

# TUTORIAL DE CSS

## COMENZANDO CON HTML + CSS

---

Este breve tutorial está pensado para aquellas personas que quieren comenzar a utilizar CSS y nunca han escrito una hoja de estilos CSS.

No explica mucho sobre CSS. Se centra en cómo crear un archivo HTML, un archivo CSS y cómo hacer que los dos funcionen juntos. Una vez finalizado este tutorial, podréis leer cualquiera de los [otros tutoriales](#) para darle más estilo a los archivos HTML y CSS. También podréis utilizar un [editor de HTML o CSS](#), para desarrollar sitios Web más avanzados.

Al final del tutorial habrás hecho un archivo HTML como éste:

*El resultado será una página HTML, con colores y formato, todo desarrollado con CSS.*

## PASO I: ESCRIBIR EL CÓDIGO HTML

---

Para este tutorial, te sugiero que utilices las herramientas más sencillas. Por ejemplo, Notepad (Windows), TextEdit (Mac) o HTML Kit, serán suficiente. Una vez comprendido lo básico, probablemente se quieran utilizar herramientas más complicadas, o incluso programas comerciales como Style Master, Dreamweaver o GoLive. Pero para el desarrollo de una primera hoja de estilos, es mejor no distraerse con características avanzadas de otras herramientas.

No utilices procesadores de texto como Microsoft Word u OpenOffice. Con ellos, normalmente se generan archivos que los navegadores no pueden interpretar. Para HTML y CSS, lo único que necesitamos son archivos en texto plano.

El paso I consiste en abrir tu editor de texto (Notepad, TextEdit, HTML Kit o el que desees), comenzar con una ventana vacía y escribir lo siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Mi primera página con estilo</title>
```

```
</head>

<body>

<!-- Menú de navegación del sitio --> //comentario//

<ul class="navbar"> //lista no ordenada//

  <li><a href="indice.html">Página principal</a>

  <li><a href="meditaciones.html">Meditaciones</a>

  <li><a href="ciudad.html">Mi ciudad</a>

  <li><a href="enlaces.html">Enlaces</a>

</ul>

<!-- Contenido principal -->

<h1>Mi primera página con estilo</h1>

<p>¡Bienvenido a mi primera página con estilo!

<p>No tiene imágenes, pero tiene estilo.

También tiene enlaces, aunque no te lleven a
ningún sitio...

<p>Debería haber más cosas aquí, pero todavía no sé
qué poner.

