An illustration of a workspace. In the center is a laptop with a screen showing green code on a dark background. A pair of hands is shown typing on the keyboard and using a mouse. To the left of the laptop is a smartphone. Above the laptop are a yellow envelope, a pair of glasses, and a cluster of four colored pencils (yellow, green, blue, purple). To the right of the laptop are a pair of blue headphones and a white cup of coffee. In the bottom right corner, there is a blue floppy disk with a red label that says '<CODE/>'.

Tutorial para aprender CSS básico desde cero



HOSTING A TOPE

Todo sobre hosting & programación web

Tabla de contenido

¿Qué son las hojas de estilos CSS y para qué sirven?	4
Los elementos básicos CSS: reglas, selectores y propiedades	5
Las reglas	6
Los selectores	6
Las propiedades	6
Las propiedades CSS más usadas	8
Propiedades CSS básicas, comunes a todos los elementos	8
Propiedades CSS para texto	9
Propiedades CSS para colores y fondos	9
Propiedades CSS para bordes	9
Otras propiedades CSS interesantes	9
Cascadas y herencia: reutilizar código forma inteligente	11
Herencia entre elementos HTML	11
Hojas de estilo CSS en cascada	11
Cómo editar los estilos CSS en una página web	12
Estilos CSS inline	12
Hojas de estilo CSS embebidas	12
Hojas de estilo CSS externas	12
Cómo editar los estilos CSS en WordPress	14
Edición de estilos en WordPress.com	14
Edición de estilos en WordPress.org	15
Cómo editar los estilos CSS en Blogger	16
Ejemplo: una página de blog con HTML y CSS	17
Resultado antes de aplicar una hoja de estilo	17
Resultado después de aplicar una hoja de estilo	18
Diseño responsive con media queries	21
Recursos CSS interesantes	24
Manual CSS en PDF y descarga del código	25

Tanto si quieres crear páginas web directamente en HTML, como si usas una plataforma de gestión de contenidos como **WordPress** o **Blogger**, si quieres **manipular el diseño de tu web**, necesitarás usar las **hojas de estilo en cascada CSS** (siglas en inglés de Cascading Stylesheets).

Y es que, al igual que en el caso de HTML, se trata de una tecnología relativamente asequible para una persona no técnica y con un gran “efecto palanca”. Es decir, con aprender tan solo cuatro cosas básicas, se te abrirá **un mundo de posibilidades**.

En el caso de crear páginas web directamente en HTML, la necesidad de usar hojas CSS es obvia. Aquí es simplemente una necesidad y no hay más que hablar.

Pero la gracia de la cuestión está en las plataformas de blogs como **Joomla**, **WordPress** o **Blogger** que son las herramientas que todo el mundo usa hoy en día.

Aquí, para ponerle las cosas más fáciles al usuario, se ha intentado siempre **evitar** el uso de HTML y CSS. Y eso es bueno el 80% del tiempo, pero hay un 20% de ocasiones donde no saber un poquito de HTML y CSS te limitará severamente.

Por citar el ejemplo de WordPress, para el diseño se usan **plantillas** o **temas de WordPress** con **diseños prefabricados**.

Salvo que uses un tema profesional de pago muy bueno como [diviAff], por ejemplo, que te permite manipular el diseño de la plantilla con un potente interfaz visual, te encontrarás con que, en general, estas plantillas **a nivel de interfaz de usuario son bastante rígidas**, especialmente si hablamos de las gratuitas que puedes encontrar en el [repositorio de WordPress](#).

Sin embargo, **a nivel de código**, en cualquier plantilla de WordPress se puede “meter mano” añadiendo hojas de estilo CSS. Además, desde la versión 4.7 de WordPress ya no es necesario hacerlo modificando código o con plugins, sino que añadir CSS personalizado se ha incorporado como una funcionalidad estándar de WordPress. Más de esto más abajo en este post.

En este sentido, en este post no pretendo convertirte en un crack de las hojas CSS, pero sí seguir la famosa regla de Pareto del 80/20 de enseñarte con unas pocas pinceladas lo más importante del mundo CSS y que puedas empezar a aplicarlo ya mismo.

Por otra parte, este post presupone que tienes **conocimientos básicos de HTML**. En caso de que no sea así, léete por favor primero este otro post que te introduce al mundo del HTML:

[Cómo hacer una página web con HTML](#)

¿Qué son las hojas de estilos CSS y para qué sirven?

La idea más importante detrás de las hojas de estilo CSS es **separar el contenido de su presentación**, es decir, de su aspecto visual. En las páginas web el lenguaje HTML se usa para **estructurar** el contenido semánticamente (títulos, subtítulos, texto, etc.) y CSS para la **maquetación y estética** del mismo.

