por ejemplo para darle un formato especial a una sección concreta.

10.5. Nav.

Se trata de un elemento que marca a su contenido como una sección de enlaces o menú, es decir una barra de navegación. Más adelante con CSS se puede dar un formato especial a dichos enlaces. **nav** se puede escribir dentro de cualquier elemento HTML estructural (**section**, **article**, **header**, **footer**,..), pero tampoco se deben marcar todos los enlaces de navegación como parte de un elemento nav, solo los que formen un menú real y completo de navegación.

Los elementos marcados por **nav** pueden ser omitidos por los lectores digitales de páginas web que utilizan, por ejemplo, las personas invidentes. Lo cual facilita la

comprensión del texto) y así que dicho contenido quede marcado solo para utilizar los enlaces interiores a **nav**.

Lógicamente dentro de **nav** se suelen incluir numerosos elementos de tipo **a**.

10.6. Article.

Si se observan los elementos descritos anteriormente parece claro que HTML 5 utiliza, como metáfora, la forma de distribuir contenidos de un periódico. Así hay cabeceras, pies, secciones y, con este elemento, artículos.

La idea es colocar dentro de este elemento, que tiene sentido que aparezca dentro del elemento **section** o incluso aparecer de forma independiente, contenido que pueda ser entendido como un *todo* que describa un tema de forma íntegra.

Es decir dentro de **article** se colocan los elementos que sirvan para describir un mismo tema.

Es posible (aunque poco habitual) colocar elementos **article** dentro de otros elementos **article**, pero solo cuando queremos indicar artículos externos relacionados con el artículo principal.

La autoría del artículo (nombre del autor y créditos) se puede indicar con el elemento **address**.

10.7. Figure.

Sirve para agrupar los elementos relativos a una imagen como la propia imagen (elemento **img**), el título de la misma, el pie, los párrafos relativos, etc. Contenido relacionado con el que tiene alrededor, pero que se podría utilizar de forma independiente en otro documento.

10.8. Figcaption.

Permite indicar el título de una imagen, dentro de un elemento **figure**, haciendo el efecto de un pie de imagen. Por defecto la imagen y el texto quedarán alineados de la misma forma.

Ejemplo 13_figcaption.html

10.9. Aside.

Permite marcar texto dentro de un artículo para que no se tenga en cuenta como parte del texto del artículo, sino como un texto aparte que permite realizar aclaraciones al artículo, referencias, resúmenes remarcados y sobre todo cuadros de texto de estilo periodístico para destacar partes del artículo. Para que este texto aparezca de manera especial, debe dársele formato con CSS.

La idea es simular el típico recuadro que aparece en los periódicos en los que se expresan ideas interesantes o que resumen el artículo.

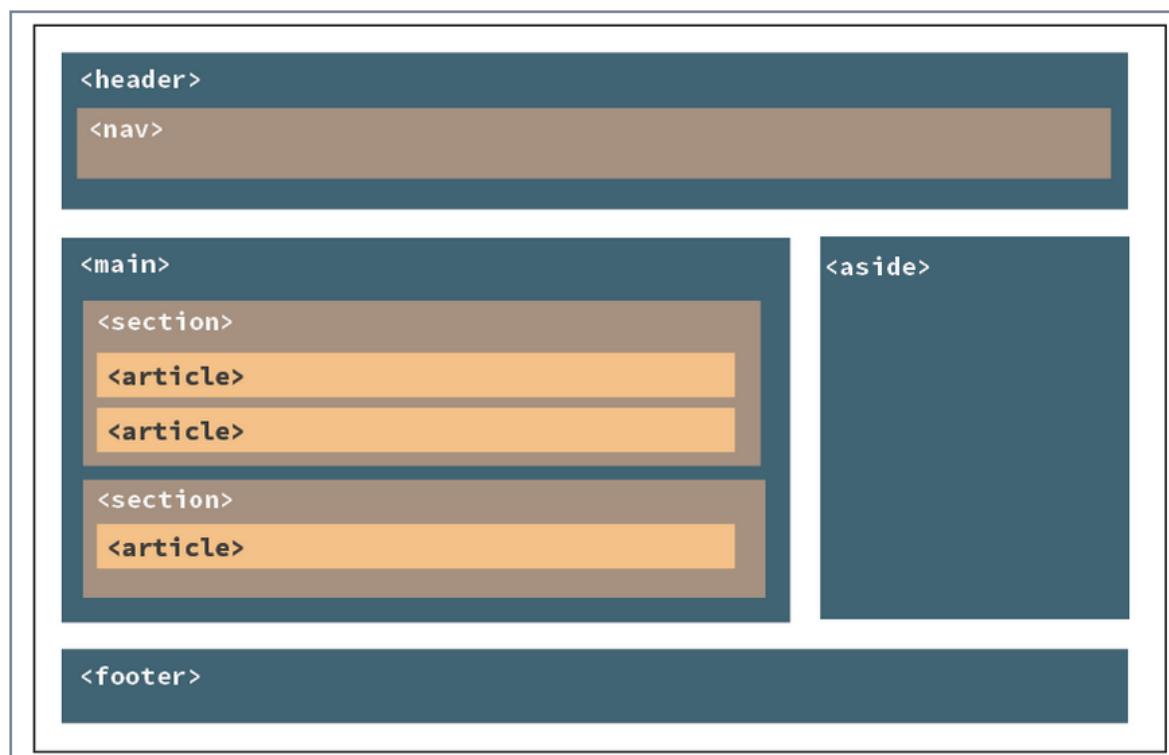
En definitiva, aside define texto que no forma parte del flujo principal de la página sino que posee contenido que debe de fluir aparte.