<!-- Firma y fecha de la página, ¡sólo por cortesía! --> //comentario//

<address>Creada el 25 de agosto de 2017<br>
  por mí mismo.</address>

</body>

</html>
```

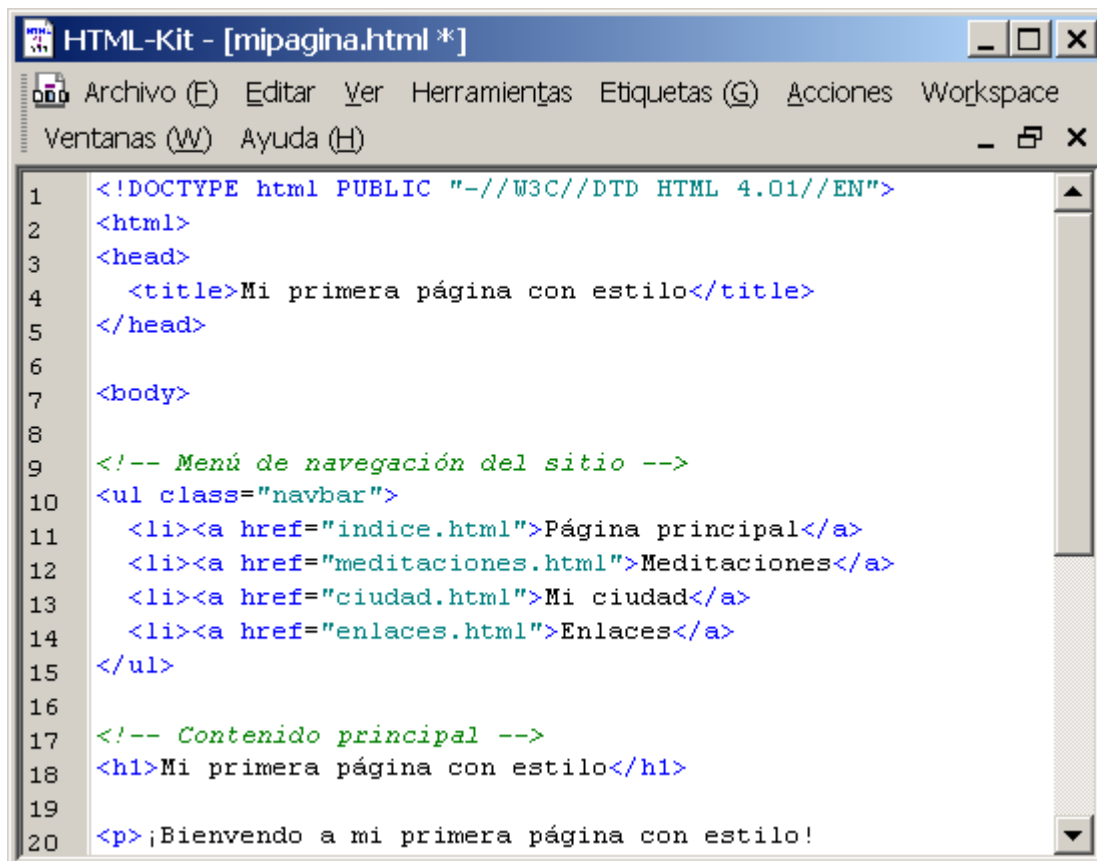
En realidad, no es necesario escribir el código: puedes copiarlo y pegarlo directamente en un editor.

La primera línea que aparece en el archivo HTML, le dice al navegador el tipo de HTML (DOCTYPE significa DOCument TYPE - en español: TIPO de DOCumento). En este caso, se trata de la versión 4.01 de HTML.

Las palabras que se encuentran entre < y > se llaman *etiquetas* (tags) y, como puedes ver, el documento está entre las etiquetas <html> y </html>. Entre <head> y </head> hay espacio para diferentes tipos de información que no aparecerán en la pantalla. Hasta ahora, el documento sólo contiene el título pero posteriormente también se añadirá la hoja de estilos de CSS.

El <body> es dónde se sitúa el texto del documento. En un principio, cualquier cosa que se coloque ahí será mostrado, excepto el texto que esté dentro de las siguientes etiquetas <!-- y -->, las cuales muestran el contenido situado en ese lugar como un comentario de referencia para nosotros mismos. El navegador la ignorará.

De las etiquetas del ejemplo, <ul> crea una “lista desordenada”, es decir, una lista en la cual los elementos no están numerados. La etiqueta <li> indica el comienzo de un “elemento lista”. <p> es un “párrafo”. Y <a> es un “enlace”, que es lo que crea un hipervínculo.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
2 <html>
3 <head>
4   <title>Mi primera página con estilo</title>
5 </head>
6
7 <body>
8
9 <!-- Menú de navegación del sitio -->
10 <ul class="navbar">
11   <li><a href="indice.html">Página principal</a>
12   <li><a href="meditaciones.html">Meditaciones</a>
13   <li><a href="ciudad.html">Mi ciudad</a>
14   <li><a href="enlaces.html">Enlaces</a>
15 </ul>
16
17 <!-- Contenido principal -->
18 <h1>Mi primera página con estilo</h1>
19
20 <p>¡Bienvenido a mi primera página con estilo!
```

Editor mostrando el código HTML.

Si quieres saber lo que significan los nombres que están entre `<...>`, un buen sitio para empezar es [Comenzando con HTML](#). Realizaré algunos comentarios sobre la estructura de la página HTML que estamos utilizando de ejemplo.