Este principio es fundamental por muchos motivos. Un ejemplo claro es el “diseño adaptativo” (**responsive design**): poder **adaptar el mismo contenido a diferentes dispositivos**. Es decir, que una misma página web se puede visualizar de una manera diferente en un PC que un móvil, optimizada para cada caso.

Los elementos básicos CSS: reglas, selectores y propiedades

Para hacer esto, el concepto fundamental de las hojas CSS son **las reglas**.

Con las reglas podemos definir cosas como éstas:

- *“El título principal de cada página (<h1>) debe estar centrado, ser de color gris oscuro y usar la fuente “Montserrat” con un tamaño de 40 pixeles”*
- *“Los subtítulos de cada página de nivel dos (<h2>) deben estar alineados a la izquierda, color negro ligeramente más claro y usar también el tipo de letra “Montserrat”, pero con un tamaño de 32 pixeles.”*
- *“Las imágenes dentro de la columna principal deben expandirse siempre al ancho máximo de dicha columna.”*
- Etc. etc., etc.

Si te fijas, es lo mismo que hacemos de manera visual con cualquier editor de texto, pero expresado de una manera literal y en un lenguaje que no es nuestro lenguaje natural, sino un **lenguaje especial**, más aptado a un ordenador.

Estas reglas de arriba concretamente traducidas al lenguaje CSS real tendrían este aspecto:

```
h1 {  
    font-family: Montserrat;  
    font-size: 40px;  
    color: #333;  
    text-align: center;  
}
```

```
h2 {  
    font-family: Montserrat;  
    font-size: 32px;  
    color: #444;  
}
```

```
div.contenido img {  
    width: 100%;  
}
```

Con este pequeño ejemplo ya podemos ver perfectamente cómo funcionan las hojas de estilo CSS.

Las reglas

En primer lugar fíjate cómo esta hoja de estilos se divide en tres bloques diferenciados, envueltos en llaves, “{...}”.

Esto son **las reglas**. Cada una implementa una directiva concreta de la pequeña lista de reglas que había formulario de una manera redactada un poco más arriba.

Los selectores

Para especificar a **qué partes de la página HTML** se aplica cada regla en cuestión, le precede **un selector**. Este selector especifica el **ámbito de aplicación de la regla**, es decir, a qué partes de la página aplica.

Si nos imaginamos un documento HTML como sus elementos como un árbol (recuerda que están organizados de manera jerárquica), este ámbito de aplicación van a ser una o varias **ramas** de ese árbol.

En las dos primeras reglas (selectores “h1” y “h2”) el ámbito de aplicación es muy amplio porque los selectores son elementos HTML. Es decir, estamos diciendo que la primera regla se aplique a **todos los elementos <h1>** y que la segunda se aplique a todos elementos <h2>.

El último selector hila un poco más fino: éste nos está diciendo que la tercera regla se aplica solamente a elementos que sean hijos de un elemento <div>. Esto se hace con la posición de los elementos, es decir, al poner “img” a la derecha de “div” estamos diciendo que img tiene que tener un <div> como padre.

Pero, además, con el sufijo de “.contenido” estamos diciendo que ese “div” no puede ser cualquier div, sino que, a su vez, este <div> tiene que pertenecer a la clase “contenido”. Esto nos es útil para diferenciar distintos bloques <div> y en el ejemplo lo usaremos para diferenciar la columna principal de contenido de la columna de la barra lateral.

Podríamos haber hecho lo mismo con cualquier otro elemento HTML. En general, las clases (atributo “class” en HTML) son muy útiles para tratar los mismos elementos HTML de una manera diferenciada según su lugar y función en la página.

Las propiedades

Por último están las propiedades, esto es lo que vemos encerrado dentro de las llaves de una regla.

Las reglas siguen la sintaxis de **[propiedad]:[valor]**.

Por ejemplo:

font-size: 40px;

Aquí estamos diciendo que a los elementos que entran dentro del ámbito de la regla se les aplique un tamaño de letra de 40 pixeles. Los elementos deben ser por tanto elementos tipo texto, es decir,

elementos como `<h1>`, `<h2>`, `<p>`, etc. Pero aunque entre algún elemento dentro del ámbito de la regla que no sea tipo texto, no pasa nada. En ese caso, se ignora esa propiedad concreta y listo.

Las propiedades CSS más usadas

Hay centenares de propiedades CSS, se te hará complicado aprenderlas todas. Pero la buena noticia es que no hace falta, una vez más se cumple la Ley de Pareto del 20/80: sabiendo solo una pequeña parte, sabrás lo más importante.

A continuación te dejo una lista con las que serían, en mi experiencia, **las imprescindibles** que ya te permitirán hacer una infinidad de cosas.

