

UF1303: Elaboración de hojas de estilo

Presentación

Identificación de la Unidad Formativa

Bienvenido a la Unidad Formativa **UF1303: Elaboración de hojas de estilo**. Esta Unidad Formativa pertenece al Módulo Formativo **MF0950_2: Construcción de páginas web** que forma parte del Certificado de Profesionalidad **IFCD0110: Confección y publicación de páginas web**, de la familia de Informática y comunicaciones.

Presentación de los contenidos

La finalidad de esta Unidad Formativa es enseñar al alumno a crear hojas de estilo para homogeneizar el aspecto de las páginas y hacerlas más atractivas, según especificaciones de diseño y desarrollo recibidas. Para ello, se estudiará en primer lugar la función de las hojas de estilo en la construcción de páginas web y posteriormente se analizará el diseño, ubicación y optimización de los contenidos de una página web.

Objetivos de la Unidad Formativa

Al finalizar esta Unidad Formativa aprenderás a:

- Describir las características de las hojas de estilo para dar formato a las páginas web y crear ficheros de estilo de acuerdo a un diseño especificado.
- Diseñar, ubicar y optimizar los contenidos de una página web, para adecuarla al formato de la misma, facilitar su manejo a los usuarios y optimizarla de acuerdo a un diseño especificado.

Índice

UD1. Hojas de estilo en la construcción de páginas web	9
1.1. Funciones y características	11
1.1.1. Descripción de estilos	12
1.1.2. Utilización de estilos	13
1.1.3. Los estilos en el lenguaje de marcas.....	16
1.1.4. Los estilos con herramientas de edición web	19
1.2. Hojas de estilo y accesibilidad	23
1.2.1. Adecuación de las hojas de estilos	26
1.3. Tipos de estilo: incrustados, enlazados, importados, en línea	31
1.3.1. Descripción de los tipos de estilo	33
1.3.2. Enlazar una hoja de estilo externa a un documento HTML.....	34
1.3.3. Incrustar un estilo dentro de un documento HTML.....	39
1.3.4. Importar una hoja de estilo desde un documento HTML	42
1.3.5. Importar una hoja de estilo a través de un archivo con estilos	43
1.3.6. Utilización y optimización de los tipos de estilos.....	44

1.4.	Selectores y reglas de estilo	45
1.4.1.	Estructura de los estilos.....	46
1.4.2.	Sintaxis básica de estilos.....	48
1.4.3.	Utilización de elementos y seudoelementos	54
1.4.4.	Utilización de clases y seudoclases	57
1.5.	Atributos de estilo para fuentes, color y fondo, texto y bloques (párrafos)	60
1.5.1.	Descripción de los atributos de estilo.....	62
1.5.2.	Utilización de los atributos de estilo	70
1.6.	Creación de ficheros de estilo	247
1.6.1.	Definición de los ficheros de estilo	248
1.6.2.	Creación de ficheros de estilo genéricos	251
1.6.3.	Adaptación de los ficheros de estilo para distintas páginas web	258
UD2.	Diseño, ubicación y optimización de los contenidos de una página web	273
2.1.	Creación de un documento funcional	275
2.1.1.	Descripción de los elementos funcionales de la página web.....	279
2.1.2.	Definición de los elementos funcionales de la página web.....	284
2.1.3.	Descripción de cada elemento	291
2.2.	Diseño de los contenidos.....	294
2.3.	Identificación de la información a ubicar en la página web....	302
2.4.	Selección de contenidos para cada elemento de la página.....	318
2.5.	Utilización del documento funcional para las especificacio- nes del diseño	324
2.6.	Tipos de página para la ubicación de contenidos	347
2.7.	Definición de los tipos de página en base a los contenidos y funcionalidades	352

Índice

2.8.	Selección de los tipos de página para la página web.....	356
2.9.	Utilización del documento funcional para las..... especificaciones del tipo de página	357
2.10.	Especificaciones de navegación.....	359
2.11.	Creación de un mapa de navegación de páginas.....	363
2.12.	Utilización del documento funcional para integrar el mapa de navegación	366
2.13.	Elementos utilizados para la navegación.....	373
2.13.1.	Definición de los elementos utilizados para navegar.....	374
2.13.2.	Utilización del documento funcional para especificar los elementos de navegación	376
2.14.	Elaboración de una guía de usuario	377
	Glosario	393
	Soluciones.....	397

Área: informática y comunicaciones

UD1

Hojas de estilo en
la construcción de
páginas web

- 1.1. Funciones y características
 - 1.1.1. Descripción de estilos
 - 1.1.2. Utilización de estilos
 - 1.1.3. Los estilos en el lenguaje de marcas
 - 1.1.4. Los estilos con herramientas de edición web
- 1.2. Hojas de estilo y accesibilidad
 - 1.2.1. Adecuación de las hojas de estilos
- 1.3. Tipos de estilo: incrustados, enlazados, importados, en línea
 - 1.3.1. Descripción de los tipos de estilo
 - 1.3.2. Enlazar una hoja de estilo externa a un documento HTML
 - 1.3.3. Incluir un estilo dentro de un documento HTML.
 - 1.3.4. Importar una hoja de estilo desde un documento HTML
 - 1.3.5. Importar una hoja de estilo a través de un archivo con estilos
 - 1.3.6. Utilización y optimización de los tipos de estilos
- 1.4. Selectores y reglas de estilo
 - 1.4.1. Estructura de los estilos
 - 1.4.2. Sintaxis básica de estilos
 - 1.4.3. Utilización de elementos y seudoelementos
 - 1.4.4. Utilización de clases y seudoclases
- 1.5. Atributos de estilo para fuentes, color y fondo, texto y bloques (párrafos)
 - 1.5.1. Descripción de los atributos de estilo
 - 1.5.2. Utilización de los atributos de estilo
- 1.6. Creación de ficheros de estilo
 - 1.6.1. Definición de los ficheros de estilo
 - 1.6.2. Creación de ficheros de estilo genéricos
 - 1.6.3. Adaptación de los ficheros de estilo para distintas páginas web

1.1. Funciones y características

Las hojas de estilo o CSS (Cascading Style Sheets) es la tecnología que nos permite controlar las propiedades visuales de los elementos que componen una página Web. Mientras HTML se preocupa por la estructura de los contenidos, CSS se hace cargo de la apariencia y el estilo.

CSS trabaja en un documento HTML modificando sus propiedades, tales como el color, el borde, el posicionamiento y el tamaño, de una forma muy precisa y exacta, los que da al desarrollador Web un control máximo sobre dónde y cómo se visualizará esa página.

A este gran control sobre las propiedades visuales de la página, debemos sumar otra gran contribución de CSS al desarrollo web: este nos permite separar el estilo del contenido. La forma del elemento. Es decir, que cuando realicemos un cambio en el estilo de nuestra web éste se aplicará en todos los elementos asignados a dicho estilo, lo que hará que tengamos mucho más ahorro de tiempo y consigamos mayor facilidad a la hora de añadir algún cambio. También nos permite crear colecciones de estilos (hojas de estilos) que podremos aplicar después a un documento HTML, permitiendo usar los mismos estilos para muchas páginas a la vez.

Cuando no existía el CSS, teníamos que repetir constantemente cierta información como el tamaño y el color de los textos. Además, HTML por si solo nos ofrecía muy pocas posibilidades de cambiar la apariencia de una página más allá de un cambio de color o tipografía de una fuente.

Es por esto que el uso de hojas de estilos nos permite ir donde HTML no puede llegar. Al usar CSS lo que hacemos es definir un mayor número de características para un elemento y dichas propiedades quedan ya establecidas sin necesidad de repetirlas.

Veamos un ejemplo:

```
DIV { font-family: Verdana;  
      font-size: 12px;  
      color: #000000  
}
```

Con CSS no es necesario repetir este código por cada marca <DIV>. Al definirlo como una regla CSS, el navegador lo interpretará cambiando las características del texto de los elementos <DIV>, con el consiguiente ahorro de trabajo y tiempo. No hay que olvidar que la mayoría de propiedades que nos ofrece CSS no sería posible definirlas con HTML.



1.1.1. Descripción de estilos

En resumen, CSS nació como un estándar para complementar HTML, lo que permitió un control mucho mayor sobre la visualización de los elementos. La misión de CSS en principio era sencilla: dar estilo y mejorar la apariencia de HTML, ya que éste dio mucha importancia a la estructura de la página web, pero dejó de lado un aspecto muy importante para los diseñadores como es la presentación.

Con CSS podremos hacer más atractiva la visualización de un documento HTML sin modificar el contenido que se utiliza.

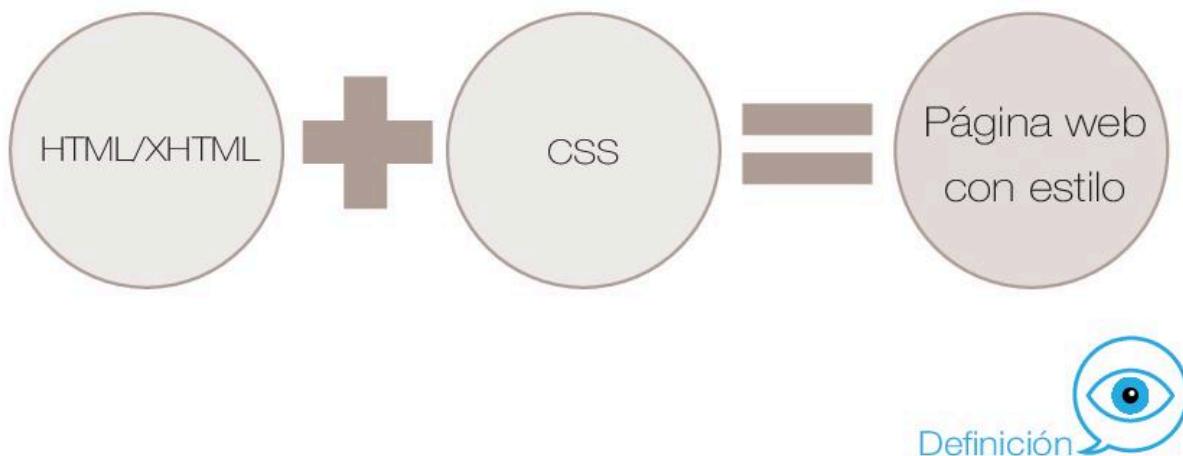
Las ventajas que existen al realizar una separación entre los contenidos y la definición de su aspecto son numerosas, ya que esto hace que se creen documentos HTML/XHTML muy bien definidos y con un significado mucho

más completo. A este tipo de documentos se les conoce como "documentos semánticos".

Por otro lado, la accesibilidad del documento es mucho mejor, el mantenimiento de la página es mucho más fácil de llevar a cabo y podremos visualizar el documento web en numerosos dispositivos diferentes.

Cuando creamos una página web, lo que hacemos primero es utilizar el lenguaje HTML/XHTML para marcar los contenidos, es decir, a cada elemento se le designa una función dentro de la página, como por ejemplo un párrafo, un titular, una tabla, una lista de otros elementos, un texto destacado, etc.

Cuando el desarrollador web ya ha creado los contenidos en HTML/XHTML, utilizará el lenguaje CSS para poder dar definición a cada uno de los elementos creados, tales como color, tamaño y tipo de letra del texto, la separación tanto horizontal como vertical que tendrán los diferentes elementos, la posición de cada uno de ellos dentro de la página, etc.



Las hojas de estilo (Cascading Style Sheets) es el lenguaje de hojas de estilo que se usa para conformar el aspecto y el formato de un documento web escrito en un lenguaje de marcas, lo que incluye varios lenguajes basados en XML como son XHTML, HTML o SVG.

1.1.2. Utilización de estilos

Antes de que entre los desarrolladores web se normalizara el uso de CSS, se utilizaban etiquetas especiales de HTML para poder cambiar el aspecto de los elementos que constitúan una página. En el siguiente ejemplo podemos

ver una página HTML con algunos estilos definidos, donde no se ha utilizado CSS:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1" />

    <title>Ejemplo de estilos sin CSS</title>

</head>

<body>

    <h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>

    <p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>

</body>

</html>
```

Como puedes observar en el ejemplo, se ha utilizado la etiqueta `` con sus atributos `color`, `face` y `size`, lo que definirá el color, tipo y tamaño de la letra de cada elemento de la página.

La problemática a la que nos enfrentamos al hacer uso de este método para definir el aspecto de los distintos elementos de nuestra página es la economía de los elementos. Imaginemos que tuviéramos 50 elementos diferentes, donde tuviéramos que insertar 50 etiquetas ``. Si la página se compusiera de 10.000 páginas diferentes, tendríamos que definir 500.000 etiquetas ``, y ésta, al tener tres atributos diferentes, tendríamos que definirla con 1.5 millones de atributos.

Como el diseño de los sitios web está en constante evolución, es habitual modificar cada cierto tiempo el aspecto de las páginas del sitio. Siguiendo con el ejemplo anterior, cambiar el aspecto del sitio requeriría modificar 500.000 etiquetas y 1.5 millones de atributos.

También tendríamos que tener en cuenta que el diseño de una página web siempre está en constante cambio, por lo que es algo habitual modificar el aspecto de los documentos web del sitio de vez en cuando. Es por esto que usaremos CSS, ya que si tuviéramos que cambiar miles de etiquetas y atributos, nos sería un trabajo demasiado largo en el tiempo. Veamos en el siguiente ejemplo cómo hemos hecho uso del CSS para mejorar este aspecto en el documento web:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1" />
    <title>Ejemplo de estilos con CSS</title>
    <style type="text/css">
      h1 { color: red; font-family: Arial; font-size: large; }
      p { color: gray; font-family: Verdana; font-size: medium; }
    </style>
  </head>
  <body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto no muy largo.</p>
  </body>
</html>
```

Como podemos ver en el ejemplo, CSS lo que hace es separar los contenidos de la página por un lado, y la información sobre el aspecto que esta tendrá por otro. Dentro del documento HTML, se ha creado una zona en concreto

donde se incluye toda la información relacionada en concreto con los estilos que tendrá la página.

Concretamente, podemos observar en el ejemplo que, dentro de la zona delimitada para el CSS se han indicado que todas las etiquetas `<h1>` de la página se deben ver de color rojo, en letra Arial y en un tamaño grande. Por otro lado, se indica que las etiquetas `<p>` del documento irán en color gris, en letra Verdana y con un tamaño medio.

Utilizando CSS, se pueden establecer los mismos estilos con menos esfuerzo y sin ensuciar el código HTML de los contenidos con etiquetas ``. Como se verá más adelante, la etiqueta `<style>` crea una zona especial donde se incluyen todas las reglas CSS que se aplican en la página.

Las ventajas que obtendremos utilizando CSS en un documento web son múltiples:

Podremos establecer los mismos estilos con mucho menos esfuerzo.

- El código HTML se ensuciará menos con etiquetas ``.
- En el código se visualizará claramente donde se encuentra la zona en la que se incluyen las reglas CSS que se aplicarán posteriormente en la página.

Si comparamos los dos ejemplos, nos damos cuenta que haciendo uso de las hojas de estilo, podremos ahorrar miles de etiquetas, millones de atributos, pero aún así sigue siendo una solución no del todo cómoda.

Más adelante sabremos que los estilos CSS sólo se aplican en la página que los incluye, por lo que veremos algunas soluciones que nos dan las hojas de estilo para no tener que copiar las reglas CSS en cada documento web.

1.1.3. Los estilos en el lenguaje de marcas

Los estilos sólo hacen una cosa: dar formato al contenido de una página web. Son archivos de texto, como los archivos HTML, pero cuya extensión es ".css". Por lo tanto, podemos redactarlos en el mismo editor de texto que usaremos para una página web.

CSS es un lenguaje que permite aplicar definiciones de estilos a los objetos (o bloques) de los que se compone la página web: el texto, una imagen, un párrafo, un título, una tabla, etc. Todo esto se puede modificar gracias a CSS.

Para poner una analogía, el HTML sería el esqueleto de la página y las hojas CSS su silueta, que puede ser más o menos seductora, más o menos sofisticada. En función de nuestros gustos y necesidades.

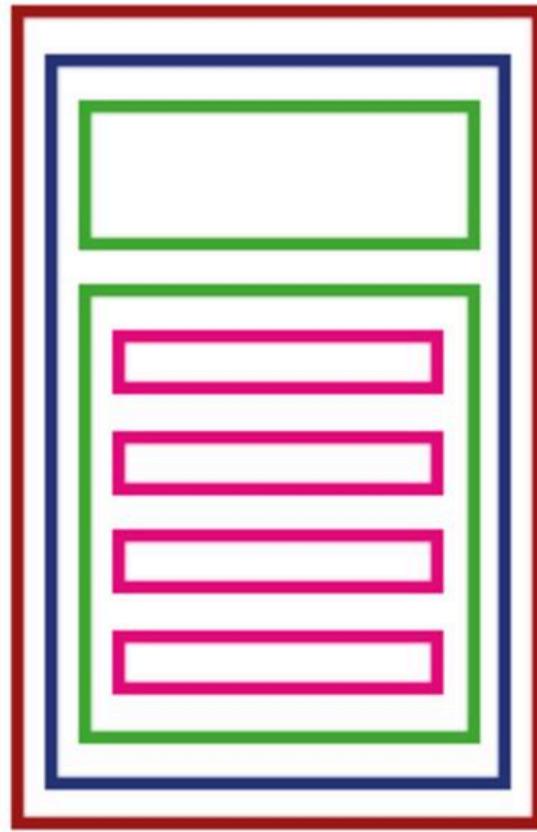
CSS son las siglas de Cascading Style Sheets en inglés, u hojas de estilo en cascada. "Cascada" se refiere al hecho de que los estilos se aplican en cascada, unos detrás de otros.

Por otro lado, el contenido de la página está organizado en bloques. La página en sí misma está contenida en un solo bloque, el body, que a su vez contiene los demás bloques: los párrafos, los títulos, las imágenes dentro de los párrafos, las tablas, etc.

Imaginemos que un bloque es como una caja rectangular que actúa como contenedor. Las hojas de estilos CSS permiten justamente aplicar efectos al contenido de las distintas cajas.

Por tanto, cada caja puede contener otras cajas, como si encajasen una dentro de otra. Para caracterizar esta relación entre los bloques se suele hablar de padres e hijos. Un parent o parente es un bloque que contiene otros bloques, y un hijo es el que está contenido dentro de otro. Todos los bloques, o casi todos, pueden ser padres de otros bloques y al mismo tiempo hijos de otros bloques.

Veamos a continuación una imagen donde podemos observar gráficamente las diferentes partes de una página web dividida en bloques y la relación entre éstos como padres e hijos:



El bloque azul es hijo del bloque rojo, que a su vez es padre de los dos bloques verdes e, indirectamente, también de los bloques rosas contenidos dentro del segundo bloque verde, etc.

El bloque rojo es el body. Él no es hijos de nadie, pero es padre de todos los demás. Si le aplicamos un estilo, dicho estilo se aplicará en cascada de un bloque a otro más pequeño. Esto se conoce como la herencia: un estilo siempre se aplicará de padres a hijos, pero no al revés.

Sin embargo, si el estilo se aplica automáticamente a los hijos, ¿cómo podemos aplicarles un estilo distinto al de los padres? Lo veremos con calma más adelante. Sólo decir que la cascada de los estilos también respeta una regla de precedencia: una definición heredada de un parent siempre será ignorada por el navegador si ya existe directamente a nivel del hijo. Por tanto, es posible definir un tipo de fuente para toda la página, luego definir otra específica para los títulos, o incluso una para un solo título, por ejemplo.

1.1.4. Los estilos con herramientas de edición web

Existen numerosas formas de diseñar el estilo de una página web. Ya sabemos que podemos realizarlo directamente desde un archivo de texto, sin embargo, también podremos aplicar estilos a través de otras herramientas de edición web. Estas herramientas facilitan bastante el trabajo, en el caso concreto de CSS, algunas nos ayudan a escribir el código y otras a utilizarlo.

A continuación, nombraremos algunas de estas herramientas y cómo comenzar a utilizarlas.

Hoy en día la depuración del código tanto en css como en javascript esta incorporado en los navegadores de ultima generación. Firefox DDeveloper, Chrome developer

Web Developer Toolbar

El WDT o Web Developer Toolbar (Barra de herramientas para desarrollo Web) es considerado otro elemento esencial entre los desarrolladores Web. Se trata de otra extensión gratuita de Firefox. Podremos conseguirla visitando el enlace siguiente: <http://www.addons.mozilla.org> y buscar Web Developer e instalarla desde la página de WDT, tal y como puedes ver en la siguiente imagen:



No podemos hablar de la multitud de opciones que nos ofrece WDT, sin embargo, hablaremos sobre algunos menús importantes. Es recomendable dedicarle tiempo para explorar todos los menús que nos ofrece, para conocer mejor todas las opciones de las que dispondremos para así aprovechar todas sus posibilidades.

Se recomienda deshabilitar el almacenamiento temporal de la página, especialmente si vamos a hacer actualizaciones menores y la caché del navegador soporta muchos archivos. Podemos deshabilitar JavaScript para averiguar

qué le sucede a la página cuando falle el script, o no se abra el entorno de JavaScript, como podemos ver en la siguiente imagen:



Desde el menú de CSS de la WDT podemos desactivar sólo los estilos incrustados, las hojas de estilo enlazadas, o los estilos en línea. También podremos desactivar los estilos por defecto del navegador, aunque esto no se recomienda.

Desde el menú Información, tendremos la posibilidad de mostrar la información de clase e ID del documento, el orden de los div de la página o un resumen de los colores que se han utilizado. Esta herramienta también es muy útil cuando queramos conocer información sobre los atributos y valores de un elemento, su posición en la página, información de fuente, elemento anterior y posterior, etc., simplemente haciendo clic sobre él.

Desde el menú Resaltar podremos resaltar clases de elementos, como por ejemplo elementos de bloque, desaprobados, posicionados, del nivel actual, tablas, además de poder establecer los elementos y colores propios. En definitiva, se trata de una herramienta de diagnóstico del diseño, ya que podemos resaltar conjuntos de elementos para ver con rapidez cómo están dispuestos en relación unos con otros, y encontrar posibles problemas.

A través del menú Herramientas conseguiremos un acceso rápido a un número de validadores, comprobadores de errores y consolas de depuración.

Dragonfly

En el caso de que nuestro navegador principal sea Opera, podremos utilizar Dragonfly. Se trata de un entorno de desarrollo integrado en Opera 58 (based on Chromium 71) y versiones posteriores. Podremos obtener más información en el siguiente enlace: <http://www.opera.com/dragonfly>.

Podremos instalar un menú de depuración a través del enlace siguiente: <http://www.opera.com/dragonfly> y pulsando Descargue Opera. Una vez que lo hayamos instalado, obtendremos un fácil acceso a Dragonfly y a sus variadas opciones a través del menú Página>Herramientas de desarrollo>Opera Dragonfly, o con el atajo del teclado Control+Mayús.+I.

El atajo del teclado sólo funcionará para su activación. En el caso de que Dragonfly ya estuviese abierto no lo cerrará. Para cerrarlo, podremos hacerlo con el ratón o a través del comando Control+W. éste comando funciona bien si Dragonfly está abierto en una ventana aparte, pero si estuviese acoplado en la ventana del navegador, éste sólo lo cierra si hemos hecho clic en algún elemento incluido en él. De otra forma, cerrará la pantalla completa.

Para activar diferentes modos de diseño de Opera, podemos hacerlo desde Ver>Barras de herramientas>Personalizar>Decorado.

Es verdad que Dragonfly se parece a Firefox developer, sin embargo, existen algunas diferencias notables, como la ficha Estilos, que puede mostrar los estilos calculados junto con los declarados y los grupos se pueden expandir o contraer. Por otro lado, la ficha Diseño muestra el cuadro de diseño del elemento inspeccionado. Nos indica las dimensiones del cuadro, además de indicar los valores en píxeles de diferentes propiedades.



Inspector Web

En el caso de que nuestro navegador principal de desarrollo Web sea Safari podemos usar la herramienta Inspector Web.

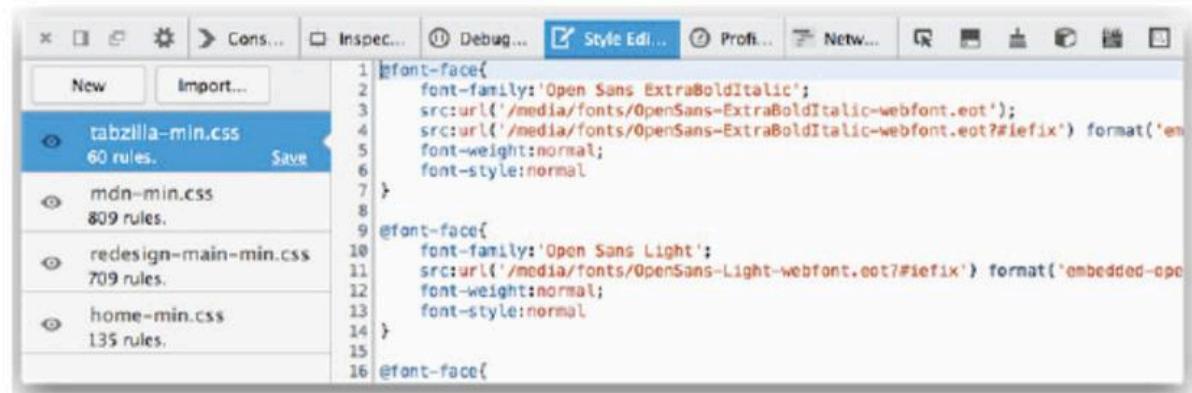
Para activarlo, sólo tendremos que seleccionar Preferencias de Safari>Avanzado y activar la casilla de verificación Mostrar el menú Desarrollo en la barra de menús. Una vez hecho esto, podremos activar el Inspector Web con el comando Control+Mayús+I (como sucede con Dragonfly, este método no se puede utilizar para cerrarlo, se hará a través del ratón).

El método abreviado Control+W sólo funciona cuando Inspector Web se encuentra en una ventana aparte o cuando éste se encuentre acoplado a la ventana principal.

Inspector Web

Firefox Developer Tools es un conjunto de herramientas de desarrollo web integradas en Firefox. Puede usarse para examinar, editar y depurar HTML, CSS y JavaScript.

Para abrir el Editor de estilos, elegiremos la opción "Editor de estilos" en el menú "Desarrollador web" (que es un submenú en el menú "Herramientas" en Mac). La Caja de herramientas aparecerá en la parte inferior de la ventana del navegador, con el Editor de estilos activado:

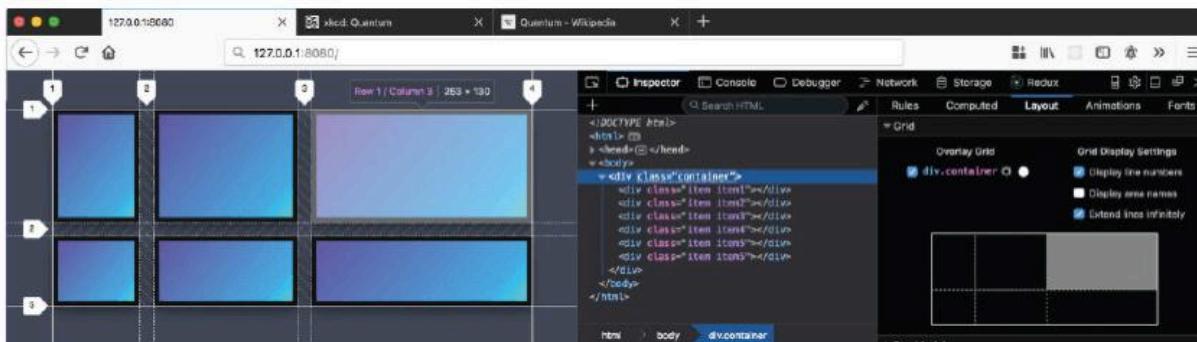


Navegador Firefox Developer Edition

Es el navegador hecho para desarrolladores.

Incluye todas las herramientas para desarrolladores más recientes de la versión beta, además de funcionalidades experimentales como el editor multilínea de consola y el inspector de WebSocket.

Un perfil y ruta separados para que se pueda ejecutar fácilmente en paralelo a la versión oficial o beta de Firefox.



1.2. Hojas de estilo y accesibilidad

A la hora de realizar nuestra página web deberemos tener en cuenta ciertas pautas de accesibilidad. Debemos tener en cuenta que los usuarios pueden entrar en nuestra página web desde contextos muy diferentes, por lo que deberemos realizar tanto la estructura de la página como el diseño de ésta acorde a las pautas que nos recomiendan desde la W3C. Centrándonos en el diseño, los problemas ante los que se puede enfrentar un usuario son los siguientes:

No pueden ver, escuchar, moverse o procesar información fácilmente o en absoluto.

Tienen dificultad de lectura o comprensión de un texto

Es posible que no tengan o no sean capaces de utilizar un teclado o ratón.

Es posible que tengan una pantalla donde sólo se visualice texto, una pantalla pequeña o una conexión lenta.

Es posible que desconozca el idioma en que esté redactado el documento o no pueda entenderlo con fluidez.

Es posible que esté navegando en situaciones en la que sus ojos, oídos o manos están ocupados o no pueda usarlos (trabajando en entornos con ruido, por ejemplo).

Es posible que estén usando un navegador en una versión antigua, diferente o un navegador de voz, o usando un sistema operativo diferente.

Si el desarrollador web tiene en cuenta estas situaciones diferentes a la hora de diseñar la página, estará realizando una página web totalmente accesible

para diferentes grupos de personas con discapacidad. De hecho, si de entrada ya hacemos nuestras páginas web usando hojas de estilo para controlar ciertos aspectos de la página, como por ejemplo el tipo de fuente (hará que eliminemos el elemento del documento HTML, haremos que la web sea más accesible para todos los usuarios, tendremos un mayor control sobre ésta y acortaremos el tiempo de carga de la misma).

Los motivos principales por los que se recomienda hacer una página accesible son 2:

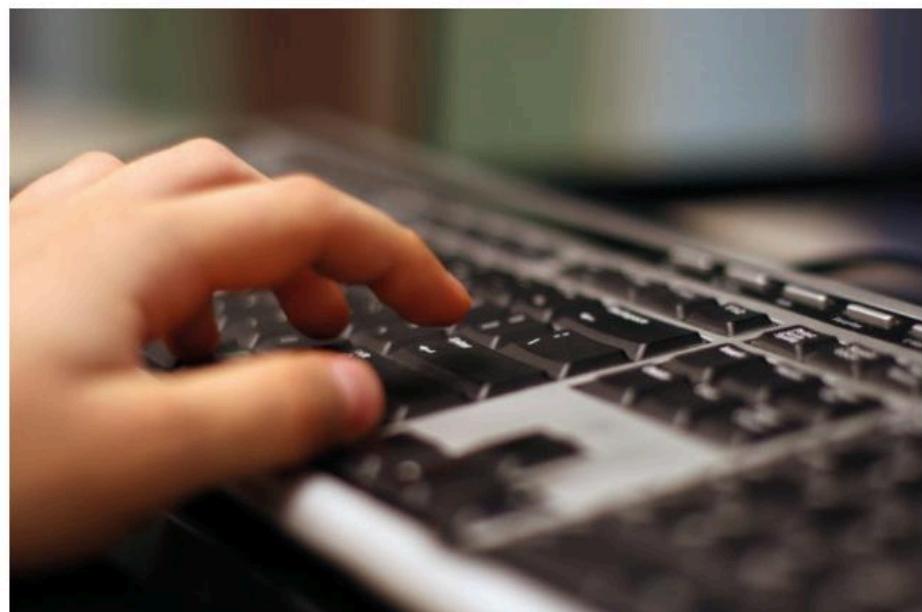
- Asegurar una transformación correcta: es decir, podemos crear páginas que se transformen de manera correcta, por ejemplo, separando el contenido de la estructura y de la presentación, proporcionar textos, proporcionar alternativas a los contenidos multimedia que no puedan ser visualizados o crear documentos que puedan funcionar con diferentes tipos de hardware.
- Hacer comprensible y navegable el contenido: es decir, que el contenido sea altamente comprensible y navegable de una forma fácil y sencilla, por ejemplo, proporcionando mapas del sitio, barras de desplazamiento, etc.

A continuación, vamos a ver cuáles son las pautas que se recomiendan para que una página web sea accesible.

- Proporcionar alternativas equivalentes para el contenido visual y auditivo.
- No nos basaremos en el color. Los textos y gráficos deberán ser comprensibles cuando se vean sin color.
- Usar marcadores y hojas de estilo apropiadamente. Diferenciar entre contenido, estructura y presentación.
- Identificar el idioma usado. Usar marcadores que faciliten la interpretación de abreviaturas o extranjero.
- Hacer tablas que puedan transformarse correctamente, mediante navegadores accesibles y otras aplicaciones de usuario.
- Si la página incorpora nuevas tecnologías hacer que éstas se transformen correctamente.
- Hacer que el usuario tenga el control sobre objetos de la página que se desplazan, parpadean o se actualizan automáticamente, haciendo que puedan ser detenidos.

- Hacer una interfaz con un diseño accesible. En caso de incrustar un objeto que tenga su propia interfaz, ésta debe ser accesible o proporcionar una alternativa.
- Haremos un diseño para la independencia del dispositivo, es decir, que permita que los elementos de la página se activen independientemente del dispositivo de entrada que se use.
- Utilizar soluciones de accesibilidad provisionales, para que las ayudas técnicas y antiguos navegadores puedan operar de forma correcta.
- Hacer uso de las tecnologías y pautas W3C, de acuerdo con las especificaciones. En caso contrario, podemos proporcionar una versión alternativa del contenido que sea accesible.
- Dar información de contexto y orientativa, para que los usuarios puedan entender páginas o elementos complejos.
- Dar mecanismos claros de navegación claros y coherentes: mapa del sitio, barras de navegación, etc., para aumentar la probabilidad de que el usuario encuentre lo que busca en el sitio.
- Nos aseguraremos de que los documentos sean claros y simples, es decir, fácilmente comprendidos.

Como última recomendación, el desarrollador deberá validar la accesibilidad del sitio. Para ello existen diferentes herramientas que podemos encontrar en internet. Con estas herramientas podremos validar la sintaxis del código, las hojas de estilo, etc.



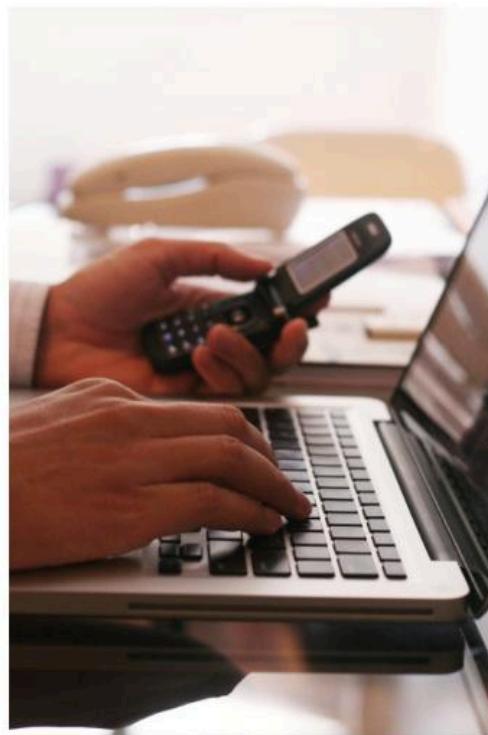
1.2.1. Adecuación de las hojas de estilos

Existen ciertas técnicas que podemos utilizar para crear hojas de estilos en cascada accesibles. Estas técnicas ayudarán a crear CSS de acuerdo con las Pautas de Accesibilidad al Contenido en la Web 1.0, sin embargo, estas técnicas no garantizan la conformidad, y por otro lado podremos realizar una web accesible de otra manera y que tenga contenidos conformes a estas pautas.

Las técnicas que se recomiendan para crear CSS accesible son las siguientes:

- Se recomienda usar un número mínimo de hojas de estilo para la página web.
- Se aconseja usar hojas de estilo vinculadas en vez de estilos incrustados (directamente en las etiquetas). Además, es mejor evitar hojas de estilo incrustadas (directamente en la página).
- En el caso de tener más de una hoja de estilo, usaremos el mismo nombre de clase ("class") para el mismo concepto en todas las hojas de estilo.
- Utilizaremos para definir el tamaño de la letra la unidad "em".
- Utilizaremos unidades de medida relativas y porcentajes. Podremos usar medidas absolutas de longitud, pero sólo en el caso de que las características físicas del medio de salida sean conocidas, como por ejemplo, imágenes de mapa de bits.
- Deberemos dar un equivalente textual para las imágenes o textos importantes generados por la hoja de estilo, por ejemplo, mediante las propiedades "background-image o "content").
- Deberemos asegurarnos de que el contenido importante de la página siempre aparezca dentro del objeto del documento. El texto generado por las hojas de estilo no forma parte del código fuente del documento y no estará disponible para las ayudas técnicas que acceden al Modelo de Objeto del Documento Nivel 1 ([DOM]).
- Se recomienda especificar siempre un tipo de letra genérico por defecto.
- Podremos controlar la información de la fuente a través de las siguientes propiedades: "font", "font-family", "font-size", "font-size-adjust", "font-stretch", "font-style", "font-variant", y "font-weight", en lugar de los siguientes elementos y atributos de tipo de letra desaconsejados en HTML: font, basefont, "face", y "size".

- Los elementos de HTML big y small que controlan la información sobre el tipo de letra pueden usarse, ya que no están desaconsejados.
- Pueden usarse las siguientes propiedades CSS para conformar estilo al texto:
 - Mayúsculas/minúsculas: "text-transform" (para mayúsculas, minúsculas y primera letra mayúscula).
 - Efectos de sombra: "text-shadow".
 - Subrayado: "text-decoration".
- Se recomienda no usar los elementos Blink o Marquee, ya que estos no aparecen en ninguna especificación W3C para HTML, es decir, no son elementos estándares.
- Se aconseja usar hojas de estilo para dar estilo al texto, en vez de representar éste a través de imágenes. De esta forma la información del documento estará disponible para un mayor número de usuarios.



- Podremos hacer uso de las siguientes propiedades para tener control sobre el formateo y la posición del texto:
 - Sangría: "text-indent". No se recomienda el uso de Blockquote u otro elemento estructural para hacer sangrías en el texto.

- Espaciado de letras o palabras: "letter-spacing", "word-spacing". Imaginemos que queremos poner un espacio entre letras y escribir "H O L A". Los usuarios normalmente reconocerán la palabra como "hola", pero un usuario con lector de pantalla leería esta palabra como letras independientes. Para evitar esto, podemos aplicar el mismo efecto visual aplicando la propiedad "word-spacing".
 - Espacio en blanco: "white-space". Con esta propiedad podremos controlar el espacio blanco que se interpreta en el contenido de un elemento.
 - Dirección del texto: "direction", "Unicode-bidi".
 - Pseudoelementos: first-letter y :first-line. Éstos permiten a los autores hacer referencia a la primera letra o línea de párrafo del texto.
- Hacer uso de números en vez de nombres para especificar colores, de la forma siguiente:
- "color", para el color de primer plano del texto.
 - "background-color", para el color de fondo.
 - "border-color", "outline-color" para colores de bordes.
 - Para los vínculos, haremos referencia a las pseudo-clases :link, :visited, y :active.
- Deberemos asegurarnos de que los colores de primer plano y de fondo tengan un buen contraste. Si especificamos el color de primer plano, deberemos siempre especificar el color de fondo (y viceversa).
- Se recomienda que la información no sólo se transmite a través del color.
- Se aconseja usar UL para listas sin ordenar y OL para las ordenadas, de forma que podamos proporcionar indicaciones contextuales junto con el CSS.
- Hasta que CSS sea algo que se utilice de forma amplia por las aplicaciones de usuario, o éstas permitan controlar la interpretación de las listas a través de otros medios, deberemos asegurar el proporcionar pistas contextuales en las listas anidadas no numeradas.

- Los contenidos tendrán que estar maquetados, ubicados, colocados en capas y alineados mediante hojas de estilo (a través de las propiedades CSS de float y colocación absoluta) y no mediante tablas:
 - Las propiedades "text-indent", "text-align", "Word-spacing" y "Font-stretch", harán que los usuarios puedan controlar el espaciado sin tener que añadir espacios adicionales. Se recomienda el uso de 'text-align:center' en vez del elemento Center.
 - No se recomienda el uso de espacios de no separación (), en lugar de éste, podremos usar las propiedad 'margin', 'margin-top', 'margin-right', 'margin-bottom' y 'margin-left' para crear espacios a los cuatro lados del contenido.
 - Podremos usar las propiedades "float", "position", "top", "right", "bottom" y "left" para que el usuario pueda controlar la posición visual de cualquier elemento, independientemente de donde aparezca éste dentro del documento.
 - A través de la propiedad "empty-cells" permitiremos que los usuarios puedan dejar vacías tablas y así proporcionarles borden en la pantalla o en papel. Una celda vacía nunca deberá llenarse con un espacio en blanco o un espacio "non-breaking" sólo para lograr un efecto visual.



- Se recomienda proporcionar textos equivalentes para todas las imágenes, incluidas las que son invisibles o transparentes. Si el diseñador no puede usar hojas de estilo y tiene que utilizar imágenes invisibles o transparentes, deberá especificar alt="" para ellas.

- Haremos uso de las hojas de estilo para crear líneas y bordes. Para la mayoría de los usuarios, estos elementos dan un efecto de separación en la página web, pero este sentido no puede ser deducido fuera de un contexto visual. Las propiedades CSS para especificar el estilo de los bordes que se recomiendan son:
 - "border", "border-width", "border-style", "border-color".
 - Para las tablas: "border-spacing" y "border-collapse".
 - Para contornos dinámicos: "outline2", "out-color", "outline-style" y "outline-width".
- Se recomienda que la presentación del contenido siempre sea comprensible cuando no se apliquen hojas de estilo.
- Deberemos tener en cuenta la necesidad de utilizar propiedades auditivas de CSS, a través de las cuales podremos proporcionar información para invidentes y usuarios de navegadores de voz.

Se recomiendan en este caso las siguientes propiedades:

- "volumen" con el que podremos controlar el volumen del texto hablado.
- "speak" determina si el contenido se pronunciará, y en caso de que así sea, si se debe deletrear o leer como palabras.
- "pause", "pause-before" y "pause-after" son las propiedades que controlan las pausas antes y después de anunciar el contenido, lo cual permite separarlos para una mejor comprensión de los mismos.
- "cue", "cue-before" y "cue-after" especifican un sonido que se reproducirá antes y después del contenido, lo que puede ser beneficioso para orientar.
- "azimuth" y "elevatio" dan diferentes dimensiones al sonido, lo que permite a los usuarios distinguir voces, por ejemplo.
- "speech-rate", "voice-family", "pitch", "pitch-range", "stress" y "richness" son propiedades que controlan la calidad del contenido hablado. De esta forma el usuario podrá ajustar al detalle la presentación sonora-auditiva del contenido.

- "speak-punctuation" y "speak-numeral" son propiedades que controlan la forma de decir los números y la puntuación, lo que aumenta la calidad de experiencia para la navegación por voz.
- "speak-header" es la propiedad que describe cómo se debe decir la información de los encabezados antes de una celda de tabla.
- Se recomienda crear diferentes hojas de estilo para adaptar la presentación del documento dependiendo del dispositivo de salida (braille, sintetizadores de voz, pantalla, móvil, etc).

Estas pautas resumen las técnicas CSS para las Pautas de Accesibilidad al Contenido en la Web 1.0. Podemos ver la información ampliada en el siguiente enlace: http://www.discapnet.es/web_accesible/tecnicas/css/WCAG10-CSS-TECHS_es.html



1.3. Tipos de estilo: incrustados, enlazados, importados, en línea

Como ya hemos visto anteriormente, las hojas de estilo se basan en una serie de reglas que definen el estilo de los elementos que han sido estructurados anteriormente en un documento HTML, y que forman la sintaxis del CSS.

De esta forma, cuando un navegador lee una hoja de estilo, el documento adquirirá el formato que hayamos insertado, y será así tal y como será mostrada.

Es por ello que CSS fue toda una revolución tras su aparición. Tanto su flexibilidad como las amplias opciones que nos ofrece para poder realizar una misma tarea, son características que hacen de las hojas de estilo un gran aporte para el HTML.

Existen numerosos beneficios que nos ofrece CSS:

Mayor control de la presentación de varios documentos desde una única hoja de estilo.

Presentación del estilo más controlada y precisa.

Posibilidad de aplicar varias presentaciones a diferentes tipos de medios (pantalla, impresión, etc.).

Técnicas diferentes avanzadas y sofisticadas.

Si atendemos a las características de las hojas de estilo, podemos observar que una de las más importantes es su flexibilidad para poder hacer una misma tarea de múltiples formas.