- “ul” representa una lista con un hipervínculo por cada elemento. Esto mostrará nuestro “menú de navegación del sitio” con enlaces a otras páginas (ficticias) del sitio Web. Supuestamente, todas las páginas de nuestro sitio Web tienen un menú similar.
- Los elementos “h1” y “p” componen el único contenido de esta página, mientras que la firma al final (“address”) será la misma para todas las páginas del sitio.

Observa que no he cerrado los elementos “li” y “p”. En HTML (pero no en XHTML), se pueden omitir las etiquetas `</li>` y `</p>`, que fue lo que hice aquí, precisamente para hacer el texto más sencillo de leer. Pero si se prefiere pueden ser añadidas.

Vamos a suponer que va a ser una página de un sitio Web que tendrán varias páginas similares. Como es frecuente en páginas Web, ésta tiene un menú con enlaces a otras páginas en el sitio ficticio, un contenido único y una firma.

Ahora, selecciona “Guardar como...” del menú Archivo, ve hasta un directorio/carpeta donde quieras guardar el documento (el escritorio puede ser una opción) y guarda el archivo como “mipagina.html”. No cierres todavía el editor, lo necesitarás otra vez.

Después, abre el archivo en un navegador de la siguiente forma: encuentra el archivo con tu administrador de archivos (Windows Explorer, Finder o Firefox) y haz clic, o doble clic, sobre el archivo “mipagina.html”. El archivo HTML debería abrirse en tu navegador predeterminado. (Si no se abre el documento, abre el navegador y arrastra el archivo sobre él).

## PASO 2: AÑADIR ALGUNOS COLORES

---

Probablemente estés viendo texto negro sobre un fondo blando, pero esto depende de cómo esté tu navegador configurado. Para que la página tenga algo más de encanto podemos añadir algunos colores. (Deja el navegador abierto, lo utilizaremos más tarde).

Comenzaremos con una hoja de estilo interna en el archivo HTML. Más adelante, pondremos el HTML y el CSS en archivos diferentes. La separación de estos archivos es recomendable ya que facilita la utilización de la misma hoja de estilo para diferentes archivos HTML: sólo tienes que escribir la hoja de estilo una vez. Pero en este paso, vamos a dejarlo todo en el mismo archivo.

Necesitamos añadir un elemento `<style>` en el archivo HTML. La hoja de estilo estará en el interior de ese elemento. Volvamos entonces a la ventana

del editor y añadamos las siguientes cinco líneas en la parte de la cabecera del archivo:

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

<html>

<head>

  <title>Mi primera página con estilo</title>

  <style type="text/css">

    body {

      color: purple;