No pretendo que con esta lista las sepas manejar, sino simplemente guiarte con las más importantes para que centres tu atención en ellas en tu proceso de aprendizaje y saques así el máximo provecho a tu tiempo.

Los detalles de cada una los puedes aprender en esta magnífica página:

[Referencia CSS del Mozilla Developer Network](#)

Como indicio de que se trata de una lista bastante completa, en el ejemplo al final de este post vas a ver que no se usa ninguna propiedad que no esté en esta lista. Pero, en cualquier caso, tus sugerencias en los comentarios para refinar esta lista serán muy bienvenidos ya que esta selección no deja de ser una visión subjetiva mía.

Propiedades CSS básicas, comunes a todos los elementos

- **width:** establece la anchura del elemento. Pueden ser valores absolutos o relativos al contenedor del elemento.
- **height:** similar a height para la altura del elemento.
- **margin:** espacio exterior al borde del elemento que mantiene con los elementos vecinos.
- **padding:** espacio entre el borde del elemento y su contenido.

Junto con las propiedades básicas, conviene comentar también el famoso modelo de caja de CSS:

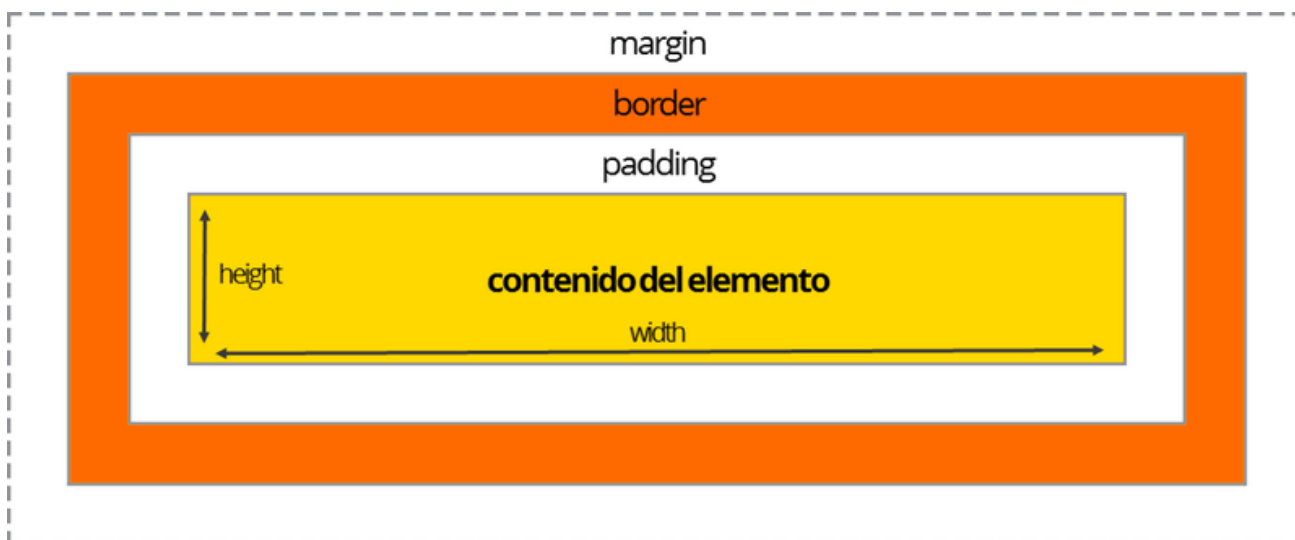


Ilustración 1 – El famoso modelo de caja de CSS.

Los elementos se encuentran en “cajas” con propiedades que son como **las capas de una cebolla** y que dan mucho juego a nivel estético con espacios y bordes (ver más abajo) para el elemento.

En [este tutorial](#) se explican a fondo las ideas detrás del modelo de caja de CSS.

Propiedades CSS para texto

- **font-family**: tipo de letra (nombre del tipo => “Montserrat”, “Open Sans”, etc.).
- **font-size**: tamaño de letra.
- **font-weight**: peso (normal, negrita, ...).
- **font-style**: estilo (normal, cursiva, ...).
- **letter-spacing**: espacio entre letras.
- **line-height**: espacio entre líneas / altura de la línea.
- **text-decoration**: cosas como subrayados, tachados, etc.
- **text-align**: alineación del texto (left, center, right).

Propiedades CSS para colores y fondos

- **color**: color del elemento. Es importante notar que existen diferentes formatos para especificar el color como palabras predefinidas (“red”, “green”, etc.), formato RGB o valor hexadecimal (ver detalles en el material de referencia).
- **background-color**: color del fondo del elemento.
- **background-image**: usar una imagen de fondo.
- **background-repeat**: usar una imagen de fondo como mosaico. Permite diferentes modos de organización de la imagen (ver detalles en material de referencia).
- **opacity**: opacidad del elemento. Va desde 0 (completamente transparente) hasta 1 (sólido). Un valor de 0.5 sería, por tanto, un nivel de transparencia del 50%.

