

## HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), 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, etc. 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.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.).

### **HTML es la "lengua materna" de tu navegador.**

HTML es un lenguaje que hace posible presentar información en Internet. Lo que ves al visualizar una página en Internet es la interpretación que hace el navegador del código HTML.

¿Para qué puedo usar HTML?

Si quieres crear sitios web, no hay otra solución que aprender HTML.

HTML es la abreviatura de "HyperText Mark-up Language", es decir, "Lenguaje de marcado hipertextual".

**Lenguaje es lo que es HTML. Este lenguaje hace uso de muchos términos en inglés.**

## Elementos y etiquetas

Los elementos proporcionan la estructura al documento HTML e indican al navegador cómo quieres que se presente tu sitio web. Por lo general, los elementos están formados por una etiqueta de inicio, el contenido, y una etiqueta de cierre.

"¿Etiquetas?"

Las etiquetas son marcas que se usan para señalar el inicio y el fin de un elemento.

Todas las etiquetas comparten el mismo formato: empiezan con el signo menor que "<" y terminan con el signo mayor que ">".

Por lo general, hay dos tipos de etiquetas: la etiquetas de inicio, por ejemplo, <html>, y las etiquetas de cierre, por ejemplo, </html>. La única diferencia entre la etiqueta de inicio y la de cierre es la barra oblicua "/". El contenido queda etiquetado al colocarlo entre una etiqueta de inicio y una etiqueta de cierre.

HTML trata esencialmente de elementos. Aprender HTML consiste en aprender a usar diferentes etiquetas.

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas.

Ejemplo: <nombre de elemento atributo="valor">Contenido</nombre-de-elemento>.

**HTML es sencillo y lógico. El navegador lee el código HTML igual que lo haces tú: de arriba abajo y de izquierda a derecha. Así pues, todo documento HTML empieza con lo que debería aparecer en primer lugar en la página y termina con lo que debería ir en último lugar.**

Lo primero que tienes que hacer es decirle al navegador que le "hablarás" usando el lenguaje HTML. Esto se consigue con la etiqueta <html>. Así que antes de hacer cualquier otra cosa, escribe "<html>" en la primera línea del documento creado con el Bloc de notas.

Lo siguiente que necesita el documento es una "cabecera", que proporcionará información relativa a este mismo documento, y un "cuerpo" donde se incluirá el contenido del documento. Puesto que HTML es de lo más lógico, la cabecera (es decir, las etiquetas <head> y </head>) se encuentra encima del cuerpo (o sea, encima de las etiquetas <body> y </body>).

## Documento HTML básico

El documento debería presentar ahora este aspecto:

```
<html>

    <head>

    </head>

    <body>

    </body>

</html>
```

Hay que prestar atención a cómo hemos estructurado las etiquetas introduciendo nuevas líneas (usando la tecla Enter) al igual que hemos usado sangrías (utilizando la tecla Tab). En principio, da igual cómo estructuras el documento HTML; sin embargo, para que te sirva de ayuda tanto para ti y como para otros a la hora de interpretar el código fuente, y con el fin tener una buena visión de conjunto, es recomendable estructurar el código HTML de una forma nítida, con saltos de línea y sangrías.

### ¿Cómo añado contenido a mi sitio web?

**Todo documento HTML tiene dos partes: un encabezado y un cuerpo. En la sección del encabezado se incluye información sobre la propia página, mientras que el cuerpo contiene la información que constituye la página en sí.**

Por ejemplo, si quieres poner a la página un título que aparecerá en la barra de títulos del navegador, deberías hacerlo en la sección del encabezado, es decir dentro de las etiquetas "head". El elemento que se usa para el título es title, o sea, escribe el título de la página entre la etiqueta de inicio <title> y la etiqueta de cierre </title>:

```
<title>Mi primer sitio web</title>
```

Ten en cuenta que este título no aparecerá en la página en sí. Cualquier cosa que quieras que aparezca en la página es contenido y debe, por la tanto, añadirse entre las etiquetas <body> y </body>.

```
<p>Este es mi primer sitio web.</p>
```

La p de <p> es la abreviatura de "párrafo", que es exactamente lo que es: un párrafo de texto.

Lo siguiente que hay que hacer es guardarlo en el disco duro y luego abrirlo en el navegador:

En el Bloc de notas elige "Guardar como..." del menú "Archivo" en el menú principal.

Elige "Todos los archivos" en la caja "Tipo". Esto es muy importante, pues, de lo contrario, se guardará como documento de texto y no como un documento HTML.

Ahora guarda el documento como "pagina1.html" (la terminación ".html" indica que es un documento HTML).

Para visualizar el documento HTML, es conveniente ir a la carpeta donde se ha guardado, usando el Explorador de Windows, y hacer doble clic sobre el archivo. Para abrir con otros programas navegadores, o el Bloc de notas para editar, hay que hacer clic derecho sobre el archivo, elegir Abrir con... y seleccionar un programa de la lista.

Empieza siempre con la plantilla básica siguiente:

```
<html>
  <head>
    <title> </title>
  </head>
  <body>
  </body>
</html>
```

En la sección del encabezado, incluye siempre un título: <title> El título de la página </title>. Fíjate cómo aparece el título en la esquina superior izquierda del navegador: el título es especialmente importante porque los motores de búsqueda (por ejemplo, Google) lo utilizan para indexar tu sitio y porque aparece reflejado en los resultados de la búsqueda.

En la sección del cuerpo, se escribe el contenido propiamente dicho de la página. Algunos de los elementos más importantes son:

#### Párrafos <p> ...

El texto de una página puede agruparse en párrafos. Para ello, el texto de cada uno de los párrafos debe insertarse entre las etiquetas <p> y </p>.

También es posible cambiar la alineación del texto de cada párrafo. Para ello se modifica el valor del atributo align, cuyos valores pueden ser left (izquierda), right (derecha), center (centrado) o justify (justificado).

Por ejemplo, para insertar el texto: Bienvenidos a mi página.

```
<p align="center">Bienvenidos a mi página. </p>
```

## Encabezados <h1> ...

Existen una serie de encabezados que suelen utilizarse para establecer títulos dentro de una página. La diferencia entre los distintos tipos de encabezado es el tamaño de la letra, el tipo de resaltado, y la separación existente entre el texto y los elementos que tiene encima y debajo de él. Hay que tener en cuenta que el navegador del usuario es el que aplicará el estilo del encabezado por lo que el mismo título se puede visualizar de forma diferente según el navegador.

Existen seis etiquetas que representan seis tipos de cabeceras distintas, desde <h1> hasta <h6>. Todas estas etiquetas precisan una etiqueta de cierre.

Ejemplo: <h1>Título principal de la página</h1>

*Para que el texto pase al renglón siguiente (o para dejar un renglón en blanco), se utiliza la etiqueta <br />. Esta etiqueta no tiene etiqueta de cierre.*

## Texto con sangrías <blockquote>

La sangría es una especie de margen que se establece a ambos lados del texto. Para que un texto aparezca separado de los bordes, se inserta entre las etiquetas <blockquote> y </blockquote>.