      background-color: #d8da3d }

    </style>

</head>

<body>

[etc.]
```

Las líneas que tienes que añadir están marcadas en rojo. La primera línea dice que eso es una hoja de estilo y que está escrita en CSS ("text/css"). La segunda línea indica que hemos añadido estilo al elemento "body". La tercera línea establece el color del texto como morado y la siguiente línea lo que hace es darle al fondo una especie de amarillo verdoso.

Las hojas de estilo en CSS están compuestas de *reglas*. Cada regla tiene tres partes:

1. el *selector* (en el ejemplo sería: "body"), el cual le dice al navegador la parte del documento que se verá afectada por la regla;
2. la *propiedad* (en el ejemplo, 'color' y 'background-color' son ambas propiedades), las cuales especifican qué aspecto del diseño va a cambiarse;
3. y el *valor* ('purple' y '#d8da3d'), el cual da el valor para la propiedad.

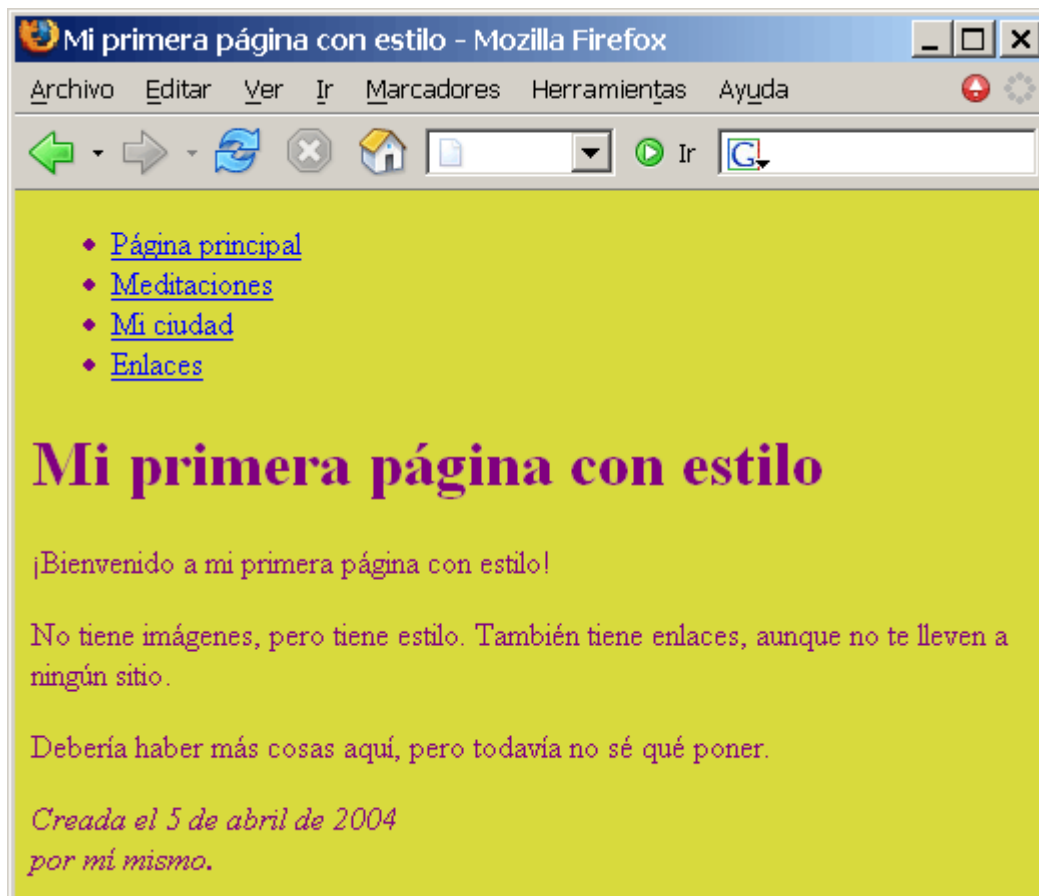
El ejemplo muestra que es posible combinar las reglas. Hemos establecido dos propiedades, por lo que podemos tener dos reglas separadas:

```
body { color: purple }  
  
body { background-color: #d8da3d }
```

ya que las dos reglas van a afectar al elemento body, sólo tenemos que escribir "body" una vez, y poner juntas las propiedades y valores.

El fondo del elemento body será el fondo para todo el documento. A los otros elementos (p, li, address...) no se les ha dado ningún fondo en concreto, por lo que de forma predeterminada no tendrán ninguno (o serán transparentes). La propiedad 'color' establece el color del texto que se encuentra en el elemento body, pero los otros elementos que se encuentran dentro de body heredarán ese color, a no ser que se especifique lo contrario. (Más adelante añadiremos más colores).

Ahora guarda el archivo (utiliza "Guardar" del menú Archivo) y vuelve a la ventana del navegador. Si haces clic en "Actualizar", la página "aburrida" que se mostraba al principio, pasará a ser una página con colores (pero aún sigue siendo muy aburrida). Aparte de la lista de enlaces que se encuentra al principio, el texto debería tener ahora el color morado sobre un fondo de color amarillo verdoso.



*Presentación de la página en el navegador tras añadirle colores.*

Los colores pueden especificarse en la CSS de formas diferentes. El siguiente ejemplo muestra dos de ellas: por el nombre (“purple”) y por el código hexadecimal (“#d8da3d”). Hay aproximadamente 140 nombres de colores y más de 16 millones de colores disponibles en código hexadecimal.

## PASO 3: AÑADIR TIPO DE LETRA

---

Otra cosa que es muy fácil de hacer es añadir diferentes tipos de letra para los diversos elementos de la página. Veamos el texto con el tipo de letra “Georgia”, excepto la cabecera h1, que irá con “Helvetica”.

En la Web, nunca puedes estar seguro de los tipos de letra que los usuarios tienen en sus ordenadores, por lo que añadiremos algunas alternativas: si Georgia no está disponible, Times New Roman o Times también pueden valer, y si ninguna de esas está presente, el navegador puede utilizar cualquier otro tipo de letra con [serifs](#). Si Helvetica no está, Geneva, Arial y SunSans-Regular son bastante similares, y si ninguna de estas funciona, el navegador puede escoger cualquier otro tipo de letra que no sea serif.

Añade en el editor de texto las siguientes líneas:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Mi primera página con estilo</title>
  <style type="text/css">
    body {
      font-family: Georgia, "Times New Roman",
        Times, serif;
      color: purple;
      background-color: #d8da3d }
    h1 {
      font-family: Helvetica, Geneva, Arial,
```

```
SunSans-Regular, sans-serif }

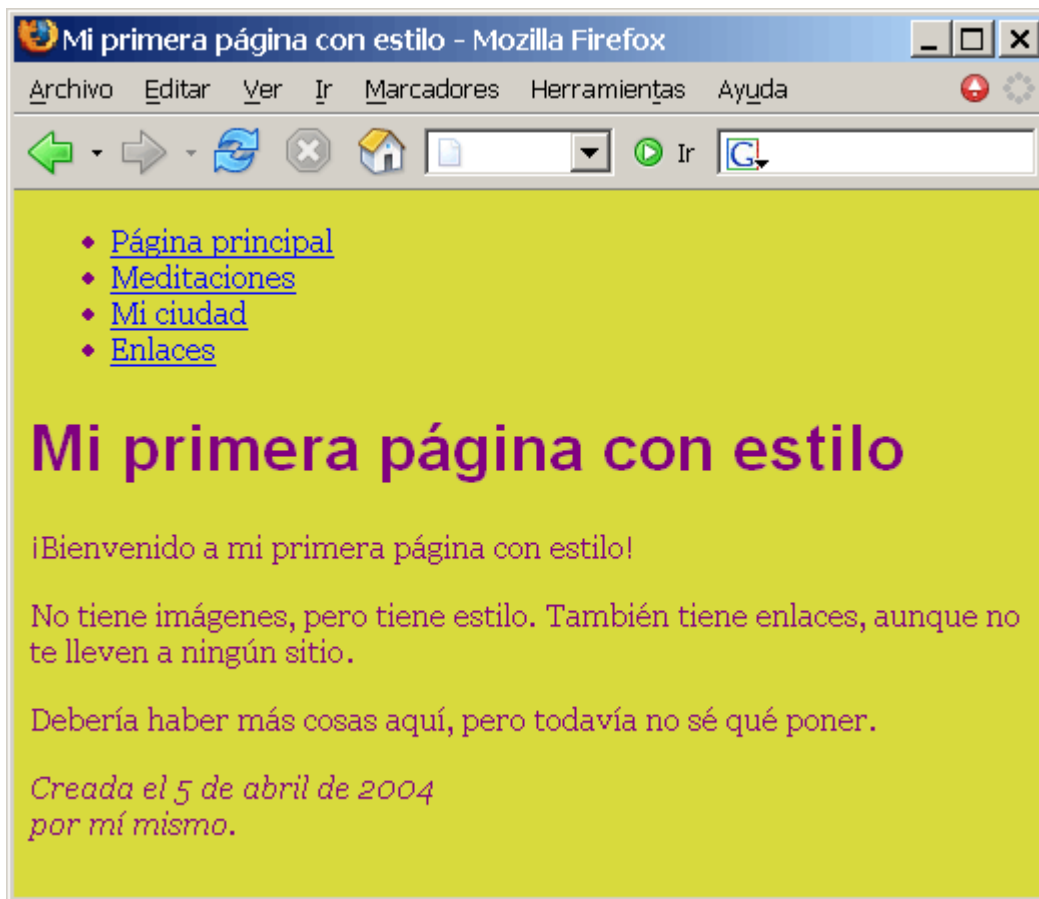
</style>

</head>

<body>

[etc.]
```

Si guardas otra vez el archivo y haces clic en "Actualizar" en el navegador, deberían aparecer tipos de letra diferentes para el encabezado y para el otro texto.