Propiedades CSS para bordes

- **border**: añade un borde a un elemento y especifica sus propiedades (grosor, estilo de línea, etc.). Ver también el modelo de caja de arriba.
- **border-color**: color del borde.
- **border-style**: diferentes estilos (sólido, rayitas, puntos, etc.).
- **border-radius**: redondear las esquinas de un elemento.

Otras propiedades CSS interesantes

- **float**: propiedad avanzada que permite una maquetación más sofisticada. Posicionar los elementos de manera “flotante”. Ver [este tutorial básico](#) sobre posicionamiento flotante.
- **clear**: controla el comportamiento de los elementos adyacentes a elementos posicionados de forma flotante. Ver también el tutorial anterior sobre posicionamiento flotante.
- **overflow**: controla el comportamiento de los contenidos que no caben en su elemento contenedor (valores: visible, hidden, scroll, auto, inherit)

- **display:** controla diferentes aspectos de la visualización de un elemento, permite incluso ocultarlo con el valor “none”.
- **list-style-image:** URL con una imagen que se debe usar como viñeta.
- **list-style-type:** diferentes estilos de viñetas y numeración para los elementos de la lista.
- **box-shadow:** aplicar un efecto de sombra a un elemento.

Cascadas y herencia: reutilizar código forma inteligente

Una de las cosas más útiles de las hojas de estilo CSS es que soportan la idea de **herencia**.

Esta idea se manifiesta básicamente a dos niveles:

1. La herencia de **elementos más generales a más específicos**.
2. El concepto de “**cascada**” de hojas de estilo.

Aquí te voy a dar sólo las pinceladas básicas, para aprender más a fondo sobre esto, te recomiendo [este artículo sobre los conceptos de herencia y cascada en CSS](#).

Herencia entre elementos HTML

Esta idea es muy simple: los elementos que contienen a otros elementos (elementos contenedores o “padres”) transmiten sus propiedades a los elementos hijos.

Por ejemplo:

El elemento **<body>** de una página HTML contiene a todos los demás elementos. Por tanto, si la propiedad CSS “background-color” es “white” en una regla para <body>, todos los demás elementos van a tener por defecto también un fondo blanco, salvo que una regla CSS más específica diga lo contrario. Es decir, **heredan la propiedad del elemento padre**.

Hojas de estilo CSS en cascada

Cuando un sitio web es grande y tiene una maquetación compleja, es inevitable que haya reglas que entren en **conflicto** con otras reglas para un determinado elemento en una o más páginas por asignar diferentes valores a las mismas propiedades.

Aquí hará falta un **mecanismo de arbitraje**, unas reglas o “meta-reglas” podríamos decir, que dictaminen en qué situaciones prevalecen qué reglas sobre un determinado elemento.

Por otra parte, la propia web querrá salirse puntualmente en algunos puntos de las líneas generales de su estilo.

Aquí lo que nos interesará será un mecanismo que nos permita **sobrescribir** o reescribir algunas reglas para algunas páginas especiales.

Ambos problemas se solucionan con el concepto de cascada que por su complejidad también se sale de este post y para el cual te remito de nuevo al [artículo sobre los conceptos de herencia y cascada en CSS](#) que ya mencioné arriba. Simplemente quiero que te quedes con la idea de que estas situaciones se pueden dar y que las hojas CSS ofrecen soluciones para ello.

Cómo editar los estilos CSS en una página web

Ahora que hemos visto qué son las reglas CSS y cómo se usan, veamos cómo se incluyen en una página web.

Para ello hay diferentes opciones:

Estilos CSS inline

Las hojas de estilo inline permiten añadir las reglas CSS directamente como atributo de un elemento HTML.

```
<h1 style="font-family:Montserrat; color:gray">Título principal</h1>
```

Aquí no hay realmente un selector de reglas puesto que la regla ya está dentro del propio elemento. Este tipo de reglas tienen **la mayor prioridad** de todas las reglas.

De todos modos, **hay que evitarlas** y dejarlas sólo para ocasiones muy justificadas porque realmente violan el espíritu reutilización de las mismas reglas para múltiples elementos.

Un ejemplo donde se justificaría es en el email en formato HTML (como el que usa Gmail, por ejemplo.). Aquí no tenemos acceso a la página del cliente de email (la página de gmail.com). No podemos añadirle nuestra propia hoja de estilo, por tanto, solo queda hacerlo desde el propio código HTML.