El uso de la etiqueta <blockquote> obliga a que el texto aparezca en una nueva línea.

Insertando el texto entre varias etiquetas <blockquote> y </blockquote> se consigue que los márgenes sean mayores.

## Separadores <hr>

El elemento que suele utilizarse para separar secciones dentro de una misma página es la regla horizontal. Para insertar una regla horizontal hay que insertar la etiqueta <hr>. Dicha etiqueta no precisa ninguna etiqueta de cierre.

Es posible especificar algunos atributos de la regla horizontal:

Atributo	Significado	Posibles valores
<b>align</b>	alineación de la regla dentro de la página	<b>left</b> (izquierda) <b>right</b> (derecha) <b>center</b> (centro)
<b>width</b>	ancho de la regla	un número, acompañado de % cuando se desee que sea en porcentaje
<b>size</b>	alto de la regla	un número
<b>noshade</b>	eliminar el sombreado de la regla	no puede tomar valores

## Listas.

Otros ejemplos de elementos son: ul, ol y li. Estos tres elementos se usan para crear listas.

ul es la forma abreviada de "unordered list" (es decir, lista no ordenada) e inserta viñetas para cada elemento. ol es la abreviatura de "ordered list" (es decir, lista ordenada) y numera cada elemento de la lista. Para crear elementos de lista use la etiqueta li ("list item", o sea, elemento de lista). Parece algo confuso, ¿no? Veamos, entonces, algunos ejemplos:

Ejemplo (lista no ordenada):

```
<ul>  
  <li>Un elemento de lista</li>  
  <li>Otro elemento de lista</li>  
</ul>
```

Ejemplo (lista ordenada – numerada):

```
<ol>  
  <li>Primer elemento de lista</li>  
  <li>Segundo elemento de  
lista</li>  
</ol>
```

## Atributos

Los elementos sirven para estructurar el contenido de todo el documento HTML e indican al navegador cómo debería presentarse el sitio web. En algunos elementos se puede añadir más información. Esa información extra se denomina atributo. Es posible añadir atributos a determinados elementos.

Los atributos se incluyen siempre dentro de la etiqueta de inicio y llevan inmediatamente detrás el signo de igual ("=") con los valores de dichos atributos entre comillas ("").

Al igual que existen muchos elementos, existen también muchos y diferentes atributos. Algunos atributos están hechos a la medida para un elemento concreto, mientras que otros se pueden usar para muchos elementos diferentes. Y viceversa: algunos elementos sólo pueden contener un tipo de atributo mientras otros pueden contener muchos.

Ejemplos:

- ✓ `<body bgcolor="#ff0000">`
- ✓ `<body bgcolor="orange">`
- ✓ `<table bgcolor="yellow">`

## Formatear el texto <font> ...

Las propiedades del texto pueden modificarse a través de la etiqueta `<font>`. Para ello, podemos insertar el texto entre las etiquetas `<font>` y `</font>`, especificando algunos de los atributos de la etiqueta:

Atributo	Significado	Posibles valores
<b>face</b>	fuelle	nombre de la fuente, o fuentes
<b>color</b>	color del texto	número hexadecimal o texto predefinido
<b>size</b>	tamaño del texto	valores del 1 al 7, siendo por defecto el 3, o desplazamiento respecto al tamaño por defecto, añadiendo + o - delante del valor

Por ejemplo, para insertar el texto:

Bienvenidos a [www.google.com](http://www.google.com)

Habría que escribir:

```
<font color="#993366" size="4" face="Comic Sans MS, Arial, MS Sans Serif">Bienvenidos a  
www.google.com</font>
```

## Comentarios

En ocasiones podemos desear añadir comentarios aclaratorios dentro del código, de manera que no sean visualizados en el navegador, pero sí a la hora de editar el documento.

Para insertar comentarios dentro del código, basta con insertar el texto entre `<!-- y //-->`.

Todo el código que se inserte entre estos símbolos no será visualizado en los navegadores.

Por ejemplo, para insertar el texto siguiente con un comentario:

¡Bienvenidos, esta es mi primera página!

Habría que escribir:

```
<!-- A continuación aparecerá una línea de texto!-->
```

¡Bienvenidos, esta es mi primera página!



## Enlaces

### *¿Qué necesito para crear un enlace?*

Para crear enlaces se usa lo que siempre usamos al codificar HTML: un elemento. Un sencillo elemento con un atributo y podrás enlazar con cualquier cosa y con todo.

El elemento `<a>` es la abreviatura de "anchor" (ancla, en español). Y el atributo `href` es la abreviatura de "hypertext reference" (es decir, referencia hipertextual), que especifica hacia dónde nos dirige el enlace; normalmente, una dirección de internet o un nombre de un archivo.

Vas a ver ahora mismo un ejemplo de cómo se codificaría un enlace a Gmail:

```
<a href="http://www.gmail.com/">Desde aquí enlazamos con tu cuenta de Google</a>
```

En el ejemplo anterior el atributo `href` contiene el valor `"http://www.gmail.com"`, que es la dirección completa de la página de GMAIL y que se denomina URL (Uniform Resource Locator; en español, localizador de recursos uniforme). Ten en cuenta que `"http://"` siempre debe incluirse en las URLs. El texto `" Desde aquí enlazamos con tu cuenta de Google "` es el que aparece en el navegador como enlace.

### *¿Y se pueden crear enlaces entre mis propias páginas?*

Si quieres crear enlaces entre las páginas de un mismo sitio web, no es necesario incluir la dirección completa (el URL) del documento. Por ejemplo, si has creado dos páginas (llamémoslas `pagina1.htm` y `pagina2.htm`) y las has guardado en la misma carpeta, puedes enlazar una página con la otra con sólo escribir el nombre del fichero en el enlace. Si este es el caso, enlazar la `pagina1.htm` con la `pagina2.htm` tendría el siguiente aspecto:

```
<a href="pagina2.htm">Haz clic para ir a la página 2</a>
```

Si la página 2 estuviera en una subcarpeta (llamada "subcarpeta"), el enlace tendría el siguiente aspecto:

```
<a href="subcarpeta/pagina2.htm">Haz clic para ir a la página 2</a>
```

En sentido inverso, enlazar desde la página 2 (en la subcarpeta) a la página 1 se haría usando con el siguiente código:

```
<a href=" ../pagina1.htm">Enlace a la pagina1</a>
```

Los dos puntos con la barra, `"../"`, hacen referencia a una carpeta que se encuentra un nivel por encima de donde se encuentra el fichero desde el que se crea el enlace. Siguiendo este mismo sistema, se puede señalar a dos (o más) carpetas en niveles superiores escribiendo `"../../"`.

¿Y se pueden crear enlaces internos dentro de una misma página?

Una tabla de contenidos en la parte superior de la página con enlaces a cada capítulo que se encuentra más abajo puede ser muy útil. Todo lo que necesitas es usar un atributo llamado id (identification, en inglés; es decir, identificador) y el símbolo "#", conocido como almohadilla.