Si atendemos a cómo aplicar el estilo a un documento HTML, esta característica se confirma, ya que existen 4 formas diferentes de poder incluir CSS en un documento HTML.

Las diferentes formas de aplicar estilo que nos ofrece CSS para poder aplicarlo a nuestro documento web son los siguientes:

Incrustados

Enlazados

Importados

En línea

1.3.1. Descripción de los tipos de estilo

Existen cuatro formas de añadir estilos a un documento Web:

A través de hojas de estilo incrustadas

A través de hojas de estilo enlazadas

A través de hojas de estilo importadas

A través de hojas de estilo en línea

Para entender mejor cómo funcionan las hojas de estilo y cómo se comportan en una página web, primero recordaremos como lo suele hacer el código HTML.

HTML utiliza tags o etiquetas, las cuales dan una información relativa a un determinado contenido de la página, el cual realiza una acción sobre el mismo. Por ejemplo, veamos el siguiente código:

```
<H1>Hola Mundo</H1>
```

Como podemos observar en el ejemplo, hemos puesto el texto Hola Mundo entre etiquetas <H1>, las cuales darán un aspecto de encabezado al texto. Son éstas etiquetas las que directamente dan formato al texto en particular.

En el caso de integrar hojas de estilo a nuestro documento HTML, podremos ver más adelante que la forma en que se hacía con las etiquetas en HTML cambia por completo.

Observaremos que el estilo siempre quedará diferenciado de la estructura HTML, lo que ofrece numerosas ventajas, ya que por un lado ahorraremos tiempo a la hora de buscar un estilo en concreto, modificarlo o agregarlo.

De esta forma, el desarrollador web tendrá un control absoluto no sólo del estilo, sino también de la estructura HTML, la cual quedará mucho más limpia de código, ganando en claridad y con menos peso de código.

1.3.2. Enlazar una hoja de estilo externa a un documento HTML

La forma más popular y recomendada de aplicar estilo a un documento HTML es enlazando una hoja de estilo externa al documento HTML con el que estamos trabajando. Este tipo de estilo es ideal para grandes sitios web, donde se contienen todas las reglas y son enlazadas desde la cabecera de cada documento HTML que forma el sitio web.

Por lo tanto, por un lado, tendremos un archivo con nuestro documento HTML, el cual contendrá la estructura del sitio, y por otro, tendremos un archivo diferente a éste, donde escribiremos el código CSS para dar el formato al documento HTML, que será nuestra hoja de estilo externa.

Por lo tanto, tendremos dos documentos diferentes, por un lado un documento HTML y por otro lado un documento CSS (nuestra hoja de estilos).

Pero, ¿cómo van unidas estos dos documentos para que el estilo afecte a nuestro documento HTML? Éstos son los pasos que deberemos seguir para enlazar una hoja de estilos externa a nuestro documento HTML:

Deberemos tener un documento con la estructura de nuestra página web. Como ya sabemos, ésta debe estar guardada con la extensión .htm o .html.

Por otro lado, crearemos un nuevo archivo de texto. Éste será el que contenga el código CSS. Deberemos guardararlo con la extensión .css.

En nuestro documento HTML, enlazaremos el archivo CSS mediante la etiqueta <link> desde la cabecera de cada documento, de la siguiente forma:

```
<head>

    <link rel="stylesheet" type="text/css" href="miestilo.css" media=screen>

</head>
```


Lo que ocurrirá cuando el navegador cargue la página HTML, será mostrar los contenidos, pero a su vez también descargará los archivos CSS que van enlazados mediante la etiqueta `<link>`, por lo que aplicará los estilos que hemos definido en la página.

Enlazar una hoja de estilo externa a un documento HTML (continuación).

Veamos en profundidad los atributos que se incluyen en la etiqueta `<link>` cuando enlazamos un archivo CSS:

- rel: este atributo indica el tipo de relación que hay entre el recurso enlazado, en nuestro caso un archivo CSS, y el documento HTML. Como en este caso el recurso enlazado es un archivo CSS, siempre se usa el valor `stylesheet`.
- type: este atributo indica el tipo de recurso enlazado. Los valores de este atributo están estandarizados. Concretamente para los archivos CSS el valor que usará siempre es `text/css`.
- href: este atributo indica la URL del archivo CSS, el cual contendrá los estilos del documento. La URL que se indica puede ser relativa o absoluta, y puede apuntar a un recurso interno o externo al sitio web.
- media: es un atributo opcional, que especifica el medio o medios en que el estilo se aplicará. Los valores posibles son:
 - screen: es el valor por defecto, se usa para pantallas de computadoras no-paginadas.
 - print: para salida a una impresora.
 - projection: para presentaciones en proyectores.
 - aural: para sintetizadores de voz.
 - speech: para sintetizadores para navegadores de voz utilizados por personas discapacitadas.
 - handheld: para dispositivos de mano como móviles, PDA,etc.
 - embossed: para impresoras braille.
 - braille: para presentaciones en dispositivos braille.

- tty: para pantalla en celda de caracteres (utilizando una fuente de inclinación fija).
- tv: para televisores.
- all: para todos los dispositivos de salida.

Para especificar varios medios a la vez, éstos irán separados por comas o a través del valor all.

Usar un enlace a una hoja de estilos externa a un documento HTML es la forma más eficiente de programar, ya que como hemos dicho antes, tendremos por separado el estilo de la estructura básica del documento HTML.

Este tipo de estilo logrará que el mantenimiento del sitio web sea mucho más simple, ya que si queremos realizar un cambio, simplemente tendremos que abrir nuestro archivo CSS y variar al instante el estilo que deseemos, de forma rápida y sencilla.

Además de esta forma, la uniformidad y los estilos se conservan mejor en múltiples documentos HTML, sin tener que repetir las reglas para cada uno, es decir, podremos usar un único fichero CSS para definir el estilo de numerosos documentos HTML.

A continuación, puedes ver en la siguiente imagen cómo quedaría la organización de nuestras carpetas y documentos:



Como puedes observar en la imagen, la mejor forma de organizarse es la siguiente:

Por un lado, tendremos una carpeta con los documentos HTML. Por otro lado, tendremos una carpeta donde guardaremos los documentos CSS. Ambas

carpetas están relacionadas entre sí gracias al vínculo que introduciremos en el código HTML.

Por lo tanto, este vínculo que está metido en el código HTML, y que se limita a una línea de texto, llamará a la hoja de estilo que hemos guardado en otra carpeta diferente, y será la que dará formato a nuestra página web. Es este vínculo concretamente el que indica al navegador qué presentación se debe mostrar del documento HTML. Lo que realmente es práctico en sí es que podemos vincular un único fichero a múltiples ficheros HTML.

Imaginemos por ejemplo que tenemos un sitio web que consta de 100 documentos web (100 páginas web). Si quisiéramos cambiar el color de fondo del sitio, no tendríamos que hacerlo uno por uno, ya que simplemente iremos a nuestro archivo CSS y cambiaremos la propiedad que pertenece al color de fondo, en pocos segundos.

Vamos a practicar ahora para enlazar una hoja de estilos externa a nuestro documento HTML. Los pasos a seguir son los siguientes:

- Abrir el bloc de notas o el editor de texto con el que habitualmente trabajemos. Crearemos dos ficheros, uno HTML y otro CSS. El contenido de cada uno será el siguiente:

- Fichero mipagina.html

```
<html>

    <head>
        <title>Mi documento</title>

        <link rel="stylesheet" type="text/css"
 href="css/miestilo.css" />

    </head>

    <body>
        <h1>Mi primera hoja de estilo</h1>
    </body>
</html>
```

- Fichero miestilo.css

```
body {
```



```
background-color: #FF0000;
```

```
}
```

- Recuerda que hemos creado una carpeta para cada documento. Deberás tener en cuenta la URL para poder enlazar correctamente la hoja de estilo al documento HTML. La carpeta css está guardada dentro de la carpeta www. Es por esto que la ruta que hemos puesto es css/miestilo.css.
- No olvides poner siempre la extensión de cada documento en la URL.
- Al abrir el archivo mipagina.html con el navegador, observarás que la página tiene un fondo de color rojo. Esto es gracias a nuestra hoja de estilo.
- Finalmente, la página deberá quedar como vemos en la siguiente imagen:



Como puedes observar en la imagen, sólo aparece el texto que hemos puesto en nuestro documento HTML, con formato de encabezado de nivel 1. Sin embargo, el fondo es rojo, a pesar de que en el documento HTML hayamos expresado en ningún momento ninguna etiqueta para expresarlo.

Este fondo rojo se visualiza gracias a la hoja de estilos externa que hemos enlazado, donde sí que hemos escrito el código pertinente para que el fondo cambie del color predeterminado de la web a un color rojo.

De esta forma, el navegador al leer el documento HTML interpreta que existe una hoja de estilos externa enlazada y a través de ésta hace que podamos visualizar el formato que hemos escrito en nuestra hoja de estilo.

1.3.3. Incrustar un estilo dentro de un documento HTML.

Otra de las formas para definir el estilo de un documento HTML es incrustando un estilo dentro del documento. Esto quiere decir que los estilos estarán definidos dentro del propio documento HTML, por lo que a diferencia de la hoja de estilos externa, no tendremos que hacer un archivo diferente al que ya tenemos.

Lo que obtendremos con este tipo de estilos será tener separados, en un mismo documento, la información o estructura de la página del estilo.

Utilizar este método nos servirá para definir un pequeño número de estilos o cuando se quieran incluir estilos muy específicos a un determinado documento web, para completar los estilos que se incluyen por defecto en todas las páginas del sitio web.

Sin embargo, utilizar este tipo de estilo tiene un inconveniente, y es que si queremos modificar alguno de los estilos definidos, tendríamos que modificar todas las páginas que incluyan el estilo que deseemos modificar, lo cual nos hará perder bastante tiempo.

Veamos entonces cómo incrustar un estilo dentro del propio documento HTML. Éstos son los pasos a seguir:

- Imaginemos que tenemos el siguiente documento HTML básico:

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <p>Esta es una página sin estilo incrustado.</p>
  </body>
</html>
```


- Lo único que debemos hacer es emplear la etiqueta `<style>` de HTML, siempre dentro de la cabecera del documento, es decir dentro de `<head></head>`.
- A continuación de la etiqueta `<style>`, escribir los estilos que deseemos.
- De esta forma, el estilo estará definido en una parte del documento HTML, es decir, en la cabecera del documento, y a partir del cuerpo del documento sólo tendremos la estructura. Finalmente, el documento HTML quedaría así:

```
<html>

    <head>

        <title>Ejemplo</title>

        <style type="text/css">

            body {background-color: #FF0000; }

        </style>

    </head>

    <body>

        <p>Esta es una página con fondo rojo.</p>

    </body>

</html>
```

Como puedes observar en el ejemplo, hemos utilizado la etiqueta `style`, pero además va acompañada del atributo `type`, el cual indica el tipo de recurso enlazado. Los valores de este atributo están estandarizados. Concretamente para los archivos CSS el valor que usará siempre es `text/css`. Seguidamente hemos escrito el estilo, el cual en este caso haría que el fondo de la página fuera de color rojo.

Por otro lado, también podremos usar el atributo `style` para incrustar un estilo dentro de un documento HTML, pero a diferencia del uso que hemos visto antes, éste podrá ser aplicado a cualquier elemento `<body>` del documento HTML. A éste tipo de estilo se le denomina "estilo en línea".

El estilo en línea no es precisamente el estilo que más se recomienda, al contrario, se suele evitar por su inflexibilidad, y por su imposibilidad de separación entre contenido y presentación.

Veamos a continuación cuáles son los pasos que deberemos seguir para usar el estilo en línea dentro de un documento HTML:

- Tener un documento HTML con una estructura básica.
- Para poder usar estilo en línea, deberemos declarar un único lenguaje de hojas de estilo para el documento. Para esto, podemos usar la siguiente etiqueta META en la cabecera del documento:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/css">
```

- A continuación, usaremos el atributo `<style>`, el cual podrá ser aplicado a cualquier elemento del cuerpo del documento (incluyendo la misma etiqueta body), a excepción de las etiquetas `<basefont>`, `<param>` y `<script>`. La forma de usar el atributo style es la siguiente:

```
<P STYLE="color: red; font-family: 'Arial', serif">Este párrafo usa estilos en rojo con la fuente Arial, si está disponible.</P>
```

Observa en el ejemplo como hemos puesto justo después de la etiqueta que designa un párrafo (`<p>`) el atributo style, y a continuación hemos insertado cada uno de los atributos junto a su valor, cada uno de ellos separado por punto y coma (;).

Volviendo al ejemplo que hemos estado utilizando para poner un fondo rojo en la página a través de CSS, hagamos lo mismo usando el estilo en línea. El resultado en el código sería el siguiente:

```
<html>
  <head>
    <title>Ejemplo de estilo en línea</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>Esta es una página con fondo rojo</p>
  </body>
</html>
```


1.3.4. Importar una hoja de estilo desde un documento HTML

Otra de las formas de añadir estilo a nuestra página web es importando una hoja de estilo desde un documento HTML. Anteriormente hemos visto cómo enlazábamos una hoja externa a nuestro documento HTML a través de la etiqueta `<link>`. Pues bien, para poder importar una hoja de estilos, haremos lo mismo, sin embargo, no usaremos la etiqueta `<link>`.

En este casoaremos uso de la etiqueta `<style>`, con la que podremos enlazar a archivos CSS externos, y por lo tanto importar los estilos que hayamos incluido en esta hoja de estilos externa para poder aplicarlos a nuestro documento HTML.

Por otro lado, también deberemos tener en cuenta, a la hora de importar una hoja de estilos a nuestro documento, el uso de la regla `@import`. Esta regla será lo que nos permita importar una hoja de estilos externa dentro de nuestras etiquetas `style`.

Haciendo uso de la etiqueta `<style>` por un lado y de la regla `@import`, podremos importar la hoja de estilos que tengamos preparada para aplicar el estilo a nuestra página web.

Los pasos que daremos para importar una hoja de estilos desde un documento HTML utilizando estos elementos son los siguientes:

- Tener por un lado un documento HTML con una estructura simple, y por otro un documento CSS externo, ambos guardados en sus carpetas correspondientes, como hemos visto hasta ahora.

Introducir dentro del encabezado la sintaxis siguiente:

```
<style>  
@import url("/css/miestilo.css");  
</style>
```

- Esta sintaxis deberá ir siempre dentro del encabezado del documento HTML.
- Las reglas del tipo `@import` siempre preceden a cualquier otra regla CSS, a excepción de la regla `@charset`.
- También podremos usar la regla `@import` de forma abreviada, sin usar la clave URL:


```
<style>@import  
  "estilos.css";  
</style>
```

En ambos casos, el navegador cargará la hoja de estilos y definirá su contenido a la página actual.

- Por otro lado, también podremos importar un archivo dependiendo del tipo de medio para el que será destinado, es decir, si queremos una hoja de estilos concretamente para imprimir y otra hoja de estilos para la página (se vista por pantalla), lo haremos usando el siguiente código:

```
<style>  
  @import url("impresora.css") print;  
  @import url("normal.css") screen;  
</style>
```

De esta forma, tendremos dos tipos de estilos diferentes: uno para la versión imprimida de la página (a través del elemento `print`) y otro para la versión vista en pantalla (a través del elemento `screen`).

1.3.5. Importar una hoja de estilo a través de un archivo con estilos

El método `@import` también puede usarse para importar una hoja de estilos dentro de otra hoja de estilos.

La regla `@import` nos permitirá importar hojas de estilo desde otras hojas de estilo. De esta forma, tendremos dos documentos. Por un lado, el documento HTML donde tendremos la estructura de la página web, y por otro, la hoja de estilos con los estilos que hayamos definido, y donde habremos introducido la regla `@import` para aportar aún más estilos al documento CSS.

Esta regla se puede usar de dos formas diferentes:

Regla 1:

```
@import "/css/miestilo.css";
```


Regla 2.

```
@import url("/css/miestilo.css");
```

Esta regla siempre deberá ir antes que cualquier otra regla en la hoja de estilo, por el contrario, cualquier regla especificada en la misma hoja de estilo prevalecerá sobre reglas contradictorias en las hojas de estilo importadas.

1.3.6. Utilización y optimización de los tipos de estilos

Ya hemos visto todas las formas que existen para poder dar estilo a nuestra página web. Sin embargo, deberemos saber cuál deberemos usar en cada momento, dependiendo del uso que queramos darle.

Por un lado, tenemos la opción de definir CSS a través de un enlace a un archivo externo a través de la etiqueta `<link>`. Este es el método que se recomienda utilizar para dar estilos al documento HTML por su flexibilidad. Además, este método es el más elegido entre la comunidad de diseñadores, ya que les permite aplicar de una forma muy homogénea los estilos a todas las páginas que forman el sitio web.

Por otro lado, mantener la página web se convierte en algo fácil, convirtiendo cualquier trabajo de cambio de estilos en algo instantáneo y simple, independientemente de que se quiera cambiar un estilo en particular o un estilo que afecte a múltiples documentos de una misma página.

Otra de las formas que tenemos de enlazar a un estilo es a través del método `@import`, aunque este método no es tan popular como el anterior. Se suele utilizar cuando queremos importar un estilo a una hoja de estilos, de forma que podremos tener varias opciones en la página para visualizar un mismo documento. Por ejemplo, podemos tener un estilo básico, uno imprimible y uno para visualizar en pantalla, y el usuario podría elegir el que mejor le convenga según le convenga.

También hemos visto que podemos añadir estilos, pero sin enlazar a ningún documento externo, es decir, aplicando el estilo en el mismo documento HTML, a través de la etiqueta `<style>`. Este método suele utilizarse cuando vamos a aplicar un número muy pequeño de estilos, o cuando queramos especificar un estilo en una determinada página HTML del sitio, pero no se recomienda ya que siempre intentaremos separar el contenido de la presentación.

Podemos utilizar la etiqueta `<style>` en el encabezado de la página, donde irán los estilos que queremos aplicar a ese documento web en concreto, o

podemos aplicar la etiqueta `<style>` en línea, es decir, directamente en los elementos del cuerpo del documento.

Aplicar CSS con la etiqueta `<style>` en el encabezado nos servirá para modificar ciertos estilos que ya estaban definidos. El inconveniente de utilizar este tipo de estilos, es que si se trata de un sitio con un gran número de páginas web, tendremos que cambiar ese estilo en cada una de las páginas que aparezca, por lo que el principal motivo de usar estilos que es facilitar el cambio de los mismos desaparece.

El tipo de estilo en línea es más bien recomendado cuando tenemos que cambiar algo muy concreto de todo el sitio web.

1.4. Selectores y reglas de estilo

Para poder entender el funcionamiento de CSS, comprender el concepto de selector es algo básico. Como hemos visto anteriormente, a través de las hojas de estilo podemos separar la visualización estética del contenido y estructura. Pero ¿cómo se asignan los estilos a diferentes partes del contenido, como párrafos, tablas o formularios?

Podremos asignar estilos al contenido y estructura de una página a través de los selectores. Los selectores servirán para unir los estilos que hemos definido a los elementos HTML, de forma que quedarán unidos la visualización y el contenido.

Los selectores se encargan de especificar a qué elementos van asignados los estilos que hemos creado. Por ejemplo, si hemos creado un estilo para que sea utilizado por todos los párrafos o si queremos que sea utilizado por un párrafo en concreto, tendremos que especificarlo con los selectores.

Los selectores pueden ser de dos tipos:

Selectores Simples

Selectores Contextuales

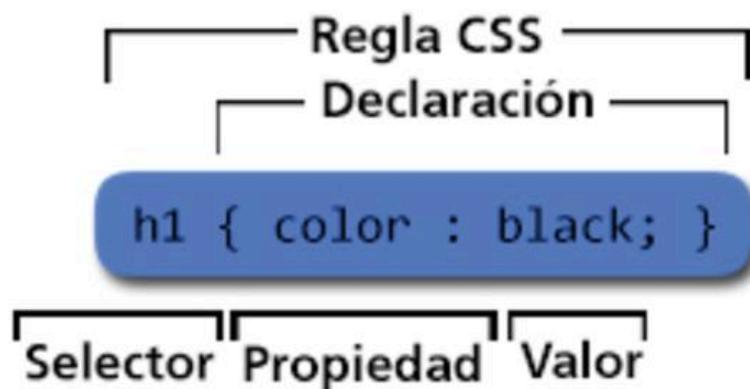
Los selectores simples se refieren al tipo de elemento, y los selectores contextuales se refieren a su posición dentro de la página.

Podemos decir entonces que los selectores son el corazón de CSS, ya que sin ellos no podríamos asignar estilos a los elementos excepto incorporándolos a sus atributos, lo que resultaría bastante laborioso.

Gracias a los selectores podemos seleccionar familias enteras de elementos a los podremos aplicar estilo, y realizar grandes cambios en el documento a través de unas escasas líneas de CSS.

1.4.1. Estructura de los estilos

Dentro de CSS utilizaremos una serie de términos que nos ayudarán a describir cada una de las partes que utilizaremos a la hora de realizar una hoja de estilos. Veamos a continuación un estilo muy básico junto a las partes que lo forman.



Como puedes ver en la imagen, un estilo CSS básico se compone de diferentes términos, como Regla, Selector, Propiedad o Valor. Pues bien, si nos fijamos bien en la Regla del estilo, veremos que esta parte se refiere a la estructura que compone en sí un estilo de CSS.

Cada Regla estará compuesta por las siguientes partes:

- Selector.
- Símbolo de llave de apertura ({).
- Declaración, finalizada en punto y coma (;).
- Símbolo de llave de cierre (}).

Una hoja de estilos puede contener multitud de reglas CSS, estas reglas podrán aplicarse a varios selectores diferentes y cada declaración podrá incluir tantos pares de propiedad/valor como deseemos.

El estándar CSS 2.1 ha llegado a definir 115 propiedades en total, cada una de ellas con su lista de valores correspondientes permitidos. En los últimos borradores del estándar CSS 3 se han contabilizado 239 propiedades.

Para estudiar la estructura de los estilos hemos visto cómo se componen los selectores. Pero, además, deberemos tener en cuenta otras dos características muy importantes a tener en cuenta en la estructura de éstos:

- Herencia.

En CSS es muy importante el parentesco entre los elementos y la herencia de los estilos y las propiedades. En CSS, los hijos, la mayoría de las veces, heredan los estilos de sus padres.

Es decir, un elemento que contenga a otro le pasará sus características siempre que el elemento contenido no tenga ningún estilo que le indique lo contrario. El ejemplo más común es el tamaño del texto. Se define en el BODY y se hereda al resto de elementos, por lo que si cambiamos una sola regla, estamos cambiando el tamaño de todos los textos de una página.

Debemos tener en cuenta que no todas las propiedades se heredan, pero podemos utilizar la palabra "inherit" ("heredado"), como valor para asegurarnos de que así sea. Por lo tanto, si el valor de una propiedad es inherit, significará que se hereda el valor de esta propiedad de padres a hijos.

- Cascada.

Cuando se hace referencia a CSS como "hojas de estilo en cascada" la comparación que se hace con una "cascada" no es accidental. Al igual que la cascada de un río arrastra hacia abajo todo lo que encuentra a su paso, en CSS las propiedades de los elementos situados más arriba en la jerarquía de la página ceden esas propiedades a sus hijos. Al final, un elemento recibe estilos propios de la página, hay otros estilos que influyen, definidos en el navegador, por ejemplo.

El navegador será el que debe decidir entre toda esta información cual se debe aplicar. La cascada intenta poner orden en todo este lío. Es el conjunto de reglas lo que decide qué estilos se van a mostrar antes que otros.

En CSS podemos hacer trampas y saltarnos la cascada si es necesario. Usando la expresión !important, definida en la declaración de un estilo,

haremos que esa propiedad tiene preferencia sobre cualquier otro que, por ejemplo, sea heredada. Por ejemplo:

```
BODY P { color: #FFCCFF }

P.black { color: #000000 !important }
```

En este caso, aunque el color del párrafo heredado por la primera regla fuese otro, los párrafos de la página se visualizarán de color negro.

1.4.2. Sintaxis básica de estilos

Anteriormente hemos mencionado que los Selectores pueden dividirse en Selectores simples y Selectores contextuales. Veamos en profundidad cuáles pertenecen a cada uno de éstos:

Selectores simples

Los selectores simples identifican los elementos HTML a los que se asignará el estilo según el tipo de elemento que sea o un atributo que posea. Esto podemos entenderlo mejor con el siguiente ejemplo:

```
TD {color: RED}
```

Este selector hará que en todas las celdas (elementos `<td>`) de la página el texto aparezca en color rojo, es decir, el criterio para el selector ha sido que se aplique a un elemento HTML tipo TD y de esta manera asignar a todos los elementos TD que haya en la página ése estilo CSS.

Existen varios tipos de selectores simples:



- Selectores de Tipos

En los selectores de tipos el criterio de selección es un elemento HTML. Es el caso del ejemplo anterior, en el que el estilo es asignado a todos los elementos TD. Este elemento puede ser cualquiera, incluido el BODY con lo que ese estilo se aplicaría a todo el contenido de la página.

```
BODY {font-size: 12px;}
```

Según este selector, todos los textos dentro de la etiqueta BODY de nuestro documento tendrán un tamaño de 12 píxeles hasta que no se indique lo contrario.

El Selector de tipos (o etiqueta), lo que hará será seleccionar todos los elementos de la página donde coincidan una etiqueta HTML con el valor del selector. Por ejemplo:

```
p {  
    ...  
}
```

En este ejemplo, se seleccionan todos los párrafos de la página. Para utilizar este selector, lo único que haremos será indicar el nombre de una etiqueta HTML (sin los caracteres <>) que corresponden con los elementos que se quieren seleccionar.

En el siguiente ejemplo, se aplican diferentes estilos a los encabezados y a los párrafos de un sitio:

```
h1 {color: red;}  
  
h2 {color: blue;}  
  
p {color: black;}
```

También podremos aplicar el mismo estilo a dos etiquetas diferentes, ya que podemos encadenar los selectores. Veamos un ejemplo:

```
h1 {  
    color: #8A8E27;  
    font-weight: normal;
```



```
font-family: Arial, Helvetica, sans-serif;  
}  
  
h2 {  
  
color: #8A8E27;  
  
font-weight: normal;  
  
font-family: Arial, Helvetica, sans-serif;  
}  
  
h3 {  
  
color: #8A8E27;  
  
font-weight: normal;  
  
font-family: Arial, Helvetica, sans-serif;  
}
```

Como podemos observar en el ejemplo, cada uno de los encabezados tienen exactamente los mismos estilos. Para esto, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Esto se puede hacer incluyendo todos los selectores separados por una coma (,) y el resultado es que la siguiente regla CSS equivale a las tres reglas anteriores:

```
h1, h2, h3 {  
  
color: #8A8E27;  
  
font-weight: normal;  
  
font-family: Arial, Helvetica, sans-serif;  
}
```

Deberemos tener en cuenta que en las hojas de estilo más complejas, lo normal es agrupar las propiedades comunes de varios elementos en una única regla, como hemos visto en el ejemplo anterior, pero podemos definir posteriormente las propiedades específicas de esos mismos elementos en el caso de que las tuvieran.

Esto lo podemos ver en el siguiente ejemplo:

```
h1, h2, h3 {  
    color: #8A8E27;  
    font-weight: normal;  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
h1 { font-size: 2em; }  
  
h2 { font-size: 1.5em; }  
  
h3 { font-size: 1.2em; }
```

Como puedes ver, primero hemos establecido las propiedades comunes de los encabezados, su color y su tipo de letra, y seguidamente, hemos establecido el tamaño de letra de cada uno de ellos por separado. Por un lado, tenemos un bloque con las propiedades comunes y por otro las propiedades específicas de cada uno.

- Selectores de Clases.

Son los selectores más utilizados ya que permiten que se asigne un estilo a cualquier elemento HTML utilizando el atributo CLASS. Este atributo permite asignar a cualquier elemento HTML un estilo que nosotros definimos en CSS. La sintaxis del selector es con un punto delante del nombre con el que nombremos al estilo, y la regla será la siguiente:

```
.colorRojo {color: red}
```

Mientras en el código HTML se utilizará el atributo class en tantos elementos como sea necesario:

```
<font class="colorRojo">texto en rojo por CSS</font>  
  
<td class="colorRojo">texto de una celda</td>  
  
<input type="text" class="colorRojo"  
      value="Texto defecto">
```


Veamos un ejemplo más para entender el Selector de clase. Consideremos el siguiente código HTML:

```
<body>

<p>Lorem ipsum dolor sit amet...</p>

<p>Nunc sed lacus et est adipiscing accumsan...</p>

<p>Class aptent taciti sociosqu ad litora...</p>

</body>
```

Imaginemos que queremos aplicar un estilo sólo al primer párrafo. Para ello usaríamos el atributo class sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

```
<body>

<p class="destacado">Lorem ipsum dolor sit amet...</p>

<p>Nunc sed lacus et est adipiscing accumsan...</p>

<p>Class aptent taciti sociosqu ad litora...</p>

</body>
```

Seguidamente en nuestro archivo CSS, crearemos una nueva regla llamada destacado, con los estilos que queramos aplicar a ese elemento en concreto. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.), de la siguiente forma:

```
.destacado { color: red; }
```

Lo que ocurrirá será que el navegador interpretará el selector destacado como cualquier elemento de la página cuyo atributo class sea igual a destacado, y en este caso es el primer párrafo el que cumple esa condición, por lo que será al que se le aplique el estilo definido.

Otra de las características principales de los Selectores de clase es que varios elementos diferentes de una misma página HTML pueden hacer uso del mismo valor en el atributo class. Podemos ver esto de forma clara en el siguiente ejemplo:

```
<body>
```



```
<p class="destacado">Lorem ipsum dolor sit amet...</p>

<p>Nunc sed lacus et <a href="#" class="destacado">est
adipiscing</a> accumsan...</p>

<p>Class aptent taciti <em class="destacado">sociosqu
ad</em> litora...</p>

</body>
```

Cuando estemos diseñando páginas web complejas, los Selectores de clase serán imprescindibles para nuestro trabajo, ya que éstos nos permiten controlar con una alta precisión al seleccionar los elementos.

Por otro lado, los Selectores de clase también pueden reutilizar los mismos estilos para varios elementos diferentes, como hemos podido observar en el ejemplo anterior.

- Selectores de ID.

Estos selectores se basan en el atributo ID de los elementos, y asignan el estilo a cualquier elemento que tenga ese identificador. Por lo tanto, solo sirven para un elemento. Su sintaxis es la siguiente:

```
#parrafo3 {font-size: 16px}
```

Y en el código HTML lo identificaremos así:

```
<p id="parrafo3">Hola Mundo</p>
```

Estos selectores pueden parecer limitados al ser de "un solo uso", pero son muy cómodos para trabajar con capas.

El selector de ID lo utilizaremos cuando tengamos necesidad de aplicar estilos CSS solo a un elemento de la página. Ya sabemos que podemos utilizar el selector de clase para esto, pero el selector de ID es más eficiente en este aspecto.

El selector de ID hará que podamos seleccionar un elemento de la página a través del valor de su atributo id. Así conseguiremos seleccionar un solo elemento de la página, ya que el valor del atributo id no puede repetirse en dos elementos diferentes de una misma página.

La sintaxis en nuestra hoja de estilo de los selectores de Id es muy parecida a la de los selectores de clase, pero cambia en que se usa el símbolo

de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#destacado { color: red; }

<p>Primer párrafo</p>

<p id="destacado">Segundo párrafo</p>

<p>Tercer párrafo</p>
```

Como podemos observar en el ejemplo anterior, el selector #destacado solo selecciona el segundo párrafo, donde su atributo id es igual a destacado.

- Selector Universal.

Como su nombre indica, este selector asignará el estilo a todos los elementos dentro de la página.

```
* {color: RED}
```

Cualquier elemento dentro de la página tendrá asignado ese estilo.

A continuación podemos ver otro ejemplo para ver cuál es la sintaxis para usar el selector universal en nuestra hoja de estilo:

```
* {

margin: 0;

padding: 0;
```

1.4.3. Utilización de elementos y seudoelementos

Una vez que sabemos cuál es la sintaxis y estructura básica de los estilos, vamos a conocer mejor cómo utilizar los elementos y seudoelementos en CSS.

Elementos

Como diseñadores, en ocasiones nos encontraremos con que tendremos que restringir mucho más el alcance del Selector de ID para un elemento en concreto.

Al igual que ocurre con los selectores de clase, también podremos restringir el alcance del selector combinando los selectores. Veamos en el siguiente ejemplo como lo haríamos:

```
p#aviso { color: blue; }
```

Como puedes ver, en el ejemplo sólo se aplica la regla al elemento de tipo `<p>` que tenga un atributo `id` igual a `aviso`.

A priori, restringir el alcance de un selector ID puede no parecer lógico, pero un selector del tipo `p#aviso` sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.

Si se diera este caso, algunas páginas pueden disponer de elementos con un atributo `id` igual a `aviso` y que no sean párrafos, por lo tanto, esta regla no se podrá aplicar sobre esos elementos.

Seudoelementos

Los Seudoelementos nos permiten acceder a elementos específicos que de otra manera no serían identificables a través de HTML. Son los siguientes:



- `:first-line`

Tal y como su nombre indica se refieren a la primera línea del elemento que especifiquemos. Es muy útil para remarcar la primera línea de cada párrafo o cambiar el color de esta. Podemos verlo en un ejemplo fácil:

```
TD:first-line { color: red }
```

Como puedes ver en el ejemplo anterior, con este seudoelemento todas las primeras líneas de una celda se verán en color rojo, sin tener que separarlas del resto de líneas utilizando la etiqueta `` o ``.

- :first-letter

Con este seudoelemento se accede a la primera letra de un elemento, normalmente un párrafo. Como en algunos libros antiguos donde la primera letra de cada página es claramente distinta al resto, en el siguiente ejemplo, vemos como podemos modificar el tamaño de cada primera letra de cada párrafo:

```
P:first-letter { font-size: 15px }
```

- :before y :after

Permiten acceder al elemento anterior y posterior de otro. Si una página posee un listado de múltiples títulos podemos ahorrar tiempo y equivocaciones con estos seudoelementos. En el siguiente ejemplo podemos ver como los párrafos anteriores a un título (H1) se ven a un tamaño más pequeño que el resto para resaltar el título:

```
P { font-size: 12pt }  
H1:before { font-size: 90% }
```



Los seudoelementos :before y :after solo funcionan en navegadores 6.x ya que pertenecen a CSS2, mientras que :first-letter y :first-line son ya compatibles con todos los navegadores actuales.



En todo momento debes tener presente la compatibilidad de tu código con los navegadores. Por suerte hoy en día contamos con un buen número de herramientas que pueden ayudarnos a ello.

1.4.4. Utilización de clases y seudoclases

Una vez que sabemos cuál es la sintaxis y estructura básica de las clases, vamos a conocer mejor cómo utilizar las clases y seudoclases en CSS.

Clases

Como diseñadores, en ocasiones nos encontraremos con que tendremos que restringir mucho más el alcance del Selector de clase para un elemento en concreto. Para hacer esto, consideremos el ejemplo anterior:

```
<body>

    <p class="destacado">Lorem ipsum dolor sit amet...</p>

    <p>Nunc sed lacus et <a href="#" class="destacado">est
adipiscing</a> accumsan...</p>

    <p>Class aptent taciti <em class="destacado">sociosqu
ad</em> litora...</p>

</body>
```

Imaginemos que queremos aplicar estilos concretamente al párrafo cuyo atributo class es igual a destacado. La solución es la siguiente: podremos combinar el selector de tipo y el selector de clase, de manera que obtendremos un selector mucho más concreto y específico:

```
p.destacado { color: red }
```

Analicemos esta combinación: el selector p.destacado es interpretado por el navegador como aquellos elementos de tipo `<p>` que dispongan de un atributo class con el valor destacado.

Igualmente, el selector a.destacado solamente seleccionará los enlaces cuyo atributo class sea igual a destacado.

Por lo tanto, podemos llegar a la conclusión de que el atributo .destacado es lo mismo que *.destacado, aunque por esto mismo la comunidad de diseñadores obvian el símbolo * al escribir un selector de clase normal.

En otras ocasiones, nos encontraremos con la labor de aplicar estilos de varias clases CSS sobre un mismo elemento.

Para realizar esto, utilizaremos la misma sintaxis, pero los valores diferentes del atributo class irán separados con espacios en blanco, como podemos ver en el siguiente ejemplo:

```
<p class="especial destacado error">Párrafo de texto...</p>
```

Podemos observar cómo se aplicarán los estilos que hayamos definido en las reglas .especial, .destacado y .error en el siguiente ejemplo. De esta forma, el párrafo se visualizará en este caso de color rojo, en negrita y con un tamaño de letra de 12 píxeles:

```
.error { color: red; }

.destacado { font-size: 15px; }

.especial { font-weight: bold; }

<p class="especial destacado error">Párrafo de texto...</p>
```

En este mismo caso, si un elemento en concreto debe disponer de un atributo class con más de un valor, podremos acotar más su estilo haciendo uso de un selector más avanzado:

```
.error { color: red; }

.error.destacado { color: blue; }

.destacado { font-size: 15px; }

.especial { font-weight: bold; }

<p class="especial destacado error">Párrafo de texto...</p>
```

Lo que ocurriría, sería que el color de la letra del texto sería azul y no rojo, ya que se ha utilizado un selector de clase múltiple .error.destacado, el cual es interpretado por el navegador como aquellos elementos de la página que dispongan de un atributo class con al menos los valores error y destacado.

Seudoclases

Las seudoclases nos hacen referencia a elementos que están más allá de la estructura del documento y que se escapan al desarrollador, tales como los enlaces visitados o el primer h1o de cada elemento. Las seudoclases son:


```

graph TD
    A[:first-child] --- B[:link]
    A --- C[:visited]
    C --- D[:hover]

```

- `:first-child`

Hace referencia al primer hijo de otro elemento, pudiendo definir cual tiene que ser ese contenedor. Es útil para destacar títulos que pueden ir anidados en otros elementos de una página y que no se quiere que pierdan protagonismo, por ejemplo:

```
H1 :first-child { font-weight: bold }
```

- `:link` y `:visited`

Definen el estilo por defecto de un enlace sin visitar y otro visitado. Combinándolo con el selector de tipo para los elementos de enlace A:

```
A:link { color: blue }
```

```
A:visited { color: red }
```

- `:hover`

Permite accede al enlace por el que pasa el ratón, normalmente, `<h ref>`. Es mediante esta seudoclase como se logra el efecto de quitar el subrayado del enlace al pasar el ratón por encima, pero debemos recordar que para ello la regla de estilo deberá ser asignada al elemento enlace (A) y no al contenedor del mismo. Veamos un ejemplo:

```
A:hover { text-decoration: none }
```

Utilizar las seudoclases en todos los elementos HTML es posible, pero debemos tener en cuenta las posibles incompatibilidades de los navegadores, por lo que se recomienda utilizarlos únicamente para formatear enlaces.

Veamos un ejemplo fácil donde se cambia la negrita del texto al pasar el ratón por encima del enlace:

```
<html>
```

```
<head>
```



```
<title>seudoclases</title>

<style type="text/css">
A {text-decoration: none;}
A:hover {font-weight: bold}
</style>