Hojas de estilo CSS embebidas

Con esta opción, escribimos las reglas dentro de un elemento **<style>** que iría anidado dentro del elemento **<head>** de la página HTML.

Aquí ya logramos reutilizar el código CSS para los elementos de la página, pero tampoco es una solución óptima porque las reglas **sólo serán visibles para esa página** en las que se han declarado y no para las demás, de modo que tendríamos que repetir lo mismo también en las demás páginas.

En realidad, el uso de estilos embebidos sólo se justificaría si tenemos una o unas pocas páginas especiales con estilos que sólo aplican a ellas.

Hojas de estilo CSS externas

Las hojas de estilo externas son la manera de vincular estilos CSS a una página HTML que **se debería utilizar siempre**, salvo que, por razones especiales como las arriba mencionadas, no sea posible. Será también la fórmula que usaremos en el ejemplo completo que viene un poco más abajo.

Esto se hace creando un fichero .css con las reglas CSS (como el del ejemplo que acompaña a este post) y vinculándolo a la página insertando en la página HTML (como hijo del elemento **<head>**) una línea CSS como ésta:

```
<link rel="stylesheet" type="text/css" href="/css/mis-estilos.css" />
```

De este modo se puede usar el mismo fichero en todas las páginas que se quiera y **reutilizamos el código CSS de manera óptima**. Podríamos darle la vuelta por completo a la estética del sitio cambiando tan solo este fichero CSS.

Se pueden usar también varios ficheros a la vez incluyendo varias líneas como la anterior. Como esto puede dar lugar a conflictos si se repitiese alguna regla, pero con diferentes valores para las propiedades, la regla que sigue CSS es que el último fichero (la línea de más abajo) es el más prioritario.

Cómo editar los estilos CSS en WordPress

Si usas WordPress para tu web, la medida en la que vas a poder manipular los estilos CSS de tu sitio van a depender, una vez más, de la versión de WordPress que uses.

Edición de estilos en WordPress.com

Si usas WordPress.com ya sabes que se encuentra **muy limitado** en el plan gratuito e incluso también en los planes de pago básicos.

El tema CSS no es ninguna excepción, aquí te vas a encontrar con las opciones de “*Personalización básica del sitio*” en el plan gratuito y en el personal, y que parecen sugerir la posibilidad de meterle mano a los estilos CSS.

Gratis	Personal	Premium	Business
¿No lo tienes claro? Empieza por aquí.	Ideal para bloggers	Ideal para emprendedores y trabajadores por cuenta propia	Ideal para los propietarios de empresas
€0 para siempre	€2.99 al mes, con facturación anual	€8.25 al mes, con facturación anual	€24.92 al mes, con facturación anual
Pruébalo gratis	Compra de Personal	Compra de Premium	Compra de Business
Ocultar detalles	Ocultar detalles	Ocultar detalles	Ocultar detalles
<ul style="list-style-type: none">✓ Subdominio de WordPress.com✓ Funcionalidades Esenciales de Jetpack✓ Ayuda de la comunidad✓ Cientos de temas gratuitos✓ Personalización básica del diseño	<ul style="list-style-type: none">✓ Nombre de dominio personalizado✓ Funcionalidades Esenciales de Jetpack✓ Ayuda por correo electrónico y chat en directo✓ Cientos de temas gratuitos✓ Personalización básica del diseño	<ul style="list-style-type: none">✓ Nombre de dominio personalizado✓ Funcionalidades Esenciales de Jetpack✓ Ayuda por correo electrónico y chat en directo✓ Temas premium ilimitados✓ Personalización del diseño avanzada	<ul style="list-style-type: none">✓ Nombre de dominio personalizado✓ Funcionalidades Esenciales de Jetpack✓ Ayuda por correo electrónico y chat en directo✓ Temas premium ilimitados✓ Personalización del diseño avanzada

Ilustración 2 - Hasta el plan Premium no se pueden modificar los estilos CSS en WordPress.com

Lamentablemente no es así. Esta opción se refiere solamente a una serie de opciones básicas que se pueden configurar en el tema (más o menos dependiendo del tema elegido), cosas como poder elegir un esquema de colores entre unos pocos o poder cambiar la imagen de cabecera.

Por tanto, aquí no se puede manipular ningún tipo de CSS.

Será a partir del plan Premium cuando se pueden manipular los estilos CSS de tu sitio.

Edición de estilos en WordPress.org

En WordPress.org, por suerte, la cosa cambia radicalmente. Como en todo lo demás, **en WordPress.org la libertad para meterle mano a las CSS es total.**

A esto hay que añadir una mejora que ha introducido la versión 4.7 en la que se ha incluido en el escritorio de WordPress la opción de **añadir directamente CSS personalizado** con una prioridad superior que el CSS por defecto del tema de modo que, en caso de conflicto, prevalecen tus reglas. Es decir, tu código CSS sobrescribe el código por defecto.