11. Títulos en los elementos de sección.

Podemos entender que hay contenedores que permiten estructurar un contenido de forma completa y que este contenido es, a su vez, parte de un todo más grande. Es decir, una página actual se puede entender como un conjunto de secciones cada sección, a su vez, se puede dividir en más secciones.

De este modo, los títulos h1, h2, h3, etc., permiten establecer una jerarquía extra. Es decir: un elemento h1 dentro directamente del elemento body se consideraba un título principal de toda la página web. Ahora, normalmente, se colocará h1 dentro de una sección concreta, lo que significará que ese h1 es el título principal de esa sección. Incluso esa sección se puede dividir en subsecciones, cada una de las cuales tendrán sus propios elementos h1 que serán títulos principales pero de cada subsección, y por lo tanto de menor nivel que los h1 de la sección.

Esto suena a galimatías, pero lo cierto es que da una mayor flexibilidad a la hora de estructurar el contenido de las páginas y, también, a la hora de darla formato.



13. Elementos para acciones interactivas.

13.1 Elementos para indicar secciones de tipo mostrar/ocultar.

Se trata de secciones que están dentro de un elemento llamado **details**. Dentro de ese elemento se puede poner una sección de tipo **summary** la cual muestra un texto de resumen de la sección. Al hacer clic en el apartado summary, se muestra todo el contenido de la sección.

Es decir en definitiva consigue crear en un documento HTML un apartado de tipo resumen/detalle.

Ejemplo 14_details.html

Summary se coloca dentro del anterior. Es un texto que se puede poner en los elementos de tipo **detail**. Ese texto aparece en lugar del texto *Detalles* que aparece por defecto en una sección de tipo *detail*.

13.2 Elementos de cuadro de diálogo.

Se suelen llamar cuadros de diálogo a las ventanas que los sistemas muestran a los usuarios para darles mensajes, o para permitirles configurar una determinada respuesta u operación.

En HTML 5 se ha incorporado una nueva etiqueta llamada **dialog**, precisamente con esta finalidad. Todavía está en fase de pruebas. El contenido del elemento dialog se muestra en un recuadro. La gracia es que desde JavaScript tiene muchas opciones de manipulación.

Hay un atributo llamado **open** (no se le pone valor alguno cuando se usa) que hace que se muestre el cuadro de diálogo (sin ese atributo, el cuadro no se muestra).

Ejemplo 15_dialog.html

Realmente sus principales posibilidades se consiguen combinando su uso con el lenguaje JavaScript.

14. Crear enlaces en las páginas web.

14.1 Enlaces globales.

Los enlaces permiten colocar un texto (u otro elemento, como una imagen o un botón) resaltado de forma especial, de modo que cuando se le hace clic, el navegador web nos llevará al destino de la URL (si está disponible).

La etiqueta que permite realizar enlaces es la etiqueta **a**. El atributo **href** permite indicar la URL a la que se realiza el salto.

Ejemplo 16_enlaces.html

14.2 Enlaces locales relativos.

```
<!-- Salto a la página tiberio.html que estará en el mismo directorio
que la actual -->
<a href="tiberio.html">Tiberio</a>
<!-- Salto a la página tiberio.html que estará dentro del directorio emperadores
<a href="emperadores/tiberio.html">Tiberio</a>
<!-- Salto a la página tiberio.html que estará en el directorio padre, es decir
<a href="../tiberio.html">Tiberio</a>
<!-- Salto a la página tiberio.html que estará en el directorio emperadores,
<a href="../../emperadores/tiberio.html">Tiberio</a>
```

Crea una página HTML con una lista de enlaces que permitan acceder a los ejemplos hechos hasta ahora en esta unidad.

15. Enlaces internos.

ENLACES A UNA POSICIÓN INTERNA DENTRO DEL PROPIO DOCUMENTO

Hay un tipo de enlace que permite posicionarnos en un punto concreto del documento. Este tipo de enlace se denomina interno, ya que salta dentro del propio documento.

Este salto requiere primero marcar la posición a la que deseamos saltar. Antes de HTML 5 se hacía usando el atributo **name** del propio elemento **a**. Pero con HTML 5 se prohíbe el uso de este atributo con este fin y, en su lugar, se utiliza el atributo **id**.

id es un atributo presente en cualquier elemento de la página web. Sirve para identificar a dicho elemento dentro de la página, por ello no podemos repetir valor para ese atributo en ningún otro elemento.

Ejemplo 17_anclas.html

ENLACES A UNA POSICIÓN INTERNA DENTRO DE OTRO DOCUMENTO

También podemos realizar un salto a una posición interna a otro documento. Por ejemplo si este enlace:

```
<a href="manual.html">Abrir manual</a>
```

Nos permite abrir una página web llamada **manual.html** (que estará en el mismo directorio que la página actual). Siempre que se abre un documento se muestra el inicio del mismo. Sin embargo este otro enlace:

```
<a href="manual.html#comentarios">  
Abrir manual por la zona de comentarios</a>
```

Abre el mismo documento, pero se intenta colocar en la zona marcada por el identificador *comentarios*. Lógicamente, para que esto sea posible, debe existir un elemento marcado con ese identificador en el documento.