Usa el atributo id para marcar el elemento con el que quieras enlazar. Por ejemplo:

```
<h1 id="encabezado1">Primer encabezado</h1>
```

Ahora ya puedes crear un enlace a dicho elemento usando la almohadilla "#" en el atributo "href" del elemento a. El símbolo "#" debe ir seguido del id de la etiqueta con la que quieres enlazar. Por ejemplo:

```
<a href="#encabezado1">Enlace con el primer encabezado</a>
```

También puedes crear enlaces a una dirección de correo electrónico. Se hace casi igual que cuando se crea un enlace a un documento:

```
<a href="mailto:domingonozzi@gmail.com">Envíe un mensaje a Domingo Nozzi</a>
```

Nota: al hacer clic en el vínculo se abrirá el programa predeterminado para enviar mensajes de correo electrónico, como por ejemplo: Outlook.

¿Hay más atributos que debería conocer?

Para crear un enlace, siempre tienes que usar el atributo href. Además, también puedes poner un título (atributo title) al enlace:

```
<a href="http://www.google.com.ar/" title="Visita Google y obtiene todas tus respuestas">Página de Google en Argentina</a>
```

El atributo title se usa para incluir una breve descripción del enlace. Si se coloca el cursor sobre el enlace -sin pulsarlo- aparecerá el texto.

### Destino del enlace

El destino del enlace determina en qué ventana va a ser abierta la página vinculada, se especifica a través del atributo target al que se le puede asignar los siguientes valores:

\_blank: Abre el documento vinculado en una nueva ventana o pestaña del navegador.

\_self: Es la opción predeterminada. Abre el documento vinculado en el mismo marco o ventana que el vínculo.

Para insertar el enlace: Visita [www.youtube.com](http://www.youtube.com) en una ventana nueva habría que escribir:

```
<a href="http://www.youtube.com" target = "_blank">Visita www.youtube.com en una ventana nueva</a>
```

Es interesante utilizar esta opción cuando la página de destino está fuera de nuestro sitio para que cuando el usuario cierre la ventana del explorador, se encuentre automáticamente en la página desde la que había salido (que vuelva a nuestro sitio).

Nota: según el navegador utilizado, la página puede mostrarse en una nueva ventana o en una nueva pestaña.

### Formato de los enlaces

En general, un texto que tiene un vínculo asociado suele aparecer subrayado. Normalmente los vínculos de texto cambian de color cuando el enlace ha sido ya pulsado o cuando el puntero del ratón se posiciona sobre él, estos cambios están predefinidos en cada navegador, pero nosotros podemos cambiar esos colores.

Los colores de los vínculos pueden especificarse a través de las propiedades de la página, en la etiqueta <body>. Estos colores se asignan a través de los atributos link (vínculo), alink (vínculo activo), y vlink (vínculo visitado).

- ❖ link permite determinar el color de los enlaces sin visitar (enlace que no ha sido pulsado ninguna vez).
- ❖ alink permite determinar el color del enlace activo (enlace que acaba de ser pulsado).
- ❖ vlink permite determinar el color de los enlaces visitados (enlaces que ya han sido pulsados).

Por ejemplo, al insertar el siguiente código:

```
<body link="#FF0000" vlink="#FF0099" alink="#FF9900">
```

Mientras no se visite la página [www.google.com](http://www.google.com), el enlace será de color rojo (#FF0000), cuando la página [www.google.com](http://www.google.com) sea la última visitada, el enlace será de color fucsia (#FF0099) y cuando se haya visitado la página [www.google.com](http://www.google.com), el enlace será de color naranja (#FF9900).

### Vínculo a archivos para descarga

El valor del atributo href normalmente será una página web (con extensión htm, html, asp, php...) pero también puede ser un fichero comprimido, una hoja de Excel, un documento Word, un documento con extensión .pdf. Cuando el enlace no es a una página Web nos aparecerá el cuadro de diálogo que seguro habrás visto alguna vez en el que el navegador le pide al usuario permiso para descargar el archivo en su ordenador.

Por ejemplo, en la carpeta donde se encuentra esta página tenemos el documento de Word [carta.doc](#) y queremos que nuestros visitantes puedan visualizar e incluso descargar el archivo en su disco rígido, en este caso definiremos el enlace:

*Haz clic aquí para descargarte el documento de la siguiente forma:*

```
<a href="carta.doc" tarjet=_blank > haz clic aquí para descargarte el documento</a>
```

En este caso hemos utilizado una referencia relativa al documento ya que la carta se encuentra en la misma carpeta que nuestra página.

## Imágenes

Todas las páginas web acostumbran a tener un cierto número de imágenes, que permiten mejorar su apariencia, o dotarla de una mayor información visual.

Para insertar una imagen es necesario insertar la etiqueta <img>. Dicha etiqueta no necesita etiqueta de cierre.

El nombre de la imagen ha de especificarse a través del atributo src (en inglés source, que significa origen).

Ejemplo: 

Teniendo en cuenta que la imagen se llama yahoo.jpg y que está dentro de la carpeta imagenes, que se encuentra en el mismo directorio que el documento actual (referencia relativa al documento).

Para trabajar de una forma más sencilla y ordenada, es recomendable que todos los documentos html se encuentren en un mismo directorio, y que dentro de este directorio existan diferentes carpetas para agrupar otros objetos, como puede ser una carpeta destinada a almacenar imágenes, o una carpeta destinada a almacenar archivos de audio, etc.

Hay tres clases diferentes de tipos de archivos de imagen que puedes insertar en tus páginas:

- ✓ GIF (Graphics Interchange Format)
- ✓ JPG o JPEG (Joint Photographic Experts Group)
- ✓ PNG (Portable Network Graphics)

Normalmente, las imágenes GIF son mejores para gráficos y dibujos, mientras que las imágenes JPEG son mejores para fotografías. El formato PNG contiene, en muchos aspectos, lo mejor de ambos formatos: millones de colores y una compresión efectiva.

Las imágenes también se pueden usar como enlaces, ejemplo:

<a href="http://www.google.com.ar"> </a>

### *Texto alternativo*

Cuando una imagen no puede ser visualizada en el navegador, cosa que puede ocurrir al especificar mal el valor del atributo src, aparece un recuadro blanco con una X roja en su lugar, junto con el nombre de la imagen.

Podemos hacer que en lugar de mostrarse el nombre de la imagen aparezca el texto que nosotros deseemos, gracias al atributo alt.

Ejemplo: ``

El texto alternativo es muy útil a la hora de diseñar páginas más asequibles a los invidentes ya que los programas lectores son capaces de leer el texto alternativo.

El atributo title se puede usar para añadir información a la imagen.

Ejemplo: ``

Si colocas el cursor sobre la imagen -sin llegar a hacer clic- verás aparecer el texto "Correo electrónico de Google" en forma de caja desplegable.

### *Borde de una imagen*

En general, al visualizar una página en un navegador las imágenes aparecen sin ningún borde alrededor, pero es posible establecer uno a través del atributo border.

El atributo border puede tomar valores numéricos, que indican el grosor en píxeles del borde.