</head>

<body>
<a href="">Texto de enlace</a>
</body>
</html>
```

1.5. Atributos de estilo para fuentes, color y fondo, texto y bloques (párrafos)

Las hojas de estilo son definitivamente el código ideal para definir los atributos visuales en documento HTML. Con CSS los diseñadores tienen todo lo que necesitan para poder diseñar y definir el aspecto visual del documento, y a su vez pueden dejar por separado la estructura y contenido del estilo.

Ya hemos visto que, gracias a las propiedades de estilo, justo a sus valores, podremos definir la sintaxis de una hoja de estilo sobre un elemento de la página y poder cambiar el aspecto de éste. De esta forma, podremos definir todos los pares de valores que queramos para definir un estilo sobre un elemento en concreto o sobre un elemento en bloque.

Por todo esto, podemos deducir que a prácticamente todos los elementos de los que se compone una página se le pueden aplicar atributos de estilo para cambiar el aspecto como deseemos. Por esto mismo, deberemos estar familiarizados con la sintaxis que se suele utilizar a la hora de asignar un valor a los atributos id y class.

Esto hará que nuestro trabajo sea mucho más profesional, y nos evitaremos dolores de cabeza cuando queramos buscar un elemento en concreto, cuanto más si trabajamos con un equipo y será otra persona la que tenga que lidiar con nuestro vocabulario. Es por ello que se recomienda que los valores que se asignen a los atributos id y class indiquen en concreto la función del

elemento, y que éstos no estén relacionados con el aspecto que pueda tener o la posición en el documento.

En la siguiente tabla podemos ver qué valores se recomiendan y cuales debemos evitar en la medida de lo posible.

Valores no recomendados	Valores recomendados
negrita	importante
arial10	titular
verdanaPequena	normal
menulzquierdo	menuSecundario
letraRoja	error

Podrá parecer que elegir un valor correcto para los atributos id o class sea complicado, pero nada más lejos de la realidad. Realmente es muy sencillo: si el aspecto de un elemento cambia, el valor de id o class deberá seguir siendo el adecuado, por lo que si evitamos utilizar valores con su posición (izquierdo, derecho, primero, segundo, superior, etc.), color (textoAzul, subrayadoNegro, etc.) o tipo de letra (Arial9, Helvetica15px, etc.).

Para aclarar mejor algunas cosas sobre la nomenclatura recomendada del atributo, también debemos tener en cuenta que tanto id como class deben cumplir ciertas reglas o restricciones. Son las siguientes:

Solo pueden comenzar por un guion medio (-), guiones bajos (_) y letras.

El resto de caracteres podrán ser números, guiones medios (-), guiones bajos (_) y letras.

Deberemos tener en cuenta que los navegadores distinguen entre mayúsculas y minúsculas.

Se podrá usar letras como ñ y acentos, pero no se recomienda su uso ya que no es seguro que funcionen de forma correcta en todas las versiones de los navegadores.

Una vez que ya conocemos las reglas sobre el uso y nomenclatura de los atributos de estilo, deberemos conocer qué atributos son los que se suelen utilizar. Veremos las propiedades definidas en las recomendaciones de hojas de estilo, las cuales incluyen las propiedades de CSS 2, excepto las relacionadas con las hojas de estilo auditiva (aural), los valores de las propiedades añadidos en la recomendación CSS 2.1 y las propiedades CSS 3 que admiten los navegadores Google Chrome, Firefox.

Las propiedades que veremos en profundidad serán las que pertenecen a los siguientes elementos que podemos encontrar en una página web:

Tipo de letra	Texto	Bordes	Margen exterior
Margen interior	Fondos	Listas	Tablas
Tamaño	Posicionamiento	Impresora	Interface de usuario
Pseudo-elementos	Pseudo-clases	Tipos	CSS 3

1.5.1. Descripción de los atributos de estilo

A continuación, veremos la descripción de cada uno de los atributos de estilo.

Tipo de letra

Tipo de letra (Fuente)	
Propiedad	Descripción
font	propiedad compuesta
font-family	tipo de letra (fuente)
font-size	tamaño
font-style	inclinación (cursiva)
font-variant	versalitas
font-weight	grosor del trazo (negrita)

- Font-family: propiedad que permite seleccionar la familia de la tipografía que va a ser utilizada o lo que es lo mismo, la fuente del texto. Se pueden especificar tantas familias como se deseen separándolas por espacios en blanco. Para que el usuario sea capaz de visualizar esa tipografía deberá tenerla instalada.

- Font-size: Cuando se utiliza HTML para crear una página Web, sorprende que para dar un tamaño a un texto se tienen tan sólo 5 posibles valores (de menor a mayor: 1, 2, 3, 4 y 5). La razón de esta simplicidad es el compromiso de HTML para ser un estándar compatible con la mayor cantidad de plataformas posibles. Pero gracias a CSS se puede mantener esta compatibilidad y además nos dejará que seamos más precisos con las características de los elementos que introduzcamos en el site. En este caso, el tamaño de la fuente puede especificarse de diversos modos:
 - Usando constantes predefinidas por CSS, como: xx-small, x-small, small, medium, large, x-large, xx-large.
 - Aunque no especifican el tamaño de la fuente, permiten saber que una es mayor que la otra sin lugar a confusión entre píxeles y puntos.
 - También se puede indicar el tamaño de manera relativa a la herencia (larger-smaller). Estos dos valores dependen de la herencia para adquirir su tamaño definitivo, siendo mayores menores que el valor heredado. En el caso de larger, siempre pintará una tipografía mayor a la heredada, mientras que smaller hará lo contrario, mostrará una tipografía más pequeña que la de sus elementos predecesores. Esta propiedad es muy cómoda en grandes textos para resaltar ciertas frases o listados.
 - Se puede utilizar el seudoelemento :after para especificar que todos los elementos que sigan a un título se vean con una tipografía más pequeña.
 - El método más utilizado es indicar el tamaño exacto de la fuente. Es la manera más precisa de especificar el tamaño de la tipografía, además de ser la manera de que el texto se vea igual en todos los navegadores. Se suele especificar en puntos y píxeles.
 - El tamaño también se puede indicar en tanto por ciento (respecto al valor original). Cuando se utilizan tantos porcientos como valor de una propiedad, el cálculo se realiza en base al valor original de la propiedad y si no tiene ninguna especificada, el valor por defecto.
- Font-style: Usamos esta propiedad para especificar el estilo de la fuente, con tres posibles valores: normal, italic y oblique.
- Font-weight: Permite especificar el ancho de la tipografía. Pero no sólo a mayor (obteniendo el efecto de negrita), sino también a menor, consiguiendo el efecto "letras finas" (si la letra lo permite). Permite dos valores:
 - Constantes predefinidas: bold, bolder, lighter y normal.

- O bien con los valores: 100, 200, 300, 400, 500, 600, 700, 800 ó 900. Estas centenas indican el grosor de la tipografía, siendo 400 normal y 700 el equivalente al valor bold.

Texto

Propiedad	Descripción
color	color del texto
direction	dirección del texto
letter-spacing	espacio entre caracteres
line-height	espaciado entre líneas
text-align	alineación del texto
text-decoration	decoración del texto
text-indent	sangría de la primera línea
text-transform	mayúsculas / minúsculas
unicode-bidi	dirección del texto
vertical-align	alineación vertical
white-space	espacios en blanco, saltos de línea y wrap
word-spacing	espacio entre palabras

Bordes

Bordes	
Propiedad	Descripción
border	cuatro bordes simultáneamente
border-top	borde superior
border-right	borde derecho
border-bottom	borde inferior
border-left	borde izquierdo
border-color	color de los bordes
border-width	grosor de los bordes
border-style	estilos de los bordes
border-top-color	color del borde superior
border-top-width	grosor del borde superior
border-top-style	estilo del borde superior

border-right-color	color del borde derecho
border-right-width	grosor del borde derecho
border-right-style	estilo del borde derecho
border-bottom-color	color del borde inferior
border-bottom-width	grosor del borde inferior
border-bottom-style	estilo del borde inferior
border-left-color	color del borde izquierdo
border-left-width	grosor del borde izquierdo
border-left-style	estilo del borde izquierdo

Las distintas propiedades que usamos para formatear los bordes de los elementos son:

- Border-width: Permite especificar el tamaño del borde de dos maneras:
 - Con los valores thin, medium y thick.
 - Especificando nosotros mismo el valor del ancho de varias formas, principalmente especificando los cuatro bordes con el orden siguiente: superior, derecho, inferior, izquierdo.
- Border-color: Con esta propiedad podremos cambiar el color del borde. Podremos especificar un color distinto para cada borde de la misma forma que se aplica el ancho con la propiedad border-width. Los valores pueden ser o bien el nombre del color o en su forma hexadecimal.
- Border-style: Esta propiedad define el estilo del borde, es decir, si será sólido, punteado, rayado, etc.

Margen exterior

Margen exterior	
Propiedad	Descripción
margin	cuatro márgenes exteriores simultáneamente
margin-top	margin exterior superior
margin-right	margin exterior derecho
margin-bottom	margin exterior inferior
margin-left	margin exterior izquierdo

Margin es una propiedad que se utiliza para especificar un margen. El margen es la distancia entre el borde de un elemento y el siguiente elemento, es decir, la distancia que hay de un elemento hacia afuera. Un único valor se interpretará para los cuatro lados, y cuatro valores indicarán cada uno de los márgenes de forma independiente.

Margen interior

Margen interior	
Propiedad	Descripción
padding	cuatro márgenes interiores simultáneamente
padding-top	margen interior superior
padding-right	margen interior derecho
padding-bottom	margen interior inferior
padding-left	margen interior izquierdo

Padding indica la distancia que existe entre el borde y el contenido del elemento, es decir, la distancia desde el borde para adentro. También es conocido como "relleno". Esta propiedad se suele utilizar para que el texto no se pegue a los bordes de una tabla o capa. Sus valores pueden especificar un valor global para todos los lados o para cada uno por separado.

Fondos

Fondos (Modelo de Caja CSS)	
Propiedad	Descripción
background	propiedad compuesta
background-attachment	atadura de la imagen
background-color	color de fondo
background-image	imagen de fondo
background-position	posición de la imagen de fondo
background-repeat	repetición de la imagen de fondo

Las propiedades del Modelo de Caja CSS se pueden aplicar a todos los elementos de una página y funcionará en todos los navegadores que soporten CSS 1. Son la base del modelo de caja CSS, que nos explica cómo se construyen los elementos web en el navegador.

- Background: la propiedad background en sí, define cómo es el fondo de un elemento. Aunque se suele asociar a marcas contenedoras de otros elementos, como body o table, se puede utilizar para todos los elementos HTML, como el texto o los párrafos. Sin embargo, lo mejor de esta propiedad es la posibilidad que nos brinda de definir los fondos como imagen y especificar la posición del fondo.
- Background-color: Su valor puede ser un color o “transparent”, que es la que tiene por defecto. Se puede aplicar a cualquier elemento HTML que tenga fondo.
- Background-image: Permite especificar una imagen de fondo para un elemento de la página. El valor de esta propiedad será la ruta a la imagen que deseemos que sea el fondo del elemento.
- Background-repeat: Cuando se aplica una imagen de fondo a un elemento y ésta es más pequeña que el elemento, por defecto ese fondo se repetirá por todo el espacio del elemento. Podemos especificar si el contenido se repite o no, y si deseamos que se repita, concretar el eje (horizontal o vertical). Sus valores son repeat, no-repeat, repeat-x y repeat-y.
- Background-attachment: Con esta propiedad podremos definir si la imagen de fondo de una imagen permanece fija o se desplaza con la barra de desplazamiento. Tiene dos valores: scroll (por defecto) y fixed (fijado).
- Background-position: Con esta propiedad podremos controlar la posición de la imagen del fondo respecto al margen izquierdo y superior de su contenedor, lo que nos permitirá ajustar perfectamente una imagen de fondo a un elemento sin tener que utilizar tablas. Utiliza los valores siguientes: top, center, bottom, left, center, right. Se puede especificar la distancia con valores exactos o en tantos por cientos.

Listas

Listas	
Propiedad	Descripción
list-style	propiedad compuesta
list-style-image	imagen del marcador
list-style-position	posición del marcador
list-style-type	tipo de marcador

Tablas

Tablas	
Propiedad	Descripción
border-collapse	modo de bordes
border-spacing	separación entre celdas
caption-side	posición de la leyenda
empty-cells	borde de casillas vacías
table-layout	algoritmo de ancho de tabla

Tamaño

Tamaño	
Propiedad	Descripción
width	anchura
min-width	anchura mínima
max-width	anchura máxima
height	altura
min-height	altura máxima
max-height	anchura máxima
overflow	si el contenido desborda al elemento

Posicionamiento

Posicionamiento	
Propiedad	Descripción
float	modo de posicionamiento flotante
clear	lado en el que no puede haber elementos flotantes
position	modo de posicionamiento
top	posición del borde superior del elemento
right	posición del borde derecho del elemento
bottom	posición del borde inferior del elemento
left	posición del borde izquierdo del elemento
clip	recorta el elemento
display	tipo de caja

visibility	visibilidad
z-index	apilamiento

Impresora

Impresora	
Propiedad	Descripción
page-break-after	salto de página después de
page-break-before	salto de página antes de
page-break-inside	salto de página dentro de
orphans	número de líneas al final de página
widows	número de líneas al principio de la página

Interface de usuario

Interface de usuario	
Propiedad	Descripción
cursor	tipo de cursor
outline	cuatro bordes simultáneamente
outline-color	color de los bordes
outline-width	grosor de los bordes
outline-style	estilo de los bordes

Seudoelementos

Seudoelementos	
Propiedad	Descripción
:after	inserta contenido después del elemento
:before	inserta contenido antes del elemento
:first-letter	primera letra
:first-line	primera línea de texto
content	contenido generado
counter-increment	puesto a cero de contador
counter-reset	puesta a cero de contador
quotes	comillas

Seudoclases

Seudoclases	
Propiedad	Descripción
:active	cuento se hace clic sobre el elemento
.first-child	primer elemento hijo
:focus	cuento el elemento tiene el foco
:hover	cuento el ratón pasa sobre el elemento
:lang	idioma
:link	enlaces no visitados
:visited	enlaces ya visitados

CSS 3

CSS 3	
Propiedad	Descripción
border-radius	bordes redondeados
border-top-right-radius	borde superior derecho redondeado
border-bottom-right-radius	borde inferior derecho redondeado
border-bottom-left-radius	borde inferior izquierdo redondeado
border-top-left-radius	borde superior izquierdo redondeado

1.5.2. Utilización de los atributos de estilo

A continuación, veremos la utilización de cada uno de los atributos de estilo. Los valores de propiedades añadidos o eliminados en la recomendación CSS 2.1 están indicados con los símbolos (+) o (-).

Tipo de letra (Fuente)	
Propiedad	Valores
font	[font-style font-variant font-weight] font-size [/ line-height]? font-family] caption icon menu message-box small-caption status-bar
font-family	[nombre-fuente familia-genérica] ,[nombre-fuente familia-genérica]
font-size	tamaño-absoluto tamaño-relativo distancia porcentaje

font-style	normal italic oblique
font-variant	normal small-caps
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900

Texto	
Propiedad	Valores
color	color
direction	ltr rtl
letter-spacing	normal distancia
line-height	normal número distancia porcentaje
text-align	center justify left right
text-decoration	none blink line-through overline underline
text-indent	distancia porcentaje
text-transform	none capitalize lowercase uppercase
unicode-bidi	normal embed bidi-override
vertical-align	baseline bottom middle sub super text-bottom text-top top distancia porcentaje
white-space	normal nowrap pre pre-line(+) pre-wrap(+)
word-spacing	normal distancia

Veamos algunos ejemplos de los atributos de estilo aplicados a la fuente de una página web.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>
```



```
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  <a href="index.html">Inicio</a>  
  <a href="fotos.html">Fotos</a>  
  <a href="curriculum.html">Curriculum</a>  
  <a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  <p><span id="capitular">L</span><em ipsum dolor</p>  
  </div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```

```
.contenido {  
  font-family: arial;  
  position: relative;  
  padding-top: 20px;  
  padding-right: 15px;  
  padding-bottom: 15px;  
  padding-left: 15px;  
  border-style: solid;  
  border-color: #0D0802;  
  border-width: 2px;  
  text-align: left;  
  min-height: 200px;
```


}

En este caso, el texto del contenido tendría un tipo de fuente de la familia Arial.

Ejemplo 2:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {
```



```
font-size: 10px;  
position: relative;  
padding-top: 20px;  
padding-right: 15px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el texto del contenido tendría un tamaño de fuente de 10 píxeles.

Ejemplo 3:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    font-style: italic;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;

    min-height: 200px;

}
```

En este caso, el texto del contenido tendría un estilo de fuente en cursiva.

Ejemplo 4:

```
<!DOCTYPE html>
```



```
<HTML>

<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    font-variant: small-caps;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el texto del contenido tendría un estilo de fuente en mayúsculas.

Ejemplo 5:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    font-weight: bold;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el texto del contenido tendría un estilo de fuente en negrita.

Ejemplo 6:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
</head>
```



```
<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    font-weight: normal;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;
```



```
min-height: 200px;  
}
```

En este caso, el texto del contenido tendría un estilo de fuente normal.

Ejemplo 7:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```

```
.contenido {  
    font-weight: bold;  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

En este caso, el texto del contenido tendría un estilo de fuente en negrita.

Ejemplo 8:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>
```



```
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    font-weight: normal;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el texto del contenido tendría un estilo de fuente normal.

Texto

Veamos ahora algunos ejemplos de los atributos de texto aplicados al texto de una página web.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {
```



```
color: navy;  
position: relative;  
padding-top: 20px;  
padding-right: 15px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el texto del contenido tendría un estilo de color azul.

Ejemplo 2:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
      styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    line-height: 2%;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;

    min-height: 200px;

}
```

En este caso, el texto tiene una altura de un 2%.

Ejemplo 3:

```
<!DOCTYPE html>
```



```
<HTML>

<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
</head>

<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    text-align: left;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
min-height: 200px;  
}
```

En este caso, el texto estaría alineado a la izquierda.

Ejemplo 4:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    text-decoration: none;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el texto no tendría decoración alguna.

Ejemplo 5:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
```



```
</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    text-indent: 2%;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;
```



```
min-height: 200px;  
}
```

En este caso, el texto tendría una tabulación de un 2%.

Ejemplo 6:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```

```
.contenido {  
    text-transform: capitalize;  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

Ejemplo 7:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
    styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span><em ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    unicode-bidi: embed;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    min-height: 200px;
}
```

En este caso, el texto tendría un nivel adicional, el cual quedaría embebido por otro texto. Se suele usar cuando queremos establecer varios idiomas en una misma página web.

Ejemplo 8:

```
<!DOCTYPE html>
```



```
<HTML>

<head>
    <title>Mi página web</title>
    <link rel="stylesheet" type="text/css" href="style/
    styles.css"/>
</head>

<body>
    <div class="todo">
        <div class="titulo">Mi página web</div>
        <div class="menu">
            <a href="index.html">Inicio</a>
            <a href="fotos.html">Fotos</a>
            <a href="curriculum.html">Curriculum</a>
            <a href="contacto.html">Contacto</a></div>
        <div class="contenido">
            <p><span id="capitular">L</span>orem ipsum dolor</p>
        </div>
        <div class="pie"><p><a href="mapaweb.html" target="_
        blank">Mapa Web</a></p></div>
    </div>
</body>
</HTML>
```

```
.contenido {
    vertical-align: 25%;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el texto quedaría alineado verticalmente un 25% con respecto a la esquina superior izquierda del espacio.

Ejemplo 9:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>  
<a href="contacto.html">Contacto</a></div>  
<div class="contenido">  
<p><span id="capitular">L</span>orem ipsum dolor</p>
```



```
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    white-space: nowrap;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    min-height: 200px;

}
```

En este caso, el texto quedaría en una sola línea, sin espacios en el salto de línea.

Ejemplo 10:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>
```



```
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    word-spacing: 25px;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;
```



```

border-color: #0D0802;
border-width: 2px;
text-align: left;
min-height: 200px;
}

```

En este caso, el texto quedaría espaciado letra por letra con un espacio de 25 píxeles entre cada letra.

Bordes

Bordes	
Propiedad	Valores
border	border-color border-width border-style
border-top	border-color border-width border-style
border-right	border-color border-width border-style
border-bottom	border-color border-width border-style
border-left	border-color border-width border-style
border-color	[color transparent] {1, 4}
border-width	[medium thick thin distancia] {1, 4}
border-style	[none hidden dashed dotted double groove inset outset ridge solid] {1, 4}
border-top-color	[color transparent] {1, 4}
border-top-width	[medium thick thin distancia] {1, 4}
border-top-style	[none hidden dashed dotted double groove inset outset ridge solid] {1, 4}
border-right-color	[color transparent] {1, 4}
border-right-width	[medium thick thin distancia] {1, 4}
border-right-style	[none hidden dashed dotted double groove inset outset ridge solid] {1, 4}
border-bottom-color	[color transparent] {1, 4}
border-bottom-width	[medium thick thin distancia] {1, 4}
border-bottom-style	[none hidden dashed dotted double groove inset outset ridge solid] {1, 4}

Bordes	
Propiedad	Valores
border-left-color	[color transparent] {1, 4}
border-left-width	[medium thick thin distancia] {1, 4}
border-left-style	[none hidden dashed dotted double groove inset outset ridge solid] {1, 4}

Veamos algunos ejemplos de los atributos de estilo aplicados a los bordes.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><span id="contenid
o">orem ipsum dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
```



```
</body>  
</HTML>  
-----  
.contenido {  
    border-top-color: black;  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

En este caso, el borde sería de color negro.

Ejemplo 2:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">
```



```
<div class="titulo">Mi página web</div>

<div class="menu">
  <a href="index.html">Inicio</a>
  <a href="fotos.html">Fotos</a>
  <a href="curriculum.html">Curriculum</a>
  <a href="contacto.html">Contacto</a></div>

<div class="contenido">
  <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
</body>
</HTML>
```

```
.contenido {
  border-right-width: 10px;
  position: relative;
  padding-top: 20px;
  padding-right: 15px;
  padding-bottom: 15px;
  padding-left: 15px;
  border-style: solid;
  border-color: #0D0802;
  border-width: 2px;
  text-align: left;
  min-height: 200px;
}
```


En este caso, el borde derecho tendría un grosor de 10 píxeles.

Ejemplo 3:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

border-bottom-color: grey;
```



```
        position: relative;  
  
        padding-top: 20px;  
  
        padding-right: 15px;  
  
        padding-bottom: 15px;  
  
        padding-left: 15px;  
  
        border-style: solid;  
  
        border-color: #0D0802;  
  
        border-width: 2px;  
  
        text-align: left;  
  
        min-height: 200px;  
  
    }
```

En este caso, el borde inferior sería de color gris.

Ejemplo 4:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>

</body>
</HTML>

-----
.contenido {
    border-left-width: 10px;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el borde izquierdo tendría un grosor de 10 píxeles.

Ejemplo 5:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><span id="contenido">orem ipsum dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    border-color: #0D0802;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;
```



```
padding-left: 15px;  
border-style: solid;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el color de borde en general sería marrón oscuro.

Ejemplo 6:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>
```



```
</div>

</body>

</HTML>

-----
.contenido {
    border-width: 2px;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el borde en general tendría un grosor de 2 píxeles.

Ejemplo 7:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">
```



```
<div class="titulo">Mi página web</div>

<div class="menu">
    <a href="index.html">Inicio</a>
    <a href="fotos.html">Fotos</a>
    <a href="curriculum.html">Curriculum</a>
    <a href="contacto.html">Contacto</a></div>

<div class="contenido">
    <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
</body>
</HTML>

-----
.contenido {
    border-style: solid;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el estilo de borde sería sólido.

Ejemplo 8:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

border-top-color: blue;

position: relative;
```



```
padding-top: 20px;  
padding-right: 15px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el color de borde superior sería azul.

Ejemplo 9:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">
```



```
<p><span id="capitular">L</span><em ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    border-top-width: 4px;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el ancho superior del borde sería de 4 píxeles.

Ejemplo 10:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
```



```
</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    border-top-style: dotted;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    text-align: left;
```



```
min-height: 200px;  
}
```

En este caso, el estilo de borde superior sería punteado.

Ejemplo 11:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```

```
.contenido {  
    border-right-color: pink;  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

En este caso, el color del borde derecho sería rosa.

Ejemplo 12:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span><em ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    border-right-width: 5px;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el grosor de borde derecho sería de 5 píxeles.

Ejemplo 13:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    border-right-style: double;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;
```



```
padding-left: 15px;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el estilo del borde derecho sería doble.

Ejemplo 14:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span><span style="font-size: 2em; color: red; font-weight: bold; margin-right: 10px;">ore  
m ipsum dolor</span></p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>
```



```
</div>

</body>

</HTML>

-----
.contenido {

    border-bottom-color: #ff0000;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    text-align: left;

    min-height: 200px;

}
```

En este caso, el color del borde inferior sería rojo.

Ejemplo 15:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">
```



```
<div class="titulo">Mi página web</div>

<div class="menu">
  <a href="index.html">Inicio</a>
  <a href="fotos.html">Fotos</a>
  <a href="curriculum.html">Curriculum</a>
  <a href="contacto.html">Contacto</a></div>

<div class="contenido">
  <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
</body>
</HTML>
```

```
.contenido {
  border-bottom-width: 3px;
  position: relative;
  padding-top: 20px;
  padding-right: 15px;
  padding-bottom: 15px;
  padding-left: 15px;
  border-color: #0D0802;
  border-width: 2px;
  text-align: left;
  min-height: 200px;
}
```

En este caso, el grosor del borde inferior sería de 3 píxeles.

Ejemplo 16:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

border-bottom-style: dashed;

position: relative;
```



```
padding-top: 20px;  
padding-right: 15px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el estilo inferior del borde sería una línea discontinua.

Ejemplo 17:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">
```



```
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    border-left-color: #00ff00;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el color del borde izquierdo sería verde.

Ejemplo 18:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
```



```
</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><em ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    border-left-width: 7px;

    position: relative;

    padding-top: 20px;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    text-align: left;
```



```
min-height: 200px;  
}
```

En este caso, el grosor del borde izquierdo tendrá 7 píxeles.

Ejemplo 19:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>  
  
-----  
.contenido {
```



```

border-left-style: solid;
position: relative;
padding-top: 20px;
padding-right: 15px;
padding-bottom: 15px;
padding-left: 15px;
border-style: solid;
border-color: #0D0802;
text-align: left;
min-height: 200px;
}

```

En este caso, el estilo del borde izquierdo será sólido.

Margen exterior

Margen exterior	
Propiedad	Valores
margin	[auto distancia porcentaje] {1, 4}
margin-top	auto distancia porcentaje
margin-right	auto distancia porcentaje
margin-bottom	auto distancia porcentaje
margin-left	auto distancia porcentaje

Veamos algunos ejemplos de los atributos de estilo aplicados al margen exterior.

Ejemplo 1:

```

<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

```



```
</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

margin-top: 25px;

position: relative;

padding-top: 20px;

padding-right: 15px;

padding-bottom: 15px;

padding-left: 15px;

border-style: solid;

border-color: #0D0802;

border-width: 2px;
```



```
    text-align: left;  
  
    min-height: 200px;  
  
}
```

En este caso, el margen superior tendría un espacio de 25 píxeles.

Ejemplo 2:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span><span id="restante">orem ipsum dolor</span></p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
</HTML>
```

```
.contenido {  
    margin-right: 10px;  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

En este caso, el margen derecho tendría espacio de 10 píxeles.

Ejemplo 3:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
    styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">
```



```
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    margin-bottom: 25px;
    position: relative;
    padding-top: 20px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el margen inferior tendría un espacio de 25 píxeles.

Ejemplo 4:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

margin-left: 10px;

position: relative;
```



```

padding-top: 20px;
padding-right: 15px;
padding-bottom: 15px;
padding-left: 15px;
border-style: solid;
border-color: #0D0802;
border-width: 2px;
text-align: left;
min-height: 200px;
}

```

En este caso, el margen izquierdo tendría espacio de 10 píxeles.

Margen interior

Margen interior	
Propiedad	Valores
padding	[distancia porcentaje] {1, 4}
padding-top	distancia porcentaje
padding-right	distancia porcentaje
padding-bottom	distancia porcentaje
padding-left	distancia porcentaje

Veamos algunos ejemplos de los atributos de estilo aplicados al margen interior.

Ejemplo 1:

```

<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

```



```
<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-top: 20px;

    position: relative;

    padding-right: 15px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;

    min-height: 200px;
```


}

En este caso, el margen interior superior tendría un espacio de 20 píxeles.

Ejemplo 2:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><span style="font-size: 2em; color: red; margin-left: 10px;">orem ipsum dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {
```



```
padding-right: 15px;  
position: relative;  
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el margen interior derecho tendría espacio de 15 píxeles.

Ejemplo 3:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>

</body>
</HTML>

-----
.contenido {
    padding-bottom: 20px;
    position: relative;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el margen interior inferior tendría un espacio de 20 píxeles.

Ejemplo 4:

```
<!DOCTYPE html>
<HTML>
<head>
```



```
<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
```



```

border-color: #0D0802;

border-width: 2px;

text-align: left;

min-height: 200px;

}

```

En este caso, el margen interior izquierdo tendría espacio de 15 píxeles.

Fondos

Fondos	
Propiedad	Valores
background	background-attachment background-color background-image background-position background-repeat
background-attachment	fixed scroll
background-color	transparent color
background-image	none url
background-position	[[left center right distancia porcentaje] [top center bottom distancia porcentaje]]? [[left center right] [top center bottom]]
background-repeat	no-repeat repeat repeat-x repeat-y

Veamos algunos ejemplos de los atributos de estilo aplicados al fondo.

Ejemplo 1:

```

<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

```



```
<div class="todo">  
    <div class="titulo">Mi página web</div>  
    <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>  
        <a href="curriculum.html">Curriculum</a>  
        <a href="contacto.html">Contacto</a></div>  
        <div class="contenido">  
            <p><span id="capitular">L</span>orem ipsum dolor</p>  
        </div>  
        <div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>  
    </div>  
</body>  
</HTML>  
-----  
.contenido {  
    background-attachment: fixed;  
    padding-top: 20px;  
    position: relative;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;
```


}

En este caso, el fondo queda fijo.

Ejemplo 2:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><span style="background-color: #ccc; padding: 0 5px; font-size: 1.5em; vertical-align: middle;"></span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {
```



```
background-color: white;  
padding-right: 15px;  
position: relative;  
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el fondo quedaría de color blanco.

Ejemplo 3:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span><em ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    background-image: url(fondo.gif');
    padding-top: 20px;
    position: relative;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el fondo tendría una imagen en formato .gif de fondo.

Ejemplo 4:

```
<!DOCTYPE html>
```



```
<HTML>

<head>
    <title>Mi página web</title>
    <link rel="stylesheet" type="text/css" href="style/
    styles.css"/>
</head>
<body>
    <div class="todo">
        <div class="titulo">Mi página web</div>
        <div class="menu">
            <a href="index.html">Inicio</a>
            <a href="fotos.html">Fotos</a>
            <a href="curriculum.html">Curriculum</a>
            <a href="contacto.html">Contacto</a></div>
        <div class="contenido">
            <p><span id="capitular">L</span>orem ipsum dolor</p>
        </div>
        <div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
    </div>
</body>
</HTML>
```

```
.contenido {
    background-position: center;
    background-image: url(fondo.gif');
    padding-right: 15px;
    position: relative;
```



```
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el fondo quedaría posicionado al centro.

Ejemplo 5:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">
```



```
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
-----
.contenido {
    background-repeat: repeat-x;
    background-image: url(fondo.gif');
    padding-right: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el fondo se repetirá horizontalmente.

Listas

Listas	
Propiedad	Valores
list-style	list-style-image list-style-position list-style-type

Listas	
Propiedad	Valores
list-style-image	none url
list-style-position	inside outside
list-style-type	none circle disc square decimal decimal-leading-zero lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman armenian georgian hebrew(-) cjk-ideographic(-) hiragana (-) katakana (-) hiragana-iroha(-) katakana-iroha(-)

Veamos algunos ejemplos de los atributos de estilo aplicados a las listas.

Ejemplo 1:

```
<HTML>

<head>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></
p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></
p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_
blank">Imagen 1</a></p></li>
```



```
<li><p><a href="images/image-2.jpg" target="_blank">Imagen 2</a></p></li>

<li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>

</p>

</body>

</HTML>
```



```
-----  
.mapaweb {  
  
    list-style-image: url(imagen.gif');  
  
    font-family: Helvetica, Gill Sans, Lucida;  
  
    border-style: none;  
  
    width:800px;  
  
    margin:0 auto;  
  
    position: relative;  
  
    padding-left: 15px;  
  
}  
  
.mapawebposicion {  
  
    padding-top: 40px;  
  
}  
  
.mapaweb a{  
  
    text-decoration: none;  
  
    color: black;
```

En este caso, se reemplazaría el marcador de objetos de lista con una imagen.

Ejemplo 2:

```
<HTML>  
  
<head>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
<title>Mapa Web</title>  
  
</head>  
  
<body>  
  
<p class="mapawebposicion">  
  
<ul class="mapaweb">
```



```
<li><p><a href="index.html" target="_blank">Inicio</a></p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_blank">Imagen 2</a></p></li>

<li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>
```



```
<ul class="mapaweb">  
    <li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>  
</ul>  
</p>  
</body>  
</HTML>  
-----  
.mapaweb {  
    list-style-position: inside;  
    font-family: Helvetica, Gill Sans, Lucida;  
    border-style: none;  
    width:800px;  
    margin:0 auto;  
    position: relative;  
    padding-left: 15px;  
}  
.mapawebposicion {  
    padding-top: 40px;  
}  
.mapaweb a{  
    text-decoration: none;  
    color: black;
```

En este caso, se especifica que el marcador debe aparecer dentro del flujo de contenido.

Ejemplo 3:

```
<HTML>  
<head>
```



```
<link rel="stylesheet" type="text/css" href="style/styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_blank">Imagen 2</a></p></li>

<li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">
```



```
<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>

</p>

</body>

</HTML>
```

```
.mapaweb {

    list-style-type: circle;
    font-family: Helvetica, Gill Sans, Lucida;
    border-style: none;
    width:800px;
    margin:0 auto;
    position: relative;
    padding-left: 15px;
}

.mapawebposicion {

    padding-top: 40px;
}

.mapaweb a{

    text-decoration: none;
```



```
color: black;
```

En este caso, se especifica que el tipo de marcador de objetos de lista será un círculo.

Tablas

Tablas	
Propiedad	Valores
border-collapse	collapse separate
border-spacing	distancia distancia?
caption-side	top bottom left(-) right(-)
empty-cells	hide show
table-layout	auto fixed

Veamos algunos ejemplos de los atributos de estilo aplicados a las tablas.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>


```



```
<div class="contenido">

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank" ></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2" target="_blank" ></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3" target="_blank" ></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4" target="_blank" ></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg" title="Imagen 5" target="_blank" ></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6" target="_blank" ></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7" target="_blank" ></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8" target="nueva"></a><br></td>

</tr>

</table>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
```



```
</body>

</HTML>

-----
table {

    border-collapse: collapse;
    width: 80%;

    margin-left: auto;
    margin-right: auto;
    height: 200px;
    border-style: none;
}

td {

    width: 30%;

    margin-left: auto;
    margin-right: auto;
}

.fotos {

    border-width: 1px;
    border-color: #3D4859;
    border-style: solid;
}
```

En este caso, los bordes se colapsan hacia el borde si fuera posible.

Ejemplo 2:

```
<HTML>

<head>
```



```
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></
p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></
p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_
blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_
blank">Imagen 2</a><p></li>

<li><p><a href="images/image-3.jpg" target="_
blank">Imagen 3</a><p></li>

<li><p><a href="images/image-4.jpg" target="_
blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_
blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_
blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_
blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_
blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">
```



```
<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>

</p>

</body>

</HTML>
```

```
.mapaweb {

    border-spacing: 10px 50px;
    font-family: Helvetica, Gill Sans, Lucida;
    border-style: none;
    width:800px;
    margin:0 auto;
    position: relative;
    padding-left: 15px;
}

.mapawebposicion {

    padding-top: 40px;
}

.mapaweb a{

    text-decoration: none;
```



```
color: black;
```

En este caso, se define la distancia entre los bordes de las celdas adyacentes en 10 y 50 píxeles.

Ejemplo 3:

```
<HTML>

<head>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></
p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></
p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_
blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_
blank">Imagen 2</a></p></li>

<li><p><a href="images/image-3.jpg" target="_
blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_
blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_
blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_
blank">Imagen 6</a></p></li>
```



```
<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>

</p>

</body>

</HTML>
```

```
.mapaweb {

    caption-side: bottom;

    font-family: Helvetica, Gill Sans, Lucida;

    border-style: none;

    width: 800px;

    margin: 0 auto;

    position: relative;

    padding-left: 15px;
```



```
}

.mapawebposicion {
    padding-top: 40px;
}

.mapaweb a {
    text-decoration: none;
    color: black;
```

En este caso, se especifica la colocación del título de la tabla en la parte inferior.

Ejemplo 4:

```
<HTML>

<head>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></
p></li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></
p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_
blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_
blank">Imagen 2</a></p></li>
```



```

<li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>

</p>

</body>

</HTML>
-----
```

```
.mapaweb {
    empty-cells: hide;
```



```
    font-family: Helvetica, Gill Sans, Lucida;  
  
    border-style: none;  
  
    width:800px;  
  
    margin:0 auto;  
  
    position: relative;  
  
    padding-left: 15px;  
  
}  
  
.mapawebposicion {  
  
    padding-top: 40px;  
  
}  
  
.mapaweb a{  
  
    text-decoration: none;  
  
    color: black;
```

En este caso, no se mostrarán backgrounds o bordes en celdas vacías.

Ejemplo 5:

```
<HTML>  
  
<head>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
<title>Mapa Web</title>  
  
</head>  
  
<body>  
  
<p class="mapawebposicion">  
  
<ul class="mapaweb">  
  
<li><p><a href="index.html" target="_blank">Inicio</a></  
p></li>  
  
</ul>  
  
<ul class="mapaweb">
```



```
<li><p><a href="fotos.html" target="_blank">Fotos</a></p></li>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_blank">Imagen 1</a></p></li>

<li><p><a href="images/image-2.jpg" target="_blank">Imagen 2</a></p></li>

<li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>

<li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>

<li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>

<li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>

<li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>

<li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>

<ul class="mapaweb">

<li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>

</ul>

</ul>

<ul class="mapaweb">

<li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>

</ul>
```



```
</p>  
</body>  
</HTML>
```

```
.mapaweb {  
    table-layout: fixed;  
    font-family: Helvetica, Gill Sans, Lucida;  
    border-style: none;  
    width: 800px;  
    margin: 0 auto;  
    position: relative;  
    padding-left: 15px;  
}  
.mapawebposicion {  
    padding-top: 40px;  
}  
.mapaweb a {  
    text-decoration: none;  
    color: black;
```

En este caso, la disposición de la celda no dependerá del contenido de las celdas, si no de su propia dimensión.

Tamaño

Tamaño	
Propiedad	Valores
width	auto distancia porcentaje
min-width	distancia porcentaje
max-width	distancia porcentaje
height	auto distancia porcentaje

Tamaño	
Propiedad	Valores
min-height	distancia porcentaje
max-height	distancia porcentaje
overflow	auto hidden scroll visible

Veamos algunos ejemplos de los atributos de estilo aplicados al tamaño.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>
```



```
</body>  
</HTML>  
-----  
.contenido {  
  
    width: 600px;  
  
    padding-left: 15px;  
  
    position: relative;  
  
    padding-top: 20px;  
  
    padding-bottom: 15px;  
  
    padding-left: 15px;  
  
    border-style: solid;  
  
    border-color: #0D0802;  
  
    border-width: 2px;  
  
    text-align: left;  
  
    min-height: 200px;  
  
}
```

En este caso, el ancho sería de 600 píxeles.

Ejemplo 2:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">
```



```
<div class="titulo">Mi página web</div>

<div class="menu">
    <a href="index.html">Inicio</a>
    <a href="fotos.html">Fotos</a>
    <a href="curriculum.html">Curriculum</a>
    <a href="contacto.html">Contacto</a></div>

<div class="contenido">
    <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>

</body>
</HTML>

-----
.contenido {
    min-width: 600px
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```


En este caso, el ancho mínimo establecido sería de 600 píxeles.

Ejemplo 3:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    max-width: 800px;
```



```
padding-left: 15px;  
position: relative;  
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el ancho máximo sería de 800 píxeles.

Ejemplo 4:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>

</body>
</HTML>

-----
.contenido {
    height: 400px;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, la altura sería de 400 píxeles.

Ejemplo 5:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    min-height: 400px;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;
```



```
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el alto mínimo sería de 400 píxeles.

Ejemplo 6:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
      styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>  
        <a href="curriculum.html">Curriculum</a>  
        <a href="contacto.html">Contacto</a></div>  
      <div class="contenido">  
        <p><span id="capitular">L</span><span id="contar">orem ipsum dolor</span></p>  
      </div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
-----
.contenido {
    max-height: 500px;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, la alturamáxima establecida sería de 500 píxeles.

Ejemplo 7:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
</head>
```



```
<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    overflow: scroll;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;
```



```
min-height: 200px;  
}
```

En este caso, si existe desbordamiento, éste se recorta, pero una barra de desplazamiento se añade para ver el resto del contenido.

Posicionamiento

Posicionamiento	
Propiedad	Valores
float	none left right
clear	none both left right
position	absolute fixed relative static
top	auto distancia porcentaje
right	auto distancia porcentaje
bottom	auto distancia porcentaje
left	auto distancia porcentaje
clip	auto forma
display	none block compact(-) inline inline-block(+) inline-table list-item marker(-) run-in table table-caption table-cell table-column table-column-group table-footer-group table-header-group table-row table-row-group
visibility	hidden collapse visible
z-index	auto número-entero

Veamos algunos ejemplos de los atributos de estilo aplicados al posicionamiento.

Ejemplo 1:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>
```



```
<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    float: right;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;
```



```
min-height: 200px;  
}
```

En este caso, se especifica que un elemento debe flotar a la derecha.

Ejemplo 2:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```

```
.contenido {  
    clear: left;  
    float: right;;  
    padding-left: 15px;  
    position: relative;  
    padding-top: 20px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```

En este caso, no se permiten elemento flotantes a la izquierda del elemento.

Ejemplo 3:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">
```



```
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
-----
.contenido {
    position: static;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el elemento se presenta con su valor por defecto estático.

Ejemplo 4:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
</head>
<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
-----
.contenido {
    top: 15px;
    float: right;
```



```
padding-left: 15px;  
position: relative;  
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el elemento se encuentra a 15 píxeles desde la altura de su ubicación.

Ejemplo 5:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>

</body>
</HTML>

-----
.contenido {
    right: 100px;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

En este caso, el elemento se presenta con a 100 píxeles de la derecha de su ubicación.

Ejemplo 6:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    bottom: 200px;

    float: right;;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, el elemento se encuentra a 200 píxeles desde la parte inferior de su ubicación.

Ejemplo 7:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>  
<a href="contacto.html">Contacto</a></div>  
<div class="contenido">  
<p><span id="capitular">L</span>orem ipsum dolor</p>
```



```
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    left: 50px;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;

    min-height: 200px;

}
```

En este caso, el elemento se presenta con a 15 píxeles de la izquierda de su ubicación.

Ejemplo 8:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>
```



```
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    clip: rect(0px, 60px, 200px, 0px);

    float: right;;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;
```



```
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

En este caso, la propiedad clip especifica un rectángulo para recortar un elemento, con cuatro coordenadas desde la esquina superior izquierda.

Ejemplo 9:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

p.capitular
{
    display:block;
}
```

En este caso, se especifica el tipo de caja utilizada, en este caso en bloque, para un elemento de párrafo.

Ejemplo 10:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    float: right;;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
```



```
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
p.capitular  
{  
visibility:hidden;  
}
```

En este caso, indica que un elemento no será visible, pero ocupará su espacio.

Ejemplo 11:

```
<!DOCTYPE html>  
<HTML>  
 <head>  
   <title>Mi página web</title>  
   <link rel="stylesheet" type="text/css" href="style/  
     styles.css"/>  
 </head>  
 <body>  
   <div class="todo">  
     <div class="titulo">Mi página web</div>  
     <div class="menu">  
       <a href="index.html">Inicio</a>  
       <a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<div class="contacto">

<div id="fotocv">

</div>

<span id="capitular">Gracias por tu visita.

Puedes ponerte en contacto conmigo a través del
formulario y contarme qué te parecen mis trabajos,
o preguntarme las dudas que tengas.

<form method="post" action="mailto:correo@mail.com"
enctype="text/plain">

<textarea rows="5" cols="30" name="comentarios"></texta-
rea>

<button type="submit">Enviar</button>&nbsp;&nbsp;
<button type="reset">Cancelar</button>

</form>

</div>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

```
img  
{  
position:absolute;  
left:0px;  
top:0px;  
z-index:-1;  
}
```

Impresora

Impresora	
Propiedad	Valores
page-break-after	auto always avoid left right
page-break-before	auto always avoid left right
page-break-inside	auto avoid
orphans	número-entero
widows	número-entero

Veamos algunos ejemplos de los atributos de estilo aplicados a la impresora.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;

    position: relative;
```



```
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
@media print {  
    pie {page-break-after: always;}  
}
```

En este caso, se especifica que se insertará un salto de página después del elemento.

Ejemplo 2:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
      styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">  
        <a href="index.html">Inicio</a>
```



```
<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<h1>Hola</h1>

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    float: right;;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;

    text-align: left;

    min-height: 200px;

}

@media print {

    h1 {page-break-before: always;}
```


}

En este caso, se insertará siempre un salto de página antes del elemento.

Ejemplo 3:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span><rem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {
```



```
padding-left: 15px;  
position: relative;  
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
@media print {  
    p {page-break-inside: avoid;}  
}
```

En este caso, se especifica evita un salto de página dentro del elemento siempre que sea posible.

Ejemplo 4:

```
<!DOCTYPE html>  
<HTML>  
    <head>  
        <title>Mi página web</title>  
        <link rel="stylesheet" type="text/css" href="style/  
        styles.css"/>  
    </head>  
    <body>  
        <div class="todo">  
            <div class="titulo">Mi página web</div>
```



```
<div class="menu">
  <a href="index.html">Inicio</a>
  <a href="fotos.html">Fotos</a>
  <a href="curriculum.html">Curriculum</a>
  <a href="contacto.html">Contacto</a></div>
<div class="contenido">
  <h1>Hola</h1>
  <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
  float: right;;
  padding-left: 15px;
  position: relative;
  padding-top: 20px;
  padding-bottom: 15px;
  padding-left: 15px;
  border-style: solid;
  border-color: #0D0802;
  border-width: 2px;
  text-align: left;
  min-height: 200px;
}
```



```
@media print {  
    orphans: 2;  
}
```

En este caso, se establece el número mínimo de líneas que se dejan en la parte inferior de una página cuando se produce un salto de página dentro de un elemento.

Ejemplo 5:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<h1>Hola</h1>  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>
```



```
</body>  
</HTML>  
-----  
.contenido {  
    float: right;;  
    padding-left: 15px;  
    position: relative;  
    padding-top: 20px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}  
  
@media print {  
    widows: 2;  
}
```

En este caso, se establece el número mínimo de líneas que se dejan en la parte superior de una página cuando se produce un salto de página dentro de un elemento.

Interface de usuario

Interface de usuario	
Propiedad	Valores
cursor	[uri,]* auto crosshair default help move pointer progress n-resize ne-resize e-resize se-resize s-resize sw-resize w-resize nw-resize text wait

Interface de usuario	
Propiedad	Valores
outline	outline-color outline-width outline-style
outline-color	color
outline-width	border-width
outline-style	border-style

Veamos algunos ejemplos de los atributos de estilo aplicados a la interface de usuario.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>L orem ipsum <span>dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>
```



```
</div>
</body>
</HTML>
-----
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

span.help {cursor:help}
```

En este caso, se especifica que el cursor indicará que existe ayuda disponible.

Ejemplo 2:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
</head>
```



```
<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<h1>Hola</h1>

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    float: right;;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;

    border-width: 2px;
```



```
    text-align: left;  
  
    min-height: 200px;  
  
}  
  
p  
  
{  
  
outline:#00FF00 dotted thick;  
  
}
```

En este caso, se indica que hay un esquema (línea que se dibuja alrededor de los elementos) con un contorno punteado de color verde.

Ejemplo 3:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p>L orem ipsum <span>dolor</span></p>  
  
</div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

```
p {

    outline-color:#00ff00;
}
```

En este caso, se indica que hay un esquema (línea que se dibuja alrededor de los elementos) con un contorno de color verde.

Ejemplo 4:

```
<!DOCTYPE html>

<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<h1>Hola</h1>

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    float: right;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;
```



```
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
p  
{  
outline-width:2px;  
}
```

En este caso, se indica que hay un esquema (línea que se dibuja alrededor de los elementos) con un contorno de un ancho de 2 píxeles.

Ejemplo 5:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<h1>Hola</h1>
<p><span id="capitular">L</span><em ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    float: right;;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

p {
    outline-style: dotted;
```


}

En este caso, se indica que hay un esquema (línea que se dibuja alrededor de los elementos) con un contorno punteado.

Seudoelementos

Seudoelementos	
Propiedad	Valores
:after	content: contenido
:before	content: contenido
:first-letter	
:first-line	
content	normal(+) none(+) [texto url contador attr(atributo) open-quote close-quote no-open-quote no-close-quote]+
counter-increment	none [identificador número-entero?]+
counter-reset	none [identificador número-entero?]+
quotes	none [texto-apertura texto-cierre]+

Veamos algunos ejemplos de los atributos de estilo aplicados a los seudoelementos.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">
```



```
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p>Lorem ipsum <span>dolor</span></p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

```
p::after
```

```
{
```



```
content:"- Remember this";  
}
```

En este caso, se especifica que se insertará contenido después de cada elemento <p>.