Esto se puede hacer en el menú Apariencia/Personalizar en cuya pantalla de edición se ha incluido la opción de “CSS Adicional” y que permite editar el código CSS que quieras añadir.

Cómo editar los estilos CSS en Blogger

Blogger también da una libertad absoluta a la hora de querer manipular los estilos CSS del sitio.

Los pasos son:

1. Entra en Blogger con tu usuario.
2. Elige el blog a actualizar.
3. En el menú de la izquierda, haz clic en Tema.
4. Haz clic en el botón “Personalizar” debajo de "El blog ahora".
5. En el menú arriba a la izquierda, haz clic en “Avanzado”
6. Ahora haz clic en “Añadir CSS”.
7. Añade tus reglas CSS.
8. Pulsa el botón de “Aplicar al blog” arriba en la derecha.



Ilustración 3 - Aplicación de estilos CSS en Blogger.

Ejemplo: una página de blog con HTML y CSS

Como punto final a este post, vamos a ver **un ejemplo sencillo**, pero creo que bastante completo e instructivo para empezar con esto.

Es el mismo ejemplo que ya usamos en el [post de introducción a HTML](#), pero esta vez en lo que nos vamos a fijar es en la parte del CSS.

En el ejemplo he maquetado una página web con el típico estilo de blog: con su cabecera, su columna de contenido principal y una barra lateral (sidebar).

Resultado antes de aplicar una hoja de estilo

De no existir las hojas de estilo CSS, el aspecto de esta página sería bastante “cutre”. Si has vivido la época de los comienzos de Internet a mediados de los noventa, el aspecto de la página te recordará algo a esos tiempos en los que las páginas no usaban CSS:

Cómo crear páginas web bien hechas

1. No te limites a herramientas visuales

Existen [muchas herramientas](#), incluso gratuitas, que te permiten crear paginas web sin saber realmente HTML.

Esto es una manera muy rápida de crear paginas web porque te “oculta” el código y trabajas de una manera similar a como ya estas acostumbrado a hacerlo en los editores de texto. Y realmente está muy bien para hacer, por ejemplo, una primera versión de una página web o partes de ella.

Pero no caigas en la tentación de dejar de aprender HTML a fondo por ello. Esto seria pan para hoy y hambre para mañana porque así siempre tendrás un techo muy bajo, cosas un poco más avanzadas no se pueden hacer así y, por tanto, siempre tendrás un techo muy bajo.

2. Piensa una buena estructura común

Antes de crear una pagina web, procura pensar una estructura lógica común compuesta por etiquetas de divisiones.

Un ejemplo puede ser, por ejemplo, una estructura basica de cabecera / cuerpo /pie.

Cuando quieres aplicar diseño con CSS a la página resultara mucho más fácil y verás cómo una estructura bien pensada te dará mucho más juego paara crear un buen diseño.

3. Diseña "responsive" desde el principio

El % de visitas que se hacen desde dispositivos móviles es cada vez mayor. En muchas webs ya ronda el 25-30% con tendencia al alza. Esto lo deberias tener muy en cuenta.

Por suerte, hacer diseños "responsive" (adaptables a diferentes tipos de pantalla) basicos no es muy difícil. Esta misma página que es muy simple

Para ello lo unico que tienes que hacer es estructurar bien la páginas y usar dimensiones relativas (%) en las divisiones. Esto ya te permitirá hacer páginas con un diseño responsive básico.

Y si quieres saber cómo hacer páginas responsive avanzadas como esas que ocultan o trasladan a otra parte ciertas divisiones segun el tamaño, entonces échala un vistazo a [este enlace](#).



Ilustración 4 – La página antes de aplicarse una hoja de estilo CSS.

Fíjate también como se nos ha ido el banner de [webempresaAff] hacia abajo cuando debería estar en el lado derecho.

Resultado después de aplicar una hoja de estilo

Sin embargo, con una sencilla hoja de estilo que se limita a utilizar las propiedades CSS que he listado en este mismo post, la cosa cambia radicalmente:



Cómo crear páginas web bien hechas

1. No te limites a herramientas visuales

Existen [muchas herramientas](#), incluso gratuitas, que te permiten crear paginas web sin saber realmente HTML.

Esto es una manera muy rápida de crear paginas web porque te "oculta" el código y trabajas de una manera similar a como ya estas acostumbrado a hacerlo en los editores de texto. Y realmente está muy bien para hacer, por ejemplo, una primera versión de una página web o partes de ella.