16. Atributos del elemento a.

atributo	significado
hreflang	Permite indicar un código de lenguaje (es, fr, en,...) indicando el lenguaje en el que está escrito el destino del enlace. Ejemplo: <pre> Información sobre Londres</pre>
media	Sólo válido en HTML5, permite indicar el medio idóneo para mostrar el contenido del enlace. Ejemplo: <pre> Descargar manual</pre>
target	Permite indicar cómo se muestra la página de destino. Posibilidades: <ul style="list-style-type: none"> ▪ <u>blank</u>. Abre el enlace en una nueva página. Es una opción muy utilizada al crear enlaces. ▪ <u>parent</u>. Abre el enlace en el marco de la página padre de ella. ▪ <u>top</u>. Abre la página en el marco superior ▪ <u>self</u>. Abre la página en el marco actual ▪ <i>nombre</i>. El nombre indicado será el del marco en el que se abrirá la página Salvo el primero, el resto no se usan por referirse a marcos.
type	Atributo añadido en HTML 5, soportado por la mayoría de navegadores. Permite indicar el tipo de contenido (según la normativa oficial de tipos de la IANA, véase: http://www.iana.org/assignments/media-types/media-types.xhtml Ejemplo: <pre>Abrir imagen nº 98</pre> En el ejemplo se indicaría que el enlace hace referencia a una imagen de tipo PNG

rel	<p>Informa sobre la función del enlace. Puede ser:</p> <ul style="list-style-type: none"> ▪ alternate. Enlace alternativo ▪ author. ▪ bookmark. Página de marcadores ▪ help. Página de ayuda ▪ license. Información sobre la licencia ▪ next. Si nuestra página pertenece a una serie ordenada, el enlace nos lleva al siguiente elemento dentro de la serie. ▪ nofollow. Marca que los robots de búsqueda de empresas como Google no tengan en cuenta los enlaces externos y así evitar que dichos enlaces en las páginas se utilicen para subir su calificación en los buscadores. Así se ignoran por los robots los enlaces marcados de esta forma. ▪ noreferrer. Un <i>referrer</i> es la información relativa a la página desde la que procede el visitante a un sitio. Con este valor en los enlaces, no se indicará al destino URL la página desde la que procedía el usuario. ▪ prefetch. Permite descargar el enlace antes de que el usuario haga clic en él y así acelerar su carga. Se usa (aunque pocos navegadores soportan este valor) en enlaces de uso habitual. ▪ prev. Si nuestra página pertenece a una serie ordenada, el enlace nos lleva al elemento anterior dentro de la serie. ▪ search. Página de búsqueda dentro de nuestro sitio web. ▪ tag. Página con etiquetas de temas (tags) de nuestro sitio web.
download	<p>Nuevo atributo de HTML 5 (no funciona ni en Internet Explorer ni en Safari, ni en cualquier versión de navegador que no sea bastante moderna) que permite indicar que al hacer clic en el enlace, no se abrirá en el navegador, sino que se descargará en nuestro directorio predeterminado de descargas.</p> <p>Ejemplo:</p> <pre> Descargar manual</pre> <p>Se puede indicar un valor para el atributo que se tomará como el nuevo nombre del archivo descargado:</p> <pre>Descargar manual</pre> <p>Aunque el nombre original del archivo que se descarga en el servidor es <i>manual3232001.pdf</i>, al descargar el nombre del archivo será <i>manual.pdf</i></p>

Ejercicio enlaces

A partir del texto que se te proporciona, debes crear una página web que tenga el siguiente aspecto:

Los tres pilares de la Web

Tim Berners-Lee es considerado el padre de la Web porque desarrolló los tres elementos básicos para el funcionamiento de la Web:

- [HTML](#)
- [HTTP](#)
- [URL](#)

HTML

HTML, siglas de **HyperText Markup Language**, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web.

Fuente: [HTML](#), Wikipedia

HTTP

Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1.

Fuente: [HTTP](#), Wikipedia

URL

Un **localizador de recursos uniforme** o URL —siglas en inglés de *Uniform Resource Locator*— es un identificador de recursos uniforme (URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres, de acuerdo a un formato modelico y estándar, que designa recursos en una red, como Internet.

Fuente: [URL](#), Wikipedia

Además, tienes que tener en cuenta los siguientes requisitos:

- El título de la página debe ser Los tres pilares de la Web.
- Los enlaces que aparecen en la página deben tener los siguientes destinos:
 - Tim Berners-Lee → http://es.wikipedia.org/wiki/Tim_Berners-Lee
 - Web → http://es.wikipedia.org/wiki/World_Wide_Web
 - HTML → enlace intradocumental al epígrafe HTML
 - HTTP → enlace intradocumental al epígrafe HTTP
 - URL → enlace intradocumental al epígrafe URL
 - Fuente: HTML → <http://es.wikipedia.org/wiki/HTML>
 - Fuente: HTTP →
http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol
 - Fuente: URL →
http://es.wikipedia.org/wiki/Localizador_de_recursos_uniforme

Consejo: para comprobar que los enlaces intradocumentales funcionan correctamente, aumenta el tamaño o zoom de la página con Ctrl+

Código base

Los tres pilares de la Web

Tim Berners-Lee es considerado el padre de la Web porque desarrolló los tres elementos básicos para el funcionamiento de la Web:

HTML

HTTP

URL

HTML

HTML, siglas de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web.