Ejemplo 2:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<h1>Hola</h1>  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>  
  
</HTML>
```



```
-----  
.contenido {  
    float: right;;  
    padding-left: 15px;  
    position: relative;  
    padding-top: 20px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}  
  
p::before  
{  
    content:"Read this: ";  
}
```

En este caso, se indica que se insertará contenido antes de cada contenido <p> del elemento.

Ejemplo 3:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>
```



```
<div class="todo">  
    <div class="titulo">Mi página web</div>  
    <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>  
        <a href="curriculum.html">Curriculum</a>  
        <a href="contacto.html">Contacto</a></div>  
        <div class="contenido">  
            <p>L orem ipsum <span>dolor</span></p>  
        </div>  
        <div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>  
    </div>  
</body>  
</HTML>  
-----  
.contenido {  
    padding-left: 15px;  
    position: relative;  
    padding-top: 20px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}
```



```
p::first-letter
{
    font-size:200%;
    color:#8A2BE2;
}
```

En este caso, se especifica que se dará un estilo a la primera letra del selector especificado, en este caso un tamaño de un 200% y un color lila.

Ejemplo 4:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<h1>Hola</h1>

<p><span id="capitular">L</span><span>orem ipsum dolor</span></p>

</div>
```



```
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
p::first-line
{
    background-color: yellow;
}
```

En este caso, se indica que se dará un estilo a la primera línea del selector especificado, en este caso un color de fondo amarillo.

Ejemplo 5:

```
<!DOCTYPE html>
<HTML>
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>L orem ipsum <span>dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;
```



```
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
a:after  
{  
content: " (" attr(href) ")";  
}
```

En este caso, se especifica que el contenido se establece como uno de los atributos del selector.

Ejemplo 6:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">
    <h1>Hola</h1>
    <p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>
</body>
</HTML>
```

```
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

h1:before
{
    counter-increment: section;
    content: "Section " counter(section) ". ";
}
```



```
}

h2:before

{
    counter-increment:subsection;
    content:counter(section) "."
    counter(subsection) " ";
}
```

En este caso, se indica una forma de enumerar las secciones y sub-secciones con "Sección 1", "1.1", "1.2", etc.

Ejemplo 7:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>L orem ipsum <span>dolor</span></p>
```



```
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

body
{
    counter-reset:section;
}

h1
{
    counter-reset:subsection;
```



```
}

h1:before

{
    counter-increment:section;
    content:"Section " counter(section) ". ";
}

h2:before

{
    counter-increment:subsection;
    content:counter(section) "."
    counter(subsection) " ";
}
```

En este caso, se especifica una manera de enumerar las secciones y subsecciones con "Sección 1", "1.1", "1.2", etc.

Ejemplo 8:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">
```



```
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<h1>Hola</h1>
<p><span id="capitular">L</span>orem ipsum dolor</p>
<q>Texto con cita</q>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```



```

q
{
  quotes: "«" "»";
}

```

En este caso, se indica el tipo de valor que se usará para las citas, en este caso los símbolos ““ “”.

Seudoclases

Seudoclases	
Propiedad	Valores
:active	
.first-child	
:focus	
:hover	
:lang	:lang(en) :lang(es) :lang(fr) etc.
:link	
:visited	

Veamos algunos ejemplos de los atributos de estilo aplicados a las seudoclases.

Ejemplo 1:

```

<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

```



```
<div class="menu">
    <a href="index.html">Inicio</a>
    <a href="fotos.html">Fotos</a>
    <a href="curriculum.html">Curriculum</a>
    <a href="contacto.html">Contacto</a></div>
<div class="contenido">
    <p>Lorem ipsum <span>dolor</span></p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

a:active
```



```
{  
background-color:yellow;  
}
```

En este caso, se especifica que el enlace en estado activo tendrá un fondo de color amarillo.

Ejemplo 2:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<h1>Hola</h1>  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>
```



```
</body>
</HTML>
-----
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
p:first-child
{
    background-color: yellow;
}
```

En este caso, se selecciona y da estilo a cada párrafo que sea el primer hijo de su padre.

Ejemplo 3:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
```



```
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>L orem ipsum <span>dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
```



```
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}
```

```
input:focus  
{  
background-color:yellow;  
}
```

En este caso, se selecciona el elemento que tiene el foco y se le da un fondo amarillo.

Ejemplo 4:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>  
<a href="contacto.html">Contacto</a></div>  
<div class="contenido">
```



```
<h1>Hola</h1>

<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

a:hover
{
    background-color: yellow;
}
```

En este caso, se da formato al selector de un enlace cuando el ratón pasa sobre éste, el cual cambiará el fondo al color amarillo.

Ejemplo 5:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>Lorem ipsum <span>dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;

    position: relative;
```



```
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
p:lang(it)  
{  
background:yellow;  
}
```

En este caso, se selecciona y da estilo a cada párrafo con un valor de atributo lang igual a "it" (italiano).

Ejemplo 6:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">
```



```
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<h1>Hola</h1>
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>

-----
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

:link
```



```
{  
color: green;  
}
```

En este caso, se da formato al selector de un enlace no visitado.

Ejemplo 7:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<h1>Hola</h1>  
  
<p><span id="capitular">L</span>orem ipsum dolor</p>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_  
blank">Mapa Web</a></p></div>  
  
</div>  
  
</body>
```



```
</HTML>  
-----  
.contenido {  
    float: right;  
    padding-left: 15px;  
    position: relative;  
    padding-top: 20px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 200px;  
}  
  
a:visited  
{  
    background-color: yellow;  
}
```

En este caso, selecciona y da estilo a enlaces visitados, en este caso, tendría un fondo amarillo.

CSS 3

CSS 3	
Propiedad	Valores
border-radius	[distancia porcentaje] {1, 4}
border-top-right-radius	distancia porcentaje
border-bottom-right-radius	distancia porcentaje
border-bottom-left-radius	distancia porcentaje

CSS 3	
Propiedad	Valores
border-top-left-radius	distancia porcentaje

Veamos algunos ejemplos de los atributos de estilo aplicados a los nuevos atributos de CSS3.

Ejemplo 1:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p>Lorem ipsum <span>dolor</span></p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>
```



```
</HTML>
-----
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

div {
    border: 2px solid;
    border-radius: 25px;
}
```

En este caso, se especifica que borde del div será de 2 píxeles de grosor, además de redondear las esquinas.

Ejemplo 2:

```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>
```



```
</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<h1>Hola</h1>

<p><span id="capitular">L</span>orem ipsum dolor</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

```
.contenido {

    float: right;

    padding-left: 15px;

    position: relative;

    padding-top: 20px;

    padding-bottom: 15px;

    padding-left: 15px;

    border-style: solid;

    border-color: #0D0802;
```



```
        border-width: 2px;  
  
        text-align: left;  
  
        min-height: 200px;  
  
    }  
  
div {  
  
    border: 2px solid;  
  
    border-top-right-radius: 2em;  
  
}
```

En este caso, se redondea la forma de la esquina superior derecha del div.

Ejemplo 3:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<p>L orem ipsum <span>dolor</span></p>
```



```
</div>

<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>

-----
.contenido {

    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}

div {

    border: 2px solid;
    border-bottom-right-radius: 2em;
}
```

En este caso, se redondela la esquina inferior derecha del div.

Ejemplo 4:

```
<!DOCTYPE html>
```



```
<HTML>

<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<h1>Hola</h1>
<p><span id="capitular">L</span>orem ipsum dolor</p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    float: right;
    padding-left: 15px;
    position: relative;
```



```
padding-top: 20px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 200px;  
}  
  
div {  
border: 2px solid;  
border-bottom-left-radius: 2em;  
}
```

En este caso, se redondea la forma de la esquina inferior izquierda del div.

Ejemplo 5:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>
```



```
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p>L orem ipsum <span>dolor</span></p>
</div>
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div>
</div>
</body>
</HTML>
```

```
.contenido {
    padding-left: 15px;
    position: relative;
    padding-top: 20px;
    padding-bottom: 15px;
    padding-left: 15px;
    border-style: solid;
    border-color: #0D0802;
    border-width: 2px;
    text-align: left;
    min-height: 200px;
}
```

```
div {
    border: 2px solid;
    border-top-left-radius: 2em;
```


}

En este caso, se redondela la esquina superior izquierda del div.



Una vez vistos los valores y descripciones de los atributos de estilo o propiedades, tendremos que tener en cuenta que en CSS se indican los valores de formas diferentes, como colores, medidas o la URL de una imagen, por lo que vamos a ver a continuación que tipo de unidades y valores podemos usar en CSS para un conocimiento más profundo del código.

- Números.

Algunas propiedades tienen como valor un número entero o un decimal. Es el caso de la propiedad z-index, la cual acepta como valor un número entero.

Por ejemplo:

```
z-index: 13;
```

```
z-index: -13;
```

```
z-index: 0;
```

- Cadenas de texto.

Otras propiedades tienen como valor un texto. Es el caso de la propiedad content, que usamos para introducir contenido a un elemento de la página.

Para insertar valores que sean cadenas de texto utilizaremos indiferentemente las comillas dobles o las comillas simples, pero nunca podemos incluir comillas dobles dentro de otras comillas dobles, como podemos ver en el siguiente ejemplo:

```
content: "un texto "cualquiera"";  
/* sintaxis incorrecta*/  
  
content: "un texto 'cualquiera'";  
/*sintaxis correcta*/  
  
content: "un texto \"cualquiera\"";  
/*sintaxis correcta*/  
  
content: 'un texto 'cualquiera'';  
/* sintaxis incorrecta*/  
  
content: 'un texto cualquiera';  
/* sintaxis correcta*/
```

Como puedes observar en el ejemplo, hemos usado el valor escapado \' para insertar dobles comillas dentro de dobles comillas. Podemos hacer lo mismo con comillas simples \'.

- Colores.

Los colores los podremos aplicar a textos, fondos, bordes y otros aspectos visuales de nuestro diseño. En CSS se pueden especificar los colores de dos formas diferentes:

- usando las palabras reservadas de HTML 4.0 o bien,
- especificando un color RGB.

Los colores que podemos usar con palabras reservadas son:

Aqua	Black	Blue	Fuchsia
Gray	Green	Lime	Maroon
Navy	Olive	Purple	Red
Silver		Teal	

Como puedes observar, son los nombres de los principales colores en inglés y se usan como valor, no como cadena, es decir, sin comillas, tal y como podemos ver en el siguiente ejemplo:

```
color: aqua;  
  
background-color: silver;  
  
color: "black" /* incorrect */
```

Estos colores no son suficientes para un diseño completo, ya que como diseñadores necesitaremos más control sobre nuestra página. Por eso, para definir un color usaremos la escala RGB, usaremos una almohadilla (#) seguida de tres pares de valores, que serán la cantidad del rojo, verde y azul del color. Por ejemplo, para obtener rojo usaremos el siguiente código:

En ocasiones puede que encontramos hojas de estilo en las que han simplificado el color definiéndolo por sólo 3 caracteres en vez de 6 de RGB. Esto se puede hacer ya que CSS duplica los 3 caracteres, por ejemplo:

```
color: #FCD; /* es lo mismo que #FFCCDD */
```

También se podrá definir un color RGB indicando los tantos por ciento de la cantidad de cada color:

```
color: rgb(100%, 0%, 0%);  
  
color: rgb(45%, 30%, 30%);
```

- Tamaños y otras unidades de medida.

Junto a los colores, lo que más definiremos en nuestra página web será los tamaños y las unidades de medida de nuestros elementos como del tamaño del texto, además de la posición de un elemento, anchos, altos, distancia de margen, etc.

Para los tamaños usaremos enteros y decimales acompañados de una unidad que dirá a qué hace referencia el número previo (si se trata de número es en centímetros, píxeles u otra unidad de medida). Este tamaño o media puede ser de dos tipos:

Absoluta

Cuando conocemos de manera exacta el medio en el que se reproducirá el contenido, por ejemplo una impresora. En este caso podemos especificar sus medidas de manera absoluta, por ejemplo, 15 cm.

Relativa

El tamaño del elemento dependerá de la información que hereda. Un valor de 80%, por ejemplo, en algunos casos será 14pt y en otros 16pt (dependiendo del valor heredado al que se aplica el tanto por ciento).

Las unidades de medida de las que disponemos en CSS son las siguientes:

- Centímetros: cm

La unidad de medida básica en el sistema métrico. Por ejemplo:

```
DIV {  
    width: 30cm  
}
```

- Pulgadas: in

Una pulgada son 2.54 centímetros en el sistema métrico. Es poco común encontrarla en páginas web, incluso en los países anglosajones, pero si se utiliza mucho en hojas de estilo con el media:print.

```
DIV {  
    width: 4in  
}
```

- Puntos: pt

Un punto es una 1/72 parte de una pulgada. Se suele utilizar para definir tamaños de textos en páginas.

```
TD {
```



```
    font-size:12pt;  
}  
.
```

- Picas: PC

Una pica equivale a 12 puntos. Esta unidad apenas se utiliza en Europa.

```
UL LI {  
    font-size:3pc;  
}
```

Existen otras medidas alternativas, sin embargo serán las que más vamos a utilizar en nuestra hoja de estilo:

- Píxel: PX

Possiblemente fue la más usada en el pasado. Un píxel es la unidad más pequeña de la que se compone una imagen gráfica mostrada por pantalla, pero no hay que confundirlo con los puntos de la pantalla. El píxel no es un punto ni un cuadrado, es un muestreo digital que se visualiza. El tamaño del píxel depende de la pantalla y la resolución. Es por esto que al definir un tamaño en píxeles el elemento siempre medirá lo mismo, aunque lo veremos más pequeño a resolución mayor (hay más píxeles y son más pequeños). El píxel es la unidad más pequeña en la que se puede dividir un elemento, por lo que no podremos usar decimales.

```
P {  
    font-size: 5.5px /* incorrecto */;  
}
```

Tampoco podrá ser un valor negativo si lo usamos para indicar un tamaño, pero sí cuando indique una posición, como podemos ver en el siguiente ejemplo:

```
DIV {  
    top: -50px /* correcto */  
    height: -15px /* incorrect */
```


- }
 - EM

Este tipo de unidad se utiliza cada vez más, y nos será muy útil para cierto tipo de diseños en CSS que se deben adaptar al tamaño de la pantalla de una forma dinámica. De las unidades de medida, es la más relativa de las relativas, ya que no sólo depende del medio en el que se muestra, sino que no será la misma en todos los elementos de nuestra página. Un em es igual al valor de la propiedad Font-size que se aplica a un elemento.

También podemos utilizar em para definir tamaños, no sólo del texto, sino de otras propiedades.

 - EX

Esta unidad es parecida al em, pero en vez de tomar como base el valor de Font-size, usa la altura de la letra x en minúsculas. Es otra forma de especificar el tamaño de una fuente. este valor no siempre es exacto, ya que se usa en tipografías que no tienen la letra x.

```
<style type="text/css">
    DIV {
        font-size: 20pt;
    }
    P {
        font-size: 4ex;
    }
</style>
```

 - Tantos por ciento: %

Otra forma de insertar valores es usando el tantos por ciento. Así podremos indicar que el valor sea relativo respecto al valor heredado del elemento padre. Por ejemplo, si un elemento hijo tiene un ancho del 50% está indicando que el ancho será la mitad que el del padre.

- URL

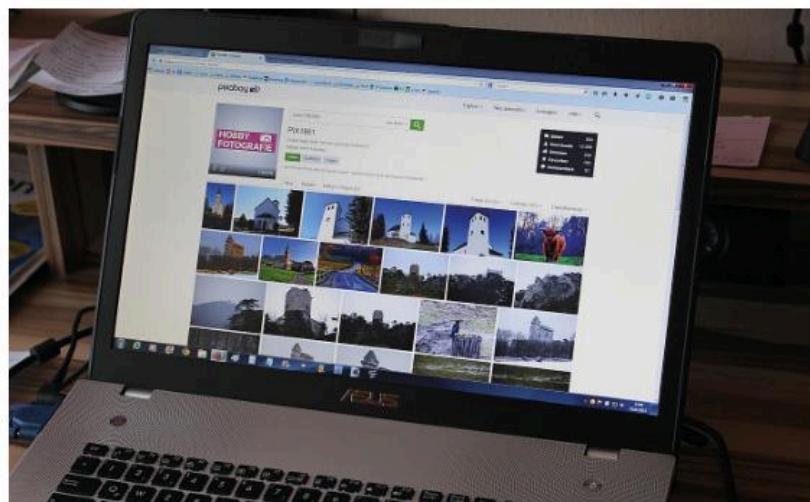
Hay algunas propiedades en CSS que reciben como valor un URL. La URL indica la localización de un recurso en internet. Puede estar en nuestro sitio (dirección relativa):

```
background-image: url('images/fondo1.jpg');
```

O en una dirección absoluta:

```
background-image: url (http://www.mipagina.com/images/fon-  
do1.jpg);
```

En cuanto a su sintaxis, las comillas simples o dobles son opcionales.



1.6. Creación de ficheros de estilo

Ya tenemos suficiente información para comenzar a realizar nuestros propios ficheros en los que comenzar a añadir estilos a nuestra página web a través del CSS.

Crear un fichero de estilo es muy fácil, pero previamente tendremos que tener claro qué estilos vamos a aplicar a nuestra página web. Para ello lo que se recomienda es dejar de un lado el CSS, el código HTML y el ordenador y coger lápiz y papel.

Comenzar teniendo las ideas claras de lo que queremos y cómo lo queremos es tan importante como el resto. Tener claros conceptos como los colores básicos de la página, los tipos de letra que vamos a usar, su tamaño y su

color, tener preparadas las imágenes a una buena resolución o pensar en las dimensiones de nuestra página y el tamaño de sus elementos es esencial. Una vez claros estos conceptos, ya podemos comenzar a aplicarlos a nuestra hoja de estilos.

Ahora vamos a conocer paso a paso cómo se crea un fichero de estilo. Un fichero de estilo no es más que la propia hoja de estilos, la cual será la que dé el estilo al documento HTML. Describiremos siempre como crear una hoja de estilos externa, ya que es lo más recomendado.

Estos son los elementos que veremos a continuación:

- definiremos qué es un fichero de estilos (o hoja de estilos),
- cómo crearlo y
- cómo adaptar un fichero de estilos a distintas páginas web.



1.6.1. Definición de los ficheros de estilo

Las hojas de estilo aparecieron sobre el año 1970 atendiendo a la necesidad de cubrir un mecanismo para poder aplicar de una forma consistente estilos a los documentos electrónicos. Cuando Internet comenzó a utilizarse masivamente fue cuando las hojas de estilo se impusieron y crecieron exponencialmente, pero a falta de un estándar, era difícil ver una misma apariencia en diferentes navegadores.

El organismo W3C fue el que inició una propuesta seria para la creación de estándares web, entre ellos, la creación de un lenguaje de hojas de estilo específico y para el HTML. Tras varias propuestas, en 1996 se creó el CSS tal y como lo conocemos ahora y se publicó la primera recomendación como CSS1.

En 1998 se publicó una segunda recomendación como CSS2, aunque actualmente la versión que utilizan los navegadores es una revisión posterior llamada CSS 2.1.

Actualmente, las últimas recomendaciones de W3C es el CSS3, aunque hasta ahora sólo han publicado algunos borradores.

Como diseñadores, deberemos tener en cuenta las reglas del W3C para poder llegar a todos los navegadores y que la visualización de nuestra página web sea óptima. Es por esto mismo por lo que no está de más conocer cierta información sobre el soporte que tienen los navegadores de CSS.

Los navegadores están divididos en numerosos componentes, y entre ellos la parte que se encarga de interpretar el código HTML y el CSS se llama motor. Para los diseñadores, esta es la parte más importante, más incluso que la versión del navegador.

A continuación, podemos ver una tabla con los navegadores más usados entre los usuarios que muestra el motor que usa, además del soporte que tiene para las reglas CSS.

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo desde la versión 85 del motor.	Completo.	Todos los selectores, seudoclases y muchas propiedades.
Firefox	Gecko	Completo desde la versión 1.0 del navegador.	Completo.	Todos los selectores, seudoclases y muchas propiedades.
Safari	WebKit	Completo desde la versión 85 del motor.	Completo.	Todos los selectores, seudoclases y muchas propiedades.
Opera	Presto	Completo desde la versión 1.0 del navegador.	Completo.	Todos los selectores, seudoclases y muchas propiedades.

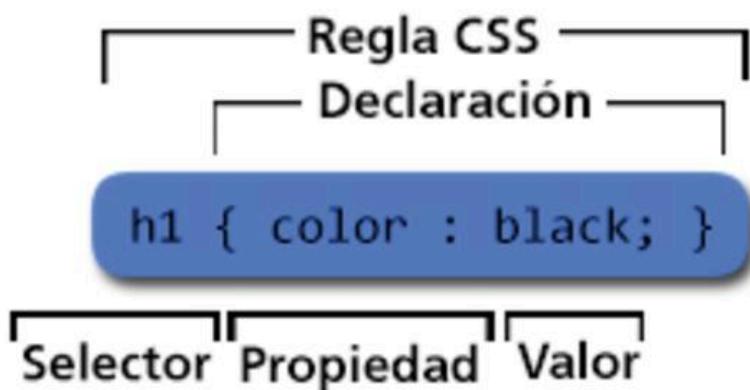
(Fuente: [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(Cascading_Style_Sheets\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(Cascading_Style_Sheets)))

Como puedes ver, es muy importante atender a las especificaciones del W3C. Este organismo trabaja actualmente en la elaboración de los nuevos estándares, para sacar a la luz la versión de CSS 3. En las especificaciones de la versión CSS 3 hay numerosos cambios en comparación con las versiones anteriores, es mucho más avanzada y compleja. Para estar al día y conocer todas las actualizaciones que va realizando el W3C a este aspecto, podemos visitar su blog o su página web:

Web	http://www.w3.org/Style/CSS/current-work
Blog	http://www.w3.org/blog/CSS

Conociendo las especificaciones del W3C estaremos al día podremos crear un fichero flexible y acorde a las actuales normas, con lo que tendremos la certeza de que nuestra página se visualizará correctamente, tal y como deseamos, en la mayoría de los navegadores.

Por otro lado, CSS utiliza ciertos términos que sería interesante recordar para manejarlos mejor con el código. Anteriormente hemos podido ver las partes de un estilo básico. Como vamos a profundizar un poco más en éste, vamos a recordarlo:



Como puedes observar en la imagen, se usan diferentes palabras para cada una de las partes que forman un estilo básico en CSS.

Veamos a continuación una definición de cada uno de los términos.

Regla: se trata de los estilos que componen una hoja de estilos. Cada una de estas reglas se compone por selectores y una declaración encerrada entre llaves ({}).

Selector: indica el elemento o elementos HTML a los que se aplicará la regla CSS.

Declaración: esta parte especifica qué estilos son los que se aplicarán a los elementos. Puede estar compuesta por una sola propiedad o por varias.

Propiedad: es la característica que se modifica en el elemento seleccionado, por ejemplo, el tamaño de la letra, el tipo o el color, etc.

Valor: es el término que establece el nuevo valor que tendrá la característica modificada en el elemento.



Un fichero de estilo podrá contener infinidad de reglas CSS, las cuales pueden aplicarse a varios selectores diferentes a la vez y cada declaración podrá incluir tantos pares propiedad/valor como queramos.

1.6.2. Creación de ficheros de estilo genéricos

Ya sabemos que podemos aplicar estilos a una página HTML de diversas formas, pero la más recomendada es a través de una hoja externa. Veamos entonces cómo hacer paso a paso un fichero de estilo genérico externo al documento HTML.

Hacer una hoja de estilos externa nos permitirá poder aplicar estilos a un sitio Web completo, de una vez y con la característica de que se podrá modificar de una forma rápida y flexible. Los pasos para hacer nuestro fichero de estilo genérico serán los siguientes:

- Abriremos un archivo nuevo en nuestro editor de textos.
- Lo guardaremos con el nombre que deseemos en la ubicación que creamos más conveniente.
- El archivo se guardará siempre con la extensión .CSS. Deberemos prestar especialmente atención a este paso, pues de lo contrario, será un archivo .txt que no servirá para aplicar estilos a nuestro documento HTML.

Para escribir dentro de nuestro fichero, no deberemos hacer nada especial, es decir, no habrá que declarar el archivo previamente con alguna declaración (como ocurre con HTML), ni habrá que introducir ninguna etiqueta especial en cada regla CSS que introduzcamos.

Simplemente podemos empezar a introducir las reglas que queramos aplicar a nuestra web y nada más. Esto ocurre porque los navegadores interpretan un archivo con extensión .CSS como un archivo de formato, por lo tanto tendrán el mismo significado que las etiquetas de apertura y cierre de CSS.

Para que nuestro fichero de estilo funcione correctamente, no podemos olvidar que en nuestra página HTML debemos colocar el código que enlazará con nuestra hoja de estilos. Este código debe ir siempre dentro del encabezado de la página, es decir entre las etiquetas `<head> </head>`, y la sintaxis que debemos utilizar es la siguiente:

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

donde `miestilo.css` será el nombre del archivo que hemos puesto a nuestra hoja de estilos previamente.

Resumiendo, son dos los pasos esenciales para crear un fichero de estilo:

- Crear la hoja de estilos en un documento de texto con la extensión .CSS
- Crear el enlace en el documento HTML

Una vez que ya tenemos tanto el fichero de estilo como el enlace en nuestro documento HTML, ¿por dónde empezar?

Primero deberemos tener en cuenta algunas recomendaciones generales sobre CSS antes de empezar a escribir los estilos en nuestro fichero de estilo.

La primera recomendación se refiere a los atributos `ID` y `class` y su correcto uso en las hojas de estilo.

El atributo `id` se emplea para identificar a cada elemento HTML, por lo que los identificadores deben ser únicos en una misma página. En otras palabras, dos elementos HTML diferentes de una misma página no pueden tener un mismo valor en el atributo `id`.

Por otra parte, el atributo `class` se emplea para indicar la clase o clases a las que pertenece el elemento. Una misma clase se puede aplicar a varios elementos diferentes y un único elemento puede tener asignadas varias clases (se indican separadas por espacios en blanco).

Aunque los dos atributos tienen muchos otros propósitos (sobre todo el atributo `id`), CSS los emplea principalmente con los selectores para indicar los elementos sobre los que se aplican los diferentes estilos.

En el siguiente ejemplo, las dos listas están formadas por un mismo elemento HTML ``, pero sus atributos `id` las distinguen de forma adecuada:

```
<ul id="menu">
```



```
...
</ul>
<ul id="enlaces">
...
</ul>
```

Una de las principales recomendaciones del diseño de páginas XHTML y hojas de estilos CSS está relacionada con los valores asignados a los atributos `id` y `class`. Siempre que sea posible, estos atributos se deben utilizar para mejorar la semántica del documento, es decir, para añadir significado a cada elemento de la página.

Por esto mismo, la siguiente recomendación es referida a los valores que debemos asignar a `id` y `class`, los cuales deberán indicar la función del elemento y no deberán estar relacionados con su aspecto o posición (como hemos visto ya anteriormente). Recordemos estas recomendaciones:

Valores no recomendados	Valores recomendados
negrita	importante
arial	titular
verdanaPequena	normal
menuizquierdo	menuSecundario
letraRoja	error

Recordemos también la recomendación sobre elegir el valor adecuado para los atributos `id` o `class`: si el aspecto de un elemento cambiase, el valor de `id` o `class` debe seguir siendo el adecuado. Evitaremos usar valores relacionados con la posición, color o tipo de letra.

También debemos recordar las restricciones que los atributos `id` y `class` deben cumplir para un correcto uso en el fichero de estilo:

- Solo pueden empezar por un guion medio (`-`), un guion bajo (`_`) o una letra.
- El resto de caracteres pueden ser números, guiones medios, guiones bajos y letras.

- Tendremos en cuenta que los navegadores distinguen entre mayúsculas y minúsculas.
- Es posible usar letras como ñ y acentos, pero no es recomendado hacerlo ya que no es seguro que funcione en todas las versiones de todos los navegadores de forma correcta.

Otra de las recomendaciones que debemos tener en cuenta es el buen uso de los atributos id y class:

Un error común al comenzar a desarrollar páginas con estilos CSS es la utilización excesiva de etiquetas `<div>` y atributos class.

A continuación, puedes ver un ejemplo de esta práctica común, donde existe un uso excesivo de las etiquetas div y class:

```
<div id="menu">

  <ul class="menu">

    <li class="elemento_menu"><span class="texto_elemento_menu">...</span></li>

    <li class="elemento_menu"><span class="texto_elemento_menu">...</span></li>

    <li class="elemento_menu"><span class="texto_elemento_menu">...</span></li>

    <li class="elemento_menu"><span class="texto_elemento_menu">...</span></li>

  </ul>

</div>
```

Los selectores de CSS permiten prescindir de la mayoría de etiquetas `<div>` y atributos id y class. Diseñar páginas con exceso de etiquetas `<div>` no mejora la semántica del documento y sólo consigue complicar el código HTML.

A continuación, veamos las recomendaciones que se dan para estructurar de la mejor forma un fichero de estilo:

La posibilidad de incluir todo el código CSS en archivos externos exclusivamente dedicados a contener las reglas CSS, permite ordenar de forma lógica las reglas, mejorando su legibilidad y facilitando su actualización.

Las reglas CSS de las hojas de estilos complejas se suelen agrupar según su funcionalidad y se suelen incluir en el siguiente orden:

- Estilos básicos (<body> tipo de letra por defecto, márgenes de , y , etc.).
- Estilos de la estructura o layout (anchura, altura y posición de la cabecera, pie de página, zonas de contenidos, menús de navegación, etc.).
- Enlaces (estilos normales, estilos :hover, etc.).
- Estilos de cada una de las zonas (elementos de la cabecera, titulares y texto de la zona de contenidos, enlaces, listas e imágenes de las zonas laterales, etc.).

Otra característica común de los mejores sitios web es el uso de comentarios CSS para mejorar la estructura de las hojas de estilos muy largas.

```
#subtitle {  
    font: normal 18px "Trebuchet MS";  
    letter-spacing: 4px;  
    word-spacing: 18px;  
    color: #6f6f6f;  
    margin: 0;  
    padding: 0 12px;  
}
```

Hoja de estilos ordenada

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 */
```

Los comentarios pueden ocupar tantas líneas como sea necesario, pero no se puede incluir un comentario dentro de otro comentario:

```
/* Este es un  
comentario CSS de varias  
lineas */
```

Aunque los navegadores ignoran los comentarios, su contenido se envía junto con el resto de estilos, por lo que no se debe incluir en ellos ninguna información sensible o confidencial.

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 -->  
<!-- Este es un  
comentario HTML de varias  
lineas -->
```

Con todas las recomendaciones en nuestra mano, y conociendo ya los atributos de los que podremos hacer uso para cambiar el estilo de nuestra página web, pasemos a la acción.

Ya hemos creado nuestro documento CSS, y tenemos el documento web con los contenidos y la estructura que deseamos que tenga nuestra página web con el enlace conveniente a la hoja de estilos. Ya con todo esto, es la hora de conocer cómo se relacionan entre sí los selectores de los elementos con la página web.

Uno de los grandes dilemas de los diseñadores Web es enfrentarse a elegir entre ID o class.

La respuesta es muy fácil: elegiremos class para las etiquetas que deben aparecer más de una vez en la página, e id para las que aparecen sólo una vez.

Por etiqueta se entiende una palabra descriptiva que deseamos aplicar a un elemento, que es para lo que se utilizan class e id.

En nuestra hoja CSS por lo tanto, escribiremos los atributos de class o id. Como hemos visto anteriormente, class e id son los selectores que van relacionados al elemento html que queremos cambiar.

Es decir, en nuestro documento HTML añadimos un selector class o id junto a su nombre. Este nombre deberá coincidir con los atributos de clase que irán en nuestra hoja de estilos.



Importante

Recordamos que los selectores de clase deben ir precedidos de un punto (.), y los selectores de id deben ir precedidos con una almohadilla (#) en la hoja de estilos.

Por último, no debemos olvidar una de las prácticas que deberemos realizar de vez en cuando, a lo largo del proceso de creación de la página y del diseño de la misma. Y no es otra que validar.

Es muy importante validar y sin embargo es una de las cosas que más se olvidan hacer o más se retrasan en nuestra agenda. Pero hay que recordar que es uno de los pasos importantes, y es una práctica que debemos realizar, y acostumbrarnos a hacerlo, cada cierto tiempo.

Es mejor detectar posibles problemas en la página a tiempo, y así poder resolverlos, antes de que éstos infecten toda la estructura de la página, con lo que tendríamos que deshacer mucho del trabajo realizado, con la consecuente pérdida de tiempo.

Podemos acceder a diferentes validadores para HTML y CSS:

- HTML: <http://validator.w3.org/>
- CSS: <http://jigsaw.w3.org/css-validator/>

Si estuviésemos desarrollando tras un firewall en nuestro portátil con un servidor web local, podremos utilizar las funciones de validación locas de Firebug y otras herramientas de desarrollador. Si tenemos acceso a la web, podemos

validar cualquier página que estemos visualizando, sea accesible para la navegación o no. Sin embargo, es preferible recurrir a la validación local, aunque la página se encuentre en un sitio público, ya que así conseguiremos adquirir el hábito de usar este servicio.

1.6.3. Adaptación de los ficheros de estilo para distintas páginas web

Una de las características más importantes de las hojas de estilos CSS es que permiten definir diferentes estilos para diferentes medios o dispositivos: pantallas, impresoras, móviles, proyectores, etc.

Además, CSS define algunas propiedades específicamente para determinados medios, como por ejemplo la paginación y los saltos de página para los medios impresos o el volumen y tipo de voz para los medios de audio. La siguiente tabla muestra el nombre que CSS utiliza para identificar cada medio y su descripción:

Medio	Descripción
all	Todos los medios definidos
braille	Dispositivos táctiles que emplean el sistema braille
embossed	Impresoras braille
handheld	Dispositivos de mano: móviles, PDA, etc.
print	Impresoras y navegadores en el modo "Vista Previa para Imprimir"
projection	Proyectores y dispositivos para presentaciones
screen	Pantallas de ordenador
speech	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
tty	Dispositivos textuales limitados como teletipos y terminales de texto
tv	Televisores y dispositivos con resolución baja

Los medios más utilizados actualmente son screen (para definir el aspecto de la página en pantalla) y print (para definir el aspecto de la página cuando se imprime), seguidos de handheld (que define el aspecto de la página cuando se visualiza mediante un dispositivo móvil).

Además, CSS clasifica a los medios en diferentes grupos según sus características. La siguiente tabla resume todos los grupos definidos en el estándar:

Medio	Continuo / Paginado	Visual / Audi- tivo / Táctil / Vocal	Mapa de bits / Caracteres	Interactivo / Estático
braille	continuo	táctil	caracteres	ambos
embos- sed	paginado	táctil	caracteres	estático
handheld	ambos	visual, auditivo, vocal	ambos	ambos
print	paginado	visual	mapa de bits	estático
projec- tion	paginado	visual	mapa de bits	interactivo
screen	continuo	visual, auditivo	mapa de bits	ambos
speech	continuo	vocal	(no tiene sen- tido)	ambos
tty	continuo	visual	caracteres	ambos
tv	ambos	visual, auditivo	mapa de bits	ambos

La gran ventaja de CSS es que permite modificar los estilos de una página en función del medio en el que se visualiza. Existen cuatro formas diferentes de indicar el medio en el que se deben aplicar los estilos CSS.

Las reglas @media son un tipo especial de regla CSS que permiten indicar de forma directa el medio o medios en los que se aplicarán los estilos incluidos en la regla. Para especificar el medio en el que se aplican los estilos, se incluye su nombre después de @media. Si los estilos se aplican a varios medios, se incluyen los nombres de todos los medios separados por comas.

A continuación, se muestra un ejemplo sencillo:

```
@media print {
    body { font-size: 10pt }
}

@media screen {
```



```

body { font-size: 13px }

}

@media screen, print {

    body { line-height: 1.2 }

}

```

El ejemplo anterior establece que el tamaño de letra de la página cuando se visualiza en una pantalla debe ser 13 píxel. Sin embargo, cuando se imprimen los contenidos de la página, su tamaño de letra debe ser de 10 puntos. Por último, tanto cuando la página se visualiza en una pantalla como cuando se imprimen sus contenidos, el interlineado del texto debe ser de 1.2veces el tamaño de letra del texto.

Cuando se utilizan reglas de tipo @import para enlazar archivos CSS externos, se puede especificar el medio en el que se aplican los estilos indicando el nombre del medio después de la URL del archivo CSS:

```

@import url("estilos_basicos.css") screen;

@import url("estilos_impresora.css") print;

```

Las reglas del ejemplo anterior establecen que cuando la página se visualiza por pantalla, se cargan los estilos definidos en el primer archivo CSS. Por otra parte, cuando la página se imprime, se tienen en cuenta los estilos que define el segundo archivo CSS.

Si los estilos del archivo CSS externo deben aplicarse en varios medios, se indican los nombres de todos los medios separados por comas. Si no se indica el medio en una regla de tipo @import, el navegador sobreentiende que el medio es all, es decir, que los estilos se aplican en todos los medios.

Si se utiliza la etiqueta <link> para enlazar los archivos CSS externos, se puede utilizar el atributo media para indicar el medio o medios en los que se aplican los estilos de cada archivo:

```

<link rel="stylesheet" type="text/css" media="screen"
      href="basico.css" />

<link rel="stylesheet" type="text/css" media="print, handheld"
      href="especial.css" />

```

En este ejemplo, el primer archivo CSS se tiene en cuenta cuando la página se visualiza en la pantalla (media="screen"). Los estilos indicados en el segundo

archivo CSS, se aplican al imprimir la página (`media="print"`) o al visualizarla en un dispositivo móvil (`media="handheld"`), como por ejemplo en un iPhone.

Si la etiqueta `<link>` no indica el medio CSS, se sobreentiende que los estilos se deben aplicar a todos los medios, por lo que es equivalente a indicar `media="all"`.

CSS también permite mezclar los tres métodos anteriores para indicar los medios en los que se aplica cada archivo CSS externo:

```
<link rel="stylesheet" type="text/css" media="screen"
      href="basico.css" />

@import url("estilos_seccion.css") screen;