Por ejemplo, para insertar la siguiente imagen con borde, habría que escribir:

```

```

### *Tamaño de una imagen*

Cuando insertamos una imagen, esta se muestra en los navegadores con su tamaño original, pero por diversos motivos puede interesarnos modificar dicho tamaño.

A través de los atributos width (anchura) y height (altura) puede modificarse el tamaño de la imagen. Dicho cambio de tamaño no se aplica directamente sobre el archivo de imagen, sino que lo que varía es la visualización de la imagen en el navegador.

El valor que pueden tomar los atributos width y height ha de ser un número, acompañado de % cuando se desee que sea en porcentaje con respecto a la página.

Es conveniente mirar el tamaño original de la imagen, para escalarla proporcionalmente y evitar que se deforme. [Hacer una tabla de Excel para calcular valores posibles sin deformar la imagen.](#)

## Tablas

Es posible insertar tablas, filas y columnas, y modificar sus propiedades. Las tablas se usan cuando necesitamos mostrar información que se presenta de forma lógica en filas y columnas.

Resultan de gran utilidad al mejorar notablemente las opciones de diseño. Gracias a las tablas es posible distribuir el texto en columnas, colocar imágenes al lado de un bloque de texto.

Para insertar tablas se usan 3 elementos diferentes:

- ❖ La etiqueta de inicio `<table>` y la etiqueta de cierre `</table>` inician y finalizan la tabla.
- ❖ `<tr>` es la abreviatura de "table row" (es decir, fila de la tabla) e inicia y finaliza las filas horizontales.
- ❖ `<td>` es la abreviatura de "table data" (es decir, datos de la tabla). Esta etiqueta inicia y finaliza cada una de las celdas que componen las filas de la tabla. La cantidad de `<td>` y `</td>` que tenga la fila determinará el número de columnas.

### Atributos del elemento <table>

- ✓ El atributo `border` se usa para especificar el grosor del borde de la tabla.

Ejemplo: `<table border="1">`

- ✓ Se puede establecer el ancho de la tabla en píxeles, o, de modo alternativo, por medio de un porcentaje del ancho de la pantalla.

Ejemplo: `<table border="1" width="60%">`

- ✓ Dos atributos más:
  - ❖ `align`: especifica la alineación horizontal del contenido de toda la tabla, de una fila o de una sola celda. Por ejemplo, `left` (izquierda), `center` (centrada) o `right` (derecha).
  - ❖ `valign`: especifica la alineación vertical del contenido de una celda. Por ejemplo, `top` (arriba), `middle` (medio) o `bottom` (abajo).

Ejemplo: `<table align="center" valign="middle">`

- ✓ Los atributos `colspan` y `rowspan` se usan cuando queremos crear tablas elaboradas.

`Colspan` es la abreviatura de "column span" (expandir columna). `Colspan` se usa con la etiqueta `<td>` para especificar cuántas columnas ocupará la celda.

Ejemplo:

```
<table border="1">
  <tr>
    <td colspan="3">Celda 1</td>
  </tr>
  <tr>
    <td>Celda 2</td>
    <td>Celda 3</td>
    <td>Celda 4</td>
  </tr>
</table>
```

Al establecer el atributo colspan a "3", la celda de la primera fila ocupa tres columnas.

El atributo rowspan especifica cuántas filas debería ocupar una celda.

Ejemplo:

```
<table border="1">
  <tr>
    <td rowspan="3">Celda 1</td>
    <td>Celda 2</td>
  </tr>
  <tr>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
  </tr>
</table>
```

Esto especifica que la celda debe expandirse a lo largo de tres filas.

### Formato de la tabla

Es posible modificar los siguientes atributos de una tabla:

Atributo	Significado	Posibles valores
<b>width</b>	ancho de la tabla	un número, acompañado de % cuando se desee que sea en porcentaje
<b>height</b>	altura de la tabla	un número, acompañado de % cuando se desee que sea en porcentaje
<b>cellpadding</b>	espacio entre el contenido de las celdas y el borde	un número
<b>cellspacing</b>	espacio entre celdas	un número
<b>border</b>	grosor del borde	un número
<b>align</b>	alineación de la tabla dentro de la página	<b>left</b> (izquierda) <b>right</b> (derecha) <b>center</b> (centro)
<b>bgcolor</b>	color de fondo	número hexadecimal
<b>background</b>	imagen de fondo	<b>Ruta de la imagen</b>
<b>bordercolor</b>	color del borde	número hexadecimal

Ejemplos (no hace falta copiar esto, practicar todo después del próximo tema):

```
<table width="50%" border="2" align="center" cellspacing="0" bordercolor="#000000" bgcolor="#FFCC99"> ... </table>
```

Con esto indicamos que el ancho (width) de la tabla debe ocupar el 50% del ancho de la ventana (lo puedes comprobar cambiando el ancho de la ventana de tu navegador y verás que el ancho de la tabla varía para ocupar siempre la mitad de la ventana), que el borde (border) tiene un grosor de dos píxeles (es más ancho que las separaciones internas de la tabla que tienen grosor 1), que la tabla está alineada al centro (center) de la ventana, que no hay espacio entre las celdas (cellspacing="0"), que el borde la tabla es negro (#000000) y el fondo naranja (#FFCC99).

### Formato de las celdas

Es posible modificar los siguientes atributos de una celda:

Atributo	Significado	Posibles valores
<b>width</b>	ancho de la tabla	un número, acompañado de % cuando se desee que sea en porcentaje
<b>height</b>	altura de la tabla	un número, acompañado de % cuando se desee que sea en porcentaje
<b>align</b>	alineación horizontal del contenido de la celda	<b>left</b> (izquierda) <b>right</b> (derecha) <b>center</b> (centro)
<b>valign</b>	alineación vertical del contenido de la celda	<b>baseline</b> (línea de base) <b>bottom</b> (inferior) <b>middle</b> (medio) <b>top</b> (superior)
<b>bgcolor</b>	color de fondo	número hexadecimal
<b>background</b>	imagen de fondo	Ruta de la imagen
<b>bordercolor</b>	color del borde	número hexadecimal

También es posible modificar estos atributos de toda una fila, especificándolos en la etiqueta <tr>, en lugar de en la etiqueta <td>.

Hay que tener en cuenta que los atributos tienen más prioridad cuando son establecidos para una celda que cuando son establecidos para una fila completa. Al mismo tiempo, tienen más prioridad los atributos establecidos para una fila que los establecidos para toda la tabla.

### Combinar celdas

Para las etiquetas <td> y <th> existen los atributos colspan y rowspan, que se utilizan para combinar celdas.

A través del atributo colspan se especifica el número de columnas por las que se extenderá la celda, y a través del atributo rowspan se especifica el número de filas por las que se extenderá la celda.

Ejercitación: Hacer entre todas algunas tablas sencillas combinando primero celdas a lo ancho, luego celdas a lo alto y por último generar una tabla que contenga una celda con varias filas y columnas. Luego hacer el ejercicio Tablas.docx.