Fuente: HTML, Wikipedia

HTTP

Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1.

Fuente: [HTTP](#), Wikipedia

URL

Un localizador de recursos uniforme o URL –siglas en inglés de Uniform Resource Locator– es un identificador de recursos uniforme (URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que designa recursos en una red, como Internet.

Fuente: [URL](#), Wikipedia

17. Imágenes.

Las imágenes son fundamentales para que una página web sea más atractiva. Son el primer contenido no textual que se planteó poder incorporar al estándar HTML. Prácticamente no hay páginas sin imágenes.

Los navegadores tienen capacidad de mostrar imágenes, pero sólo las que pertenezcan a tipos concretos. Los tipos de imágenes reconocidos por la mayoría de navegadores son:

- **Formato jpg.** Son imágenes que ocupan muy poco gracias a su alta compresión. No admiten animaciones ni zonas marcadas con transparencia. Son el formato habitual de la imagen digital en una página web.

- **Formato gif.** Imágenes con hasta 256 colores, por lo que son buenas para dibujos y logotipos, pero no para fotografías. Pueden incluso contener animaciones (uno de sus usos habituales) y colores marcados como transparentes, a través de los cuales se mostraría lo que esté *por debajo* de la imagen.
- **Formato png.** Imágenes fotográficas comprimidas al estilo de jpg pero con más calidad (ocupan más, normalmente) Permite la posibilidad además de utilizar canales alfa, es decir marcar opacidades con hasta 255 niveles. De esta forma habrá partes de la imagen que se mezclarán con el fondo. También hay posibilidad de hacer animaciones estilo gif.
- **Formato webp.** Formato de imagen auspiciado por **Google** para conseguir mejorar a PNG y JPG haciendo que ocupen menos sin pérdida de calidad respecto a estos formatos. Sin embargo, solo los navegadores **Chrome y Opera** son capaces de interpretar este formato.
- **Formato SVG.** Es un formato vectorial de imágenes. Las imágenes vectoriales tienen la ventaja de que nunca pierden calidad independientemente de cuanto las ampliemos o las reduzcamos. La desventaja es que no se pueden utilizar para imágenes fotográficas. Tenía problemas de compatibilidad en algunos navegadores, pero en los últimos años ha ganado mucha popularidad y es compatible con todos los navegadores.

Descartando a las imágenes **SVG**, el tamaño en disco de las imágenes puede ser mayor o menor dependiendo de su tamaño y su compresión. De modo que un tamaño grande implica más tardanza al cargar la página, pero una mayor nitidez en la imagen.

Las imágenes son fundamentales en las páginas web, su elección resulta vital para la estética de la misma, nunca se considera un mero acompañamiento ya que el impacto visual y las sensaciones sobre la profesionalidad de la página siempre le consiguen las imágenes con ayuda de la tipografía (pero ahí ya entra el lenguaje CSS) y la disposición o maquetación (para lo que también necesitamos CSS).

Para minimizar el efecto de la incompatibilidad de algunos formatos de imagen se dispone del elemento **picture** que se explica más adelante.

18. Inserción de imágenes.

```

```

img es una etiqueta sin cierre, la imagen se coloca directamente en la posición de esta etiqueta. Se la considera como texto al maquetarla. De modo que aparece entre el texto si ponemos la etiqueta entre el texto. Colocar de forma adecuada las imágenes respecto al texto es todo un arte que se consigue (una vez más) con ayuda de CSS.

Atributos:

atributo	significado
alt	Obligatorio. Indica un texto alternativo. Ese texto aparece cuando la imagen no se ha podido cargar (o durante la carga). También suele aparecer cuando arrimamos el cursor a la imagen a fin de informarnos sobre ella. Es un texto también tenido en cuenta por los buscadores a fin de identificar lo que muestra la imagen. Deberíamos tomarnos este atributo como obligatorio
width	Anchura de la imagen. No es aconsejable su uso, ya que si la ampliamos no se verá en buena calidad y si la reducimos estaremos cargando una imagen grande para luego mostrarla en pequeño; sería más inteligente reducirla primero con un editor de imágenes. En cualquier caso es importante utilizar este atributo (junto con height) para que el navegador sepa de antemano el tamaño de la imagen y así que prepare la página correctamente. De este modo si la imagen no se carga, al menos veremos el rectángulo que la misma ocuparía y la página no se desbarata.
height	Altura de la imagen. Tiene las mismas connotaciones que el atributo anterior.

19. Elemento picture.

HTML5 introdujo un nuevo elemento llamado **picture** para dar mayor versatilidad al uso de imágenes. El navegador **Internet Explorer** no reconoce este elemento, pero no es un problema, ya todos los elementos picture incorporan dentro elementos de tipo **img**.

El elemento **picture** se trata, en realidad, de un contenedor que dentro posee elementos capaces de mostrar imágenes. Concretamente puede contener elementos de tipo **source** y elementos de tipo **img**.

La idea es mostrar un tipo u otro de imágenes dependiendo de una serie de condiciones. Por ejemplo, podremos mostrar imágenes de alta resolución si la pantalla las soporta o mostrar imágenes en formatos especiales si el navegador las reconoce.