Pero no caigas en la tentación de dejar de aprender HTML a fondo por ello. Esto seria pan para hoy y hambre para mañana porque así siempre tendrás un techo muy bajo, cosas un poco más avanzadas no se pueden hacer así y, por tanto, siempre tendrás un techo muy bajo.

2. Piensa una buena estructura común

Antes de crear una página web, procura pensar una estructura lógica común compuesta por etiquetas de divisiones.

Un ejemplo puede ser, por ejemplo, una estructura basica de cabecera / cuerpo /pie.



Título del widget

- Item A
- Item B
- Item C

Ilustración 5 - La misma página después de aplicar una hoja de estilos CSS.

¿Cuáles son las diferencias?

Posicionando los bloques de contenido

En primer lugar, hemos preparado el terreno desde la estructura del contenido HTML para crear los diferentes bloques que queremos diferenciar luego en la maquetación usando CSS.

Es decir:

- Una cabecera para el logo
- Un cuerpo con el contenido
 - La columna principal de contenido a la izquierda
 - La columna secundaria o barra lateral a la derecha
- Un pie de página (que no se ve en las capturas, lo puedes ver cuando te descargues el ejemplo)

La manera de modelar esto es usar elementos <div>. Con esto la estructura básica del HTML dentro del <body> de la página (sin el contenido en sí) quedaría así:

```
<body>

  <div class="cuerpo">

    <div class="contenido"></div>
```

```

        <div class="barra-lateral"></div>
    </div>
    <div class="pie">
    </div>
</body>

```

A través de las clases de estos divs aplicamos reglas con propiedades como **float:left** o **float:right** para ir colocando los divs en su sitio. Puedes verlo todo en detalle en el código dl ejemplo.

Descarga el código

Por ejemplo, la columna del contenido principal la ajustamos a un 65% del ancho disponible del <div> padre ("cuerpo") y la "flotamos" a lado izquierdo:

```

.contenido {
    float: left;
    width: 65%;
}

```

Alineando contenidos

Pero no solamente hay que alinear los bloques, sino también los contenidos.

Hemos alineado la cabecera y los títulos principales (<h1>) de manera centrada con la propiedad **text-align:center**.

Cambiando colores y tipos de letra

Una de las cosas que más **impactan** el diseño de una página son los **tipos de letra utilizados**. Una simple página con fondo blanco, pero con unos tipos de letra bien escogidos y ajustando un poco su color, sin más elementos, ya transmite elegancia y buen diseño.

Para el ejemplo he incluido dos tipos de letra de [Google Fonts](#) muy populares y que, además, son gratuitos al 100%, como el resto de fuentes de Google Fonts. Se trata de "Montserrat" y "Open Sans".

Para poder utilizarlos es importante incluir estas líneas al principio de la hoja de estilos:

```

@import url('https://fonts.googleapis.com/css?family=Montserrat');
@import url('https://fonts.googleapis.com/css?family=Open+Sans');

```

Luego utilizando las propiedades **font-family: Montserrat** para los títulos de nivel <h1> y <h2> he asignado esta fuente a estos títulos y usando las propiedades **font-family: Open Sans** con <body> he conseguido que Open Sans sea el tipo de letra por defecto para cualquier otro elemento de texto de la página.

Echa un vistazo a este extracto del código completo (que te puedes descargar):

```
body {  
    font-family: Open Sans;  
    margin: 0px;  
}  
  
h1 {  
    font-family: Montserrat;  
    font-size: 40px;  
    color: #333;  
    font-weight: 600;  
    letter-spacing: -3px;  
    text-align: center;  
}
```

Y por último, fíjate también en el uso de las propiedades **color** y **background-color** en varios elementos como la cabecera y los títulos con las que hemos fijado expresamente los colores usados para diferenciar un poco más la cabecera y para rebajar un poco la intensidad del color negro en los títulos ya que queda más agradable a la vista.

Diseño responsive con media queries

Y para rematar, tenemos que hablar también del **diseño adaptativo** (“responsive design” en inglés).

Hoy en el uso de los móviles y las tablets para navegar por la red está tan extendido que un gran % del tráfico proviene de dispositivos móviles.

En nuestras webs, por ejemplo, según la hora del día, estamos hablando siempre de, mínimo, un 25% con picos cercanos al 50% en determinadas horas del día (por la noche, sobre todo).

Por tanto, si quieres hacer las cosas bien ya no se puede hacer una web sin tener en cuenta la adaptación de la misma a **diferentes tamaños de pantalla**, algo para lo cual las hojas de estilo CSS también están preparadas.

De todos modos, aquí entramos también en temas más avanzados de diseño web. Sin llegar ser complicadísimo, ya es lo suficientemente complejo para salirse claramente del alcance de este post, por eso, quiero limitarme simplemente a explicarte el concepto clave en esto, la **media query**, junto con un ejemplo muy simple.