### Formulario <form>

Un formulario es un elemento que permite recoger datos introducidos por el usuario.

Los formularios se utilizan para conocer las opiniones, dudas, y otra serie de datos sobre los usuarios, para introducir pedidos a través de la red, tienen multitud de aplicaciones.



Un formulario está formado, entre otras cosas, por etiquetas, campos de texto, menús desplegables, y botones.

Es muy recomendable utilizar tablas para organizar los elementos de los formularios. Utilizando tablas se consigue una mejor distribución de los elementos del formulario, lo que facilita su comprensión y mejora su apariencia.

Los formularios se insertan a través de las etiquetas `<form>` y `</form>`. Entre dichas etiquetas habrá que insertar los diferentes objetos que formarán el formulario. la etiqueta `<form>` tiene los siguientes atributos:

- El atributo `action` indica una dirección de correo electrónico a la que mandar el formulario, o la dirección del programa que se encargará de procesar el contenido del formulario.
- El atributo `enctype` indica el modo en que será cifrada la información para su envío. Por defecto tiene el valor `application/x-www-form-urlencoded`.
- El atributo `method` indica el método mediante el que se transferirán las variables del formulario. Se recomienda utilizar el valor `post`, ya que se utiliza cuando el usuario manda datos que deben ser almacenados en una base de datos.

Por ejemplo, podríamos insertar un formulario escribiendo el siguiente código:

```
<form action="mailto:domingonozzi@gmail.com" method="post" enctype="text/plain" >  
...  
</form>
```

A continuación veamos los distintos elementos que se pueden incluir en un formulario.

### Elementos de entrada `<input>`

Para insertar un elemento de entrada es necesario incluir la etiqueta `<input>` entre las etiquetas `<form>` y `</form>` del formulario.

El atributo `name` indica el nombre que se desea dar al elemento de entrada, mediante el cual será evaluado, y el atributo `type` indica el tipo de elemento de entrada.

Vamos a ver los diferentes tipos de elementos de entrada, y el resto de atributos que pueden definirse para cada uno de ellos.

#### *Campo de texto:*

- Para insertar un campo de texto, el atributo `type` debe tener el valor `text`.
- El atributo `size` indica el número de caracteres que podrán ser visualizados en el campo de texto, determina el ancho de la caja.

- El atributo `maxlength` indica el número de caracteres que podrán ser insertados en el campo de texto.
- El atributo `value` indica el valor inicial del campo de texto.

Por ejemplo: `<input name="campo" type="text" value="Campo de texto" size="20" maxlength="15">`

#### *Campo de contraseña:*

- Para insertar un campo de contraseña, el atributo `type` debe tener el valor `password`.
- El resto de atributos son los mismos que para un campo de texto normal. La única diferencia es que todas las letras escritas en el campo de contraseña serán visualizadas como asteriscos.

Por ejemplo: `<input name="contra" type="password" value="contraseña" size="20" maxlength="15">`

#### *Casilla de verificación:*

Para insertar una casilla de verificación, el atributo `type` debe tener el valor `checkbox`.

- El atributo `value` indica el valor asociado a la casilla de verificación. Es necesario poner este atributo, aunque el usuario no pueda ver su valor. Es el valor a enviar.
- La aparición del atributo `checked` indica que la casilla aparecerá activada inicialmente. Este atributo no toma valores.

Por ejemplo: `<input name="casilla" type="checkbox" value="acepto" checked>`

#### *Botón de opción:*

Para insertar un botón de opción, el atributo `type` debe tener el valor `radio`.

- El atributo `value` indica el valor asociado al botón de opción. Es necesario poner este atributo, aunque el usuario no pueda ver su valor. Es el valor a enviar.
- La aparición del atributo `checked` indica que el botón aparecerá activado inicialmente. Este atributo no toma valores.

Los botones de opción se utilizan cuando se desea que una variable del formulario pueda tomar un solo valor de entre varios posibles. Para ello, se insertan varios botones de opción con el mismo nombre (que indica la variable) y con distintos valores. Sólomente uno de estos botones podrá estar activado, el que esté activado cuando se envía el formulario, su valor será el que tendrá la variable.

Por ejemplo:

```
<input name="estadocivil" type="radio" value="Casado">
```

```
<input name="estadocivil" type="radio" value="Soltero" checked>
```

Cuando se envíe el formulario, si el primer botón está activado la variable `prefiere` será igual a `Casado`, si es el segundo el activado, la variable `prefiere` valdrá `Soltero`. Observa que lo que ponemos como valor no aparece escrito en la página es un dato interno.

### *Campos de selección <select> ...*

Los campos de selección se utilizan para insertar menús y listas desplegables.

Para insertar uno de estos menús o listas es necesario insertar las etiquetas `<select>` y `</select>` en un formulario.

- El atributo `name` indica el nombre del menú o lista será el nombre de la variable que contendrá el valor seleccionado.
- El atributo `size` indica el número de elementos de la lista que pueden ser visualizados al mismo tiempo, determina el alto de la lista.
- La aparición del atributo `disabled` indica que la lista estará desactivada, por lo que el usuario no podrá seleccionar sus elementos. Este atributo tampoco toma valores.

Cada uno de los elementos de la lista ha de insertarse entre las etiquetas `<option>` y `</option>`.

El atributo `value` indica el valor a enviar si se selecciona el elemento. Si no se especifica este atributo, se enviará el texto de la opción, que se encuentra entre las etiquetas `<option>` y `</option>`.

La aparición del atributo `selected` indica que el elemento aparecerá seleccionado. Este atributo no toma valores.

Por ejemplo:

```
<select name="mascota">
  <option selected>--- Elige animal ---</option>
  <option>Perro</option>
  <option>Gato</option>
</select>
```

### *Áreas de texto <textarea>*

Las áreas de texto permiten a los usuarios insertar varias líneas de texto. Por ello, suelen utilizarse para que incluyan comentarios.

Para que el área de texto tenga algún valor inicial, habrá que insertarlo entre las etiquetas `<textarea>` y `</textarea>`.

- El atributo `name` indica el nombre del área de texto. Es necesario dar nombres diferentes a cada uno de los elementos de un formulario, para poder identificarlos a la hora de procesarlos.
- El atributo `rows` indica el número de líneas que podrán ser visualizadas en el área de texto por lo que determina al alto del área de texto.
- El atributo `cols` indica el número de caracteres por línea que podrán ser visualizados en el área de texto por lo que determina al ancho del área de texto.

Ejemplo: `<textarea name="ejemploarea" cols="30" rows="3">Escribe el texto que quieras</textarea>`

#### *Botón:*

Para insertar un botón, el atributo `type` debe tener el valor `submit`.

- Si el valor es `submit`, al pulsar sobre el botón se enviará el formulario.
- El atributo `value` indica el texto que mostrará el botón.

Por ejemplo: `<input name="boton" type="submit" value="Enviar">`

## Estándares WEB

HTML se puede codificar de muchas maneras diferentes. Y los navegadores pueden interpretar HTML de todas esas mismas maneras. Se podría decir que HTML tiene muchos dialectos.