Este tipo de acciones, antes de la aparición de HTML5, estaban relegadas a JavaScript. Con picture podemos realizarlas desde el propio HTML.

20. Elemento source.

Permite mostrar medios (imágenes, audio, vídeo, etc.) en una página web. Si se usa dentro de picture puede utilizar hasta 3 atributos:

- **srcset.** Dirección al archivo de imagen que queremos mostrar.
- **type.** Tipo MIME del recurso. Los tipos MIME son indicaciones estándar reconocibles por navegadores (y por servidores web) que hacen referencia al tipo de información que queremos mostrar. Si el tipo no se reconoce, entonces no se mostrará.
- **media.** Permite consultar el tipo de medio en el que se está mostrando la página (pantalla, impresora,etc.) e incluso su tamaño, resolución, orientación, etc. Más adelante se describirá como utilizar este atributo.

21. Uso de picture.

```
<picture>
  <source srcset="logo.webp" type="image/webp">
  <source srcset="logo.png" type="image/png">
  
</picture>
```

Inicialmente se intenta mostrar la imagen con formato **webp**. Si el navegador no reconoce este formato, entonces se mostrará imagen **png**. Si el navegador no reconociera el formato **png** (hecho muy1 improbable hoy en día) se mostraría la imagen **jpg**.

Hay que observar que el último elemento siempre es una imagen dentro de un elemento img, así incluso los navegadores que no comprenden la etiqueta picture, mostrarán correctamente la imagen.

Los atributos de imagen (**alt**, **width**, **height**, etc.) se indican solo en la última etiqueta. De modo que, en el caso del código anterior, independientemente del tamaño original de las imágenes en formato png o webp, se mostrarán con un tamaño de 500x300 (indicado en la etiqueta img).

No solo se utiliza picture para usar diferentes formatos. También usar diferentes imágenes en función del tipo de pantalla que usa el usuario. Ejemplo:

```
<picture>
  <source srcset="logo-1000.png"
          media="(min-width:1000px)">
  <source srcset="logo-600.png"
          media="(min-width:600px)">
  
</picture>
```

En este caso, se muestra la imagen *logo-1000.png* solo si la anchura de la ventana del usuario en la que se muestra la página mide más de 1000 píxeles. Se mostrará la imagen *logo-600.png* en ventanas de 600 a 999 píxeles de anchura y si mide menos de 600 se mostrará la imagen *logo.png*.

En este caso la ventaja es cargar imágenes de diferente resolución (y por lo tanto de diferente tamaño en disco) en función del medio que posee el usuario. Parece absurdo cargar una imagen de mucha resolución para pantallas de baja, pero sí es lógico hacerlo en pantallas de alta resolución.

El atributo media es muy poderoso y se discutirá su uso más adelante ya que implica conocimientos de lenguaje CSS.

22. Mapas de imagen.

Se trata de una técnica que permite dividir en trozos a una imagen, de modo que podremos asignar un enlace a cada trozo.

Se utiliza en mapas propiamente dichos, en los que el usuario selecciona regiones del mapa y en imágenes donde hay elementos claramente destacados que al hacer clic nos permitan ir a una nueva URL.

Hoy en día está en desuso ya que el uso del lenguaje JavaScript ofrece capacidades muy superiores a los mapas de imágenes; pero la ventaja de los mapas es que no requieren conocer ningún lenguaje aparte de HTML.

ELEMENTO MAP

Los mapas se basan en una imagen previamente colocada en la página web mediante la etiqueta **img**. A partir de esa imagen se crea un elemento nuevo en la página de tipo **map**.

El elemento **map** al menos debe indicar estos atributos:

- **name**. Nombre del mapa de imágenes. Identifica el mapa de imágenes (no se debe de utilizar para este fin el atributo **id**)

ATRIBUTOS PARA MAPAS EN EL ELEMENTO IMG

Por otro lado, en la etiqueta **img** asociada a la imagen del mapa se debe utilizar este atributo:

- **usemap**. A la que se le da como valor el nombre del mapa (el que se utilice en el atributo **name** del elemento **map**), pero anteponiendo el símbolo **#**.

También es posible usar un segundo atributo (apenas se usa en la práctica)

- **ismap**. Se le puede dejar sin valor o ponerle valor **true**, en cuyo caso se está diciendo que el mapa de imágenes es un mapa del lado del servidor. Es decir será el servidor de la imagen el que maneje los clics en la imagen que haga el usuario. Para ello debemos saber crear imágenes de servidor, algo que va más allá de la pretensión de este manual.