*Ahora el texto tiene un tipo de letra diferente al del encabezado.*

## **PASO 4: AÑADIR UNA BARRA DE NAVEGACIÓN**



La lista que aparece al principio de la página HTML será más adelante un menú de navegación. Algunos sitios Web tienen un menú al principio o en un lado de la página, y éste debería tenerlo también. Pondremos uno a la izquierda, ya que es más interesante ponerlo ahí que arriba...

El menú ya está en la página HTML. Es la lista <ul> que se encuentra al principio. Los enlaces que tiene no funcionan ya que nuestro "sitio Web", hasta el momento, sólo tiene una página, pero eso no es importante ahora. Por supuesto, en un sitio Web real no debería haber ningún enlace roto.

Tenemos que mover la lista a la izquierda y colocar el resto del texto un poco más hacia la derecha, para dejar espacio para el menú. Las propiedades de CSS que vamos a utilizar son 'padding-left' (para mover el texto del elemento body) y 'position' (para mover el menú).

Hay otras formas de hacerlo. Si buscas "column" y "layout" en la página [Aprender CSS](#), encontrarás algunas plantillas listas para ser utilizadas. Pero, por el momento, ésta está bien.

En la ventana del editor, añade las siguientes líneas al archivo HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Mi primera página con estilo</title>
  <style type="text/css">
    body {
      padding-left: 11em;

      font-family: Georgia, "Times New Roman",
        Times, serif;

      color: purple;

      background-color: #d8da3d }

    ul.navbar {

      position: absolute;

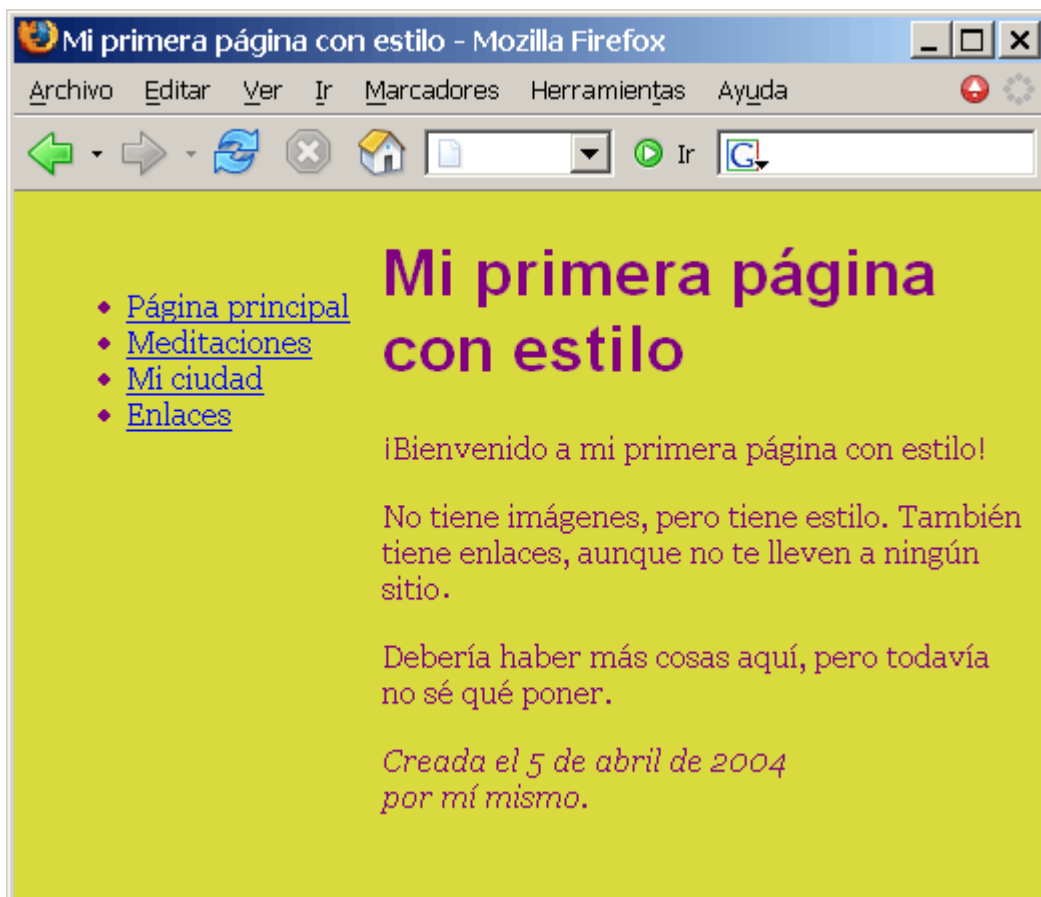
      top: 2em;