@media print {

    /* Estilos específicos para impresora */

}
```

Los estilos CSS que se aplican cuando se visualiza la página en una pantalla se obtienen mediante el recurso enlazado con la etiqueta `<link>` y mediante el archivo CSS externo incluido con la regla de tipo `@import`. Además, los estilos aplicados cuando se imprime la página se indican directamente en la página HTML mediante la regla de tipo `@media`.

En el caso de incluir en una misma hoja de estilos varios estilos según el dispositivo donde se vaya a visualizar el contenido de la página, los estilos por defecto se definirán al principio, seguidos de las reglas y estilos que satisfacen la consulta de medios. Los estilos definidos al principio se aplicarán en cascada sobre los estilos coincidentes con la regla, pudiendo sobrescribirse por completo.

Área: informática y comunicaciones

UD1

Bibliografía de ampliación

Referencias bibliográficas

- Estrada, A. y Berisso, G. (2019). La guerra de los estilos. La hoja de estilo vs. El estilo editorial. [Comunicación en congreso]. 5º Congreso Internacional de correctores de texto en español. El corrector, entre el texto y el lector.

En este artículo se habla sobre la hoja de estilo editorial como instrumento utilizado por autores, académicos y correctores. Por ello, en este documento se muestra la evolución de la hoja de estilo de varios productos de una misma editorial, así como las propias tensiones creadas entre la normativa y los factores ajenos a ella.

- Tosi, C. (2020). ¿Hojas de estilo para el lenguaje inclusivo? Un análisis acerca de las prácticas de corrección de estilo en el ámbito editorial. *Exlibris*, 9, 169-179. [http://revistas.filо.uba.ar/index.php/exlibris/article/view/3345](http://revistas.filو.uba.ar/index.php/exlibris/article/view/3345)

En este artículo se pretende analizar las prácticas de la corrección del estilo en relación con el lenguaje inclusivo por medio de una serie de encuestas realizadas a editoriales comerciales y organismos gubernamentales. Se pretende contribuir al lenguaje reflexivo del uso del español en las tareas de escritura, corrección y edición.

- Hernández, C. V. (2018). Introducción a HTML y CSS. [Fundación Universitaria Konrad Lorenz]. Repositorio digital Konrad Lorenz. <https://repositorio.konradlorenz.edu.co/handle/001/136>

En este documento se presenta la importancia y las funciones necesarias para poder mostrar una serie de aplicaciones web al público, es decir, el estilo de maquetación que se presentara ante una serie de aplicaciones. Por otra parte, se comenta sobre la estructura que se realizaría en el HTML.

- Jaimez-González, CR. (2017). Portal web con recursos didácticos digitales para el aprendizaje de HTML y CSS. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 8 (5).

En este artículo se recoge cómo se ha creado una página web como forma de apoyar el aprendizaje del lenguaje HTML y las hojas de estilo en cascada, CSS. Por ello, para el desarrollo de este portal, se discute sobre la organización del contenido, los objetivos y los recursos didácticos.

Enlaces de interés

- Munoz, E. (4 de junio, 2017). Las hojas de estilo editorial. *Wordpress*. <https://munozeric.wordpress.com/2017/06/04/las-hojas-de-estilo-editorial/>

En esta página se habla sobre las funciones que cumple la hoja de estilo, así como los diferentes componentes editoriales que intervienen en su desarrollo.

- Delgado, H. (21 de noviembre, 2020). Formularios HTML- ejemplos y estilos con CSS para controles. *Diseño web*. <https://disenowebakus.net/formularios-html.php>

En esta página se habla sobre la importancia que tiene de mantener los formularios actualizados para su posterior uso. Por otra parte, se destaca la estructura básica que presenta un formulario al igual de diferentes atributos con los que se suele configurar.

- Echevarria, I. (8 de mayo, 2021). Los conceptos básicos sobre programación más requeridos por las empresas en la actualidad. *Radio agricultura*. <https://www.radioagricultura.cl/tendencias/2021/05/08/los-conceptos-basicos-sobre-programacion-mas-requeridos-por-las-empresas-en-la-actualidad.html>

En este artículo se explica la importancia que ha ido adquiriendo en estos últimos meses la necesidad de conocer formatos de programación y ejecución de programas y aplicaciones, por ello, en este se muestra una breve explicación de las funciones que cumplen algunos de los recursos básicos.

- Alvares, M. A. (11 de febrero, 2020). Distintas formas de incluir estilos CSS en una página web. *Desarrolloweb*. <https://desarrolloweb.com/articulos/183.php>

En esta página se presenta las distintas formas con las que se pueden incluir estilos de CSS en las páginas web, desde una forma general hasta otra más concreta.

UD1

Lo más importante

- CSS (u hojas de estilo en cascada) nació como un estándar para complementar HTML y la visualización de los elementos.
- Deberemos tener en cuenta la accesibilidad de los usuarios para que vean en condiciones óptimas la página web.
- Existen 4 formas de aplicar estilos: incrustados, enlazados, importados y en línea.
- A través de las hojas de estilos, podremos aplicar atributos al texto, al color, al fondo, y a los párrafos.

UD1

Autoevaluación

1. ¿Cómo enlazamos un archivo CSS a nuestro documento HTML?
 - a. Con la etiqueta <a>.
 - b. Usando la etiqueta <link> desde la cabecera de cada documento.
 - c. Con la etiqueta <a href>.
 - d. No se puede enlazar.
2. ¿Qué determina el atributo rel dentro del enlace que debemos poner para enlazar una hoja de estilos?
 - a. La URL.
 - b. El tipo de recurso enlazado.
 - c. El tipo de relación que hay entre el recurso enlazado y el documento HTML.
 - d. La ruta.

3. ¿Qué determina el atributo type dentro del enlace que debemos poner para enlazar una hoja de estilos?
 - a. La URL.
 - b. El tipo de relación.
 - c. La dirección.
 - d. El tipo de recurso enlazado.
4. ¿Qué determina el atributo href dentro del enlace que debemos poner para enlazar una hoja de estilos?
 - a. El tipo de relación.
 - b. El medio en el que el estilo se aplicará.
 - c. La URL del archivo CSS.
 - d. El tipo de recurso enlazado.
5. ¿Dónde se integra el estilo cuando está incrustado?
 - a. Enlazada a una hoja en línea.
 - b. En una hoja de estilos externa.
 - c. Dentro del documento HTML, en cualquier parte.
 - d. Dentro del documento HTML, en el encabezado.
6. ¿Qué regla se usa para importar una hoja de estilos y aplicarla a un documento HTML?
 - a. :import.
 - b. @import.
 - c. import.
 - d. import@.

7. ¿De qué tipos pueden ser los selectores?
 - a. Selectores simples y Selectores contextuales.
 - b. Selectores simples y Selectores compuestos.
 - c. Selectores interiores y exteriores.
 - d. Selectores de Regla y de Valores.
8. ¿Qué heredan los hijos en la propiedad de Herencia propia de los CSS?
 - a. Los hijos siempre heredan los estilos de sus padres.
 - b. Los hijos no heredan nada.
 - c. Los hijos heredan, la mayoría de las veces, los estilos de sus padres.
 - d. Los hijos heredan los estilos de otros hijos.
9. ¿Qué atributo usamos para aplicar un estilos con el selector de clases?
 - a. El atributo clas.
 - b. El selector universal *.
 - c. El atributo ID.
 - d. El atributo class.
10. ¿Qué propiedad se utiliza para cambiar la fuente del texto?
 - a. font-weight.
 - b. font-family.
 - c. font-size.
 - d. font-style.

UD2

Diseño, ubicación
y optimización de
los contenidos de
una página web

- 2.1. Creación de un documento funcional
 - 2.1.1. Descripción de los elementos funcionales de la página web
 - 2.1.2. Definición de los elementos funcionales de la página web
 - 2.1.3. Descripción de cada elemento
- 2.2. Diseño de los contenidos
- 2.3. Identificación de la información a ubicar en la página web
- 2.4. Selección de contenidos para cada elemento de la página
- 2.5. Utilización del documento funcional para las especificaciones del diseño
- 2.6. Tipos de página para la ubicación de contenidos
- 2.7. Definición de los tipos de página en base a los contenidos y funcionalidades
- 2.8. Selección de los tipos de página para la página web
- 2.9. Utilización del documento funcional para las especificaciones del tipo de página
- 2.10. Especificaciones de navegación
- 2.11. Creación de un mapa de navegación de páginas
- 2.12. Utilización del documento funcional para integrar el mapa de navegación
- 2.13. Elementos utilizados para la navegación
 - 2.13.1. Definición de los elementos utilizados para navegar
 - 2.13.2. Utilización del documento funcional para especificar los elementos de navegación
- 2.14. Elaboración de una guía de usuario

2.1. Creación de un documento funcional

Para crear un documento funcional en el que aplicaremos los conocimientos adquiridos de CSS no se necesitan cosas demasiado especiales. Lo imprescindible para comenzar a trabajar es lo siguiente:

- Un ordenador

Cualquiera servirá, ya sea con Windows, Mac o Linux. Incluso un PDA serviría. En otras palabras, es inútil comprar un equipo potente para crear páginas web. Las máquinas que se usaban para el código HTML hace diez años eran menos potentes que cualquier PDA actual.

Una gran pantalla sí que nos será muy útil, ya que te permitirá mostrar más ventanas en pantalla o verlas a un tamaño más grande. Un monitor de 15 pulgadas será suficiente, y uno de 17 pulgadas o más sería ideal.

- Conexión a Internet

Este elemento realmente es opcional, ya que uno podría crear páginas web en el equipo local para uso personal. Evidentemente será necesario si se desea compartir las páginas web con otros usuarios.

- Un editor de texto

Un archivo HTML es un sencillo archivo de texto, cuya extensión es .html o .htm en lugar de .txt. Igualmente, un archivo CSS es también un archivo de texto, cuya extensión es .css.

El Bloc de notas de Windows servirá de sobra para escribir una página HTML o CSS. Sin embargo, recomendamos el uso de un programa mejor, y gratuito: Notepad 2. Lo podemos descargar y comenzar a trabajar con él en poco tiempo.

La principal ventaja es que reconoce el lenguaje HTML y CSS, y lo coloreará automáticamente. Esto puede facilitarnos mucho el trabajo.

Además, también nos permite numerar las líneas, los que nos será muy útil para localizar errores o determinados bloques de código.

El bloc de notas equivalente para Mac es el Textedit, y para Linux es Gedit.

¿Por qué no utilizar un editor WYSIWYG? Los editores WYSIWYG (lo que ves es lo que obtienes), nos permiten codificar páginas sin manipular el código HTML, de forma similar a como utilizamos el procesador de textos para aplicar negrita, cursiva o insertar imágenes en una página. Es decir, todo se puede hacer con el ratón, sin escribir una línea de código.

Aptana Studio es un entorno de desarrollo integrado de software libre basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: PHP, Python, Ruby, CSS, Ajax, HTML y ActionScript 3.0. Wikipedia

- Fundador: Paul Colton
- Fundación: 2005
- Director ejecutivo: Paul Colton (oct 2005)
- Oficinas centrales: San Mateo, California, Estados Unidos
- Licencia: Licencia Dual: Aptana Public License v1.0 GNU General Public License
- Tipo de programa: Web Development/IDE
- Organización principal: Appcelerator

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Wikipedia

- Programado en: TypeScript, JavaScript, CSS
- Desarrollador: Microsoft
- Licencia-> Código fuente-> Licencia MIT-> Binarios-> Freeware
- Lanzamiento: 29 de abril de 2015
- Plataformas admitidas: x86, x86-64 y ARM
- Tipo de programa: editor de código fuente

El más conocido es Dreamweaver, de Adobe. Como nuestro propósito es aprender CSS, no aprender a utilizar este programa, evitaremos su uso, además el uso de este programa tiene algunos inconvenientes: no es gratuito y no crea un código optimizado o fácil de entender para un principiante.

- Un programa FTP.

Para publicar los archivos en internet, es necesario disponer de un programa específico capaz de conectarse por FTP a tu servidor.

- Imágenes.

Para preparar las imágenes que usaremos en la página HTML, deberíamos disponer de un programa de retoque y edición que te permita recortar y reducir el tamaño de la imagen, además de guardarla en formato JPEG o PNG.

La mayoría de programas de edición permiten realizar estas manipulaciones. Puedes encontrar numerosos editores de imágenes gratuitos, entre ellos se encuentra Gimp. Es muy completo y muy parecido al programa tan conocido como Photoshop.

- Un navegador.

Los puntos que veremos a continuación funcionarán en la mayoría de navegadores, Google Chrome, IE, Firefox, Opera, etc. Lo normal es que utilicemos la última versión de nuestro navegador favorito. Se recomienda en concreto el navegador Firefox, para probar las páginas, ya que ofrece numerosas posibilidades en la depuración de errores de la misma.

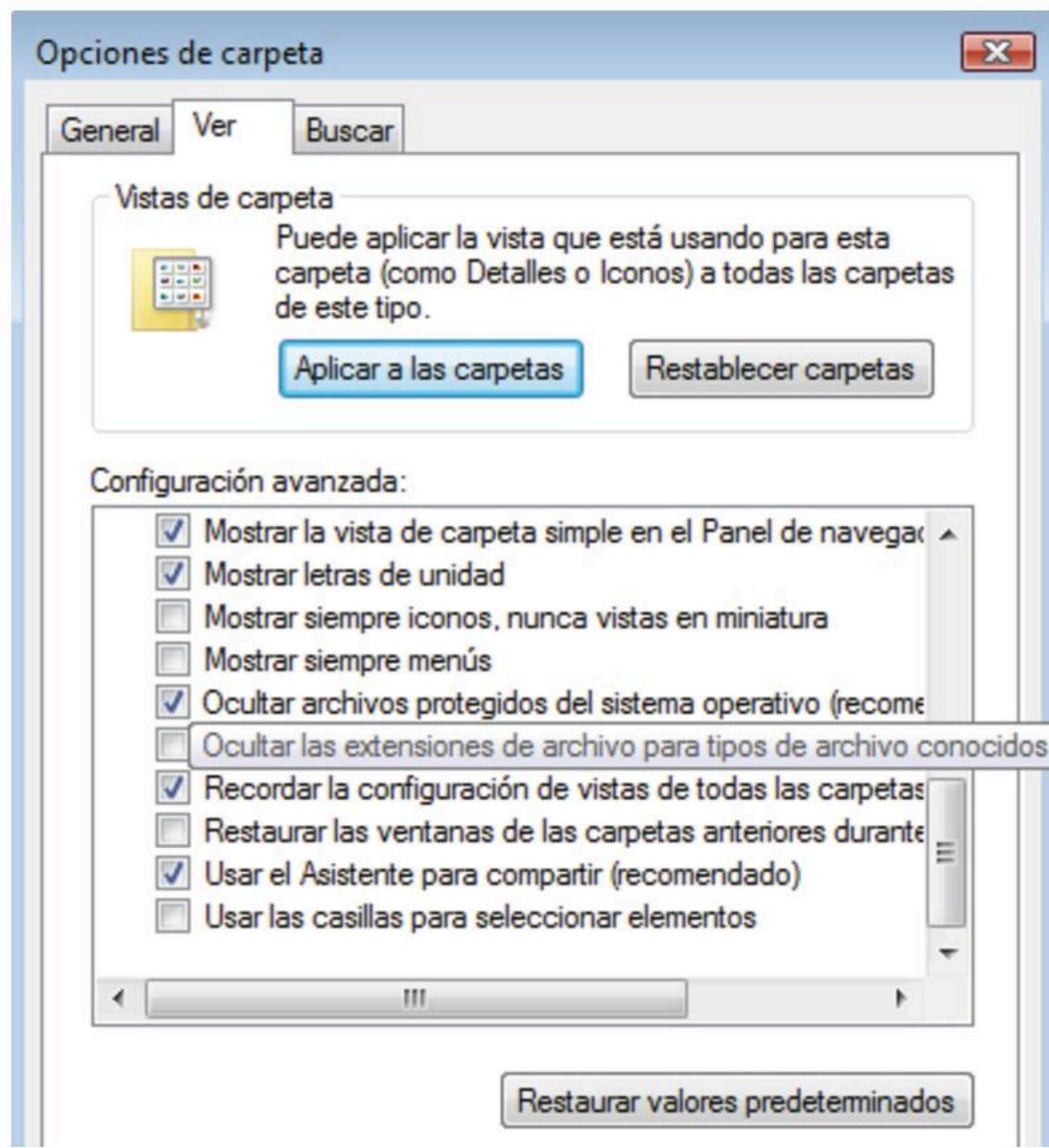
- Bloc de Notas.

Y esta vez nos referimos a un bloc de notas de verdad, un cuaderno. Tenerlo siempre cerca nos ayudará a apuntar lo que nos parezca útil, información como código, reglas, etc., o ideas a recordar. Tener esta información guardada será muy útil a medida que avanzamos en nuestro trabajo, ya que será una gran herramienta de consulta.

- Mostrar extensiones.

De forma predeterminada, Windows suele ocultar las extensiones de archivos. En ocasiones, tenemos varios archivos de imagen, por ejemplo, pero algunos están en formato distintos, dependiendo de su uso en la página web. Por esto mismo a veces es imposible saber cuál es el archivo concreto que queremos utilizar, ya que la extensión está oculta. Para evitar esto, y saber exactamente cuál es el archivo que necesitamos, haremos que Windows muestre siempre las extensiones.

Desde el explorador Windows, deberemos acceder a Herramientas > Opciones de Carpeta, y clicaremos sobre la ficha Ver. Ésta mostrará una lista con la Configuración avanzada. Es ahí donde deberemos desactivar la casilla Ocultar las extensiones de archivo para tipos de archivos conocidos, tal y como podemos ver en la siguiente imagen:



De esta forma, podremos ver el nombre de los archivos completo.

- Creación de un archivo HTML y un archivo CSS.

Se recomienda que creemos una carpeta donde guardaremos ambos archivos. Podemos crear los dos archivos desde el programa Notepad2 o desde el Bloc de notas. Simplemente crearemos dos archivos nuevos, uno de ellos lo guardaremos con el nombre que deseemos y la extensión

.html, el cual será el archivo HTML. El otro archivo lo guardaremos con la extensión .css y será nuestra hoja de estilos.

Una vez creado el archivo HTML, éste se mostrará en carpeta con el ícono del navegador. Para abrirlo con el Notepad2 o el Bloc de notas, podemos hacer clic con el botón derecho del ratón sobre el archivo y clicar en Abrir con, donde seleccionaremos el editor de texto que estemos usando. Así podremos comenzar a editar el documento web.

Para abrir el CSS, haremos lo mismo, de tal forma que podremos comenzar a editar el CSS del sitio Web.

2.1.1. Descripción de los elementos funcionales de la página web

A continuación, vamos a comenzar a realizar una página web desde cero. Será una página web básica en la que podremos aplicar distintos estilos, compuesta por una serie de elementos funcionales que nos servirán como práctica para ir practicando con los elementos que hemos estudiado hasta ahora.

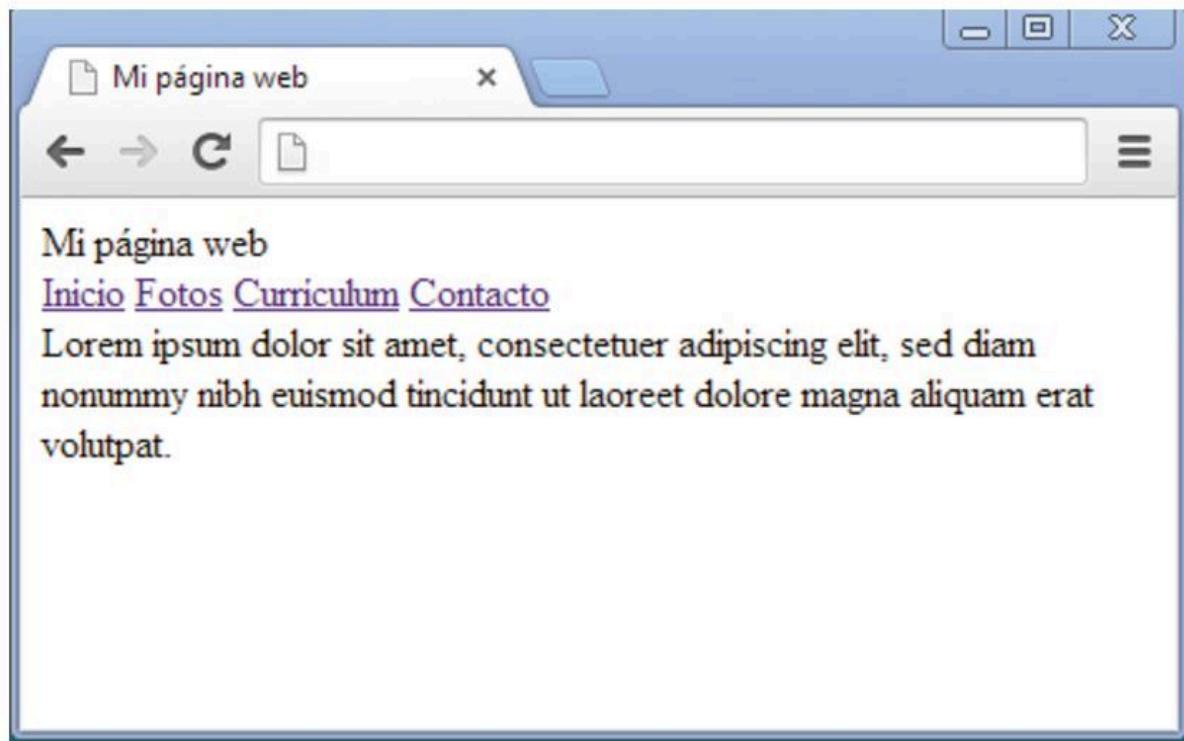
Los elementos funcionales que veremos en la página web serán:



Estos elementos son los que constituirán el documento principal o índice del documento HTML. Tanto el cuerpo del documento, como el título, menú y tabla, irán modificados en su estilo gracias al CSS que iremos modificando.

Lo que haremos será la página personal de un fotógrafo, con el fin de mostrar sus trabajos, el currículum y un apartado de contacto.

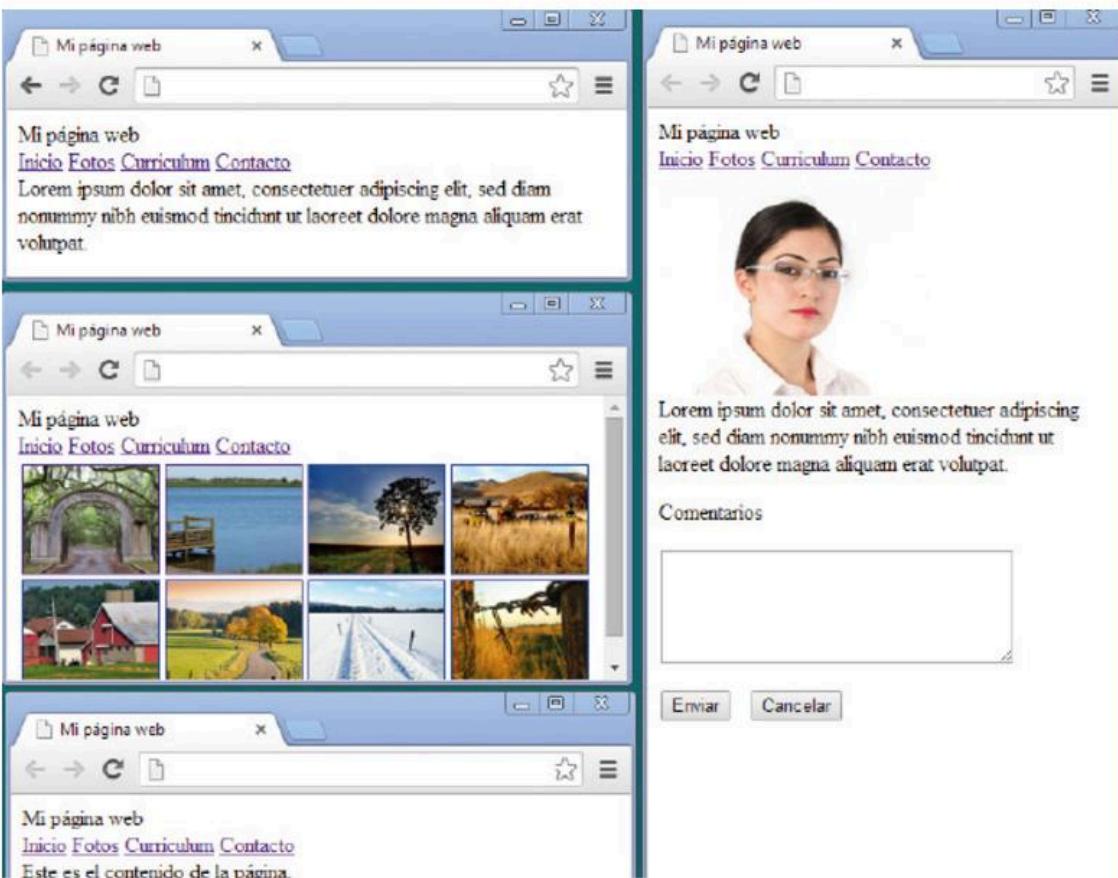
En la siguiente imagen podrás observar los elementos que tendrá la página:



Como puedes ver en la imagen, los elementos que en principio tendrá nuestra página serán muy básicos. Un menú escueto, un poco de texto y un encabezado. Poco más necesitamos para a partir de ahí comenzar a darle estilo y convertirla en una página mucho más atractiva.

En la siguiente imagen puedes ver los 4 documentos web de los que se compone la página. Cada uno de ellos pertenece a un elemento del menú:





Ya sabemos desde que elementos vamos a partir, pero ¿cuál será el resultado final? A continuación, vamos a describir cada uno de los elementos a los que llegaremos en cada documento web.

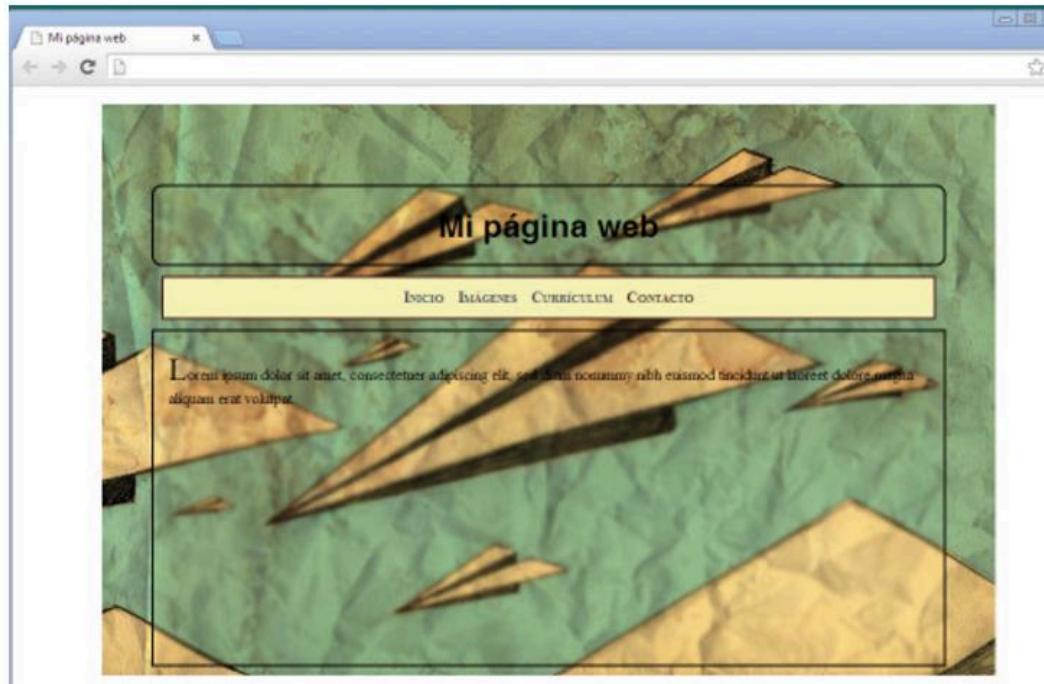
– Inicio.

En la página de Inicio llegaremos a tener varios elementos funcionales, los cuales iremos insertando poco a poco e iremos dándole el estilo que deseemos.

Los elementos que tendremos que realizar en la página son los siguientes:

- Encabezado: Pondremos un encabezado en la página, con un texto descriptivo que presente y describa la página.
- Menú: Introduciremos un menú que nos ayude a navegar por los distintos documentos de la página.
- Texto descriptivo: Para nuestra página de Inicio simplemente vamos a introducir un texto que describirá el sitio web en sí y lo que pretende.

En la imagen que puedes ver a continuación, puedes ver el primer objetivo al que queremos llegar una vez hayamos aplicado los estilos a nuestra página web:



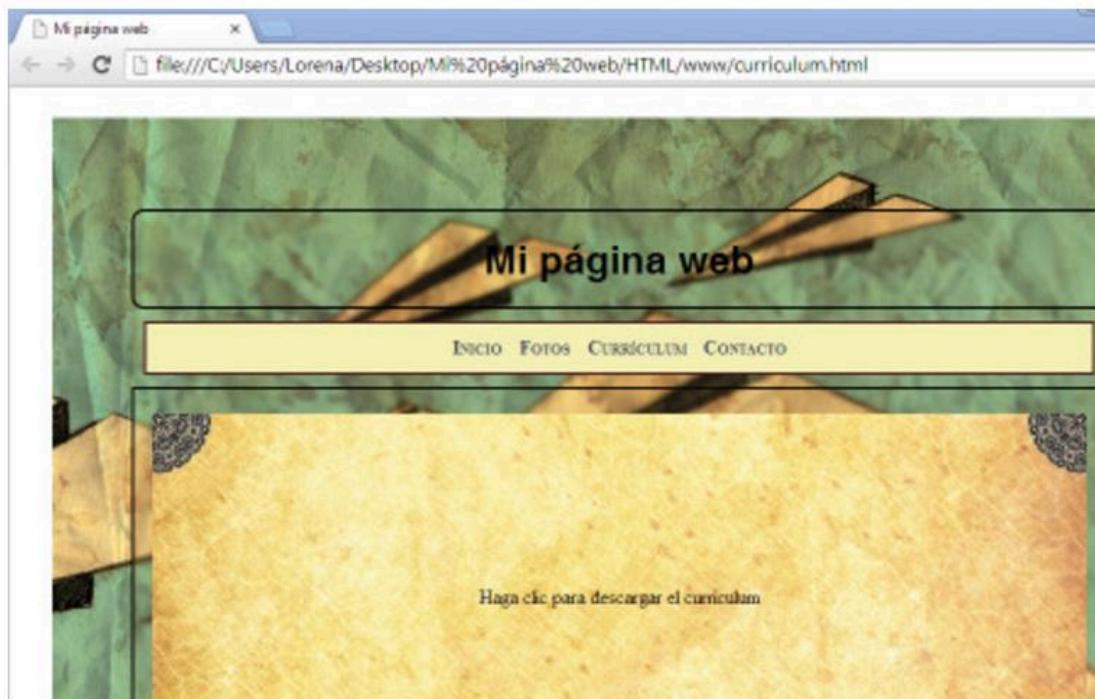
- Fotos.

Cuando clicamos en el menú en el botón Fotos, nos llevará a otro documento Web donde se muestran algunos trabajos dentro de una tabla. Las imágenes son miniaturas de 100 x 80 px. Al clicar sobre cada imagen, se abrirán en una nueva ventana en su tamaño original.



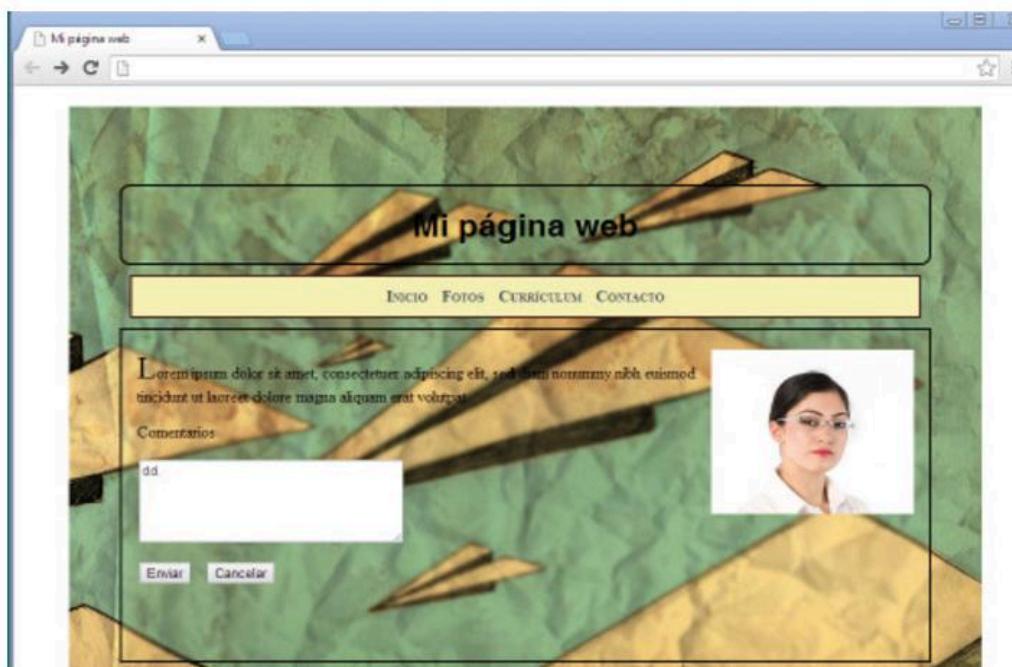
- Currículum.

En el apartado currículum habrá un enlace a un documento PDF desde donde se podrá descargar el archivo.



- Contacto.

En el apartado de Contacto, tendremos una imagen con una foto personal y un apartado para poder enviar comentarios al correo electrónico.



2.1.2. Definición de los elementos funcionales de la página web

Ahora vamos a ver cuáles son los elementos funcionales de la página web, no vistos por el navegador, si no en el código en sí. Vamos a ver cada uno de los documentos web (en total son 4), y definirlos paso a paso para formar la estructura de cada uno de ellos.

- Inicio.

Veamos en primer lugar el código que hemos usado para el Inicio en el documento HTML para definirlo después paso a paso:

```
<!DOCTYPE html>

<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

<span>L</span>orem ipsum dolor sit amet,
consectetuer adipiscing elit,
sed diam nonummy nibh euismod
tincidunt ut labore et dolore
```



```
magna aliquam erat volutpat.  
</div>  
</div>  
</body>  
</HTML>
```

Como puedes ver, hay una primera división que contiene a todo.

```
<body>  
<div>  
<div>Mi página web</div>  
<div>  
<a href="iniciosinestilos.html">Inicio</a>  
<a href="fotossinestilo.html">Fotos</a>  
<a href="curriculumsinestilo.html">Curriculum</a>  
<a href="contactosinestilos.html">Contacto</a></div>  
<div>  
<span>L</span><p>orem ipsum dolor sit amet,  
consectetuer adipiscing elit,  
sed diam nonummy nibh euismod  
tincidunt ut laoreet dolore  
magna aliquam erat volutpat.  
</div>  
</div>  
</body>
```

Seguidamente hay una división para cada elemento:

- primero una división para un banner que contiene el título que queremos mostrar:

```
<div>Mi página web</div>
```


- una segunda división para el menú:

```
<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>
```

- y finalmente una tercera división que será la que contenga el contenido que queramos mostrar en cada uno de los documentos web:

```
<div>

<span>L</span>orem ipsum dolor sit amet,
consectetuer adipiscing elit,
sed diam nonummy nibh euismod
tincidunt ut laoreet dolore
magna aliquam erat volutpat.

</div>
```

- Fotos.

El código que hemos utilizado en el documento web "Fotos" es el siguiente:

```
<!DOCTYPE html>

<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>
```



```
<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1"></a></td>

<td><a href="images/image-2.jpg]" title="Imagen 2"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4"></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg]" title="Imagen 5"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8"></a><br></td>

</tr>
```



```
</table>  
</div>  
</div>  
</body>  
</HTML>
```

Como puedes observar, este documento se compone en la primera parte de la misma forma que el índice y que el resto del documento: una primera división para el banner que llevará el título y una segunda división para integrar el menú.

El elemento diferenciador en este documento es que la tercera división contendrá una tabla dentro de la cual se enlazarán los archivos que queremos mostrar. En este caso se trata de una tabla muy simple de 4 columnas y dos filas.

En cada una de las celdas hemos enlazado a un archivo relativo a través del elemento `<a href>`, hemos introducido un título para las imágenes usando el elemento `<title>`, y hemos indicado la ruta del archivo a través del elemento `<src>`.

También hemos indicado el ancho y el alto de los archivos con `<width>` y `<height>` y hemos indicado el grosor del borde con `<border>`.

- Currículum.

El código HTML que hemos utilizado para el documento web currículum es el siguiente:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html;  
charset=iso-8859-1" />  
  
<title>Mi página web</title>  
  
</head>  
  
<body>  
  
<div>
```



```
<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

    <div><p>Haga clic aquí para descargar el currículum</p></div>

</div>

</div>

</body>

</HTML>
```

Al igual que el resto de documentos web, éste también se compone por un banner principal y el menú a través del cual podremos navegar por la página.

Seguidamente, tenemos un `<div>` donde está el contenido de este documento web en particular, el cual se compone básicamente por un texto con un enlace.

Dentro de este texto, insertaremos un enlace a un currículum, el cual será un archivo PDF. Para ello usaremos el elemento `<a>` donde estará la ruta específica hacia el documento.

- Contacto.

El código HTML que hemos utilizado para el documento web Contacto es el siguiente:

```
<!DOCTYPE html>

<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
```



```
<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

<div>



</div>

<span>Lorem ipsum dolor sit amet, consectetur
adipiscing elit,



<form method="post" action="mailto:correo@extension.com"
      enctype="text/plain">

<p>Comentarios</p>

<textarea rows="5" cols="30" name="comentarios"></texta-
rea>

<p><button type="submit">Enviar</button>&nbsp;&nbsp;
<button type="reset">Cancelar</button></p>

</form>

</div>

</div>
```



```
</body>
```

```
</HTML>
```

Como puedes ver, el documento está compuesto en la primera parte de la misma forma que el índice y que el resto de la web: una primera división para el banner que llevará el título y una segunda división para integrar el menú.

El contenido de esta parte será un formulario de contacto junto a una imagen con una foto personal. El formulario en sí será muy básico, simplemente para que el usuario pueda enviar sus impresiones y pueda contactar con nosotros a través de un correo electrónico.

Además hemos insertado una imagen con una foto personal a través del elemento ``.

2.1.3. Descripción de cada elemento

Vamos ahora a describir cada uno de los elementos que componen la página para conocerla en profundidad y así poder, posteriormente, aplicar los estilos pertinentes a cada uno de sus partes.

Empecemos por la página de Inicio o index de la misma.

- Índice.

La página de Inicio se encuentra dividida en 3 partes claramente diferenciadas. Estas divisiones se encuentran así mismo en los demás elementos del menú, por lo que se explicarán sólo una vez.

Antes de explicar las divisiones de la página, podemos observar que el cuerpo del documento contiene una imagen de fondo. Ésta imagen en sí era más oscura, sin embargo, se ha editado para que haga contraste con el texto y éste sea más legible para el usuario.

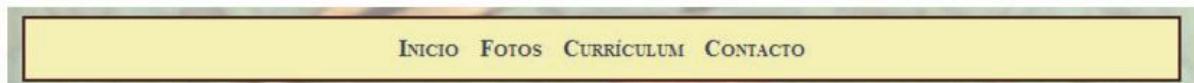
Si nos fijamos un poco más, la imagen está desenfocada dentro de las zonas donde se encuentran las divisiones o distintas partes del documento.



La primera división se ha creado como espacio para colocar el título del documento, es decir, un banner, a modo de presentación de la página.

Este espacio en sí contiene simplemente un texto. La división tiene un borde de un grosor determinado, y las esquinas de la división están redondeadas.

Seguidamente, encontramos la segunda división que contiene el menú del documento, a través del cual podremos navegar por la página.

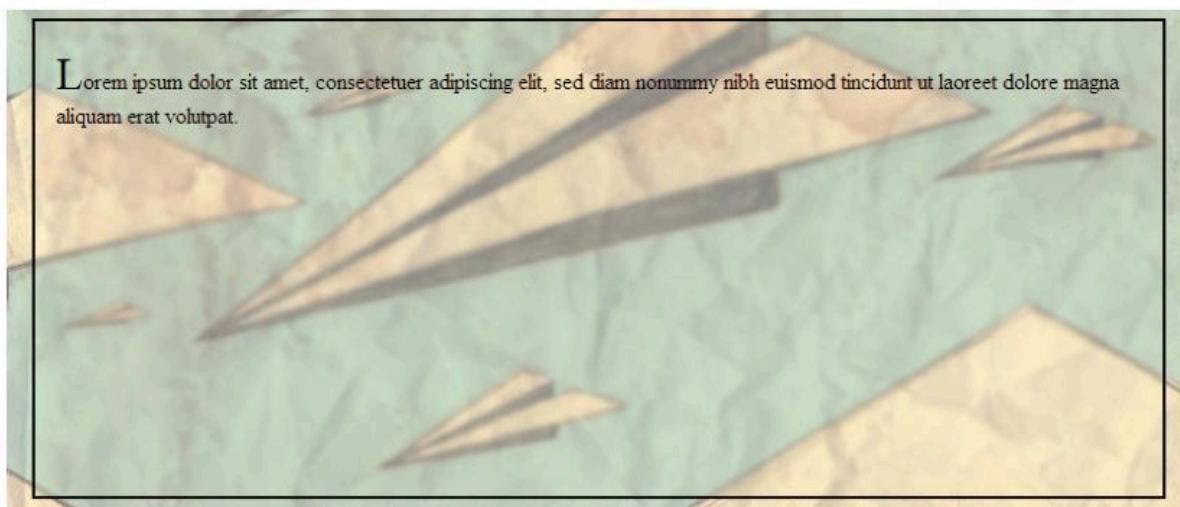


Tanto la primera división como la segunda se encuentran visibles continuamente a lo largo de todas las páginas web.

Llegamos a la tercera división del documento Inicio. Esta es la única división que se transforma a lo largo de las páginas, dependiendo de sitio en el que nos encontremos en la página. Tiene un borde con un grosor determinado y a diferencia del banner, no tiene las esquinas redondeadas.

La tercera división de la página del Inicio simplemente contiene un texto de bienvenida y explicativo sobre la página web en sí.

Como puedes observar, hemos introducido una letra capitular al principio del texto.



- Fotos.

El contenido del documento Fotos, contiene efectivamente trabajos fotográficos que se desean mostrar. Para este fin, hemos realizado una tabla básica, en la que hemos introducido miniaturas de las propias fotos. Al pulsar sobre éstas se abrirán en otra ventana las fotos en su tamaño original.

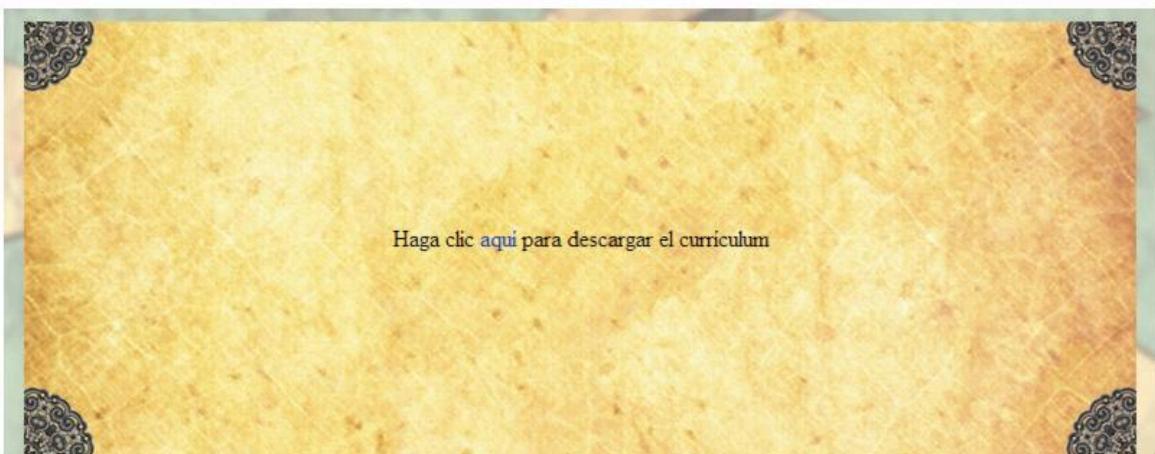
Las fotos tienen un borde con un color azul oscuro para que destaquen un poco sobre el fondo, y además nos sirve como pista para que el usuario las reconozca como enlace a un archivo distinto.



- Curículum.

El contenido del documento Curículum sólo contiene una imagen de fondo en la misma división, junto a un texto con un enlace para que el usuario pueda descargar un archivo en formato PDF. En este caso, se le ha aplicado un estilo al texto para que no tenga el subrayado típico de los enlaces de texto, pero se ha conservado el azul para que el usuario reconozca el enlace en sí.

Por otro lado, hemos aplicado un estilo para que una imagen se muestre en las esquinas, a modo de decoración.

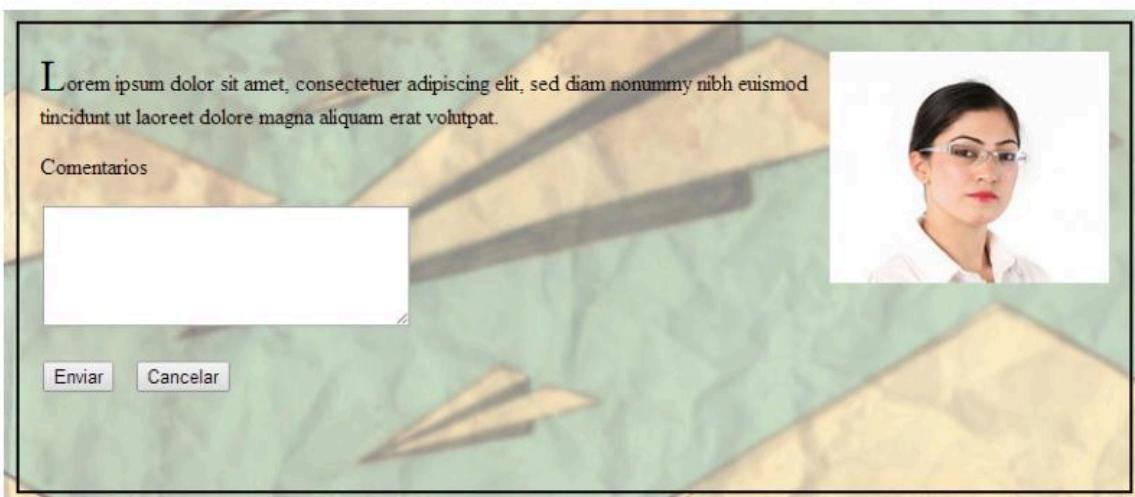


- Contacto.

Cuando entramos en el enlace Contacto del menú, observamos que el contenido consta de un texto en el que se pretende animar al usuario a que contacte con nosotros a través del formulario.

Hemos integrado una imagen de una foto personal a la derecha del texto.

Seguidamente, se encuentra el formulario, el cual se compone de un área de texto de una determinada extensión, junto con dos botones, uno para enviar el texto, y el botón cancelar para borrar el texto que haya podido introducir el usuario.



2.2. Diseño de los contenidos

El diseño que hemos aplicado en los contenidos del documento funcional es muy sencillo y fácil de hacer. Hemos elegido no recargar demasiado la página para que esta sea clara y sencilla.

A continuación, veremos una por una las páginas y ver así el diseño que se aplica en cada una de ellas.

Cuando comenzamos a diseñar la página web, lo primero que hicimos fue imaginar cual sería el aspecto del sitio. Para ello dedicamos un tiempo a esbozar con papel y lápiz en mano qué queríamos hacer y cómo lo queríamos.

Empezamos por un croquis, y cuando quedamos satisfechos con el resultado lo dimos por válido y empezamos a trabajar.

El trabajo final fue un sitio donde había un bloque principal que lo contendría todo, otro para el logotipo, banner o texto de título, otro para la barra de navegación y por último un bloque para el contenido de cada página.

Por eso, comenzaremos describiendo el diseño que hemos aplicado al div que lo contiene todo, después el título, seguido del menú y el contenido de cada página.

- Diseño de la página Inicio:

A la página de Inicio hemos aplicado estilos al cuerpo del documento. El diseño se trata de una imagen de fondo, posicionado al centro y sin repetición. El cuerpo tiene un ancho de 1000 píxeles y no tiene márgenes.

Además, hay una división que incluye todo el contenido en general, con el siguiente estilo: tiene una anchura de 800 píxeles, sin márgenes. El texto está alineado al centro, con un margen superior de 100 píxeles.

El color del borde es de color negro y las esquinas las tiene redondeadas. Por otro lado, la división tiene una imagen de fondo desenfocada que sólo se verá dentro de la división.

El diseño de la división del título (o logotipo/banner), es el siguiente: se trata de un fondo también desenfocado para que se vea dentro de la división desenfocada, y fuera de esta enfocada.

El título consta de un texto en un tamaño extra grande, del tipo Helvetica, con una anchura en negrita y de color negro. Además, se ha aplicado un espacio de 20 píxeles en su parte inferior y superior. La división del título tiene un borde de 2 píxeles visible, de color negro y con esquinas redondeadas.

Para el diseño del menú, hemos escogido un color de fondo amarillo (#f6f2b6), con un borde de 2 píxeles de ancho visible, con el color #4a1e20.

El texto está en versalitas, con un espacio de 10 píxeles en su parte superior e inferior. El texto está en negrita y tiene un espacio entre letras de 10 píxeles. Al ser el texto del menú los enlaces a las diferentes páginas web de las que se compone el sitio, aplicamos a éste un color (#4a1e20), y le quitamos el subrayado típico de los enlaces en internet.

Además, cuando se pasa el ratón sobre el enlace, éste cambia al color #510600 y una vez visitado, el color cambia a #3D4859.

Por último, el diseño de la división del contenido: se encuentra en una posición relativa, con un color de borde (#0D0802), de 2 píxeles de ancho y visible. La división tiene una altura mínima de 300 píxeles, y el texto de dentro se encuentra alineado a la izquierda, con un espacio de 20 píxeles en su parte superior, y 15 píxeles en el resto. La primera letra del texto la hemos puesto como letra capitular, por lo que se le aplica un diseño de tamaño de fuente extra grande.

A continuación, puedes ver el código que hemos utilizado para realizar todo esto a través de la hoja de estilos:

```
.todo {  
    width:800px;  
    margin:0 auto;  
    text-align: center;  
    margin-top: 100px;  
    border-color: black;  
    border-radius: 10px;  
    background-color: white;  
    background: url(..../images/avionesdesenfocadol.  
jpg);  
    background-repeat: no-repeat;  
    background-position: center;  
    background-attachment: fixed;  
}
```

```
.titulo {  
    font-size: xx-large;  
    font-family: Helvetica, Gill Sans, Lucida;  
    font-weight: bold;  
    color: black;  
    background-color: #white;
```



```
padding-top: 20px;  
padding-bottom: 20px;  
border-width: 2px;  
border-style: solid;  
border-color: black;  
background: url(..../images/avionesdesenfocadol.  
jpg);  
background-repeat: no-repeat;  
background-position: center;  
background-attachment: fixed;  
border: 2px solid #000000;  
border-radius: 10px;  
}  
-----  
.menu {
```

```
background-color: #f6f2b6;  
margin: 10px;  
border-style: solid;  
border-color: #4a1e20;  
border-width: 2px;  
padding-top: 10px;  
padding-bottom: 10px;  
font-weight: bold;  
font-variant: small-caps;  
word-spacing: 10px;  
}
```



```
.menu a {  
    color: #4a1e20;  
    text-decoration: none;  
    /*color para armonía #3D4859*/  
}  
  
.menu a:hover {  
    color: #510600;  
}  
  
.menu a:visited {  
    color: #3D4859;  
}  
  
-----  
.contenido {  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;
```



```
min-height: 300px;  
}
```

```
#capitular {  
    font-size: xx-large;  
}  
}
```

A continuación, explicaremos el diseño del resto de la página, donde sólo queda explicar el contenido en sí de cada uno de las páginas.