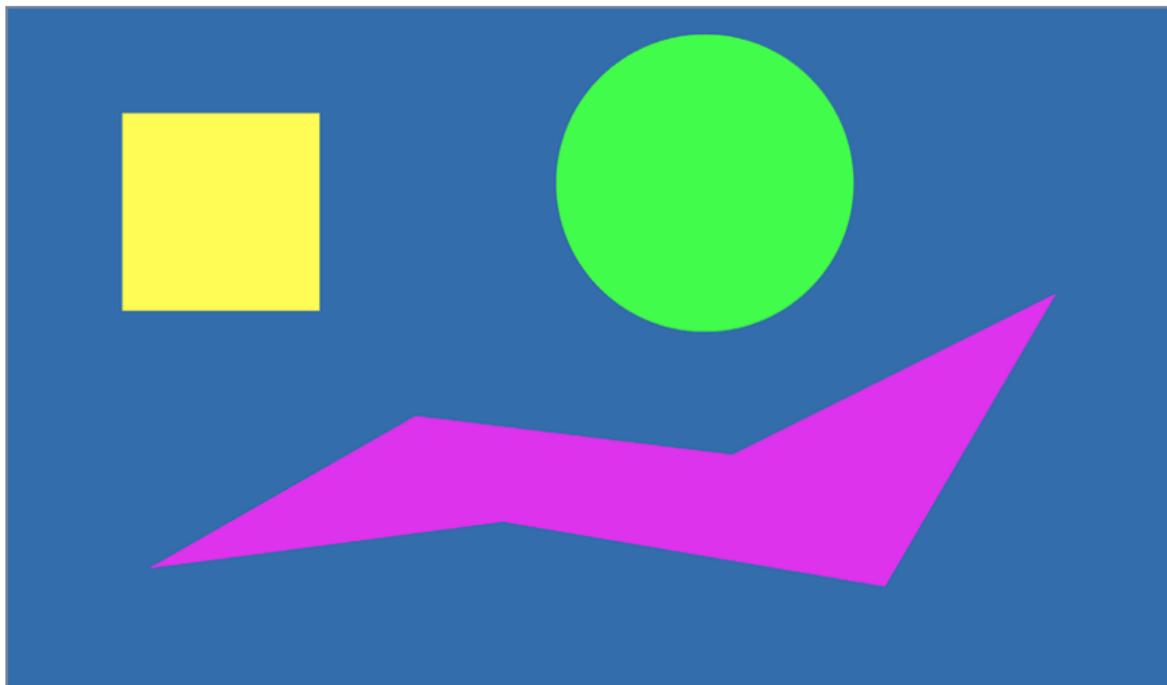
ELEMENTO AREA

Con los elementos **img** y **map** sólo se está indicando que la imagen es un mapa, pero falta decir qué zonas tiene el mapa y qué hacer cuando se hace clic en ellas. Esa es la labor de los elementos de tipo **area**.

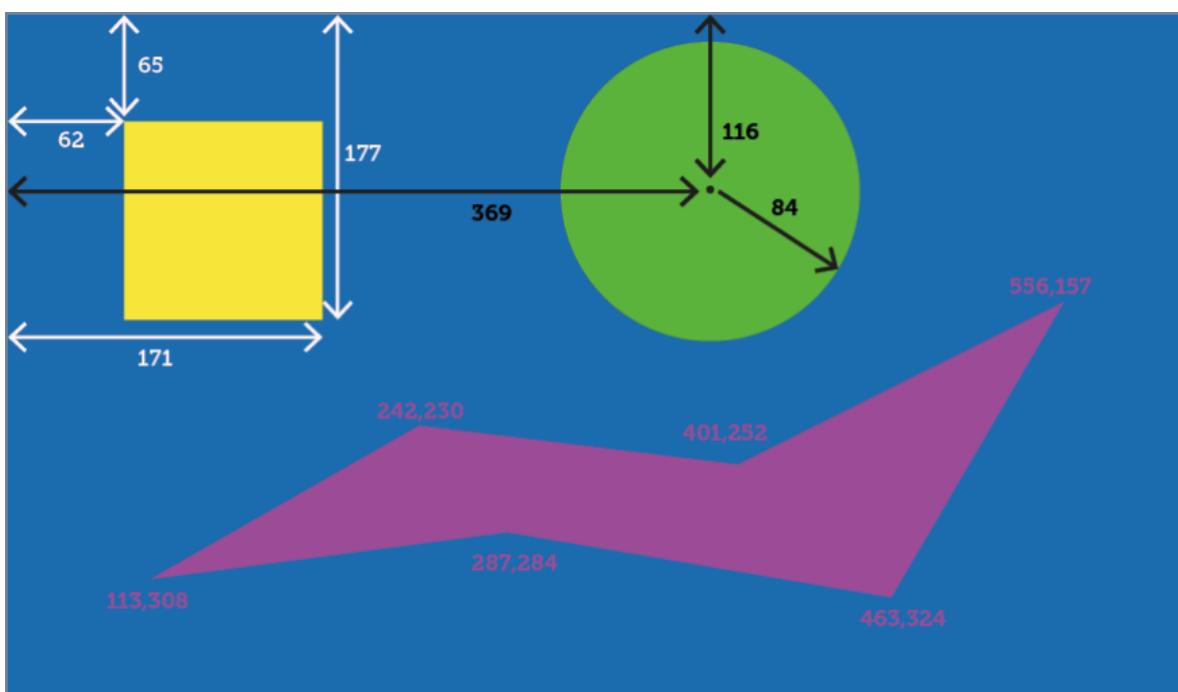
La etiqueta **map** debe contener dentro tantos elementos **area** como secciones en el mapa queramos crear. En cada elemento **area** rellenaremos estos atributos:

- **coords**. Coordenadas del área. La forma de indicarlas depende de lo que se haya elegido en el atributo **shape**.