      left: 1em;

      width: 9em }
```

```
h1 {  
    font-family: Helvetica, Geneva, Arial,  
        SunSans-Regular, sans-serif }  
  
</style>  
</head>  
  
<body>  
[etc.]
```

Si guardas otra vez el archivo y lo actualizas en el navegador, deberías tener la lista de enlaces a la izquierda del texto principal. Ahora el resultado parece mucho más interesante ¿verdad?



'position: absolute' indica que el elemento ul está situado independientemente de cualquier texto que esté antes o después en el documento. 'left' y 'top' indican cuál es la posición. En este caso, está a 2em de la parte de arriba de la ventana y a 1em del lado izquierdo.

'2em' significa 2 veces el tamaño del tipo de letra actual. Por ejemplo, si el menú se muestra con un tipo de letra de 12 puntos, entonces '2em' es 24 puntos. 'em' es una unidad muy útil en CSS ya que se puede adaptar automáticamente al tipo de letra que el usuario esté utilizando. La mayoría de los navegadores tienen un menú para aumentar o disminuir el tamaño de letra: puedes probarlo y ver la forma en la que el menú aumenta de tamaño a la vez que lo hace el tipo de letra, lo que no ocurriría si hubiéramos utilizado un tamaño en píxeles.

## PASO 5: DAR ESTILO A LOS ENLACES

---

El menú de navegación todavía parece una lista en lugar de un menú. Vamos a cambiar su aspecto. Quitaremos los puntos que indican los diferentes elementos de la lista y moveremos los elementos a la izquierda, donde estaban antes los puntos. También le daremos a cada elemento un fondo blanco y un cuadrado negro. (¿Por qué? Por nada en particular, sólo porque podemos hacerlo).

No hemos especificado el color de los enlaces, vamos a añadirlo también: utilizaremos el azul para aquellos enlaces que el usuario no ha visitado todavía, y morado para los enlaces que ya ha visitado:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Mi primera página con estilo</title>
  <style type="text/css">
    body {
      padding-left: 11em;
      font-family: Georgia, "Times New Roman",
        Times, serif;
      color: purple;
      background-color: #d8da3d }
    ul.navbar {
```

```
list-style-type: none;

padding: 0;

margin: 0;

position: absolute;

top: 2em;

left: 1em;

width: 9em }

h1 {

font-family: Helvetica, Geneva, Arial,

SunSans-Regular, sans-serif }

ul.navbar li {

background: white;

margin: 0.5em 0;

padding: 0.3em;

border-right: 1em solid black }

ul.navbar a {

text-decoration: none }

a:link {

color: blue }

a:visited {

color: purple }

</style>

</head>

<body>

[etc.]
```

Normalmente, los navegadores muestran los hipervínculos subrayados y con color. Los colores son parecidos a los que especificamos aquí: azul para enlaces a páginas que no has visitado todavía (o visitaste hace mucho tiempo), morado para páginas que ya has visitado.