- Diseño de la página Fotos:

Para el diseño de la página Fotos, hemos optado por integrar una tabla con las imágenes de las fotos dentro.

El diseño de la tabla consta de un ancho de un 80%, sin borde, con márgenes automáticos a izquierda y derecha, y con una altura de 300 píxeles.

Las celdas de la tabla tienen un ancho de un 30%, con un margen a izquierda y derecha automático (es decir, el navegador calcula el ancho).

Las fotos tienen el siguiente diseño aplicado: tienen un borde con un ancho de 1 píxel, de color #3D4859 y visible.

A continuación, puedes ver el código que hemos insertado en la hoja de estilos para estos estilos:

```
table {  
    width: 80%;  
    margin-left: auto;  
    margin-right: auto;  
    height: 300px;  
    border-style: none;  
}
```



```
td {  
    width: 30%;  
    margin-left: auto;  
    margin-right: auto;  
}  
  
.fotos {  
    border-width: 1px;  
    border-color: #3D4859;  
    border-style: solid;  
}
```

Veamos ahora los diseños que hemos aplicado a las páginas de Currículum y Contacto.

- Diseño de la página Curriculum.

El diseño que hemos aplicado en la página Currículum es bastante sencillo, ya que sólo consta de una división de 300 píxeles de alto, con un texto alineado al centro tanto vertical como horizontalmente, y con 5 fondos, una hoja de pergamino en color amarillento que hace de fondo central, y unos dibujos para cada una de las esquinas.

El enlace del texto está sin decoración para evitar el subrayado típico azul que viene por defecto.

El código que hemos insertado en nuestra hoja de estilos es el siguiente:

```
#curriculum {  
    text-align: center;  
    min-height: 300px;  
    background: url(..../images/esquina-top-left.png)  
top left,  
                url(..../images/esquina-top-  
right.png) top right,
```



```
url(..../images/esquina-bottom-
left.png) bottom left,
url(..../images/esquina-bottom-
right.png) bottom right,
url(..../images/fondo-curricu-
lum.jpg) center no-repeat;
background-repeat: no-repeat;
background-color: #ffffff;
}
```

```
#curriculum div{
    display: table;
    min-height: 300px;
    width: 100%;
}
```

```
#curriculum div p {
    display: table-cell;
    vertical-align: middle;
}
```

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

- Diseño de la página Contacto.

El único diseño que se ha aplicado a la página de contacto es para la imagen que hemos integrado junto al texto, la cual se ha colocado a la derecha del mismo, y se le ha aplicado una anchura de 200 píxeles.

El resto del contenido de la página de Contacto no tiene estilos aplicados.

El diseño que se ha aplicado en la hoja de estilos para el contenido de la página de Contacto es el siguiente:

```
#fotocv {  
    float: right;  
    width: 200px;  
}
```

En la siguiente imagen puedes ver el diseño que tendrá la página web una vez acabada.



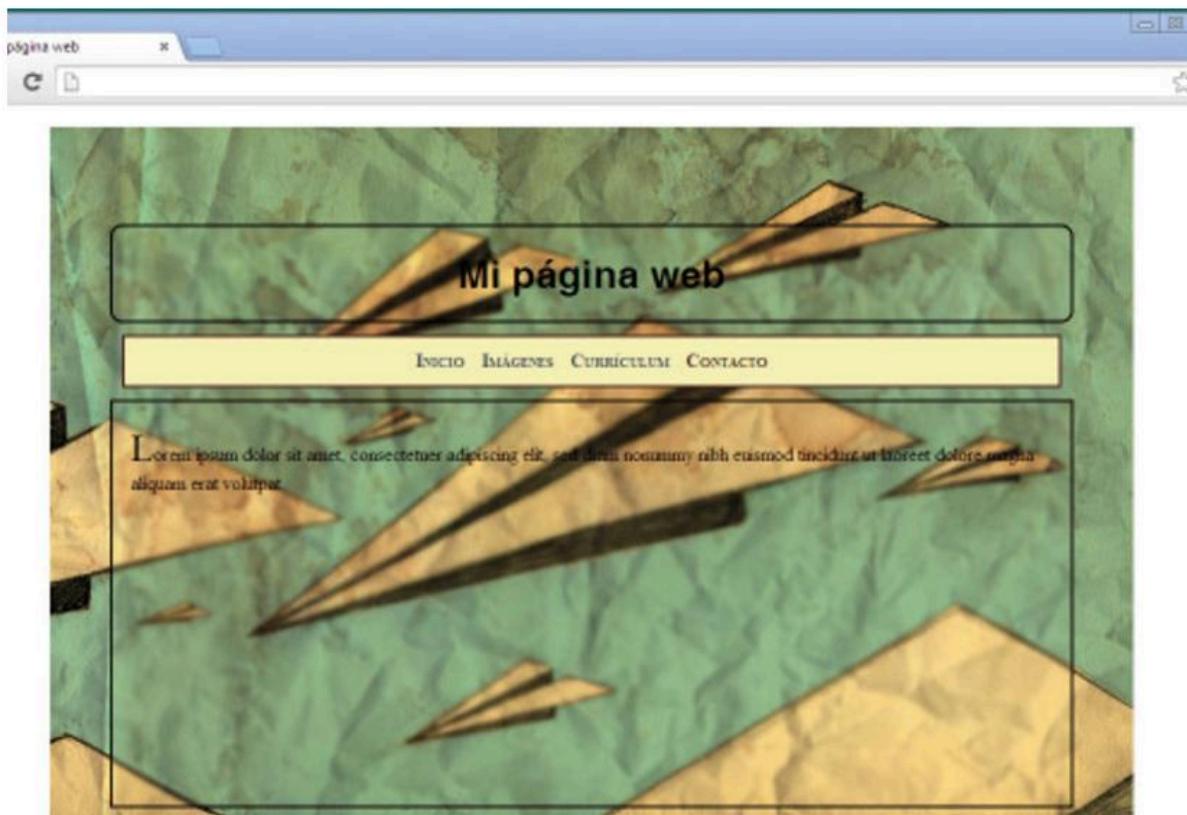
2.3. Identificación de la información a ubicar en la página web

En la página web que estamos poco a poco realizando se encuentran ubicados ciertos aspectos de la misma que ayudarán al usuario a obtener información sobre la misma.

Vamos a identificarlos de una forma más específica para así poder ubicarla e integrarla posteriormente en nuestro documento HTML.

- Inicio.

Veamos en la siguiente imagen una captura de la página Inicio para poder ubicar mejor los elementos de la misma:



Lo primero que nos llama la atención de la página es que tiene una imagen de fondo. Al ser una página web de un artista fotográfico, hemos decidido ubicar una imagen de fondo en vez de un color plano.

Esta imagen se ha seleccionado teniendo en cuenta los colores de la página y el sentido artístico de la misma. La imagen en sí tiene las dimensiones siguientes: 900x575 px, con una resolución de 96 ppp.

Seguidamente lo que podemos observar son las divisiones de la página. Éstas pueden reconocerse fácilmente ya que se ha procurado poner un grosor al borde.

Dentro de la primera división se encuentra la división que incluye el título de la página. En ella podemos incluir un banner con una imagen en movimiento, una imagen propia del autor estática, etc. Esta información será

muy importante para el usuario ya que servirá para saber en pocas palabras el tema principal de la página.

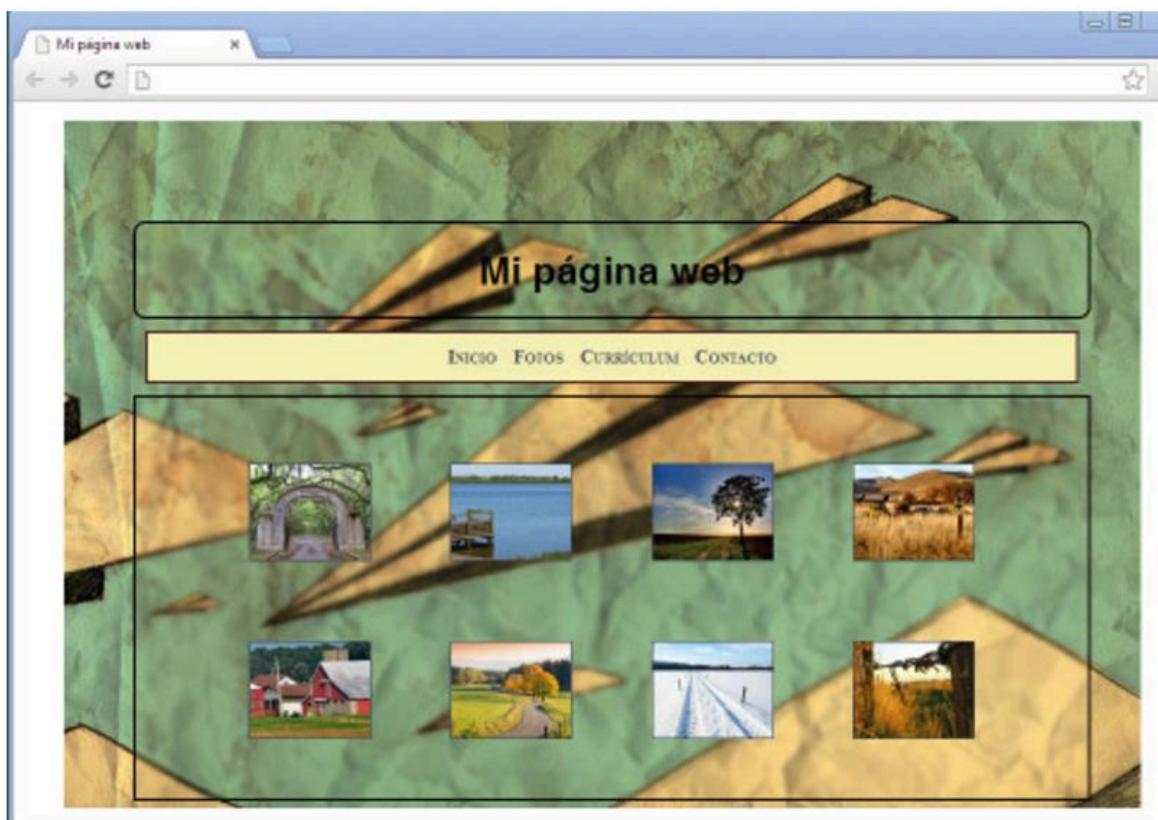
Seguidamente se encuentra el menú de la página, el cual se encuentra integrado por 4 botones o enlaces que nos llevan a los 4 documentos que conforman la página.

Por último, se encuentra la división en la que irá el contenido principal de la página Inicio, la cual tiene 300px de largo. En ella podemos incluir un texto de bienvenida para que el usuario conozca un poco más en profundidad aspectos sobre el autor, su trabajo, qué vamos a encontrar en ella, etc.

Como puedes observar, el texto no está redactado aún, ya que eso suele incluirse al final del trabajo. De momento tiene un texto de relleno.

- Fotos.

Veamos una imagen con la captura del documento Fotos para ver los elementos de los que se compone:



En la página Fotos, el elemento diferenciador es la división donde se encuentra el contenido. En esta podemos ver una tabla con diferentes imágenes.

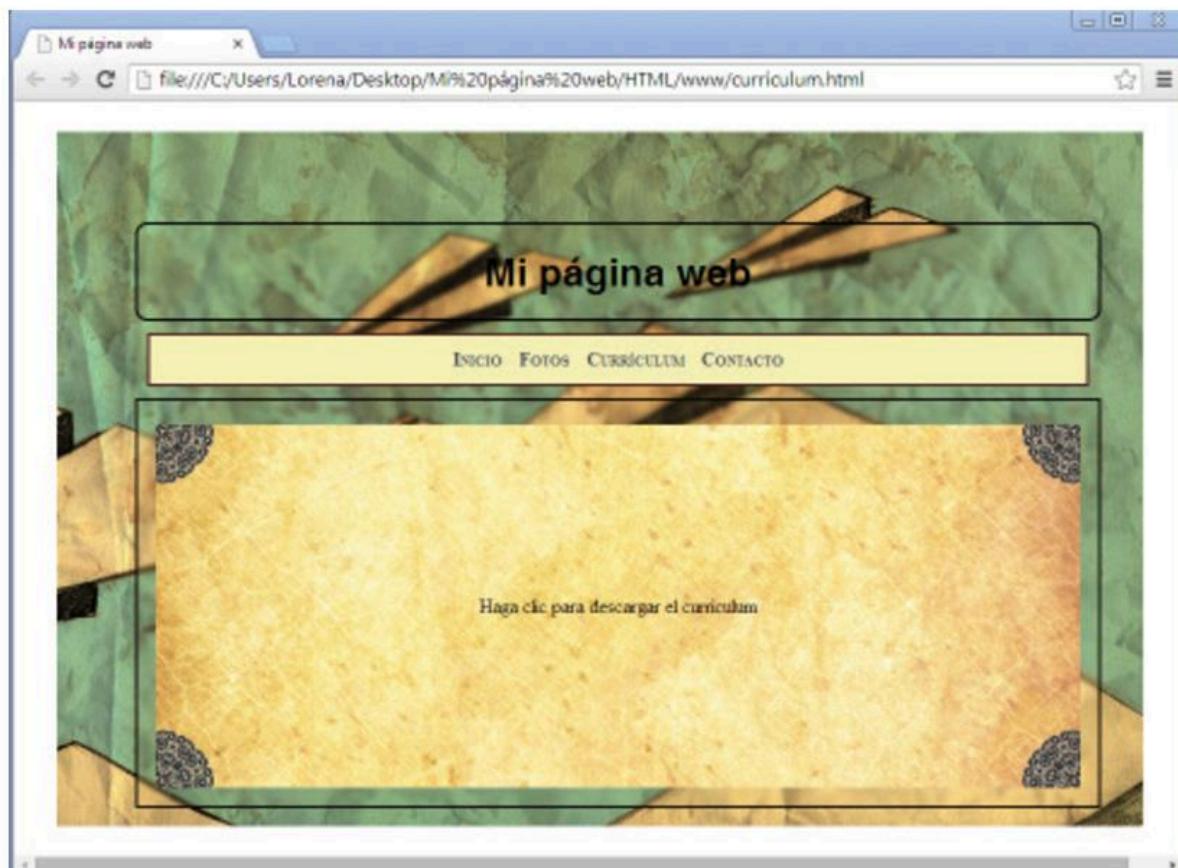
Para hacer este documento web, hemos tenido que buscar las imágenes en un banco de imágenes en Internet. Hay diversas páginas que se dedican a ello, por lo que no será difícil encontrar las imágenes.

Las imágenes las hemos editado en un editor de imágenes para redimensionarlas y poder hacer las miniaturas que van integradas en la tabla.

Las miniaturas de las imágenes tienen las dimensiones siguientes: 100x80 px, con una resolución de 96 ppp. Una vez redimensionadas las imágenes, estarán preparadas para poder mostrarlas en la página.

- Curriculum.

Veamos a continuación una imagen de la página Curriculum para poder ver todos sus elementos.



El elemento diferenciador de la página currículum a simple vista es una imagen de fondo, junto a 4 imágenes que decoran las 4 esquinas del contenido, y el texto con un enlace a un documento PDF.

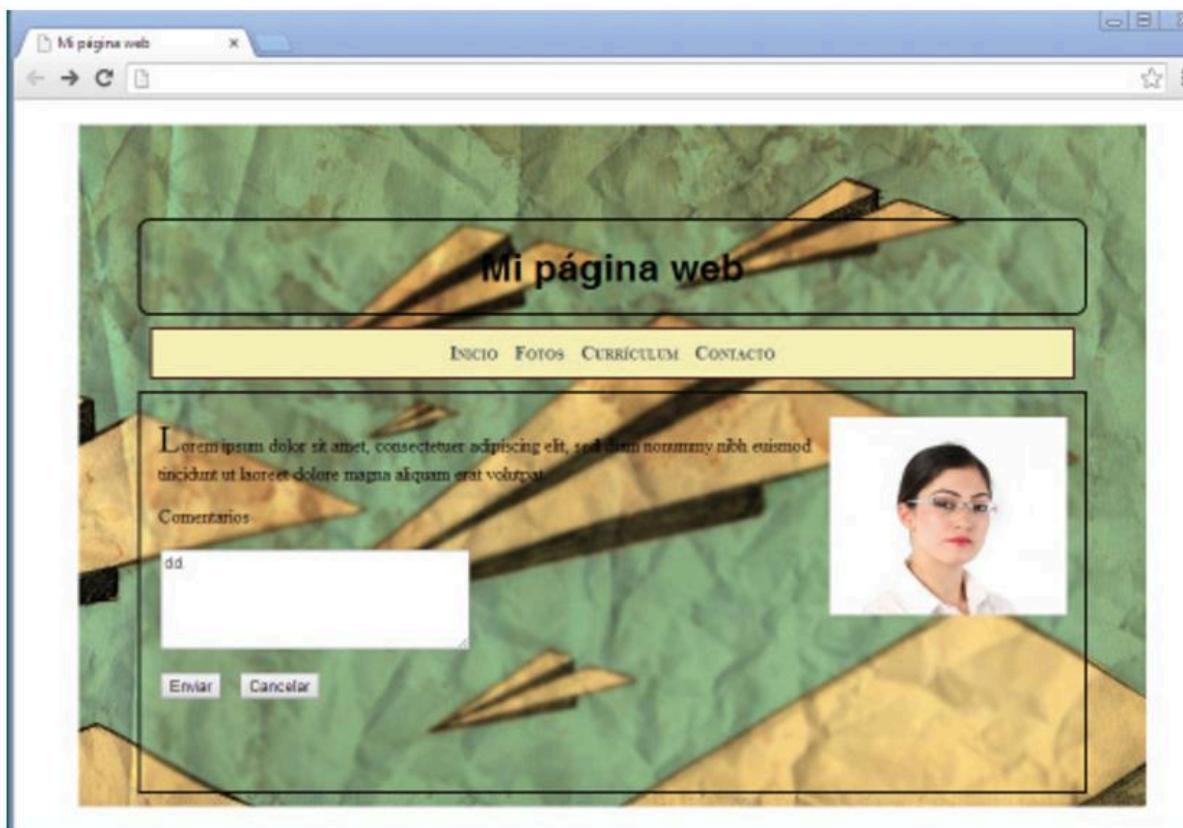
La imagen de fondo se ha integrado dentro de la división para dar un aspecto de papel. Esta imagen u otras parecidas puedes encontrarlas en los bancos de imágenes en Internet.

Las dimensiones de la imagen de fondo son: 800x1203 px, con una resolución de 96 ppp.

Las imágenes que decoran las esquinas, es la misma, la cual se ha editado en un editor de imágenes y se ha rotado hasta conseguir las 4 partes que necesitamos. Es una imagen con extensión .png para que respete las transparencias, y sus dimensiones son de 50x50 px.

- Contacto.

Veamos a continuación una imagen con la captura de la página Contacto.



Los elementos que encontramos en la división del contenido son un texto que invita a contactar con el autor, un formulario para enviar a un correo electrónico, y una imagen con una foto personal del autor.

El texto para invitar a contactar con el autor es un texto de relleno, el cual se cambiará al final del diseño de la misma.

El formulario se ha realizado en HTML, con un textarea de 5 líneas y 30 columnas, predefinidas en el código.

La imagen de la fotografía del autor se encuentra a la derecha, sin ningún estilo en concreto por ahora.

Veamos ahora los elementos de la información que estará ubicada en la página web identificados en el código HTML de la misma. Comencemos por el documento de inicio index.html:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Ilustraciones</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor sit
amet, consectetur adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat. </p>

</div>

</div>
```



```
</body>  
</HTML>
```

Este es el documento íntegro en formato HTML. Como puedes ver, ya se les han aplicado algunos estilos a las partes a las que hemos querido añadir estilos. Vayamos por partes para diferenciar mejor la información que hemos ubicado en la página de Inicio.

En primer lugar podemos observar que se encuentra el encabezado del documento:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>
```

En esta parte encontramos definidos el tipo de documento, el título de la página y el enlace a los estilos CSS.

En segundo lugar, encontramos el cuerpo del documento. Ahí encontramos la primera división que contiene toda la parte visible de la página web. A continuación, la división que pertenece a la primera división visible con el título de la página.

```
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>
```

En tercer lugar, vemos la división que pertenece al menú por donde podrá navegar el usuario.

```
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Ilustraciones</a>  
  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>
```

Y finalmente encontramos la división que pertenece al contenido del documento, junto con un estilo de clase id para la letra capitular del contenido. Seguidamente se cierra esta división y la división de "todo", y finaliza el documento HTML.

```
<div class="contenido">

<p><span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. </p>

</div>
</div>
</body>
</HTML>
```

A continuación, podremos ver la información que está ubicada en el código HTML del documento fotos.html de la página:

```
<!DOCTYPE html>

<HTML>

<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css" href="style/styles.css"/>
</head>
<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank"></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2" target="_blank"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3" target="_blank"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4" target="_blank"></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg" title="Imagen 5" target="_blank"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6" target="_blank"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7" target="_blank"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8" target="_blank"></a><br></td>

</tr>

</table>

</div>

</div>
```



```
</body>  
</HTML>
```

Veamos ahora por partes la información que va ubicada en este documento web en particular.

La primera parte ya la conocemos, se trata del comienzo del documento, junto a la división que abarca toda la parte visible del mismo, junto al banner o título, y el menú:

```
<!DOCTYPE html>  
<HTML>  
  <head>  
    <title>Mi página web</title>  
    <link rel="stylesheet" type="text/css" href="style/  
      styles.css"/>  
  </head>  
  <body>  
    <div class="todo">  
      <div class="titulo">Mi página web</div>  
      <div class="menu">  
        <a href="index.html">Inicio</a>  
        <a href="fotos.html">Fotos</a>  
        <a href="curriculum.html">Curriculum</a>  
        <a href="contacto.html">Contacto</a></div>  
    </div>
```

En la tercera división, que es donde integramos el contenido, podemos ver que consta de una tabla de 2 x 4 celdas. En cada una de las celdas va integrada una imagen pequeña denominada thumb-* en formato .jpg, de 100x80 px, al que se le ha integrado un title con el número de imagen y un borde de grosor 1.

Finalmente se finalizan los divs "todo" y "contenido" y se finaliza el documento HTML.

```
<div class="contenido">
```



```
<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank"></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2" target="_blank"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3" target="_blank"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4" target="_blank"></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg" title="Imagen 5" target="_blank"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6" target="_blank"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7" target="_blank"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8" target="_blank"></a><br></td>

</tr>

</table>

</div>

</div>

</body>

</HTML>
```


Veamos ahora el documento curriculum.html para poder ver la información que está ubicada en el mismo:

```
<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<div id="curriculum">

<div><p>Haga clic<a class="curriculum" href="documentos/
curriculum.pdf" target="_blank">aquí</a> para descargar
el currículum</p></div>

</div>

</div>

</div>

</body>

</HTML>
```

Como en los documentos web anteriores, la primera parte del documento HTML denominado curriculum.html consta de el encabezado de la página, la primera división que contiene toda la parte visible del documento, y las dos

primera divisiones que se integran dentro de ésta, que pertenecen al título o banner y a menú.

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>
```

En la división que contendrá el contenido de la página, se integra un texto con un enlace para descargar un documento PDF. Este contenido tiene sus estilos definidos en la hoja de estilos, y como puedes observar, tiene estilos de id para la división que pertenece a curriculum y un estilo de clases para el enlace. Seguidamente se finalizan el div de "todo" además de los divs de "contenido", de "curriculum" y de la división que hay dentro de "curriculum" para el párrafo. Se finaliza el documento HTML.

```
<div class="contenido">

<div id="curriculum">

<div><p>Haga clic <a href="documentos/curriculum.pdf"
target="_blank">aquí</a> para descargar el currículum</
p></div>

</div>

</div>
```



```
</div>  
</body>  
</HTML>
```

Ahora veremos los elementos de la información que se ubica en el documento contacto.html, para poder identificarlo en su documento web.

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<title>Mi página web</title>  
  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
  
</head>  
  
<body>  
  
<div class="todo">  
  
<div class="titulo">Mi página web</div>  
  
<div class="menu">  
  
<a href="index.html">Inicio</a>  
  
<a href="fotos.html">Fotos</a>  
  
<a href="curriculum.html">Curriculum</a>  
  
<a href="contacto.html">Contacto</a></div>  
  
<div class="contenido">  
  
<div id="fotocv">  
  
  
  
</div>  
  
<span id="capitular">L</span>orem ipsum dolor sit amet,  
consectetuer adipiscing elit,  
  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore  
magna aliquam erat volutpat.
```



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

<p>Comentarios</p>

<textarea rows="5" cols="30" name="comentarios"></texta-
rea>

<p><button type="submit">Enviar</button>&nbsp; &nbsp;
<button type="reset">Cancelar</button></p>

</form>

</div>

</div>

</body>

</HTML>
```

Al igual que el resto de los documentos web que componen la página web, éste también consta de las mismas partes en el principio: el encabezado de la página, y el cuerpo del documento con su división para todo lo visible, que consta de una primera división para el título o banner y de una segunda división para el menú de la página.

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>
```



```
<a href="curriculum.html">Curriculum</a>  
<a href="contacto.html">Contacto</a></div>
```

A continuación, el contenido de la página, que consta de una división denominada "fotocv" con una foto personal la cual tiene un estilo de clase id, un texto con una letra capitular con un estilo id, y un formulario para enviar por correo texto. Seguidamente se cierran los divs de "contenido" y de "todo" y finaliza el documento HTML.

```
<div class="contenido">  
  
<div id="fotocv">  
  
  
</div>  
  
<span id="capitular">L</span><p>orem ipsum dolor sit amet,  
consectetuer adipiscing elit,  
  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore  
magna aliquam erat volutpat.  
  
<form method="post" action="mailto:correo@extension.com"  
enctype="text/plain">  
  
<p>Comentarios</p>  
  
<textarea rows="5" cols="30" name="comentarios"></texta-  
rea>  
  
<p><button type="submit">Enviar</button>&nbsp;&nbsp;  
<button type="reset">Cancelar</button></p>  
  
</form>  
  
</div>  
  
</div>  
  
</body>  
  
</HTML>
```


2.4. Selección de contenidos para cada elemento de la página

Seleccionar los contenidos de una página no es tarea fácil. En ocasiones tenemos demasiada información entre la que elegir para mostrarla y no sabemos filtrarla para que la página no acabe siendo una imagen con demasiada información.

A continuación, vamos a especificar qué contenidos hemos puesto en nuestra página y porqué se han seleccionado.

- Inicio.

Veamos en el siguiente código cuál es el contenido que hemos seleccionado para el documento index.html:

```
<div class="contenido">  
  
<p><span id="capitular">L</span>orem ipsum dolor sit  
amet, consectetuer adipiscing elit,  
  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore  
magna aliquam erat volutpat. </p>  
  
</div>
```

Como puedes observar, el contenido que hemos preferido insertar en la página de Inicio es un texto de presentación. Para realizar la página en sí, hemos escogido un texto de relleno, el famoso Lorem ipsum. En internet podrás encontrar numerosas páginas donde se genera este texto.

En este momento, ha llegado la hora de introducir el contenido propio de la página. En este caso, deberemos redactar un texto de presentación. Este texto será lo primero que el usuario vea al entrar en nuestra página, por lo que debe ser atractivo y fácil de leer.

Para más información, consulta algunos consejos para desarrollar contenidos en la web en los anexos al final del libro.



Por lo que un posible contenido para la página de Inicio podría ser algo así:

```
<div class="contenido">  
  
<p><span id="capitular">B</span>ienvenido a mi página  
web.  
  
Soy un fotógrafo profesional al que le gusta hacer fotos  
de la  
  
naturaleza a través de los viajes que realizo.</p>  
  
<p>Aquí puedes ver  
  
una muestra de los mismos. Sólo tienes que pulsar en el  
menú Fotos.</p>  
  
<p> Espero que te gusten.</p>  
  
</div>
```

– Fotos.

En el documento fotos.html hemos seleccionado como contenido una muestra de varias imágenes del trabajo del autor. Para que las imágenes estén ordenadas, las hemos colocado en una tabla. De esta forma la información queda mejor estructurada.

La sintaxis que se ha usado para el contenido de fotos.html es el siguiente:

```
<div class="contenido">

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank"></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2" target="_blank"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3" target="_blank"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4" target="_blank"></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg" title="Imagen 5" target="_blank"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6" target="_blank"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7" target="_blank"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8" target="_blank"></a><br></td>

</tr>

</table>

</div>
```

Para realizar la página ficticia, podemos buscar imágenes libres de derechos en numerosas páginas de internet. Una vez que las descarguemos,

tendremos que editarlas en algún editor de imágenes para redimensionarlas sin que pierdan calidad, y colocarlas en la tabla.

No debemos olvidar poner el title a cada una de las imágenes. Para una mejor navegabilidad del usuario, hemos definido que la imagen se abra en una ventana distinta, usando el elemento target, con un valor _blank. Así evitamos que el usuario siga permaneciendo en la página cuando haga clic en las imágenes.

Veamos uno de los <td> de la tabla para poder visualizar mejor el contenido que se ha seleccionado, además de las características de cada uno de ellos:

```
<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank"></a></td>
```

Como puedes observar, los elementos que se han usado han sido los siguientes:

- A href: para enlazar a la ruta de la imagen.
- Title: para que exista un título cuando se pase el ratón sobre la imagen.
- Target: para que el vínculo se abra en una ventana nueva.
- Img src: la ruta a la imagen en sí.
- Class: el nombre del estilo de clase que dará definición a la misma.
- Width: ancho de la imagen.
- Height: alto de la imagen.
- Border: borde la imagen.
- Currículum.

En el apartado Currículum simplemente hemos integrado un texto con un enlace a un documento. Hemos seleccionado este tipo de contenido porque es el más sencillo y directo.

La sintaxis que hemos utilizado para el contenido de la página curriculum.html es la siguiente:

```
<div class="contenido">  
  <div id="curriculum">  
    <div><p>Haga clic <a href="documentos/curriculum.pdf" target="_blank">aquí</a> para descargar el currículum</p></div>  
  </div>  
</div>
```

La página curriculum.html es muy básica. Simplemente se trata de un texto corto donde podemos encontrar un enlace al currículum del autor en formato PDF.

Los elementos que se han utilizado en la sintaxis son:

- a href: para enlazar al documento pdf.
- Target: con el valor _blank para que se abra el enlace en una nueva ventana.

En este tipo de páginas, también podemos encontrar los currículums completos en pantalla, para que el usuario pueda visualizarlo, pero no suele quedar muy estético, y no se recomienda por la protección de datos. Al poner un enlace, nos aseguramos de que sólo clicará la persona que esté interesada en verlo.

Se recomienda en estos casos, no poner en el currículum datos personales.

- Contacto.

En el documento contacto.html hemos seleccionado elementos como un pequeño texto, un formulario y una foto personal. La sintaxis del documento es la siguiente:

```
<div class="contenido">  
  <div id="fotocv">  
      
  </div>
```



```
<span id="capitular">L</span>orem ipsum dolor sit amet,  
consectetuer adipiscing elit,  
  
sed diam nonummy nibh euismod tincidunt ut laoreet dolore  
magna aliquam erat volutpat.  
  
<form method="post" action="mailto:correo@extension.com"  
enctype="text/plain">  
  
<p>Comentarios</p>  
  
<textarea rows="5" cols="30" name="comentarios"></texta-  
rea>  
  
<p><button type="submit">Enviar</button>&nbsp; &nbsp;  
<button type="reset">Cancelar</button></p>  
  
</form>  
  
</div>
```

Como puedes observar, en primer lugar, hemos realizado una división para introducir la imagen, seguidamente hemos introducido el texto, que quedaría a la izquierda de la imagen, y justo después va integrado el formulario de contacto.

Estos elementos suelen ser los que se utilizan para las páginas de contacto. El formulario es un elemento muy recurrente, ya que nos permite recibir información de nuestras visitas, sin tener que exponer en la página teléfonos o correos personales.

Como puedes observar, hemos introducido en el texto un texto auxiliar. Vamos a cambiarlo y veamos como quedaría finalmente.

```
<span id="capitular">G</span>racias por tu visita.  
  
Puedes ponerte en contacto conmigo a través del  
formulario y contarme qué te parecen mis trabajos,  
o preguntarme las dudas que tengas.
```


2.5. Utilización del documento funcional para las especificaciones del diseño

Como hemos visto anteriormente, hemos utilizado un documento base HTML para aplicarle los estilos pertinentes y obtener un resultado distinto como página Web.

A continuación, vamos a explicar cómo hemos utilizado el documento funcional y cómo se le han aplicado las especificaciones de estilo que hemos estado explicando.

Comencemos por el documento index.html.

- Inicio.

Recordemos cual era el documento inicial en HTML:

```
<!DOCTYPE html>

<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

<span>L</span>orem ipsum dolor sit amet,
```



```
consectetuer adipiscing elit,  
sed diam nonummy nibh euismod  
tincidunt ut laoreet dolore  
magna aliquam erat volutpat.  
</div>  
</div>  
</body>  
</HTML>
```

Simplemente algunas divisiones y nada más.

Veamos qué estilos hemos ido aplicando a los diferentes elementos del documento index.html:

Para ello, veamos cómo queda el documento, una vez que se le han especificado nombre de los estilos para ser aplicados desde nuestra hoja de estilos. Así podremos comparar un documento con el otro:

```
<!DOCTYPE html>  
<HTML>  
<head>  
<title>Mi página web</title>  
<link rel="stylesheet" type="text/css" href="style/  
styles.css"/>  
</head>  
<body>  
<div class="todo">  
<div class="titulo">Mi página web</div>  
<div class="menu">  
<a href="index.html">Inicio</a>  
<a href="fotos.html">Fotos</a>  
<a href="curriculum.html">Curriculum</a>
```



```
<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p></p><span id="capitular">B</span>ienvienido a mi página
web .

Soy un fotógrafo profesional al que le gusta hacer fotos
de la naturaleza a través de los viajes que realizo.</p>

<p>Aquí puedes ver una muestra de los mismos. Sólo tienes
que pulsar en el menú Fotos.</p>

<p> Espero que te gusten.</p>

</div>

</div>

</body>

</HTML>
```

Como puedes ver, el primer elemento al que se le ha aplicado un estilo es a la división "todo". Sin embargo, en nuestra hoja de estilos, hemos aplicado al elemento `<body>` algunos estilos para comenzar a formar un estilo en la página.

Los estilos que hemos introducido son los siguientes:

```
body {

    background-color: white;
    background-image: url(..../images/aviones1.jpg);
    background-position: center;
    background-repeat: no-repeat;
    background-attachment: fixed;
    width:1000px;
    margin:0 auto;
}
```

- `Background-color`: con este estilo cambiamos el color de fondo de la página. Si bien la página tiene un fondo con una imagen, se recomienda colocar un color de neutro, para que éste se reproduzca en caso de que la página cargue muy lento. De esta forma, se irá viendo el color, y el usuario sabe que comenzará a cargarse la página.

- Background-image: Ahora sí, ponemos el fondo con la imagen que queramos introducir.
- Background-position: Posicionamos el documento centrado respecto a la ventada del navegador.
- Background-repeat: indicamos al estilo que no se repita la imagen que hemos insertado.
- Background-attachment: indicamos al estilo que no se mueva la imagen en caso de que exista barra de navegación.
- Width: Indicamos el ancho de la página.
- Margin: Indicamos el margen de la página.

Veamos ahora los estilos que se han añadido al primer elemento que encontramos en el documento HTML, es decir, al div con un estilo de clase denominado "todo".

```
.todo {  
    width:800px;  
    margin:0 auto;  
    text-align: center;  
    margin-top: 100px;  
    border-color: black;  
    border-radius: 10px;  
    background-color: white;  
    background: url(..../images/avionesdesenfocadol.  
jpg);  
    background-repeat: no-repeat;  
    background-position: center;  
    background-attachment: fixed;  
}
```

Los estilos que hemos aplicado a la división que contiene todo los elementos que se verán en la página son los siguientes:

- Width: para especificar el ancho de la división.
- Margin: especificamos el margen de la división.
- Text-align: para indicar que el texto quede alineado al centro.
- Margin-top: indicamos un número de píxeles del margen superior.
- Border-color: indicamos un color para el borde.
- Border-radius: indicamos un número de píxeles para redondear las esquinas del borde. Este estilo es nuevo, pertenece a CSS 3. Facilita mucho las cosas, ya que antes era mucho más complicado redondear las esquinas de un elemento, pero gracias a este estilo podemos hacerlo de una forma muy fácil, simple y rápida.
- Background-color: Al igual que antes, especificamos un color de fondo, por si no se carga la imagen de fondo como debería.

Hasta aquí todo normal. Pero hemos visto en el estilo de la página que existe una imagen de fondo normal, y sin embargo dentro del div se ve desenfocada. Para hacer esto, lo haremos de la siguiente forma:

De entrada, necesitamos 2 imágenes iguales, aunque una de ellas la editaremos para que esté desenfocada. A continuación, sólo habrá que seguir los pasos siguientes:

Tendremos que designar una imagen (la enfocada) al body y otra al fondo de un div que contenga la mayor parte del contenido de la página.

El próximo paso es especificar la situación de la imagen en el mismo sitio, tanto en el body como en el div. En este caso, se trata de:

```
background-repeat: no-repeat;  
background-position: center;
```

Por otro lado, en el div llamaremos a la imagen, esta vez, la desenfocada:

```
background: url(..../images/avionesdesenfocado1.jpg);
```

Y ahora la clave de todo: el estilo background-attachment, con valor fixed:

```
background-attachment: fixed;
```


De esta forma, conseguimos un efecto muy vistoso, ya que de fondo tenemos la imagen normal, y al traspasar el div, parezca que se distorsiona y difumine. Un efecto visual muy bonito y creativo, a la vez que fácil y muy rápido de conseguir.

Para conseguir otros efectos, también podemos usar varios fondos repetidos. Prueba a hacerlos con capas diferentes en tonos o transparencias.

A continuación, vamos a explicar en profundidad las esquinas redondeadas con CSS 3. Son muy fáciles de hacer, en especial una vez que nos hayamos familiarizado con las dimensiones de las curvas.

Una de las ventajas de utilizar esta técnica es que se controlan sólo con CSS, no necesitan ningún marcado adicional, y no se requiere ningún fondo de color que rodee al elemento redondeado como con la técnica antigua.

La desventaja es que tienen un soporte limitado. Actualmente no hay ninguna versión de IE compatible, aunque parece que así será para IE9.

Para crear el efecto esquina redondeada sólo habrá que utilizar el código siguiente:

```
border-radius: 20px;
```

Pero imaginemos que queremos redondear las esquinas en forma de óvalo. Para ello, deberemos usar dos valores. Por ejemplo:

```
border-radius: 20px / 60px;
```

De esta forma, los arcos serán elípticos y no circulares, gracias al resultado de dos valores separados por barras diagonales. Si los omitimos, las esquinas tendrían tamaños diferentes, pero cada una usará un arco circular. En el caso de eliminar las barras, las esquinas tendrán un radio desigual.

Deberemos tener en cuenta que algunos navegadores son incompatibles aun con esta técnica. Estos son Safari/Chrome y Firefox, a los que deberemos añadir un prefijo específico al código, además de dejar la versión sin prefijo para que se mantenga cuando los navegadores sean compatibles:

```
-moz-border-radius: 20px;
```

```
-webkit-border-radius: 20px;
```

```
border-radius: 20px;
```


Sin embargo, estos navegadores admiten propiedades que permiten establecer cada esquina de forma individual. Estas son prácticas en caso de utilizar Webkit, porque aunque no soporta valores como 20 px 60 px, sí acepta propiedades de esquina individuales:

```
-webkit-border-radius: 20px  
-webkit-border-top-right-radius: 60px;  
-webkit-border-bottom-left-radius: 60px;  
-moz-border-radius: 20px 60px  
  
Border-radius: 20px 60px;
```

El siguiente estilo que se aplica al documento index.html es el estilo de clase "titulo", el cual define los estilos de la primera división que puede visualizar el usuario de la página, donde hemos añadido un texto que simplemente reza: "Mi página web".

Los estilos que hemos definido para esta división son los siguientes:

```
.titulo {  
    font-size: xx-large;  
    font-family: Helvetica, Gill Sans, Lucida;  
    font-weight: bold;  
    color: #black;  
    background-color: white;  
    padding-top: 20px;  
    padding-bottom: 20px;  
    border-width: 2px;  
    border-style: solid;  
    border-color: black;  
    background: url(..../images/avionesdesenfocadol.  
jpg);  
    background-repeat: no-repeat;  
    background-position: center;
```



```
background-attachment: fixed;  
/*border: 2px solid #000000;*/  
border-radius: 10px;  
}
```

Veamos uno a uno cuáles son los estilos que hemos aplicado a esta división:

- Font-size: para añadir un tamaño de fuente al texto.
- Font-family: para determinar de qué familia debe ser el texto.
- Font-weight: para añadir un grosor al texto.
- Color: para dar color al texto.
- Background-color: añadimos un color de fondo por si no carga la página.
- Padding-top y padding-bottom: para determinar la anchura de relleno del elemento, en sus parte superior e inferior.
- Border-width: añadir una anchura de borde.
- Border-style: aplicar un estilo al borde.
- Border-color: aplicar un color al borde.
- A continuación añadimos los elementos que hemos explicado antes para aplicar el efecto de imagen desenfocada en el div (background, background-repeat y background-position).
- A continuación, en un comentario podemos ver cómo podemos aplicar en una sola línea los estilos al borde de esta forma: /*border: 2px solid #000000;*/.
- Border-radius: para redondear las esquinas del borde.

A continuación, las especificaciones de diseño que se han definido en las hojas de estilo para el documento de base index.html es para la división del menú de la página. Veamos cuales han sido los estilos que hemos especificado:


```
.menu {  
    background-color: #f6f2b6;  
    margin: 10px;  
    border-style: solid;  
    border-color: #4a1e20;  
    border-width: 2px;  
    padding-top: 10px;  
    padding-bottom: 10px;  
    font-weight: bold;  
    font-variant: small-caps;  
    word-spacing: 10px;  
}
```

Veamos uno por uno los estilos que hemos aplicado al menú del documento funcional:

- Background-color: Hemos especificado un color plano como fondo del menú.
- Margin: hemos especificado un número específicos para el margen.
- Border-style: Añadimos un estilo al borde de la división.
- Border-color: añadimos un color al borde de la división.
- Border-width: aplicamos un grosor al borde de la división.
- Padding-top: determinamos la anchura del relleno del elemento de la parte superior.
- Padding-bottom: determinamos la anchura del relleno del elemento de la parte inferior.
- Font-weight: añadimos una anchura al texto.
- Font-variant: para añadir un variante del tipo de letra. En este caso, el valor small-caps, que define la letra como versalitas.

- Word-spacing: para dar espacio con un número de píxeles entre cada una de las letras.

Por otro lado, tenemos especificaciones de estilo a lo que son los enlaces del menú. Esto lo hemos especificado en nuestra hoja de estilos de la siguiente forma:

```
.menu a {  
    color: #4a1e20;  
    text-decoration: none;  
    /*color para armonía #3D4859*/  
}  
  
.menu a:hover {  
    color: #510600;  
}  
  
.menu a:visited {  
    color: #3D4859;  
}
```

Hemos aplicado los siguientes estilos a los enlaces del menú con el código siguiente:

- .menu a: aplicamos al menú un color específico al texto y especificamos que no tenga ninguna decoración para evitar el típico subrayado de los enlaces.
- .menu a:hover: aplicamos al menú un color específico para cuando se pase el cursor por encima del enlace.
- .menu a.visited: aplicamos al menú un color específico para cuando se visite el enlace.
- Como puedes ver, se ha comentado en la hoja de estilos un color que queremos tener en cuenta para la armonía de la página.

Otro de los estilos que encontramos en el documento index.html es el de la división del contenido. En concreto, para la página de Inicio, como bien sabemos, hemos integrado sólo un texto. Pero este también tiene un

estilo añadido en nuestra hoja de estilos. Veamos pues cuál es el código que hemos utilizado para esto:

```
.contenido {  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: #0D0802;  
    border-width: 2px;  
    text-align: left;  
    min-height: 300px;  
}
```

Veamos uno por uno los diferentes estilos que se han utilizado:

- Position: para especificar la posición del contenido dentro del div.
- Padding-top: determina la anchura del relleno del elemento de la parte superior.
- Padding-right: determina la anchura del relleno del elemento de la parte derecha.
- Padding-bottom: determina la anchura del relleno del elemento de la parte inferior.
- Padding-left: determina la anchura del relleno del elemento de la parte izquierda.
- Border-style: añadimos un estilo al borde de la división.
- Border-color: agregamos un color al borde.
- Border-width: determina que anchura tendrá el borde del div.