- **shape.** Permite indicar la forma del área. Puede ser:
 - **shape="rect".** El área será rectangular. En este caso el atributo **coords** indicará cuatro coordenadas. Las dos primeras marcan la coordenada X y la coordenada Y de la esquina superior izquierda de la imagen, las dos últimas las coordenadas X e Y de la esquina inferior derecha. Las coordenadas cuentan desde la esquina superior izquierda de la imagen (que serían las coordenadas 0,0).
 - **shape="circle".** El área es una región circular en la imagen. El atributo **coords** tendría tres coordenadas: las dos primeras son la coordenada X e Y del centro del círculo y la tercera el radio del mismo.
 - **shape="poly".** Permite indicar una región poligonal. El atributo **coords** permite indicar las coordenadas X e Y de cada punto del polígono. Al final se cerrará el polígono.
 - **area="default".** Hace lo mismo que la opción **rect**.
- **alt.** Texto alternativo en el área de la imagen. Funciona igual que el atributo **alt** de la etiqueta **img**. No es obligatorio, pero sí muy interesante.
- **href.** Funciona como el atributo del mismo nombre en la etiqueta **a**: indica una URL a la que se dirigirá el navegador cuando el usuario haga clic sobre la imagen. Es decir cada área del mapa de imagen será un enlace cuyo destino lo marca este atributo (que tiene las mismas posibilidades que el atributo **href** del elemento **a**)



Si deseamos que al hacer clic en el cuadrado, el círculo y la forma libre saltemos a una URL, necesitamos conocer las coordenadas que tienen estas formas (en esto nos pueden ayudar los programas de retoque gráfico como Adobe *Photoshop* o *Adobe Fireworks*). En esta imagen se muestran las coordenadas necesarias:



```

<map name="formas" id="formas">
    <area shape="rect" coords="62,65,171,177"
          href="rectangulo.html" alt="Rectángulo">
    <area shape="circle" coords="369,116,84"
          href="circulo.html" alt="Círculo">
    <area shape="poly"
          coords="113,308,242,230,401,252,556,157,463,324,287,284"
          href="poligono.html" alt="Polígono" />
</map>
```

El uso del atributo **id** en el elemento **map** no es obligatorio para que el mapa funcione, pero sí es necesario para cumplir las nuevas normas HTML donde se intenta retirar el atributo **name** para sustituirle por **id** (aunque los mapas siguen funcionando con el atributo **name**); es decir se aconseja para una posible compatibilidad futura.

Se trata de una técnica en desuso debido a la mayor potencia que añade JavaScript para realizar acciones interactivas sobre imágenes y a la mayor facilidad que poseen las imágenes SVG para detectar áreas interactivas.

No obstante, crear mapas de imágenes mediante elementos **map** sigue siendo una técnica compatible con todos los navegadores del mercado por lo que aun se pueden observar páginas web en las que se aplica esta técnica.

Ejercicio enlaces e imágenes.

Crea una página HTML similar a:



Cada imagen debe enlazar a alguna página que deseas. Las tres imágenes deben estar centradas (pista: atributo **style**).

23. Tablas.

Una tabla normal se entiende que es un conjunto de datos presentados en forma de filas y columnas. En HTML las tablas tienen muchas posibilidades, lo que ha permitido, tradicionalmente, ser utilizadas como elemento de maquetación de páginas complejas.

En especial permiten unir varias celdas para formar tablas con una disposición irregular, lo que permite realizar acabados muy complejos en las páginas. Aún más: se permite incluso colocar una tabla dentro de una celda lo que nos proporciona aun más libertad en la maquetación de páginas.

Sin embargo, hoy en día se recomienda no usar las tablas de esta forma ya que las mejoras que proporciona CSS en el uso de capas, permiten olvidar a las tablas como recurso de maquetación.

El uso actual de las tablas es más semántico: indicar una disposición en celdas las cuales se marcan también de forma semántica (celdas de cabecera, celdas normales, filas, columnas, pie de tabla, título de tabla, etc.).

Equipo	Ciudad	Títulos NBA
Lakers	Los Ángeles	16
Celtics	Boston	17
Spurs	San Antonio	5
Bulls	Chicago	6

24. Tablas simples.

En HTML cada tabla está asociado a un elemento **table**, dentro de este elemento se indican las filas mediante el elemento **tr** y dentro de cada fila se indican las celdas mediante elementos **td**.

En HTML 5 las etiquetas **table** y **tr** no tienen atributos, aunque hay atributos, como por ejemplo el atributo **border**, que reconocen los navegadores por estar presentes en versiones anteriores de HTML. Es, como siempre, el lenguaje CSS el que determina la apariencia y formato de la tabla.

Ejemplo 18_tablasSimples.html

Ejemplo 19_tablasCabecera.html

Ejemplo 20_tablasConTitulo.html

25. Agrupación de filas.

Hay tres elementos HTML que sirven para diferenciar las tres partes principales de una tabla, son:

- **thead**. Sirve para indicar las filas que forman la cabecera de la tabla
- **tfoot**. Indica el pie de la tabla
- **tbody**. Indica el cuerpo de la tabla

De esa forma se podrá más adelante dar formato diferencial a cada parte.

Ejemplo 21_tablasAgrupacionFilas.html

26. Agrupación de columnas.

elemento colgroup

Permite realizar agrupaciones de columnas de una tabla con la finalidad de dar un formato homogéneo a cada grupo de columnas indicado. El único atributo permitido en HTML 5 es **span** que sirve para indicar el número de columnas que agrupamos.