Por parte del Consorcio del World Wide Web (W3C), fundado por Tim Berners-Lee, se han producido intentos para crear un estándar común de HTML.

Cuando se escribe código HTML conforme con los estándares del W3C, creamos sitios web que se pueden ver en todos los navegadores. Lo que hemos aprendido es una nueva versión más estricta y limpia de HTML llamada XHTML.

¿Cómo indico qué versión uso?

Es necesario indicar al navegador qué "dialecto" utiliza tu código HTML, en este caso XHTML. Para hacer esto, usa una DTD ("Document Type Declaration", declaración de tipo de documento) que debe aparecer siempre al inicio del documento:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
```

Aparte del DTD (la primera línea del ejemplo anterior), que indica al navegador que vas a usar código escrito en XHTML, también es necesario insertar información adicional en la etiqueta html por medio de los atributos xmlns y lang.

xmlns es la abreviatura de "XML-Name-Space" (espacio de nombre XML) y siempre debería tener el valor <http://www.w3.org/1999/xhtml>. Para más información sobre el tema, se puede leer sobre espacios de nombre en el sitio web del W3C.

Con el atributo lang se especifica en qué idioma se ha escrito el documento. En el ejemplo anterior el idioma usado es el inglés ("en").

Con la DTD, el navegador sabe con exactitud cómo interpretar y mostrar el código HTML. Así pues, usa el ejemplo anterior como plantilla para todos los documentos futuros que vayas a crear.

## Capas <div> ...

Las capas no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.

Podemos insertar una capa a través de las etiquetas <div> y </div>, que como ya vimos, sirven para agrupar bloques de texto.

En la hoja de estilos CSS, se crean las definiciones para una capa usando la almohadilla (#) seguido del nombre que le asignemos a la capa, y luego entre corchetes {} se escriben sus definiciones. En el documento HTML, se escribe <div id="nombre de la capa">.

A través de las propiedades left (izquierda) y top (superior) se establece la posición de la capa respecto a los márgenes izquierdo y superior de la página. Pueden tomar un número como valor, acompañado de px cuando haga referencia a píxeles, y acompañado de % cuando haga referencia a un porcentaje.

Para que la capa aparezca en la posición establecida, es necesario incluir también la propiedad position con el valor absolute. Si no se estableciera este valor, la capa se mostraría pegada al margen izquierdo, en la posición en la que hubiera sido insertada dentro del código.

A través de las propiedades width (anchura) y height (altura) se establece el tamaño de la capa. Pueden tomar un número como valor, acompañado de px cuando haga referencia a píxeles, y acompañado de % cuando haga referencia a un porcentaje.

A través de la propiedad z-index puede establecerse el índice de la capa dentro de la página. Una capa podrá ser solapada por aquellas capas cuyo índice sea mayor. Siempre es un valor numérico.

A través de la propiedad background-image se puede establecer una imagen de fondo para la capa. La ruta y el nombre de la imagen han de aparecer entre paréntesis, después de la palabra url.

A través de la propiedad background-color se puede establecer un color de fondo para la capa. Ha de ser un número hexadecimal.

Otra forma de insertar capas es utilizando las etiquetas <span> y </span>, en lugar de las etiquetas <div> y </div>.

La etiqueta <span> se utiliza del mismo modo que la etiqueta <div>, y no produce un salto de renglón en el caso de bloques de texto.

Otras propiedades para las capas:

A través de la propiedad `visibility` puede establecerse la visibilidad de la capa. Puede tomar los valores `inherit` (se muestra la capa mientras la capa a la que pertenece también se esté mostrando), `visible` (muestra la capa, aunque la capa a la que pertenece no se esté viendo) y `hidden` (la capa está oculta).

A través de la propiedad `overflow` puede establecerse si se mostrará o no el contenido de la capa cuando no pueda ser visualizado en su totalidad, por ser la capa demasiado pequeña. Puede tomar los valores `visible` (se muestra todo el contenido de la capa, aunque esto implique hacer que la capa sea más grande), `hidden` (no es posible visualizar el contenido de la capa que no quepa en ella), `scroll` (se muestra la barra de desplazamiento, aunque el contenido de la capa pueda ser visualizado totalmente) y `auto` (se muestra la barra de desplazamiento cuando sea necesario).

A través de la propiedad `clip` puede establecerse el área de la capa que podrá ser visualizado. Lo que hace es recortar la capa, haciendo que partes de ella no sean visibles. Ha de especificarse la distancia de los márgenes de la capa entre paréntesis, después de la palabra `url`.

- El primer valor ha de ser la distancia (se asume que está en píxeles) del margen superior, hasta la que no se visualizará el contenido de la capa.
- El segundo valor ha de ser la distancia del margen izquierdo, hasta la que no se visualizará el contenido de la capa.
- El tercer valor ha de ser la distancia del margen superior, hasta la que se visualizará el contenido de la capa.
- El cuarto valor ha de ser la distancia del margen izquierdo, hasta la que se visualizará el contenido de la capa.

También es posible incluir `auto` como valor de alguna de estas distancias, lo que indica que la distancia se corresponderá con los bordes de la capa.

## Hojas de estilo (CSS).

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado.

HTML se usa para estructurar el contenido; CSS se usa para formatear el contenido previamente estructurado.

Al estar la definición de los estilos en un archivo externo a las páginas y común a todas las páginas del sitio (es recomendable) el aspecto de nuestras páginas será más homogéneo y además podremos cambiar ese aspecto de manera segura e inmediata cambiando únicamente la hoja de estilos.

Una hoja de estilo puede ser creada con el bloc de notas, y se guarda con extensión .css

## El atributo STYLE

Para incluir código CSS dentro de una página Web, se puede utilizar el atributo STYLE. Este atributo se utiliza directamente acompañando a un elemento dentro de la etiqueta de inicio. Luego de escribir STYLE se coloca el signo igual, y entre comillas se escribe la regla o las reglas CSS.

Ejemplo: `<p style="color:red; text-align:center; ">Texto en color rojo y centrado</p>`

Una regla CSS se escribe de la siguiente manera dentro del atributo STYLE:

nombre\_de\_la\_propiedad, luego dos puntos (:) para asignar un valor, y luego el valor que se desea asignar terminando con un punto y coma (;). Se pueden asignar varias propiedades CSS a un elemento.



## Algunas Propiedades CSS

### Fuentes y textos

La propiedad *font-family* se usa para determinar la fuente con la que se quiere representar el texto.

Ejemplo: `font-family: Arial, Verdana, Helvetica, sans-serif ;`

La propiedad *font-style*, cuyo valor es *italic* para cursivas y *normal* (por defecto).

Ejemplo: `font-style: italic ;`

La propiedad *font-weight* definimos el grosor de la letra y puede tomar los valores *bold*, *bolder*, *900*, *800*, ..., *300*, *200*, *100* y *lighter*.

Ejemplo: `font-weight: bold ;`

La propiedad *font-size* se usa para definir el tamaño del texto, recordar incluir *px* (para píxeles).

El atributo *text-decoration*, puede tomar los valores *none*, *overline*, *underline*, *through* y *blink*.

- ❖ *none* es el valor por defecto e indica que no se representará el texto con ningún tipo de decoración.
- ❖ *overline* mostrará una línea por encima del texto escogido.
- ❖ *underline* representa el texto subrayado.
- ❖ *through* tacha el texto y
- ❖ *blink* lo muestra parpadeante.

La propiedad *text-align* que puede tomar los valores *left*, *right*, *justify* o *center* que lo alinearán respecto al lugar donde se encuentre dicho texto.

La propiedad *text-indent* permite establecer un valor para la sangría de primera línea.

Ejemplo: `text-indent: 50px;`

### Propiedad color

Esta propiedad define el color del texto de un elemento.

Ejemplo: `color: #FF0000 ;`

Color de fondo: puede ser aplicado al cuerpo, a una celda de una tabla, a un párrafo.

Ejemplo: `background-color: red;`

### Imagen de fondo

Al igual que el anterior, en vez de colocar un color de fondo coloca una imagen de fondo.

Ejemplo: `background-image: url("imagenes/fondo1.jpg");`

### Ancho.

Podemos también definir la anchura de las imágenes, de las celdas, de las tablas y otros elementos, mediante el atributo width, se suelen preferir los tantos por ciento y los pixeles, px.

Ejemplos: width:350px / width:80%

De la misma manera el alto se puede establecer con la propiedad height.

### Márgenes.

Se pueden definir cuatro atributos para esa propiedad:

- ✓ margin-top: define el margen superior del elemento al que se aplique.
- ✓ margin-bottom: define el margen inferior.
- ✓ margin-left: margen izquierdo y...
- ✓ margin-right: que define el margen derecho.

Ejemplo: Si queremos que todos los párrafos tengan unos márgenes de 10 pixeles y un color de texto azul, tendremos que definir la siguiente Regla CSS en la Hoja de Estilo:

```
p { margin-top:10px ; margin-bottom:10px; margin-left:10px; margin-right:10px; color:blue ;}
```

Si definimos los cuatro tipos (atributos) de márgenes comentados, podemos resumirlo de la siguiente manera: margin: valorsuperior valorderecho valorinferior valorizquierdo; eso sí, respetando ese orden y SIN separar esos valores con comas.

Ejemplo: p { margin: 10px 10px 10px 10px; color:blue ; }

### Padding.

CSS define cuatro propiedades para controlar cada uno de los espacios de rellenos horizontales y verticales de un elemento.

Las propiedades son: padding-top, padding-right, padding-bottom, padding-left.

Los valores pueden ser unidades de medida o porcentajes.

Ejemplo:

```
td { padding-top:10px; padding-right:15px; padding-bottom:20px; padding-left:25px; }
```

También se podría escribir, como en el caso de los márgenes:

```
td {padding: 10px 15px 20px 25px}
```

Si se utiliza el atributo padding con un solo valor permite separar el contenido de la caja usando los mismos valores para cada lado.

## Bordes.

Se trata de definir el grosor que queremos que tenga el borde de una imagen o de una celda, o incluso de un párrafo, en cuyo caso saldrá recuadrado.

El atributo para definir el grosor de borde de un elemento es `border-width` y se utiliza la unidad pixeles (px). Este atributo define el ancho de los cuatro lados que rodean el elemento en cuestión, pero si queremos definir distintos bordes para cada uno de los lados podemos usar los atributos:

- `border-top-width`: que define el grosor del borde de arriba.
- `border-bottom-width`: para el grosor del borde de abajo
- `border-left-width`: para el izquierdo y...
- `border-right-width`: para el derecho.

Es posible definir el color de ese borde mediante el atributo `border-color`.

En cuanto al estilo del borde, se define mediante el atributo `border-style` y puede tomar los siguientes valores: `solid`, `inset`, `groove`, `outset` y `double`. Probar para ver efectos.

Un ejemplo que agrupa algunas propiedades vistas hasta ahora podría ser el de un párrafo de texto rojo, con un recuadro azul rodeándolo de tamaño 10px y con apariencia de doble línea. La Regla CSS correspondiente sería la siguiente:

```
p { color:red ; border-color:blue ; border-width:10px ; border-style:double ; }
```

Otra sintaxis del ejemplo anterior: `p {color:red; border: blue 10px double; }`

## Comentarios

CSS permite incluir comentarios entre sus reglas y estilos. Los comentarios son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad. Los navegadores ignoran por completo cualquier comentario de los archivos CSS, por lo que es común utilizarlos para estructurar de forma clara los archivos CSS complejos.

El comienzo de un comentario se indica mediante los caracteres `/*` y el final del comentario se indica mediante `*/`, tal y como se muestra en el siguiente ejemplo:

```
/* Este es un comentario en CSS */
```

La sintaxis de los comentarios CSS es muy diferente a la de los comentarios HTML, por lo que no deben confundirse:

```
<!-- Este es un comentario en HTML -->
```

## Usando el elemento <STYLE> en la cabecera

Se pueden definir estilos personalizados para una página escribiéndolos dentro las etiquetas <STYLE> y </STYLE>. Esto se debe escribir dentro de la cabecera, entre las etiquetas <HEAD> y </HEAD>.

## Sintaxis de las hojas de estilo

En primer lugar se pone el nombre del estilo, y entre llaves se especifica la lista de propiedades (en minúsculas) que se corresponden con dicho estilo. Cada una de estas propiedades tiene que tener un punto y coma detrás, y los valores se asignan con el símbolo : (dos puntos).

El nombre del estilo puede ser un nombre inventado por nosotros, o el nombre de una etiqueta HTML. Para poder utilizar un nombre inventado, tiene que estar precedido por un punto, o por el nombre de una etiqueta seguida de un punto.

Por ejemplo, en una hoja de estilo podríamos tener lo siguiente:

```
body {background-color: #006699; font-family: Arial,Helvetica;}
```

```
.mitexto {font-family: Arial,Helvetica; font-size:18px;}
```

En la hoja de Estilos hay que indicar el selector y su declaración (nombre y valor), que juntos componen la Regla CSS.

## Aplicando estilos.

Si vinculáramos esta hoja de estilo a un documento, se aplicarían directamente las propiedades especificadas para la etiqueta <body>.

En cambio, no existe ninguna etiqueta <mitexto>, por lo que para aplicar este estilo a algún elemento de la página habría que indicarlo de algún modo.

Para aplicar este estilo a un elemento, habría que insertar el atributo class en su etiqueta.

Por ejemplo, para aplicar ese estilo al siguiente párrafo del documento:

```
<p>texto con estilo</p>
```

```
Habría que escribir: <p class="mitexto">texto con estilo</p>
```

Pero para aplicar el estilo únicamente a una parte del contenido de la etiqueta, como podría ser en este caso una palabra del párrafo, sería necesario incluir la etiqueta <span> (que agrupa bloques, sin producir un cambio de línea). Por ejemplo, para aplicar el estilo únicamente a la palabra estilo, habría que escribir:

```
<p>texto con <span class="mitexto">estilo</span></p>
```

### Vincular una hoja de estilo.

Para poder incluir las propiedades de una hoja de estilo en un documento, hay que vincularla a él.

Para vincular una hoja de estilo a un documento es necesario insertar la etiqueta <link> en el documento, entre las etiquetas <head> y </head>. Esta etiqueta no necesita etiqueta de cierre.

A través del atributo href se especifica la hoja de estilo que se va a vincular al documento.

A través del atributo rel se tiene que especificar que se está vinculando una hoja de estilo, por lo que su valor ha de ser stylesheet.

A través del atributo type se tiene que especificar que el archivo es de texto, con sintaxis CSS, por lo que su valor ha de ser text/css.

Por ejemplo, podríamos vincular una hoja de estilo escribiendo el siguiente código.