- Text-align: asignamos una alineación al texto, en este caso, izquierdo.
- Min-height: determinamos una altura mínima al div, en este caso, de 300 píxeles.

Estos son hasta el momento los estilos que se han aplicado en la página de Inicio del documento funcional. Una vez especificados los estilos en la misma, podemos ver el resultado y ver cómo van cambiando los estilos.

Los estilos de los siguientes documentos web que forman la página son iguales en sus partes comunes.

Las partes comunes que utiliza el documento funcional con los demás documentos web son:

- Body
- Estilos para la división "todo".
- Estilos para la división "título".
- Estilos para la división "menú".
- Estilos para la división "contenido".

El último de los estilos que encontramos en el documento de inicio es el de la letra capitular del párrafo. En la hoja de estilos el código es el siguiente:

```
#capitular {
    font-size: xx-large;
}
```

Con este sencillo estilo, podemos hacer que la principal letra tenga un valor extra grande, gracias al elemento font-size.

Para los estilos de los demás documentos web, cada uno tendrá sus diferencias dependiendo del contenido que tenga. Los veremos a continuación.

- Fotos.

El documento funcional que hemos utilizado para la página Fotos, es el siguiente:

```
<!DOCTYPE html>
```



```
<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1"></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4"></a></td>

</tr>

<tr>
```



```
<td><a href="images/image-5.jpg" title="Imagen 5"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8"></a><br></td>

</tr>

</table>

</div>

</div>

</body>

</HTML>
```

Como puedes ver, no tiene ningún estilo aplicado. Veamos ahora la diferencia del mismo documento con los estilos aplicados:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>
```



```
<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<table>

<tr>

<td><a href="images/image-1.jpg" title="Imagen 1" target="_blank"></a></td>

<td><a href="images/image-2.jpg" title="Imagen 2" target="_blank"></a></td>

<td><a href="images/image-3.jpg" title="Imagen 3" target="_blank"></a></td>

<td><a href="images/image-4.jpg" title="Imagen 4" target="_blank"></a></td>

</tr>

<tr>

<td><a href="images/image-5.jpg" title="Imagen 5" target="_blank"></a></td>

<td><a href="images/image-6.jpg" title="Imagen 6" target="_blank"></a></td>

<td><a href="images/image-7.jpg" title="Imagen 7" target="_blank"></a></td>

<td><a href="images/image-8.jpg" title="Imagen 8" target="nueva"></a><br></td>

</tr>

</table>

</div>
```



```
</div>  
  
</body>  
  
</HTML>
```

Como puedes ver, los estilos que se han aplicado específicamente a este documento son los comunes a toda la página. Sin embargo, a pesar de no parecer que hay algún estilo más, si que se ha aplicado un estilo, en concreto, a la tabla del documento.

En nuestra hoja de estilos, podemos ver cuál es el estilo que hemos aplicado a la tabla:

```
table {  
    width: 80%;  
    margin-left: auto;  
    margin-right: auto;  
    height: 300px;  
    border-style: none;  
}
```

Veamos los estilos que hemos añadido a la tabla en particular:

- Width: para establecer un ancho de la tabla.
- Margin-left: para establecer un margen a su izquierda.
- Margin-right: para establecer un margen a su derecha.
- Height: para establecer una altura en píxeles.
- Border-style: para establecer un borde, en este caso ninguno, es decir, con valor none.

Por otro lado, hemos establecido una serie de estilos para las celdas. En este caso, el código CSS que hemos insertado es el siguiente:

```
td {  
    width: 30%;  
    margin-left: auto;
```



```
    margin-right: auto;  
}
```

Los estilos que hemos añadido a las celdas de la tabla son los siguientes:

- Width: hemos añadido un ancho a la celda.
- Margin-left: hemos aplicado un margen a su izquierda.
- Margin-right: hemos aplicado un margen a su derecha.

Por otro lado, hemos aplicado estilos a las imágenes en sí que van insertadas dentro de cada celda. Los estilos son los siguientes:

```
.fotos {  
  
    border-width: 1px;  
  
    border-color: #3D4859;  
  
    border-style: solid;  
  
}
```

Los estilos que se han aplicado a las imágenes son:

- Border-width: se ha aplicado un borde de 1 píxel de grosor de ancho.
 - Border-color: se ha aplicado un color determinado para que el usuario sepa que es un enlace y puede clicar sobre la imagen.
 - Border-style: se ha añadido un estilo al borde para que este sea visible en la página.
- Currículum.

El documento funcional que hemos utilizado para la página Currículum era el siguiente:

```
<!DOCTYPE html>  
  
<HTML>  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html;  
charset=iso-8859-1" />
```



```
<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>

<a href="contactosinestilos.html">Contacto</a></div>

<div>

    <div><p>Haga clic aquí para descargar el currículum</p></div>

</div>

</div>

</body>

</HTML>
```

Sin embargo, una vez que hemos aplicado estilos al documento funcional, este quedaría de la siguiente forma:

```
<!DOCTYPE html>

<HTML>

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">
```



```
<div class="titulo">Mi página web</div>

<div class="menu">
  <a href="index.html">Inicio</a>
  <a href="fotos.html">Fotos</a>
  <a href="curriculum.html">Curriculum</a>
  <a href="contacto.html">Contacto</a></div>

<div class="contenido">
  <div id="curriculum">
    <div><p>Haga clic <a href="documentos/curriculum.pdf" target="_blank">aquí</a> para descargar el currículum</p></div>
  </div>
</div>
</body>
</HTML>
```

Los elementos que hemos utilizado para dar formato al contenido específico del documento funcional curriculum.html los podemos ver en la hoja de estilos. Son los siguientes:

```
#curriculum {
  text-align: center;
  min-height: 300px;
  background: url(..../images/esquina-top-left.png) top left,
              url(..../images/esquina-top-right.png) top right,
              url(..../images/esquina-bottom-left.png) bottom left,
              url(..../images/esquina-bottom-right.png) bottom right,
              url(..../images/fondo-curriculum.jpg) center no-repeat;
  background-repeat: no-repeat;
```



```
background-color: #ffffff;  
}
```

Los estilos que se han aplicado en el contenido de este documento en concreto son:

- Text-align: añadimos un alineado centrado al texto.
- Min-height: aplicamos una altura mínima al div.
- Background-color: aplicamos un color de fondo neutro por si tarda en cargar el fondo.

Como puedes ver, hemos añadido al CSS los estilos siguientes:

```
background: url(..../images/esquina-top-left.png) top left,  
url(..../images/esquina-top-right.png) top right,  
url(..../images/esquina-bottom-left.png) bottom left,  
url(..../images/esquina-bottom-right.png) bottom right,  
url(..../images/fondo-curriculum.jpg) center no-repeat;  
  
background-repeat: no-repeat;
```

Bien, esta técnica se utiliza para aplicar múltiples fondos a nuestra página. Es una técnica nueva de CSS 3. Se trata de una novedad muy ingeniosa, la cual soporta múltiples imágenes de fondo en un elemento determinado.

En ocasiones los diseñadores han tenido que anidar múltiples elementos div para conseguir un conjunto de decoraciones de fondo. Sin embargo, con esta técnica se puede evitar ese tedioso trabajo.

Lo único que debemos hacer para repetir un fondo en todas las esquinas, es añadir cada una de las imágenes. Simplemente habrá que separar cada uno de los valores background para conseguir múltiples fondos.

Solo deberemos tener en cuenta una cosa: dentro de los valores, hemos integrado un fondo blanco liso. Este estilo deberemos ponerlo siempre en el último de los valores, ya que, si lo colocamos en el lugar equivocado, corremos el riesgo de que reemplace a una o más imágenes.

Esto ocurre porque el fondo múltiple va desde el valor que está "más arriba" hasta los "más bajo". Por eso, es mejor aplicar la propiedad aparte.

Por otro lado, con el estilo no-repeat, ninguna de las imágenes se repetirá, ya que este estilo se aplica a todos los fondos asignados.

Por otro lado, existen otros estilos que hemos aplicado también a la división del contenido del documento funcional curriculum.html. Son los siguientes:

```
#curriculum div{  
    display: table;  
    min-height: 300px;  
    width: 100%;  
}  
  
#curriculum div p {  
    display: table-cell;  
    vertical-align: middle;  
}  
  
#curriculum a {  
    text-decoration: none;  
}
```

Como puedes ver, hemos aplicado un estilo a la división que existe dentro del div curriculum, otro al párrafo, y otro al enlace.

La división que existe dentro del div curriculum tiene los siguientes estilos:

- Display: hemos utilizado este estilo para controlar mejor el texto y poder tratarlo como si se tratara de una tabla, para posteriormente centrarlo dentro del div.
- Min-height: para añadir una altura mínima al div.
- Width: hemos aplicado un ancho al div.

Los estilos que hemos añadido al párrafo del div son:

- Display: para controlar el texto como si estuviera dentro de una celda y controlar mejor su posición dentro del div.

- Vertical-align: para alinearlo verticalmente, en este caso, en el centro del div.

El estilo que hemos aplicado al enlace del párrafo son:

- Text-decoration: con el valor none, para que el texto no tenga ese antiestético subrayado típico de los enlaces.

De esta forma, hemos controlado el texto del div del contenido del documento funcional curriculum.html.

En el caso de que la página hubiera tenido valores absolutos y no relativos como esta, el centrado del texto hubiera sido más fácil, pero en este caso hemos optado por valores relativos.

- Contacto.

Hemos utilizado el siguiente documento funcional para el documento web contacto.html:

Sin embargo, una vez aplicados los estilos, el document HTML quedaría de la siguiente forma:

```
<!DOCTYPE html>

<HTML>

<head>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<title>Mi página web</title>

</head>

<body>

<div>

<div>Mi página web</div>

<div>

<a href="iniciosinestilos.html">Inicio</a>

<a href="fotossinestilo.html">Fotos</a>

<a href="curriculumsinestilo.html">Curriculum</a>
```



```
<a href="contactosinestilos.html">Contacto</a></div>

<div>

<div>



</div>

<span>L</span>orem ipsum dolor sit amet, consectetur
adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat.

<form method="post" action="mailto:correo@extension.com"
      enctype="text/plain">

<p>Comentarios</p>

<textarea rows="5" cols="30" name="comentarios"></texta-
rea>

<p><button type="submit">Enviar</button>&nbsp;&nbsp;
<button type="reset">Cancelar</button></p>

</form>

</div>

</div>

</body>
```

Los estilos que se han utilizado para aplicarlos al documento funcional contacto.html , además de los comunes de la página en general, son los siguientes:

```
#fotocv {
    float: right;
    width: 200px;
}
```


Como puedes observar, en este caso se han aplicado estilos sólo a la imagen que se encuentra en el contenido. Los estilos que se han aplicado en la misma son los siguientes:

- Float: Hemos utilizado este elemento para mover la imagen hacia la parte derecha del documento, para que de esta forma, el texto quede alineado a la izquierda y estén en la misma línea.
- Width: hemos aplicado un ancho a la imagen de 200 píxeles.

2.6. Tipos de página para la ubicación de contenidos

Existen numerosos tipos de páginas que podemos realizar a través del código HTML. Dependiendo de cómo irán ubicados los contenidos, estas páginas serán de un tipo o de otro. Para ello, en HTML utilizamos las etiquetas `<div>` y ``, creando capas en las que ubicar el contenido de las páginas. Tener estructurado documento HTML en base a las capas tiene las siguientes ventajas:

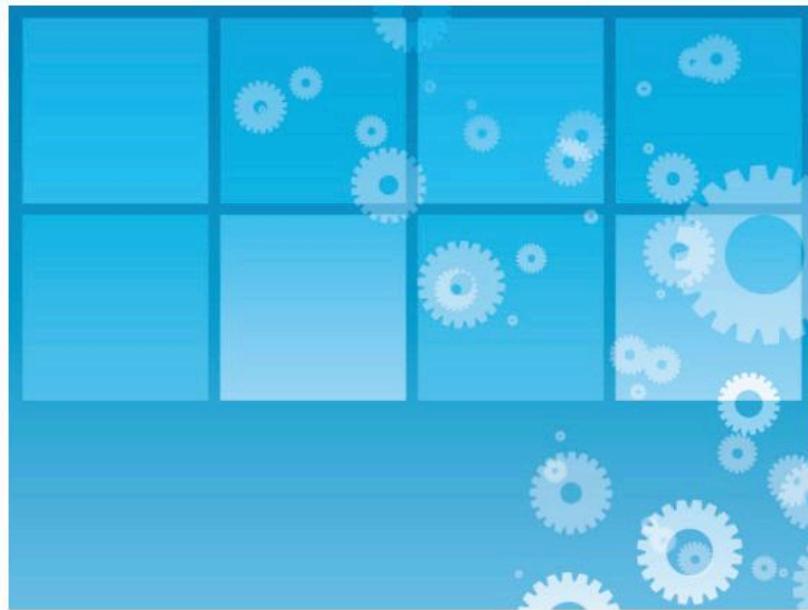
- Nos permite establecer un diseño diferente para cada contenido.
- Podemos identificar cuando un usuario hace clic sobre la página, y específicamente sobre qué contenedor.
- Podremos diseñar efectos especiales para una capa en concreto.

Existen numerosos tipos de páginas dependiendo de la ubicación del contenido, tantos como imaginación pueda tener el diseñador.

Sin embargo, existen ciertos tipos de páginas básicas que se suelen utilizar mucho. Son páginas muy funcionales, muy simples y muy rápidas de hacer.

Estas páginas se diseñan dependiendo del diseño que se vaya a realizar en la página. Podemos encontrar páginas donde el menú se encuentra a la izquierda de la página, por lo que el contenido iría a la derecha del menú. En otras ocasiones, encontramos el menú debajo de la cabecera, por lo que el contenido se encontraría justo debajo de éste.

Dependiendo del tipo de página que sea, podemos elegir un diseño u otro. Para ello, lo mejor es optar por la funcionalidad de la página. Es el usuario el que debe encontrarse cómodo en la página, encontrar la información de forma fácil y rápida. Para ello, deberemos tener claros qué tipos de páginas existen según el tipo de contenidos que vayamos a exponer en la misma.



Dependiendo del tipo de página que vamos a utilizar, los contenidos irán estructurados de una u otra forma. Para esto necesitamos algo de organización y planificación del sitio.

Crear una página web requiere tiempo y energía. Es mejor reflexionar antes de hacer cualquier diseño que rehacer toda una página por un error o un descuido.

Por eso, a continuación, puedes ver algunos consejos para comenzar a ubicar el contenido y la estructura de una página:

Se recomienda redactar una lista con los temas de los que quieras hablar. El tipo y la cantidad de temas a presentar tendrán un determinado impacto en la estructura y la apariencia del site.

No se suelen utilizar los mismos colores para una página creada para mostrar tus trabajos, que una página creada para explicar cómo se arregla el motor de un coche.

Se suele hablar de la arquitectura de un sitio web por un buen motivo: un sitio web está construido como un edificio, con las piezas situadas en su lugar correspondiente, con su acceso y su estilo.

Si imaginamos que para entrar en la cocina tuviéramos que pasar por el dormitorio, esto no sería muy práctico, al igual que con los sitios web.

El diseñador debe construir el sitio en función del uso que se le quiera dar. El objetivo es lograr que el sitio sea cómodo y accesible, usando siempre la lógica y el sentido común.



Veamos a continuación una serie de preguntas que podemos hacernos antes de comenzar a diseñar nuestro sitio web. A través de estas preguntas, podemos llegar a tener bastantes pistas de que es lo que debemos tener en cuenta y que deseamos hacer en nuestro sitio web.

¿Qué es lo más importante? ¿Es visible, fácil de acceder? Si quieres captar la atención sobre un elemento en concreto, no ocultes la información en un punto de la página.

¿Cómo se podría facilitar más el acceso?

¿La navegación es intuitiva?

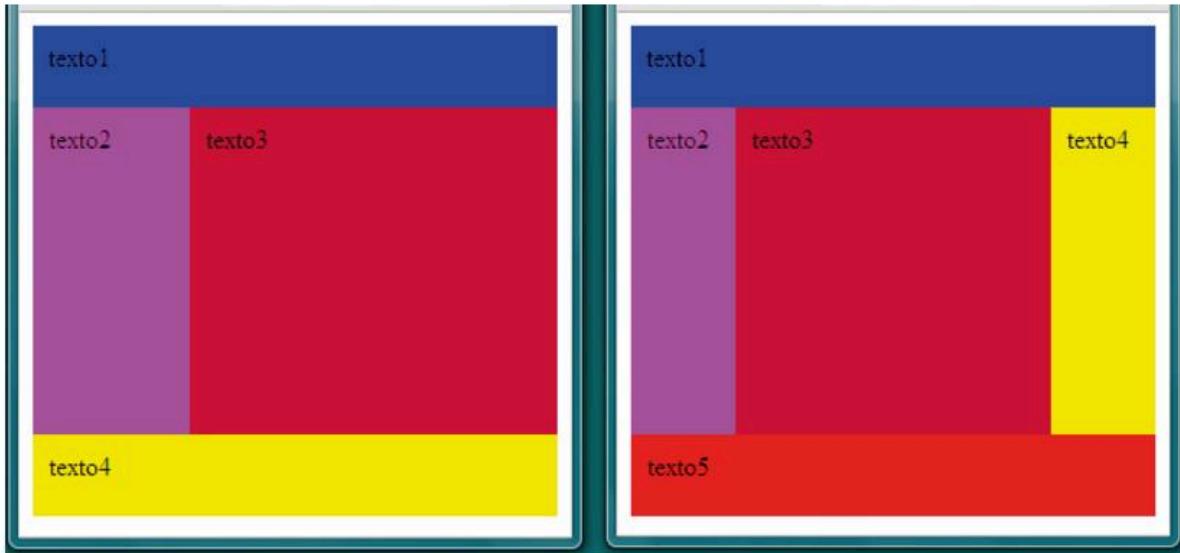
¿Los nombres de los enlaces son explícitos? ¿Fáciles de localizar?

¿Qué tamaño de texto utilizar? ¿Qué color? Se recomienda uno fácil de leer, que haga contraste con el color de fondo, así aseguramos la accesibilidad.

¿Qué tamaño dar a las imágenes? Se recomienda ofrecer una versión adaptada a la pantalla del navegador, e incluir un vínculo a la versión tamaño real si se tiene.

Dependiendo de la ubicación del contenido, tendremos que hacer la estructura de la página de una forma u otra. Podemos dividir nuestra página de muchas formas, y cada página necesitará una estructura dependiendo de a

qué esté destinada. Veamos a continuación una imagen con ejemplo de dos divisiones típicas que se suelen realizar en la construcción y diseño de una página web:



Como puedes observar, la imagen de la izquierda se compone de una estructura clásica, donde se encuentran los componentes siguientes:

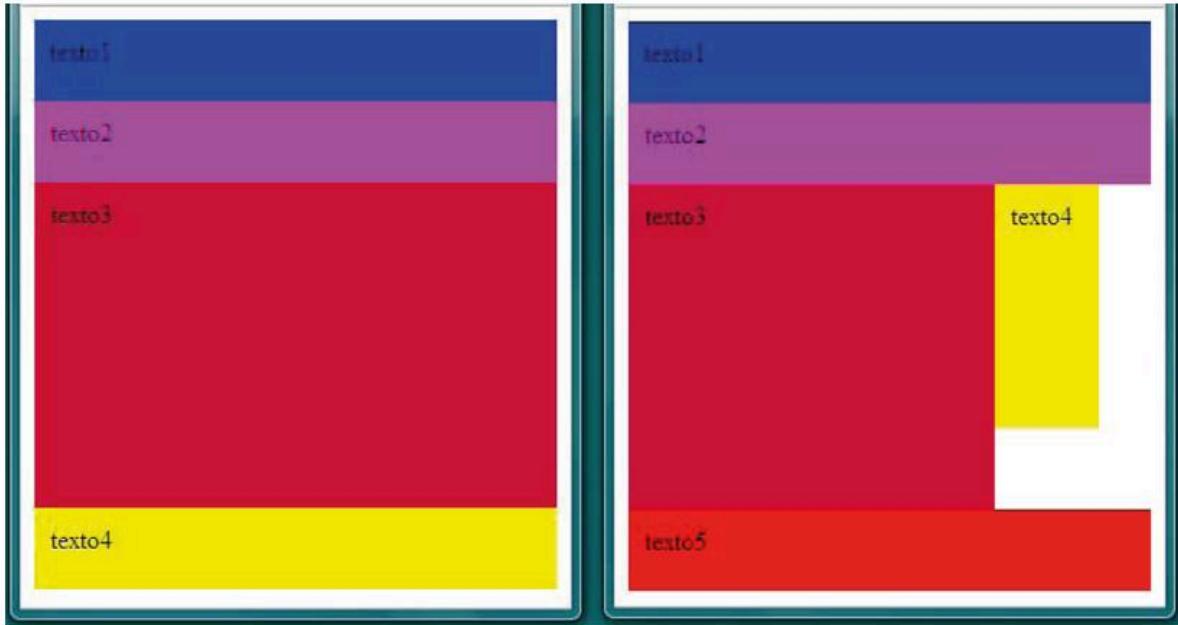
- Encabezado en color azul.
- Menú en color rosa.
- Contenido en color fucsia.
- Pie en color amarillo.

Si nos fijamos en la imagen de la derecha, la estructura es muy parecida a la que ya hemos visto, pero tiene un añadido. Veamos los componentes de los que se compone:

- Encabezado en color azul.
- Menú en color rosa.
- Contenido en color fucsia.
- Submenú o banner vertical en color amarillo.
- Pie en color rojo.

En este caso, se añade otra división a la derecha del contenido. Esto puede servirnos como submenú o como banner para añadir algún elemento.

También podemos encontrarnos divisiones de otro tipo, con un estilo más horizontal. A continuación puedes observar en la imagen otros dos estilos donde el contenido se ubica de forma diferente al anterior ejemplo:



En la imagen de la izquierda puedes observar cómo la estructura del documento en sí es horizontal en todos sus elementos. Los elementos de los que consta son los siguientes:

- Encabezado en color azul.
- Menú en color rosa.
- Contenido en color fucsia.
- Pie en color amarillo.

Si te fijas, esta es la forma que hemos elegido para el documento funcional que hemos estado viendo, a excepción del pie del documento.

En la imagen de la derecha, hay otro modelo, donde el contenido se encuentra dispuesto casi igual, pero con un elemento añadido:

- Encabezado en color azul.
- Menú en color rosa.
- Contenido en color fucsia.

- Submenú en color amarillo.
- Pie en color rojo.

En este caso, se añade junto al contenido un submenú. También puede integrar un banner o cualquier otro tipo de elemento, pero se suele utilizar como submenú del superior.

2.7. Definición de los tipos de página en base a los contenidos y funcionalidades

Cuando comenzamos a realizar la estructura de una página web, encontramos varios tipos de página en base a la posición de los contenidos, los menús de navegación y los enlaces a otras páginas.

Existen varios tipos de estructuras estándares de páginas web donde se especifican la posición de los menús de navegación y el contenido, donde estructura y diseño se mantendrán por toda la navegación del sitio web.

Lo usual entre los diseñadores es posicionar en los márgenes de la página los menús de navegación, para así posicionar el contenido en el centro.

Las estructuras estándares por defecto, posicionan los menús de navegación arriba o en los laterales, aunque hay muchas otras opciones:

- Bloque de navegación a la izquierda:

Cuando la estructura está definida con el bloque de navegación a la izquierda, esto permite presentar muchos menús o enlaces, ya que cada línea puede ser un enlace. En el caso de que los menús se extiendan demasiado, no causa ningún problema ya que el usuario podría moverse con la barra de desplazamiento vertical.

Se suele utilizar este tipo de estructura cuando los menús son administrables, ya que el diseño no se verá muy afectado si aumentara o disminuyera el número de enlaces.

- Bloque de navegación arriba:

Esta estructura se suele utilizar cuando la web no tiene demasiadas secciones. El menú se sitúa en línea y ocupa mucho espacio. Es fácil si los menús se despliegan con otros submenús (lo que facilita la navegación) o si son realmente cortos.

- Bloque de navegación a la derecha:

Tiene las mismas ventajas que el bloque de navegación a la izquierda, aunque esta estructura se usa menos.

- Combinación de los anteriores:

Esta estructura se suele utilizar cuando se tienen muchos enlaces a presentar, por lo que se pueden combinar cualquier a de las estructuras anteriores, sobre todo si los menús se pueden agrupar en dos o tres grandes grupos.

Otra forma de definir los tipos de página en base a sus funcionalidades es por la estructura de navegación, la cual viene definida por la experiencia de navegación del usuario, a través de la cual puede saltar de una página a otra dentro del sitio web usando los enlaces de la página.

La estructura de navegación se suele elegir en función del tipo de contenido. Existen distintos tipos de estructuras:

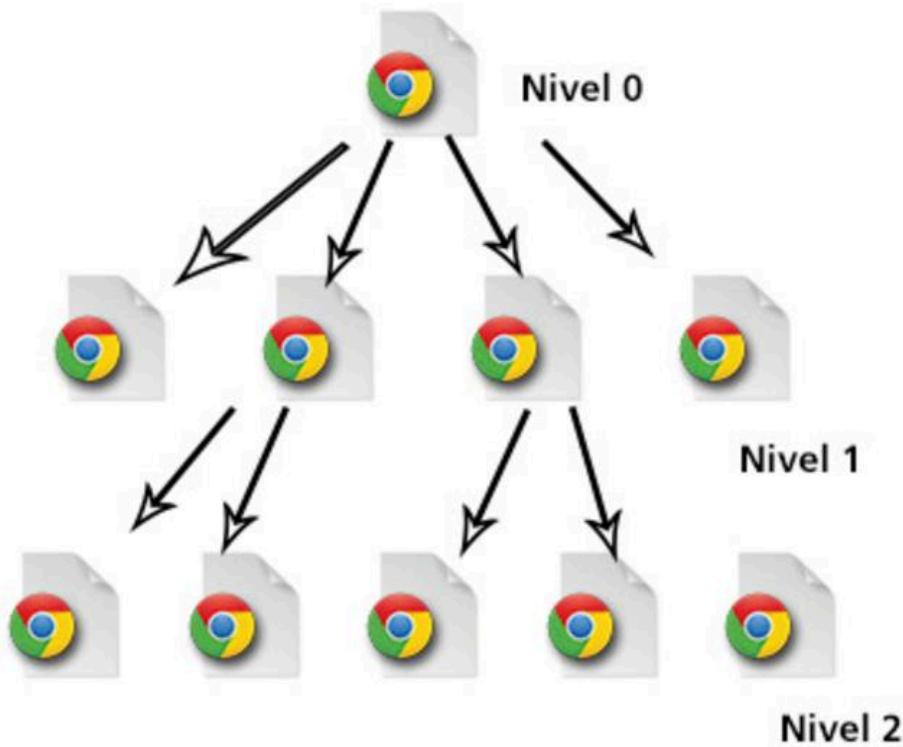
- Jerárquica:

Se trata de una estructura en forma de árbol donde existe una página principal o índice, desde el cual se puede acceder al resto de páginas. Desde estas subpáginas se puede acceder a otras y así sucesivamente, por lo que se crean sucesivamente niveles o jerarquías.

Es ideal para sitios web de centros o proyectos. Es desaconsejable utilizar más de cuatro niveles, así evitaremos que el usuario se desoriente durante la navegación.

Es recomendable situar en todas las páginas un menú para que el usuario pueda navegar de una forma fácil y directa por los niveles y páginas de cada nivel.

A continuación, puedes ver una imagen de una estructura jerárquica:



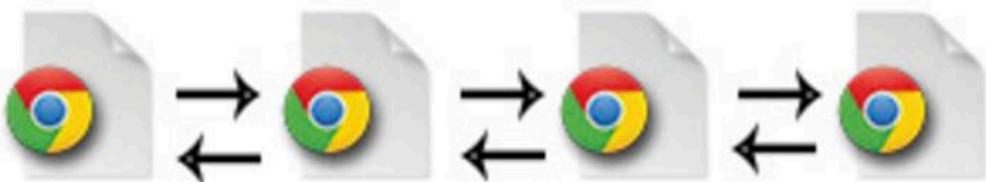
– Lineal:

Se trata de una estructura muy simple, igual a las páginas de un libro. Desde una página concreta se podrá navegar a la página siguiente o a la anterior. Este tipo de estructuras es muy útil si queremos que el usuario siga un itinerario fijo y guiado sin que pueda acceder a otras páginas que puedan distraerle.

Es útil para el tipo de páginas como guías o tutoriales de aprendizaje. No se recomienda este tipo de estructura si el número de páginas encadenadas es muy elevado, ya que puede producir sensación de fatiga y no permite retomar la secuencia que se dejó desde la última sesión.

En la siguiente imagen puedes ver cómo es una estructura lineal en una página web:

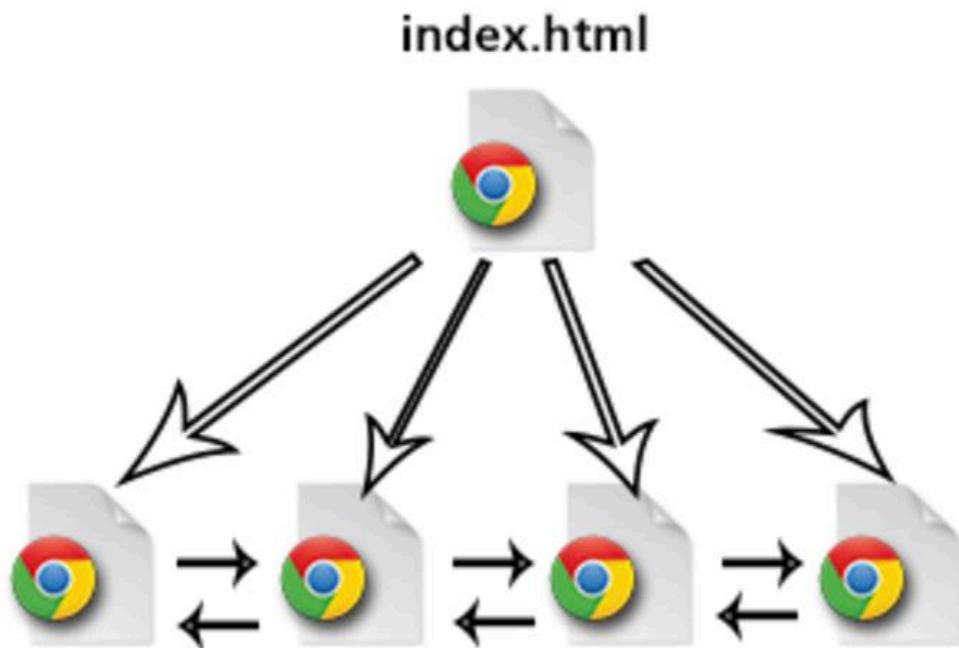
index.html



- Lineal con jerarquía.

Esta estructura es un híbrido que trata de aprovechar las ventajas de la estructura jerárquica y la lineal. Las páginas y subpáginas se organizan de forma jerárquica, pero también es posible navegar de forma lineal por las páginas de un mismo nivel.

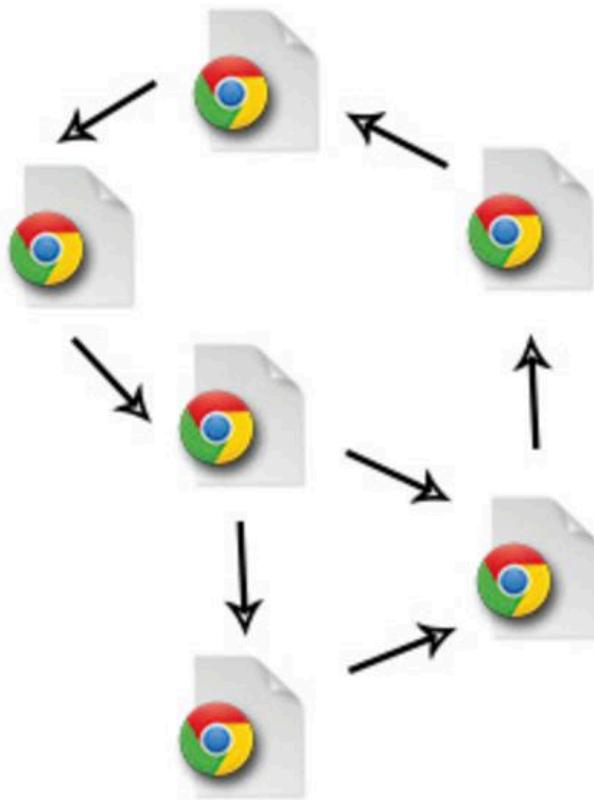
Veamos una imagen donde poder ver gráficamente este tipo de estructura:



- Red.

Con esta estructura, a partir del índice o página principal el usuario podrá navegar a otra u otras sin ningún orden aparente. Se trata de una estructura más libre, pero no se aconseja cuando el número de páginas es muy alto, ya que puede que el usuario se desoriente al no saber dónde está ni disponer de recursos para ir a alguna página en particular.

Veamos en una imagen como se formaría esta estructura:



2.8. Selección de los tipos de página para la página web

Atendiendo a las necesidades de la página web funcional que estamos realizando, deberemos seleccionar un tipo de estructura determinado. Ya que vamos a realizar un contenido básico, la estructura que mejor nos convendrá para la navegación será una estructura con el bloque de navegación arriba.

Nuestra página web tiene pocas secciones, sin embargo, si quisieramos más adelante ampliar los menús con submenús, no tendríamos problema, ya que, al estar en la línea superior, el usuario podría ampliar el menú de forma fácil y rápida.

Por otro lado, la estructura que elegiremos para la web será estructura Lineal con jerarquía. Esta estructura es perfecta para nuestro diseño, ya que las páginas están dispuestas de forma jerárquica, aun así, es posible para el usuario navegar por las páginas de un mismo nivel.



Por las características de nuestra página web funcional, hemos seleccionado una estructura con el bloque de navegación superior, en el que están los botones que nos ayudarán a navegar por los 4 documentos web que tiene nuestra página (Inicio, Fotos, Curriculum y Contacto).

Por esta forma de navegación, sabemos que la estructura del documento funcional es lineal con jerarquía, ya que es la mejor opción para el tipo de página que estamos realizando.

Elegir esta forma de estructurar la navegación es la mejor opción, ya que lo que se busca es que el usuario llegue a la información de forma sencilla y rápida. El contenido lo podrá visualizar sin dar muchos rodeos.

2.9. Utilización del documento funcional para las especificaciones del tipo de página

A continuación, podemos ver en el código de nuestro documento funcional las distintas divisiones que hemos realizado. De esta forma podrás observar cómo se han dividido los distintos elementos el código HTML:


```
<!DOCTYPE html>
<HTML>
<head>
<title>Mi página web</title>
<link rel="stylesheet" type="text/css"
href="style/styles.css"/>
</head>
<body>
<div class="todo">
<div class="titulo">Mi página web</div>
<div class="menu">
<a href="index.html">Inicio</a>
<a href="fotos.html">Fotos</a>
<a href="curriculum.html">Curriculum</a>
<a href="contacto.html">Contacto</a></div>
<div class="contenido">
<p></p><span id="capitular">B</span>ienvé-
nido a mi página web.
Soy un fotógrafo profesional al que le
gusta hacer fotos de la
naturaleza a través de los viajes que rea-
lizo.</p>
<p>Aquí puedes ver
una muestra de los mismos. Sólo tienes que
pulsar en el menú Fotos.</p>
<p> Espero que te gusten.</p>
</div>
</div>
</body>
</HTML>
```



- Div 1: se trata de la división que contiene toda la parte visible que el usuario podrá ver en el navegador.
- Div 2: esta es la división que contiene el título/banner/logo de la página web. Es decir, la primera división que vemos en la página web.
- Div 3: la tercera división contiene el menú del documento funcional.
- Div 4: en la cuarta división encontramos el espacio donde irá el contenido de la página, el cual cambiará dependiendo de donde pulse el usuario.

Como puedes ver en la imagen, los divs están dispuestos de forma lineal, por eso, cuando visualizamos la página en el navegador, podemos ver cómo el espacio de nuestra página web está dividida en partes que van de arriba hacia abajo.

Veamos una imagen de la página web vista por el navegador, donde hemos señalado sus divisiones que se han maquetado en el documento HTML:

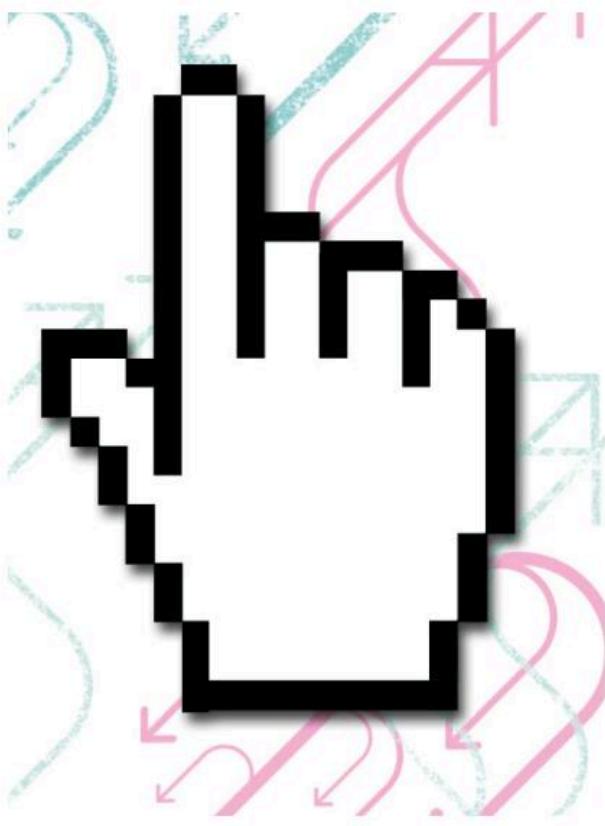


- Div 1: como puedes ver, la capa 1 en el navegador es en sí la imagen de fondo, pero al no tener borde, parece que no fuera una división en sí, ya que su función es contener las divisiones que sí serán visibles a simple vista para el usuario.
- Div 2: la capa o división 2, en el navegador se visualizará como el título/logo/banner de nuestra página web.
- Div 3: la tercera capa de nuestra página se visualizará como el menú a través del cual el usuario podrá moverse por el documento y sus diferentes páginas.
- Div 4: en el navegador podremos visualizar esta capa como el contenido que se ofrece al usuario, el cual cambiará dependiendo de la página en la que se encuentre.

2.10. Especificaciones de navegación

La navegación de un sitio web es de una gran importancia. No se puede entender una página web sin que ésta contenga enlaces a través de los cuales un usuario pueda navegar.

En el caso de que una página web no tuviera sistema de navegación, sería extremadamente difícil adivinar y localizar un contenido sin la ayuda del mismo.



Los sistemas de navegación deber ser intuitivos, ya que esto ayudará a que los usuarios obtengan el mayor beneficio al visitar un sitio web. De esta forma, los usuarios podrán encontrar lo que buscan sin perderse.

El hecho de crear un buen diseño para el menú de navegación no es una tarea fácil, ya que entran muchos puntos a tener en cuenta: desde la densidad del contenido de la página web al diseño que deseamos que tenga la página. Podemos diseñar un menú de navegación simple o complejo, y que pueda facilitar lo que el usuario busca o por el contrario, hacerlo más difícil.

Al planificar un menú de navegación para un blog que para una tienda online. Por eso, muchas veces un simple menú de navegación servirá para conseguir un diseño web bien estructurado, y otras veces no nos quedará más remedio que crear submenús.

Veamos algunos consejos para mejorar la navegación en un site:

Organización en la Estructura de Navegación Web

La parte más importante de un buen diseño es organizar los contenidos del menú, de manera que éste sea fácil e intuitivo para que los usuarios puedan encontrar la información que buscan.

Los diseñadores suelen utilizar algunas prácticas , conocidas como arquitectura de la información, que aquí recomendamos para mejorar la experiencia de usuario.

- Esquema del menú de navegación:

Es recomendable pensar en el esquema del menú de navegación como algo a largo plazo, ya que no sabemos si en un futuro habrá que desarrollarlo más. Además, también tendremos que ponernos en la piel de nuestros usuarios y hacerles la navegación fácil.

Una buena práctica es visitar las páginas de otros diseñadores, con la misma temática que la nuestra, ya que de esta forma encontraremos muchas ideas que tal vez no hubiéramos tenido en cuenta antes. También es aconsejable realizar un test de usabilidad en el caso de tener un sitio web con mucho tráfico.

Una vez que ya tenemos las ideas claras sobre cómo queremos clasificar los contenidos del site, será el momento de crear el esquema de navegación web, el cual no dará una idea visual de cómo se relacionan entre sí los contenidos. Tendremos que evitar que los contenidos estén a más de dos clics de la página de inicio.

- Menú Principal y Menú Secundario:

En el caso de tener una página web con una gran cantidad de contenidos o contenidos que no tienen relación entre sí, deberíamos pensar en hacer un menú principal y un menú secundario.

- Menú Principal:

El menú principal debe contener las opciones primordiales cara al usuario del sitio. Cada web tendrá sus vínculos en un menú principal o en uno secundario, dependiendo de lo que quiera destacar.

- Menú Secundario:

En el menú secundario podemos encontrar vínculos de valor que por algún motivo no están relacionadas o no tienen la suficiente importancia como para estar en el principal.

- Agrupación de contenidos:

En el caso de tener mucho contenido disponible y tener que crear submenús del menú principal, viene la pregunta de en qué orden clasificamos los contenidos, ¿utilizamos el orden alfabético o mejor los ordenamos por el grado de popularidad? Por ejemplo, 1stwebdesigner organizaba los contenidos de los submenús en orden alfabético, y digo organizaba porque ahora ha cambiado el sistema de navegación por uno, que, a mi manera de ver, no es tan claro ni intuitivo como el anterior.

Si nuestra web tiene mucho contenido, y hemos tenido que crear submenús, nos surge la problemática del orden en el que deben ir clasificados los contenidos. Para dar algunas ideas, a continuación, tienes una lista de los métodos de clasificación más usados:

- Orden alfabético: Este tipo de ordenación es recomendable cuando el usuario busca algo de forma rápida.
 - Popularidad: Ordenar por popularidad es recomendable cuando el usuario busca contenidos en base a intereses, aunque tiene la desventaja de dejar de lado otros contenidos que quizás queramos potenciar.
 - Orden cronológico: Clasificar por orden cronológico es una buena opción para contenidos en que la fecha de publicación es importante, como un periódico o un blog.
 - Pasos a seguir: Elegiremos esta opción en el caso que el contenido responda a la pregunta de cómo hacer algo y éste requiera una serie de pasos que el usuario deberá seguir.
- Horizontal versus vertical:

El diseñador web tiene ante sí numerosos factores que influirá en la decisión de elegir un menú horizontal o vertical, como son el diseño, la usabilidad o la cantidad de contenido, incluso añadir elementos visuales que faciliten la navegación del usuario. Puede que el diseñador se decante por una tipografía específica para darle un toque más exclusivo.

Otro factor a tener en cuenta que debe tener el diseñador es si el sitio web tendrá varios idiomas. Este factor habrá que tenerlo en cuenta, ya no es igual de extenso un menú en inglés que en alemán o español. En un idioma el menú puede quedar perfecto, y en otro destruir todo el trabajo.

- Uso de los submenús:

Los submenús nos serán muy útiles en el caso de que la página web tenga muchos contenidos. Si queremos facilitar la navegación del usuario entre estos contenidos tendremos que ser creativos y organizados, para que el usuario no se pierda entre estos.

Otra recomendación es evitar un margen pequeño con el siguiente elemento en un menú horizontal, ya que podría dificultar la navegación.

A continuación, puedes ver tres ideas a tener en cuenta a la hora de crear submenús que mejoren la experiencia de navegar por el site:

Uso de categorías y subcategorías. La buena clasificación en los submenús es primordial, por eso, para que no dé lugar a confusión se recomienda crear antes un esquema de navegación lógico antes de escoger el diseño, lo que facilitará la navegación del usuario.

Diferenciación de cada submenú: Cada categoría tiene un submenú diferente del anterior, sin perder la consistencia. El uso de colores puede ayudarnos en esta tarea.

Uso de iconos, tipografía e imágenes. Para mejorar la calidad del menú, es recomendable hacer uso de los espacios, iconos o también tipografía e imágenes, en detrimento de la letra pequeña o demasiada información en poco espacio.

2.11. Creación de un mapa de navegación de páginas

Gracias a un mapa de navegación podemos presentar de forma gráfica la estructura del hipertexto que se encuentra en una página web, para así poder indicar los principales conceptos incluidos en el espacio de la información y las interrelaciones que puedan existir entre estos.

Un mapa servirá, por lo tanto, a través de su representación del sitio web, para orientar al usuario durante el recorrido que haga éste en la página o para facilitarle un acceso directo al lugar que le interese.

El mapa de navegación, por lo tanto, deberá reflejar la estructura de la web a través de enlaces que van a los nodos principales, los cuales también pueden desarrollarse para mostrar los subnodos. Este mapa puede estar representado de forma textual, o también de una forma gráfica, o incluso una combinación de ambas.

Al igual que con un sumario, un índice o una tabla de contenido que se usan para “moverse” por los documentos en papel, un mapa de navegación es igual de útil para navegar por un hipertexto. En una Web, esta forma de navegar tradicional se puede encontrar en forma de menús.

Si queremos dar una visión global del contenido al usuario, es muy útil definir en la web un mapa de todo el contenido, que el usuario pueda verlo en una sola pantalla y que pueda acceder a él desde cualquier nodo del hiperdocumento.

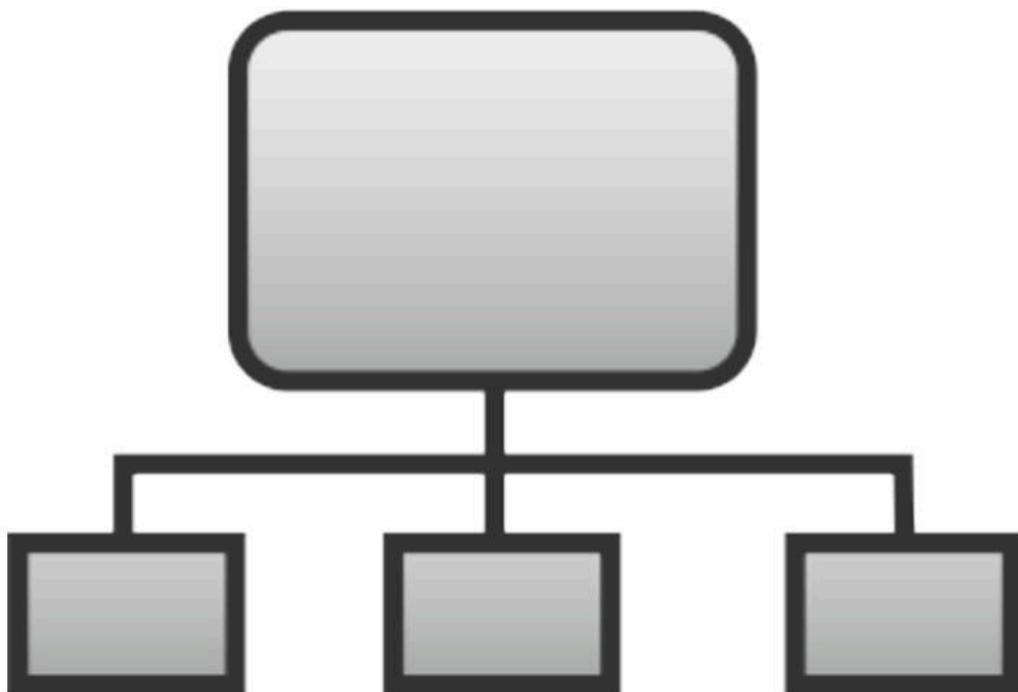
Sin embargo, mostrar todo el contenido reflejado en un mapa o esquema global en una sola pantalla cuando el espacio web o un hipertexto son demasiado grandes, es casi imposible de llevar a la práctica y tampoco tiene ninguna utilidad confeccionar una farragosa sopa de letras o imágenes.

En el caso de tener una web demasiado grande, se hará una tarea dificultosa mostrar todo el contenido reflejado en una sola pantalla o esquema, y si se hace, puede que sea demasiado complicado de entender visualmente para el usuario.

Para este problema, podemos realizar diagramas que de forma global muestren los nodos más importantes y luego desarrollar cada subnodo en diagramas locales con información más detallada. También podemos mostrar estos extensos diagramas a través de los mapas ojos de pez, con el que podremos mostrar todo el espacio hipertextual en un solo gráfico dividido en diferentes niveles de detalle. Así podremos mostrar más información sobre la posición actual del usuario y disminuir los detalles de forma gradual según estén más alejadas las partes entre sí.

Realizar este tipo de nodos es bastante laborioso de confeccionar en la tecnología web, ya que cada nodo necesita de un tratamiento específico.

En Internet encontramos multitud de programas que nos ayudan a confeccionar mapas que muestran el contexto del nodo activo, además de mostrar de una manera gráfica la estructura global del hipertexto, como por ejemplo FrontPage de Microsoft. Sin embargo, estas herramientas no son del todo útiles, ya que sólo funcionan dentro del programa de construcción de las páginas, y no cuando ésta se publica en la Web.



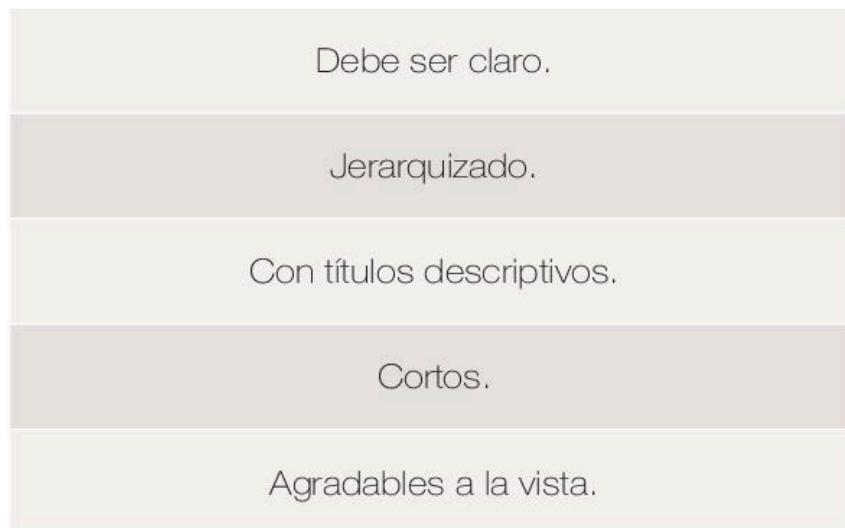
En resumen, un mapa de navegación o mapa del sitio puede ser una representación gráfica o textual de un sitio web, equivalente al índice de un libro. Se compone de una única página en HTML mostrando la estructura del sitio:



El mapa del sitio facilita al usuario la navegación, pero también a los buscadores, ya que se distinguen dos tipos de mapa de sitio, uno en HTML y otro en XML. Estos últimos sirven para el posicionamiento en buscadores.

A raíz de este uso, los sitemaps se hicieron más populares, y se suelen utilizar para facilitar el acceso a todas las secciones de nuestra página y que incluyan esos archivos en los resultados de las búsquedas.

Volviendo al uso para el HTML, veamos cuáles son las características principales de un mapa del sitio web:



2.12. Utilización del documento funcional para integrar el mapa de navegación

Veamos ahora los cambios que hemos realizado en la página web para añadir un enlace a un mapa web:

- Hemos creado un documento nuevo con los enlaces a cada uno de los elementos de nuestra web que contengan un enlace. El documento HTML que hemos creado queda finalmente de la siguiente forma:

```
<!DOCTYPE html>

<html lang="es">

<head>

<link rel="stylesheet" type="text/css" href="css/styles.css"/>

<title>Mapa Web</title>

</head>

<body>

<p class="mapawebposicion">

<ul class="mapaweb">

<li><p><a href="index.html" target="_blank">Inicio</a></p>

</li>

</ul>

<ul class="mapaweb">

<li><p><a href="fotos.html" target="_blank">Fotos</a></p>

</li>

</ul>

<ul class="mapaweb">

<li><p><a href="images/image-1.jpg" target="_blank">Imagen 1</a></p></li>
```



```
    <li><p><a href="images/image-2.jpg" target="_blank">Imagen 2</a></p></li>
    <li><p><a href="images/image-3.jpg" target="_blank">Imagen 3</a></p></li>
    <li><p><a href="images/image-4.jpg" target="_blank">Imagen 4</a></p></li>
    <li><p><a href="images/image-5.jpg" target="_blank">Imagen 5</a></p></li>
    <li><p><a href="images/image-6.jpg" target="_blank">Imagen 6</a></p></li>
    <li><p><a href="images/image-7.jpg" target="_blank">Imagen 7</a></p></li>
    <li><p><a href="images/image-8.jpg" target="_blank">Imagen 8</a></p></li>
</ul>

<ul class="mapaweb">
    <li><p><a href="curriculum.html" target="_blank">Curriculum</a></p></li>
</ul>

<ul class="mapaweb">
    <li><p><a href="documentos/curriculum.pdf" target="_blank">Curriculum PDF</a></p></li>
</ul>

<ul class="mapaweb">
    <li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>
</ul>
</body>
</HTML>
```

validación correcta W3C.

Como puedes observar, se trata de una lista desordenada, donde los elementos principales se componen por los enlaces del menú, y los elementos secundarios están compuestos por los enlaces que se encuentran dentro del contenido de cada enlace principal.

- Hemos introducido dos estilos de clase para dar formato al documento. Al introducir nuestro mapa del sitio dentro del cuerpo del documento, éste hereda el fondo de la imagen.
- Hemos añadido ciertos estilos a nuestro mapa del sitio, como por ejemplo un estilo de fuente, una anchura, una posición o un espacio de relleno de 15 píxeles. Lo podemos ver en el código CSS siguiente:

```
.mapaweb {  
    font-family: Helvetica, Gill Sans, Lucida;  
    border-style: none;  
    width: 800px;  
    margin: 0 auto;  
    position: relative;  
    padding-left: 15px;  
}  
  
.mapawebposicion {  
    padding-top: 40px;  
}  
  
¡Este documento es CSS versión 3 + SVG válido!
```

- A continuación, hemos ajustado algunos valores de la página. Al añadir el pie de página a nuestro documento HTML, hemos tenido que reducir los píxeles de la altura del div que pertenecía al bloque del contenido de la página a 200 píxeles. Esto no ha tenido ninguna repercusión en nuestra página ya que se trata de una página con muy poco contenido. Si hubiéramos tenido demasiado contenido, tendríamos que cambiar las proporciones de la imagen de fondo, o incluso cambiar la web a unas proporciones relativas.

```
<!DOCTYPE html>  
<HTML lang="es">
```



```
<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<p><span id="capitular">B</span>ienvenido a mi página
web. Soy un fotógrafo profesional al que le gusta
hacer fotos de la naturaleza a través de los via-
jes que realizo.</p> <p>Aquí puedes ver una
muestra de los mismos. Sólo tienes que pulsar en el menú
Fotos.</p>

<p> Espero que te gusten.</p>

</div>

<div class="pie"><p><a href="mapaweb.html" target="_
blank">Mapa Web</a></p></div>

</div>

</body>

</HTML>
```

Como puedes ver, simplemente hemos integrado un nuevo div al final de la página, pero va integrado dentro del div "todo".

- Específicamente en el documento contacto.html hemos tenido que realizar algunos cambios en las especificaciones del formulario, para que pueda caber el pie de página y quede en armonía con el resto del documento. Veamos los cambios que hemos realizado a continuación:

```
<!DOCTYPE html>

<HTML lang="es">

<head>

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="style/
styles.css"/>

</head>

<body>

<div class="todo">

<div class="titulo">Mi página web</div>

<div class="menu">

<a href="index.html">Inicio</a>

<a href="fotos.html">Fotos</a>

<a href="curriculum.html">Curriculum</a>

<a href="contacto.html">Contacto</a></div>

<div class="contenido">

<div class="contacto">

<div id="fotocv">

</div>

<span id="capitular">Gracias por tu visita. Puedes ponerte en contacto conmigo a través del formulario y contarme qué te parecen mis trabajos, o preguntarme las dudas que tengas.

<form method="post" action="mailto:correo@extension.com"
enctype="text/plain">

<textarea rows="5" cols="30" name="comentarios"></textarea>
```



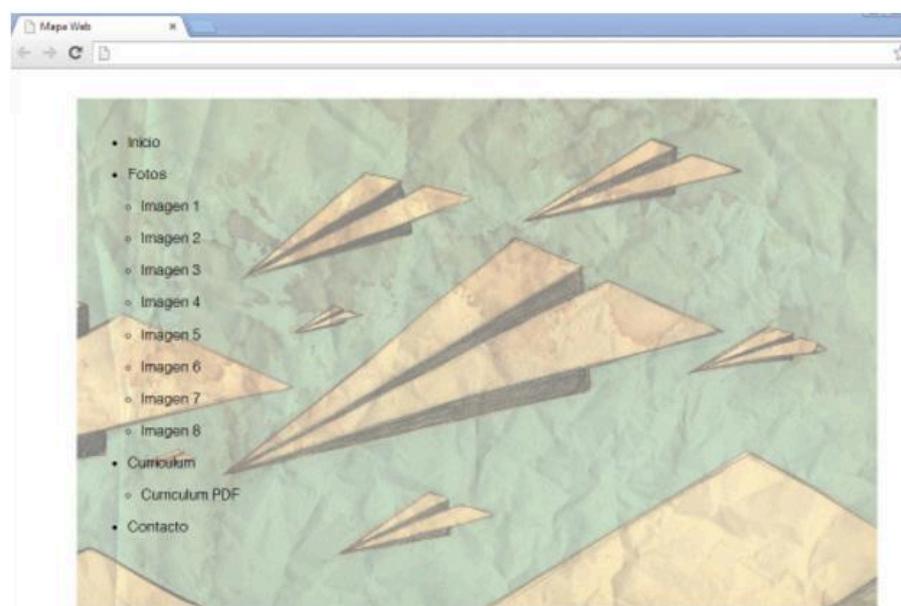
```
<button type="submit">Enviar</button>  
  
<button type="reset">Cancelar</button>  
  
</form>  
  
</div>  
  
</div>  
  
<div class="pie"><p><a href="mapaweb.html" target="_blank">Mapa Web</a></p></div> </div>  
  
</body>  
  
</HTML>
```

Como puedes observar, hemos añadido un estilo de clase contacto, para agregarle los estilos pertinentes para poder reducir el bloque a 200 píxeles.

También hemos suprimido el texto "comentarios", ya que nos ocupaba espacio y no nos dejaba subir el div hasta 200 píxeles. Al ser un texto que sobraba, ya que con el texto es suficiente, no ha sido una pérdida importante. Veamos a continuación el estilo que hemos aplicado a contacto.html:

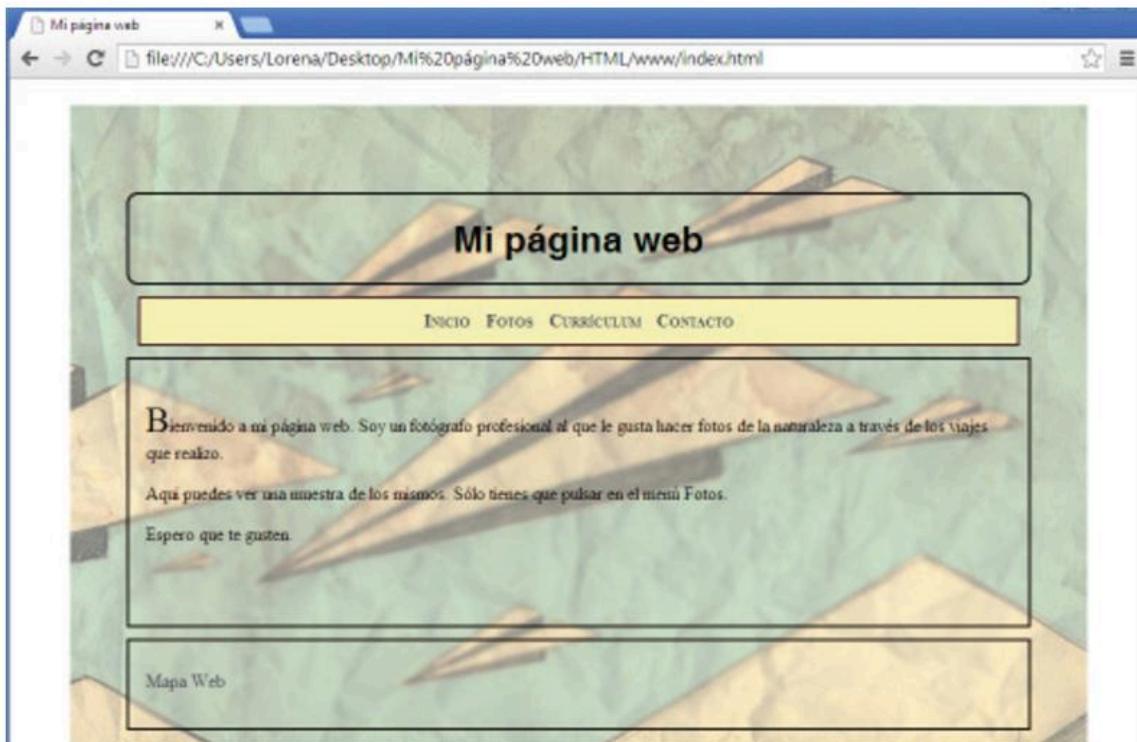
```
.contacto {  
  
    min-height: 200px;  
  
}
```

En la siguiente imagen puedes ver una imagen del documento funcional mapaweb.html visto en el navegador:



Como puedes observar, se trata de una lista desordenada con los elementos que componen nuestra página. Cada uno de los enlaces irá a su elemento en concreto. Los enlaces de las imágenes van directamente a las imágenes grandes. Todos los documentos se abrirán en una ventana nueva.

Veamos en la siguiente imagen cómo queda el pie en nuestra página de Inicio:



Como puedes ver, el pie de página tiene un pequeño número de píxeles para integrar el texto que hace de enlace, para llevarnos al documento que nos muestra el Mapa del sitio. En concreto, los estilos que hemos aplicado para el pie de página son los siguientes:

```
.pie {  
    margin-top: 10px;  
    position: relative;  
    padding-top: 10px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;
```



```
border-color: #0D0802;  
border-width: 2px;  
text-align: left;  
min-height: 30px;  
}  
-----  
.pie a {  
text-decoration: none;  
color: #4ale20;  
}  
.pie a:hover {  
color: #510600;  
}  
.pie a:visited {  
color: #3D4859;  
}
```

2.13. Elementos utilizados para la navegación

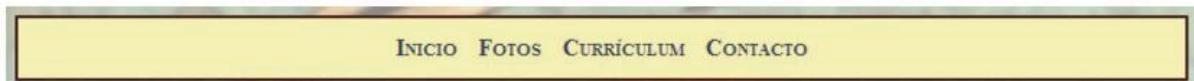
Los elementos que se han utilizado para la navegación del sitio web consisten en una serie de enlaces que van aplicados a cada uno de los elementos que componen el menú del documento.

Este menú puede verse continuamente a través de toda la navegación, por lo que el usuario puede continuamente ver el menú y desplazarse por la página sin perderse. Los elementos de los que se compone el menú del documento para la navegación son los siguientes:



Cada uno de estos enlaces nos lanza a un documento web diferente. En cada uno de estos documentos, también podremos ver el menú, y podremos seguir navegando por las páginas. En cada uno de las páginas web, lo único que cambia será el contenido de las mismas, sin embargo, tanto el menú como el título/banner de la misma no cambian, siempre permanecen igual.

A continuación puedes ver una imagen con la captura del menú del documento:



2.13.1. Definición de los elementos utilizados para navegar

Como ya sabemos, los elementos que hemos escogido para navegar por la web son los que integran el menú del documento. Estos elementos se encuentran definidos por una serie de estilos que veremos a continuación:

```
.menu {  
background-color: #f6f2b6;  
margin: 10px;  
border-style: solid;  
border-color: #4a1e20;  
border-width: 2px;  
padding-top: 10px;  
padding-bottom: 10px;  
font-weight: bold;
```



```
    font-variant: small-caps;  
    word-spacing: 10px;  
}  
-----  
.menu a {  
    color: #4a1e20;  
    text-decoration: none;  
}  
  
.menu a:hover {  
    color: #510600;  
}  
  
.menu a:visited {  
    color: #3D4859;  
}
```

Veamos los estilos que hemos aplicado al menú:

- Background-color: hemos aplicado un color de fondo de un color amarillo.
- Margin: hemos aplicado un margen de 10 píxeles.
- Border-style: hemos aplicado un borde sólido para que sea visible en el navegador.
- Border-color: hemos aplicado un color marrón al borde del menú de navegación.
- Border-width: hemos aplicado un ancho de borde de 2 píxeles.
- Padding-top: hemos aplicado un ancho de espacio de 10 píxeles en la parte superior.

- Padding-bottom: hemos aplicado un ancho de espacio de 10 píxeles en la parte inferior.
- Font-weight: hemos aplicado un ancho de fuente en negrita.
- Font-variant: hemos aplicado un estilo versalitas al texto.
- Word-spacing: hemos aplicado un espacio entre letras de 10 píxeles.

Por otro lado, hemos aplicado estilos a lo que viene siendo el texto como hiperenlace:

- Color: hemos aplicado un color al texto granate oscuro, para que haga contraste con el color de fondo.
- Text-decoration: le hemos dado el valor none para quitar el subrayado por defecto de los enlaces.
- A:hover : hemos aplicado un color granate oscuro al texto para que cambie cuando el ratón pase sobre el enlace.
- A:visited: hemos aplicado un gris azulado al enlace del menú de navegación para que cambie una vez visitado.

2.13.2. Utilización del documento funcional para especificar los elementos de navegación

A continuación, vamos a ver cuáles son las especificaciones de navegación del documento funcional que hemos estado realizando.

Para ello, veamos cuál es el tipo de menú que hemos realizado en el código de nuestra página:

```
<div class="menu">  
  <a href="index.html">Inicio</a>  
  <a href="fotos.html">Fotos</a>  
  <a href="curriculum.html">Curriculum</a>  
  <a href="contacto.html">Contacto</a></div>
```

Como puedes observar, se trata de un menú simple, con cuatro enlaces que van a los cuatro documentos web de los que se compone la página.

Hemos seleccionado este tipo de menú simple por el tipo de página que es, ya que al tratarse de una página personal con poco contenido, no necesitaremos un menú demasiado extenso, o con un submenú.

También hemos incorporado dentro de la página el menú de una forma horizontal y en la parte superior, de manera que la navegación del usuario será muy sencilla y práctica.

De esta forma, la información la encontrará de forma rápida y fácil.

Hemos utilizado un texto simple como enlace del menú. Hemos evitado imágenes o iconos, para aumentar la comprensión del menú y su fácil accesibilidad.

El orden que hemos utilizado para el menú, ha sido por prioridades.

- Inicio
- Fotos
- Curriculum
- Contacto

Lo que se pretende con esto es que el usuario, al leer de izquierda a derecha, comience a navegar por el Inicio y vaya clicando a lo largo del menú.

De todas formas, al ser un menú muy básico, se puede clicar sin tener que llevar un orden, de forma aleatoria.

Además, recordamos que, en la hoja de estilos, hemos incluido a los enlaces colores diferentes para los diferentes enlaces de los vínculos, de manera que el usuario pueda saber si ha visitado ya o no cualquiera de los enlaces, etc.

2.14. Elaboración de una guía de usuario

Los manuales de usuario son guías escritas en formatos impresos (en papel) o en documentos electrónicos (PDF o XPS) que proporcionan instrucciones de cómo hacer o utilizar algo. Si bien se piensa generalmente en las "guías de usuario" como manuales para programas de computación, las guías de usuario pueden acompañar a computadoras y a otros dispositivos electrónicos, como televisores, estéreos, sistemas telefónicos, y reproductores MP3, y también pueden acompañar a electrodomésticos y equipos de jardinería. Los

buenos manuales de usuario educan al usuario acerca de las características del producto mientras les enseñan cómo utilizar esas características de manera efectiva, y están dispuestos de tal forma para que puedan leerse y consultarse fácilmente. A continuación, se presentan algunas cosas para tener en cuenta a la hora de crear un contenido efectivo y diseñar la disposición para un manual de usuario.

– Método 1:

Define quién es el usuario. Para escribir un manual de usuario exitoso, debes desarrollar un perfil de tu usuario, ya sea de manera formal, creando un perfil escrito, o de manera informal, tomándote el tiempo de realizar suposiciones razonables acerca de las características del usuario. Un perfil así es útil cuando formas parte de un equipo que está escribiendo la documentación de usuario y también puede ayudar a llevar al producto en sí desde un concepto hasta su forma final. Las cosas a tener en mente cuando formas un perfil de usuario son:

Dónde utilizarán los usuarios la guía de usuario, como en su casa, en la oficina, en un lugar de trabajo remoto o en el auto. Esto puede determinar, no sólo el contenido, sino también el estilo que tome el manual de usuario.

Cómo utilizarán los usuarios la guía de usuario. Si los usuarios consultarán el manual pocas veces o sólo para buscar información, debería tener un formato como el de un documento de consulta. Si, al principio, los usuarios consultarán el manual con frecuencia, la sección de consulta debería estar acompañada por una sección titulada "Cómo empezar" y por instrucciones de las tareas más simples para las que se utilizará el producto.

Cuánta experiencia tienen los usuarios con el producto o con otros productos similares. Si tu producto es nuevo o significativamente diferente de otros productos similares, deberás incluir una explicación de cómo es distinto comparado con otros productos y también instrucciones de cómo comenzar a utilizarlo. Si el producto está relacionado con algo que frecuentemente resulta problemático para los usuarios, como sucede con muchos programas de computación, deberás proporcionar una información adecuada y detallada de una forma entendible.

Escribe para ayudar a las necesidades del usuario de una manera tal que éste lo pueda entender. A menos que el usuario tenga antecedentes técnicos, es posible que sea mejor evitar el lenguaje muy técnico para lograr explicaciones más claras y simples. El texto debería estar organizado de manera similar a como piensa el usuario; a menudo, tiene más sentido listar las características de los productos agrupándolas por sus funciones que listar sólo aquellas que se utilizan más seguido.

A veces es inevitable utilizar términos técnicos, como para un programa de computación para crear gráficos que incluye gráficas de Fibonacci además de los gráficos de barras y circulares, que son más comunes. En ese caso, es útil definir el término y proporcionar algo de información básica al respecto, como explicar qué es un gráfico de Fibonacci y para qué se usa en el análisis financiero.

– Método 2:

Incluye la portada y las carátulas adecuadas. Deberás incluir una portada para cualquier guía de usuario mayor a una hoja de referencia, y una carátula para cualquier manual que cubra más que la superficie de una hoja de papel doblada (4 o más páginas de largo).

Si el manual está protegido por los derechos de autor, se debe incluir un aviso de derechos de autor en la portada y en la carátula.

Si existen términos y condiciones para usar el manual y el producto asociado a éste, colócalos en la portada interior.

Pon referencias a las documentaciones relacionadas en el prefacio. Si la documentación de usuario es más que un solo manual, coloca aquí las referencias a los demás documentos, con sus números de versiones correspondientes. El prefacio también es el lugar en donde colocar, si la hubiera, una sección "cómo utilizar esta guía".

Si el manual excede las 10 páginas, incluye un índice de contenidos.

Pon las instrucciones o los procedimientos y los materiales de consulta en el cuerpo del manual. En la mayoría de los casos, los procedimientos y los materiales de consulta tienen sus propias secciones, aunque, dentro de una sección, puedes indicarle al usuario que consulte un contenido específico de la otra. De este modo, el usuario puede encontrar más rápidamente la información que esté buscando.

Los procedimientos deben escribirse con una estructura consistente a lo largo de la sección de instrucciones del manual. Comienza con un resumen de la tarea, luego describe qué usuario debe hacer el usuario y qué resultados debe observar. Los pasos deben ser numerados y comenzar con acciones verbales del mismo modo que están escritos los pasos en cada sección de este artículo.

Los materiales de consulta pueden incluir una lista de opciones, consejos para la resolución de problemas, y las preguntas más frecuentes. Los glosarios y los índices puedenadirse cerca del final del manual, aunque se puede colocar una lista de términos utilizados frecuentemente al principio. Puede omitirse el índice si el manual tiene menos de 20 páginas.

Usa imágenes gráficas para respaldar al texto cuando sea necesario. Las imágenes gráficas, o las capturas de pantalla pueden ilustrar algunos puntos en el manual mejor que un texto, en especial en procedimientos complejos en los cuales el usuario necesita una confirmación visual de que estén llevando a cabo los pasos correctamente. Las imágenes gráficas pueden crearse utilizando programas de diseño asistido por computadora (CAD, por sus siglas en inglés) o programas de diseño gráfico, cámaras digitales y programas de edición de fotografías, o en el caso de las capturas de pantalla, puedes utilizar la función de captura que viene instalada en la computadora, o un programa de manejo de gráficos con una función de captura de pantallas.

Una vez que hayas creado una imagen gráfica, guárdala en un formato tan comprimido como tu procesador de textos o tu programa de autoedición te permita utilizar. También debes reducir el tamaño físico de la imagen hasta alcanzar uno que pueda encajar más fácilmente en la página proveyendo simultáneamente el detalle necesario para el usuario (de ser necesario, quizás debas separar la imagen en partes para mostrar las partes relevantes junto al texto que respaldan).

Si utilizas muchas imágenes gráficas en los procedimientos, hazlas de un tamaño consistente, ya sea en las mismas medidas de ancho y alto o en la misma proporción de reducción respecto al tamaño original. Esto hará que las imágenes luzcan más atractivas para el usuario. Del mismo modo, cuando tomes capturas de pantalla en una computadora, asegúrate de que la computadora muestre la misma combinación de colores cuando captures imágenes si el manual de usuario mostrará las capturas en colores. Aunque los programas de diseño gráfico como el Photoshop o el Paint Shop Pro ofrecen funciones de captura de pantalla decentes, los programas de captura especializados como el Snagit también permiten modificar, catalogar y comentar fácilmente las capturas de pantalla.

– Método 3:

Elige algunas fuentes legibles. Aunque las computadoras permiten utilizar muchas fuentes diferentes, el objetivo de un manual de usuario es ser fácilmente legible. Para lograr esto, lo mejor es elegir un número reducido de fuentes que se vean bien juntas. Las fuentes pueden dividirse en dos grupos: las fuentes serif y las fuentes sans serif

Las fuentes serif tienen pequeñas líneas de adorno al final de los trazos principales que forman la letra. Algunas fuentes serif son Times New Roman, Baskerville, y Book Antiqua. Las fuentes serif funcionan mejor en grandes fragmentos de texto impresos en el cuerpo principal del manual de usuario en un tamaño de 10 o 12 puntos.

Las fuentes sans serif solo tienen trazos que forman las letras, sin adornos. Algunas fuentes sans serif son Arial, Calibri y Century Gothic. Las fuentes sans serif se pueden utilizar en grandes fragmentos de texto en tamaños de 8 a 10 puntos en un PDF o en un manual en internet, aunque la falta de serifs hace que las oraciones escritas en tamaños de 12 puntos o más sean difíciles de leer. Sin embargo, se pueden utilizar de manera efectiva en tamaños más grandes para los títulos y encabezados, y además son muy buenas en tamaños más pequeños para pies de página y números en columnas y tablas.

Por lo general, deberías elegir fuentes simples como Arial o Times New Roman para tu manual de usuario, aunque puedes elegir una fuente decorativa para las citas apartadas o los títulos si escribes un manual de usuario para un videojuego con una temática fantástica o de ciencia ficción (en el caso de las citas al margen, a menudo puedes utilizar alguna de las fuentes sencillas y mostrar la cita apartada en cursiva).

Una vez que hayas elegido las fuentes que vayas a utilizar, crea una página de muestra para verificar que las fuentes se vean bien juntas en el papel. También deberías mostrarle esta muestra a la persona que deba aprobar la apariencia del manual antes de continuar escribiéndolo. Piensa bien la distribución. Una vez que hayas elegido las fuentes del manual de usuario, deberás decidir dónde colocas cada cosa en las páginas.

Por lo general, es conveniente poner el título del manual o del capítulo en el encabezado o en el pie de página colocando el título del manual en la página izquierda y el título del capítulo en la página derecha. También es conveniente colocar el número de página en el encabezado o en el pie de página, ya sea en el lado externo (del encabezado o del pie de página) o en el centro (únicamente en el pie de página). Es conveniente diferenciar la primera página de cada sección o capítulo de las otras páginas colocando su número de página en el centro del pie de página y en las páginas siguientes, ponerlo en la esquina exterior del encabezado.

Es conveniente colocar el texto de las referencias de las imágenes en colores o en recuadros sombreados para separarlo del resto del texto. Asegúrate de elegir un color o un nivel de sombreado que permita leer el texto con facilidad.

Deja unos márgenes razonablemente amplios a los lados, con un espacio extra en los bordes en los cuales vayas a unir todas las páginas.

Eige el tipo de encuadernación para el manual de usuario. Si el manual de usuario tiene más de 4 páginas, será necesario unir las páginas de algún modo. Aunque los documentos de uso interno pueden engramparse en

las esquinas, los manuales de usuario de uso externo que se entregan con un producto pueden encuadrarse en una de estas 3 formas:

Engrapar a los lados resulta apropiado para los manuales compuestos por hojas de papel dobladas de 21 x 27,5 cm (8,5 x 11 pulgadas), 21 x 35 cm (8,5 x 14 pulgadas) o 27,5 x 42,5 cm (11 x 17 pulgadas). La mayoría de los manuales baratos de 48 páginas o menos se encuadernan de este modo.

Es más común utilizar el cosido a caballo para las guías de consulta para terceros que para los manuales de usuario que se entregan con los productos que no sean automóviles, aunque algunas guías de usuario más largas se encuadernan de este modo (inicialmente, cuando JASC software producía el Paint Shop Pro, entregaban manuales de usuario encuadrados mediante el cosido a caballo).

La encuadernación con espirales es más adecuada para las guías de usuario que estén diseñadas para entornos más hostiles como en exteriores, en donde los manuales encuadrados con grapas o cosidos a caballo se desarmarían. Algunos manuales encuadrados con espirales también pueden tener páginas laminadas para evitar que se arruine el manual si se moja o se ensucia.

Arma una plantilla de documento para tu manual de usuario. Muchos procesadores de texto y programas de autoedición ofrecen la posibilidad de crear plantillas de documentos para los manuales de usuario, así que a medida que escribas, el texto se mostrará automáticamente en la fuente que hayas elegido para la parte del manual en la que estés trabajando (de hecho, este artículo fue escrito inicialmente utilizando una plantilla de Microsoft Word). La mayoría de estos programas también incluyen un conjunto de plantillas predefinidas que puedes modificar según lo necesites, en vez de crear la plantilla desde el comienzo.

Los procesadores de texto y los programas de autoedición también ofrecen la posibilidad de crear "estilos", que son tipos y tamaños de fuente predefinidos para los títulos, los pies de páginas, los encabezados y el cuerpo del texto. Puedes elegir uno de los nombres de estilo predefinidos (por ejemplo, "Encabezado1", "Normal", "Cita") o crea tu propio estilo con su propio nombre. Si tienes muchos estilos para la misma clase de texto, sigue las convenciones para los nombres de estilo tanto como sea posible (por ejemplo, Microsoft Word llama a los estilos de texto para los títulos como "Título1", "Título2", etc., para todos los títulos desde el principal hasta varios niveles de subtítulos). Intenta crear anticipadamente todos los estilos que puedas necesitar así no tendrás que detenerte para crearlos mientras estés escribiendo realmente el manual.

Consejos

Utiliza códigos de campo o variables de texto cuando sea posible. Puedes asignarles valores, como el nombre del producto, del manual o el título del capítulo, y utilizarlos en el documento en lugar de escribir el texto real. Cuando hagas una vista previa o imprimas el manual, el texto aparecerá en lugar de la variable. Si el nombre del producto cambia, o si cambias el nombre del manual o de un capítulo, será más fácil modificar la información cambiando el valor de la variable de texto que buscando y reemplazando las palabras en el documento.

Necesitarás:

- Un procesador de textos o un programa de autoedición.
- Un programa de diseño gráfico o de captura de pantallas.

Área: informática y comunicaciones

UD2

Bibliografía de ampliación

Referencias bibliográficas

- Aubry, C. (2018). *Cree su primer sitio web: Del diseño a la realización*. ENI.

En este libro se abordan gran cantidad de aspectos fundamentales para la elaboración de una página web como el flujo de trabajo, la interfaz de administración, la gestión de archivos multimedia, la apariencia del sitio, la base técnica, el posicionamiento, el diseño de contenido gráfico, los diferentes tipos de contenido web, la ergonomía, la navegación, la accesibilidad, la gestión y la redacción del contenido o el alojamiento.

- Sorzano-Serrano, M. (2019). La influencia de los CMS en la creación de las páginas web. *CLIC*, 3(2), 7.

En este artículo se expone cómo existen muchos puntos a tener en cuenta en el diseño web, siendo la distribución del contenido, la fase del diseño y la interfaz del diseño tres aspectos a los que hay que prestarle gran atención. Además, muestra los principios básicos de diseño que transforman correctamente y ofrecen el contenido de manera comprensible para facilitar la navegación para el sitio web.

- Velázque-Cornejo, B. I., y Hernández-Gracia, J. F. (2019). Marketing de contenidos. *Boletín Científico de la Escuela Superior Atotonilco de Tula*, 6(11), 51-53. Recuperado de: <https://repository.uaeh.edu.mx/revistas/index.php/atotonilco/article/view/3697>

En este ensayo se proporciona información básica acerca del marketing de contenidos, dándose a conocer las ventajas y los errores más comunes entre las empresas y tratando de realizar contenidos para su mercado meta, o sus futuros clientes.

Enlaces de interés

- Delgado, H. (2019, octubre 26). Contenidos en páginas Web - Diseño, redacción y lenguaje. *Akus.Net Diseño web*. Recuperado de: <https://disenowebakus.net/valor-contenidos-sitio-web.php>

Este enlace expone cómo el contenido de la página web dependerá directamente de la temática de esta. De este modo, ofrece importantes tips a tener en cuenta.

- Casas, V. (2021, febrero 12). Partes de una página web: Estructura y contenido. *LucusHost*. Recuperado de: <https://www.lucushost.com/blog/partes-de-una-pagina-web-estructura-y-contenido/>

En este enlace se explican cuáles son las partes de una página web, su estructura y las parten existentes en función del contenido.

- De Souza, I. (2019, mayo 31). Qué tipo de contenido debe tener un sitio web para conquistar a sus visitantes. *Rockcontent*. Recuperado de: <https://rockcontent.com/es/blog/contenidos-para-sitios-web/>

En este enlace se muestran consejos para pensar en el contenido de tu página web y conquistar a sus visitantes.

UD2

Lo más importante

- Hemos creado un documento funcional, al cual se le han aplicado estilos a través de una hoja externa.
- A cada uno de los contenidos se les han aplicado varios formatos, dependiendo del tipo de página para la que ha sido diseñada.
- Finalmente hemos elaborado una guía de usuario.
- Se ha comprobado la validación de los estilos CSS [El Servicio de Validación de CSS](#) y [HTML del W3C](#).

Área: informática y comunicaciones

UD2

Autoevaluación

1. ¿Con qué extensión guardamos nuestra hoja de estilos?
 - a. .html.
 - b. .css.
 - c. .htm.
 - d. .cs.
2. ¿Cómo comienza nuestro documento funcional?
 - a. Con el contenido.
 - b. Con un menú.
 - c. Con un título/logo o encabezado.
 - d. Con enlaces varios.
3. ¿Qué determina las divisiones del documento HTML?
 - a. La URL.
 - b. Las imágenes.
 - c. Los enlaces.
 - d. El elemento div.

4. ¿Cómo hemos aplicado los estilos al documento funcional?
 - a. A través de una hoja de estilos en línea.
 - b. En una hoja incrustada.
 - c. A través de una hoja de estilos externa.
 - d. A través de una hoja de estilos importada.
5. ¿Qué elemento hemos insertado en el documento funcional para que el archivo enlazado a curriculum.pdf se abra en una nueva página?
 - a. No se abre en una nueva página.
 - b. target:"_blank".
 - c. target="blank".
 - d. target=_blank".
6. ¿Qué elemento usamos para que se vea un título al pasar sobre una imagen?
 - a. título.
 - b. title.
 - c. target.
 - d. Width.
7. ¿Qué estilo se aplica al cuerpo del documento funcional para que la posición de la imagen de fondo esté centrada?
 - a. Background-position: center;.
 - b. Background-position: center.
 - c. Background-position: left;.
 - d. Background-position: fixed;.

8. ¿Cómo indicamos un estilo para redondear las esquinas de un borde en CSS 3?
 - a. Border-circle.
 - b. Borlder-round.
 - c. Border-radius.
 - d. Border-top.
9. ¿Qué atributo usamos para aplicar un estilo versalitas al texto?
 - a. Font: variant.
 - b. Font-variant: versalitas;.
 - c. Font-variant: small-caps.
 - d. Font-variant: small-caps;.
10. ¿Qué elemento hemos aplicado en la hoja de estilos para cambiar el color del enlace una vez visitado?
 - a. .menu a:hover.
 - b. .menu a:visited.
 - c. .menu a.
 - d. menu a:visited.

Área: informática y comunicaciones

Glosario

- **Alineación**

Posicionamiento de los elementos respecto a algún bloque.

- **a href**

Atributo que especifica la URL de la página a la que nos lleva un link.

- **Barra lateral**

Bloque de información, generalmente de forma rectangular, adyacente al bloque principal del contenido. Comúnmente se coloca contenido de la página web en cuestión (links internos), publicidad, barras de búsqueda, links externos, etc. También se le conoce como sidebar.

- **Buscador**

Sitios en la red que actúan como herramientas que recopilan y estructuran sistemáticamente la información contenida en ella. Ayudan a los usuarios a buscar datos específicos, para lo cual se organizan en buscadores por palabras o índices temáticos o directorios.

- **Cabecera**

Encabezamiento de una página o una sección de esta. Se suele colocar el nombre de la página, un logotipo, menús de navegación, barras de búsqueda, etc.

– **Class**

Selector CSS que nos permite especificar propiedades de estilo para un grupo de elementos. En HTML se representa como class, en CSS se representa con el símbolo del punto (.).

– **Clear**

Propiedad CSS que especifica que no se deben mostrar elementos flotantes del lado especificado con el selector: left, right o both.

– **DHTML**

Siglas de Dynamic HTML o HTML Dinámico, tecnología que permite dotar a las páginas web de mayor interactividad a través de efectos especiales y animaciones entre otros medios. DHTML integra las tecnologías: HTML tradicional, CSS, DOM y scripts.

– **ECMA**

European Computer Manufacturers Association, o Asociación Europea de Fabricantes de Ordenadores, es una organización que se dedica a aprobar estándares.

– **Elemento**

Cada documento contiene uno o más elementos, cuyos límites están delimitados por etiquetas iniciales y etiquetas finales, o para elementos vacíos por una etiqueta de elemento vacío. Cada elemento tiene un tipo, identificado por un nombre, y debe tener un conjunto de atributos. Cada atributo tiene un nombre y un valor.

– **Elementos a nivel de bloque**

Elementos que comienzan una nueva línea y con tendencia a apilarse a modo de bloque dentro del documento. Los elementos de bloque conforman los componentes principales de la estructura de un documento.

- **Herencia**

En programación orientada a objetos, la capacidad de crear nuevas clases (o interfaces) que contienen todos los métodos y propiedades de otra clase (o interfaz), más métodos adicionales. Si una clase (o interfaz) D hereda de una clase (o interfaz) B, entonces se dice que D está derivada de B. Se dice que B es una clase (o interfaz) base de D. Algunos lenguajes de programación permiten herencias múltiples, desde más de una clase o interfaz.

- **Hipervínculo**

Fragmento de texto, imagen o similar, o contenidos en una página web, que permite acceder a otra parte de esa página o de una página distinta. También se conoce como Link.

- **HTML**

Lenguaje Marcador de Hipertexto. El lenguaje de marcado usado en los documentos Web.

- **Mapa de bits**

Es un tipo de imágenes que se caracteriza por generar imágenes con un ancho y un alto fijos, además de profundidad de color. No se puede aumentar sus dimensiones sin perder calidad. Su extensión es .bmp.

- **Motor de renderizado**

Parte del navegador que se encarga de examinar e interpretar los documentos HTML, CSS, Scripts y demás para después mostrarlo en pantalla.

- **Navegador**

Programa que permite acceder a la web y muestra el contenido de las páginas Web.

- **Pie de página**

Parte inferior de una página web, en que se coloca información como datos de contacto, copyright, autor de la página, redes sociales, etc.

– **Servidor**

Cualquier ordenador en red que ejecute software que le permita responder a las peticiones de documentos y otros datos.

– **Trident**

Motor de renderizado de Internet Explorer, utilizado desde la versión 4.0 del navegador.

Área: informática y comunicaciones