Otra posibilidad, más recomendable, es usar dentro de **colgroup** elementos **col** los cuales dan un mayor sentido y son más versátiles.

Esta etiqueta debe de ser la primera de una tabla excepto si se usa título en la tabla (etiqueta **caption**) en cuyo caso será la segunda.

elemento col

Se usa dentro de la anterior para dar propiedades a cada grupo de columnas realizado. Permite el uso del atributo col para indicar cuántas columnas se incluyen en el grupo.

Ejemplo 21_tablasAgrupacionColumnas.html

27. Combinar celdas.

Es posible unir celdas y de esta forma conseguir tablas de formas caprichosas que permiten una maquetación más poderosa.

Las etiquetas de columna (**td** y **th**) son las que poseen los atributos que permiten esta operación. En concreto son los atributos:

atributo	significado
colspan	Combina la celda actual con el número de celdas a la derecha que se indique. Por ejemplo <code>colspan="3"</code> une esta celda con las dos que tiene a su derecha, formando una combinación de tres celdas en horizontal.
rowspan	Combina la celda actual con el número de celdas hacia abajo que se indique. Por ejemplo <code>rowspan="3"</code> une esta celda con las dos que tiene hacia abajo, formando una combinación de tres celdas en vertical.

Hay que tener en cuenta que se pueden usar ambos atributos a la vez, lo que permite obtener celdas combinadas de otras columnas y filas a la vez.

Ejemplo 23_tablasCombinadas.html

28. Formato CSS para tablas.

Las propiedades CSS específicas para tablas son:

- **border-collapse**. Permite indicar cómo manejar los bordes adyacentes entre elementos. Posibles valores:
 - **collapse**. Se unen los bordes adyacentes para formar un único borde. Es decir no hay espacio entre bordes adyacentes.
 - **separate**. Los bordes adyacentes se mantienen separados. Es el valor por defecto.

```
table{
    width:100%;
}
td{
    border:2px solid black;
}
```


```
table{
    width:100%;
    border-collapse:collapse;
}
td{
    border:2px solid black;
}
```


Hay que fijarse en que es en el elemento **table** (el contenedor general de la tabla) en el que se indica esta propiedad.

border-spacing. Indica el espacio entre bordes. Admite dos medidas, la primera se utiliza para la distancia horizontal y la segunda para la vertical. Si se utiliza una medida se referirá a ambas distancias.

border-spacing:10px 5px;

-
- **caption-side**. Posición del título de la tabla (elemento **caption**). Posibilidades: **top** (arriba, valor por defecto), **bottom** (abajo).
- **empty-cells**. Indica si deseamos mostrar las celdas vacías de la tabla. Valores: **show** (mostrar, valor por defecto) y **hide** (ocultar). No funciona (**hide**) con los bordes colapsados.

- **table-layout.** Propiedad presente en los últimos navegadores (no en **Internet Explorer**) permite indicar la forma en la que se calcula la anchura de las celdas. Por defecto toma el valor **auto**, en el que, aunque hayamos indicado una anchura fija para una celda, el tamaño de la misma depende del contenido. **fixed**, sin embargo deja la tabla con el tamaño fijado por las propiedades de anchura de columna (**width**).

El problema es cuando un contenido sobrepasa el tamaño de una celda (una imagen por ejemplo). En el primer caso la celda se amplía para dar cabida al contenido. Eso significa que el navegador no puede renderizar la tabla hasta no haber cargado todo su contenido.

En el segundo caso la tabla se dibuja al instante aunque los contenidos no hayan llegado. Aunque en ese caso esos contenidos sobresalgan de la tabla. Solo es interesante **fixed** si sabemos seguro el tamaño de la tabla por adelantado.

Ejemplo 24_tablasEspaciado.html

- **caption-side.** Propiedad utilizada para la etiqueta **caption** que contiene el título de una tabla. Permite indicar si queremos que el título esté por encima de la tabla (valor **top**, valor por defecto) o por debajo (valor **bottom**)
- **empty-cells.** Permite especificar si los bordes y sombreados de la tabla se aplican a las celdas vacías. Valores:
 - **show.** Se aplican bordes y sombreados a esas celdas (valor por defecto)
 - **hide.** No se aplican los bordes y sombreados a las celdas vacías.
- **table-layout.** Indica la forma en la que se determina la anchura de la tabla y las celdas. Posibles valores:
 - **auto.** Valor por defecto. Aunque hayamos indicado anchura para tabla y/o celdas, su tamaño se determina por el contenido de las mismas (si el contenido requiere mayor anchura, así se hará).
 - **fixed.** Predomina la anchura establecida por las propiedades y no por el contenido

Ejercicio tablas

Reproduce la siguiente página HTML:

					
Capacidad	2GB	4GB	8GB	32GB	2GB
Colores	Blanco	• Negro • Rosa • Dorado	Negro	• Negro • Blanco	
Pantalla	3 pulgadas			7 pulgadas	
Tiempo de Carga	1 hora			5 horas	
				30 minutos para 75%	

29. Bibliografía.

- <https://www.w3schools.com/>
- <https://desarrolloweb.com/>
- <http://jorgesanchez.net>
- <https://www.campusmvp.es>
- https://oscarmaestre.github.io/lenguajes_marcas/index.html