```
<link href="misestilos.txt" rel="stylesheet" type="text/css">
```

## Modificar enlaces usando estilos CSS.

Para controlar estos efectos se usan lo que se ha denominado como pseudo-clases. Una pseudo-clase te permite tener en cuenta diferentes condiciones o eventos al definir una propiedad para una etiqueta HTML.

Todo enlace tiene diferentes estados. Por ejemplo, visitado o no visitado. Puedes usar una pseudo-clase para asignar diferentes estilos a los enlaces visitados y no visitados.

Ejemplos:

<pre>a:link {     color: blue; }</pre>		<pre>a:visited {     color: red; }</pre>
--	--	--

La pseudo clase :link se usa para enlaces que dirigen a páginas que el usuario no ha visitado.

La pseudo clase :visited se usa para enlaces que dirigen a páginas que el usuario ya ha visitado.

La pseudo clase :active se usa para enlaces que están activos.

El código de este ejemplo hace que el color de fondo para los enlaces activos sea amarillo:

```
a:active {  
    background-color: #FFFF00;  
}
```

Pseudo-clase a:hover

La pseudo clase a:hover se usa cuando el puntero del ratón pasa por encima de un enlace.

Esta pseudo-clase se puede usar para crear efectos interesantes. Por ejemplo, si queremos que nuestros enlaces sean de color naranja y estén en cursiva cuando el cursor pase sobre ellos, el código CSS que utilizaremos será el siguiente:

```
a:hover {  
    color: orange;  
    font-style: italic;  
}
```

Ejemplo 1: Espaciado entre letras

```
a:hover {  
    letter-spacing: 10px;  
    font-weight:bold;  
    color:red;  
}
```

Ejemplo 2: MAYÚSCULAS y minúsculas

```
a:hover {  
    text-transform: uppercase;  
    font-weight:bold;  
    color:blue;  
    background-color:yellow;  
}
```

Ejemplo 3: Quitar el subrayado de los enlaces

```
a {  
    text-decoration:none;  
}
```

Ejercitación - Ejemplo 4:

En la hoja de estilos escribir:

```
.alternativo {color: #CC0000;}
```

```
.simple {text-decoration: none;}
```

```
.importante {font-weight: bold; font-size: 20px;}
```

```
.raro {text-decoration:overline;}
```

Guardar con el nombre practicaestilos.css en la subcarpeta ESTILOS CSS.



Abrir el documento básico de HTML y asignar clases:

```
<a href="#">Enlace con el estilo por defecto</a>
```

```
<a class="alternativo" href="#">Enlace de color rojo</a>
```

```
<a class="simple" href="#">Enlace sin subrayado</a>
```

```
<a class="importante" href="#">Enlace muy importante</a>
```

```
<a class="raro" href="#">Enlace con un estilo raro</a>
```

Guardar con el nombre practicaenlaces.html

Vincular el documento HTML con la Hoja de estilos y visualizar en el navegador.

### Aplicando estilos a tablas.

```
table {width:50% ; }
```

aunque también podemos definir la clase .lamitad como:

```
.lamitad { width: 50%; }
```

y aplicárselo solo a las tablas que queramos, indicando ya en la página (dentro del body) lo siguiente: <table class="lamitad">...

Nota: del mismo modo que el atributo anterior, Height se utiliza para el alto.

- *Para centrar una tabla:*

Dentro de sus propiedades en la hoja de estilos CSS, se puede utilizar la propiedad margin con el valor auto. Ejemplo: table {width:80%; margin: 0px auto; }

El valor 0 px indica 0 pixeles a lo alto, y auto indica la misma distancia (o sea centrado) a la izquierda y la derecha.

- *Colocando bordes internos y modificando sus propiedades.*

Se utilizan las propiedades de los bordes sobre las etiquetas td o th.

Hay dos modelos diferentes para el tratamiento de los bordes de las celdas. La propiedad que permite seleccionar el modelo de bordes es border-collapse y los valores posibles son collapse y separate. El modelo collapse fusiona de forma automática los bordes de las celdas adyacentes, mientras que el modelo separate fuerza a que cada celda muestre sus cuatro bordes.

Si se opta por el modelo separate (que es el que se aplica si no se indica lo contrario) se puede utilizar la propiedad border-spacing para controlar la separación entre los bordes de cada celda.

Si solamente se indica como valor una medida, se asigna ese valor como separación horizontal y vertical. Si se indican dos medidas, la primera es la separación horizontal y la segunda es la separación vertical entre celdas.

Ejemplo: table {border-collapse:separate; border-spacing:5px 25px; }

CSS define otras propiedades específicas para el control del aspecto de las tablas. Una de ellas es el tratamiento que reciben las celdas vacías de una tabla, que se controla mediante la propiedad empty-cells. Esta propiedad sólo se aplica cuando el modelo de bordes de la tabla es de tipo separate. Los valores que puede tomar esta propiedad son show (mostrar) y hide (ocultar). El valor hide indica que las celdas vacías no se deben mostrar.

Para ajustar la alineación vertical de las celdas se usa la propiedad vertical-align, sus valores pueden ser top (arriba), middle (medio) y bottom (abajo).

Los comandos padding y margin también son utilizados para tablas.

### Posicionamiento de elementos

Con posicionamiento CSS, se puede colocar un elemento en el lugar exacto que se quiera de la página.

### Posicionamiento absoluto

El elemento que se posiciona de forma absoluta no ocupa espacio alguno en el documento. Para posicionar un elemento de forma absoluta, la propiedad **position** se establece como absolute. Posteriormente puedes usar las propiedades left, right, top, y bottom para colocar la caja.

Ejemplo para posicionar elementos en las 4 esquinas de la página, en el documento CSS:

```
#box1 {
    position:absolute;
    top: 50px;
    left: 50px;
}

#box2 {
    position:absolute;
    top: 50px;
    right: 50px;
}

#box3 {
    position:absolute;
    bottom: 50px;
    right: 50px;
}

#box4 {
    position:absolute;
    bottom: 50px;
    left: 50px;
}
```

Para colocar imágenes, se puede escribir en el cuerpo del documento HTML:

```
<div id="box1"></div>
<div id="box2"></div>
<div id="box3"></div>
<div id="box4"></div>
```