Media query traducido del inglés quiere decir “consulta del medio” y se refiere a que el navegador web es capaz de saber en qué dispositivo se encuentra y, por supuesto, cuáles son las dimensiones de la ventana en la que se está visualizando una página web.

Esto lo podemos utilizar, para crear reglas CSS condicionadas a estos factores.

Por ejemplo, en la siguiente consulta vamos a suprimir la barra lateral cuando el ancho de página se reduce mucho (cómo sucedería en un móvil, por ejemplo).

Además, por estética, consideramos que los títulos y subtítulos deberían estar centrados, así como la propia columna principal que ya no debería estar alineada (“flotar”) a la izquierda, sino estar también centrada una vez que nos hemos quedado sin barra lateral.

Éste es el resultado, visto con la herramienta “Vista de diseño adaptativo” que viene de fábrica con el navegador FireFox y que se puede activar con Ctrl+Mayus+M:



Ilustración 6 - A partir de un ancho de 640 píxeles hacia abajo, la apariencia de nuestra página se adapta a este menor ancho.

Lo interesante de la herramienta de diseño adaptativo de FireFox es que podemos crear con ella una “ventana” dentro del navegador cuyo tamaño podemos ajustar a la dimensión exacta en píxeles que queramos (ver captura). Esto es simplemente genial para experimentar con diseños CSS responsive para ver si se comportan como esperamos.

Además, permite hacer capturas de pantalla del contenido de esa ventana que guarda automáticamente como imagen PNG.

En el código CSS esto se manifiesta con este bloque especial que, a su vez, contiene reglas CSS:

```
@media (max-width: 640px) { ... }
```

Pues bien, este bloque lo que hace simplemente es decirle al navegador de que en caso de que el ancho de la página sea de 640 píxeles o menos, deben aplicarse las reglas contenidas en dicho bloque y sobrescribir, en su caso, las reglas “normales” con las que entren en conflicto.

En nuestro ejemplo, con la propiedad **display: none** ocultamos la barra lateral, con **text-align: center** centramos los títulos, etc.

Aquí puedes ver todas las reglas CSS para un ancho de página máximo de 640 pixeles o inferior:

```
@media (max-width: 640px) {  
  .barra-lateral {  
    display: none;  
  }  
  
  .cuerpo {  
    max-width: 600px;  
    margin-left: auto;  
    margin-right: auto;  
  }  
  
  .contenido {  
    float: none;  
    width: 100%;  
  }  
  
  h1, h2, h3 {  
    text-align: center;  
  }  
}
```

Recursos CSS interesantes

Como ves, las hojas de estilo CSS son tan imprescindibles en una página web como el propio HTML. Por tanto, si quieres aprender a hacer código para páginas web necesitas tanto HTML como CSS.

Espero que con la lectura de este post de introducción hayas entendido la filosofía de cómo funciona el lenguaje CSS y ya puedas empezar a hacer incluso tus primeros pinitos.

No obstante, necesitarás profundizar un poco más en las propiedades que he listado aquí y practicar algunas horas con ellas antes de coger un poquito de soltura. Pero a partir de ahí, verás como ya se te abrirá **un mundo de posibilidades** en lo que se refiere a diseñar páginas web.

Especialmente interesante es la posibilidad de crearte una pequeña librería de componentes **reutilizables** de código en WordPress con un plugin como [Code Snippets](#). Esto es muy útil para hacerte, por ejemplo, determinados banners, tablas o recuadros que utilices repetidamente en tus contenidos. Nosotros usamos muchísimos estos componentes en nuestras webs.

Para profundizar en el tema, te recomiendo estos recursos:

- [Librosweb](#): un sitio gratuito fantástico para aprender todo sobre HTML y CSS, y mucho más.
- [W3CSchools](#): un sitio de referencia que viene muy bien para consultar puntualmente dudas técnicas muy concretas.
- [CSS Tricks](#): aquí encontrarás ejemplos de código CSS muy originales y muy avanzados en algunos casos. Una fuente fantástica para convertirte en un pro si lo deseas.
- [Zengarden](#): un sitio clásico con muchos diseños para inspirarte. No está tan orientado a aprender a programar los diseños como CSS Tricks, sino mucho más a inspirarte con el resultado final.

Manual CSS en PDF y descarga del código

También te recomiendo una lectura o re-lectura del post “hermano” de este post, ya que aparte de la introducción al mundo del HTML trata también herramientas y otros temas que te serán también muy útiles para trabajar con CSS:

[Cómo hacer una página web con HTML](#)

El código de los ejemplos te lo puedes descargar aquí:

Descarga el código