En HTML, los hipervínculos se crean con elementos <a>, por lo que para especificar el color, necesitamos añadir una regla de estilo para "a". Para diferenciar los enlaces visitados de los no visitados, CSS proporciona dos "pseudo-classes" (:link y :visited). Se llaman "pseudo-classes" para distinguirlas de los *atributos* de clase, que aparecen directamente en HTML, por ejemplo, `class="navbar"`.

## PASO 6: AÑADIR UNA LÍNEA HORIZONTAL

---

El último elemento a añadir en la hoja de estilo es una regla horizontal para separar el texto de la firma que se encuentra al final. Utilizaremos 'border-top' para añadir una línea punteada encima del elemento <address>:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Mi primera página con estilo</title>
  <style type="text/css">
    body {
      padding-left: 11em;
      font-family: Georgia, "Times New Roman",
        Times, serif;
      color: purple;
      background-color: #d8da3d }
    ul.navbar {
      list-style-type: none;
      padding: 0;
      margin: 0;
      position: absolute;
      top: 2em;
      left: 1em;
```

```
width: 9em }

h1 {
  font-family: Helvetica, Geneva, Arial,
              SunSans-Regular, sans-serif }

ul.navbar li {
  background: white;
  margin: 0.5em 0;
  padding: 0.3em;
  border-right: 1em solid black }

ul.navbar a {
  text-decoration: none }

a:link {
  color: blue }

a:visited {
  color: purple }

address {
  margin-top: 1em;
  padding-top: 1em;
  border-top: thin dotted }

</style>

</head>

<body>

[etc.]
```

Ahora, ya hemos terminado de darle estilo a la página. Lo siguiente es ver cómo se pone la hoja de estilo en un archivo diferente para que de esta forma otras páginas puedan utilizar el mismo estilo.

## PASO 7: PONER LA HOJA DE ESTILO EN UN ARCHIVO SEPARADO

---

Ahora tenemos un archivo HTML con una hoja de estilo en su interior. Pero si nuestro sitio crece desearemos añadir más páginas que compartan el mismo estilo. Hay un método más adecuado que copiar la hoja de estilo dentro de cada página y que es: colocar la hoja de estilo en un archivo separado, haciendo que todas las páginas estén enlazadas a esa hoja.

Para separar la hoja de estilo del archivo HTML, necesitamos crear otro archivo de texto, que esté vacío. Puedes escoger "Nuevo" del menú Archivo del editor para crear una ventana vacía.

Después, corta y pega en la nueva ventana todo lo que hay dentro del elemento `<style>` del archivo HTML. No copies las etiquetas `<style>` y `</style>`. Éstas pertenecen a HTML, no a CSS. En la nueva ventana del editor, deberías tener la hoja de estilo completa:

```
body {
    padding-left: 11em;

    font-family: Georgia, "Times New Roman",
        Times, serif;

    color: purple;

    background-color: #d8da3d }

ul.navbar {

    list-style-type: none;

    padding: 0;

    margin: 0;

    position: absolute;

    top: 2em;

    left: 1em;

    width: 9em }

h1 {

    font-family: Helvetica, Geneva, Arial,
        SunSans-Regular, sans-serif }
```

```
ul.navbar li {  
  
    background: white;  
  
    margin: 0.5em 0;  
  
    padding: 0.3em;  
  
    border-right: 1em solid black }  
  
ul.navbar a {  
  
    text-decoration: none }  
  
a:link {  
  
    color: blue }  
  
a:visited {  
  
    color: purple }  
  
address {  
  
    margin-top: 1em;  
  
    padding-top: 1em;  
  
    border-top: thin dotted }
```

Elige “Guardar como...” del menú Archivo, comprueba que está en el mismo directorio/carpeta que el archivo mipagina.html, y guarda la hoja de estilo como “miestilo.css”.

Vuelve a la ventana que muestra el código HTML. Borra todo lo que hay desde la etiqueta <style> hasta </style>, ambas inclusive, y reemplázalo por un elemento <link> de la siguiente forma:

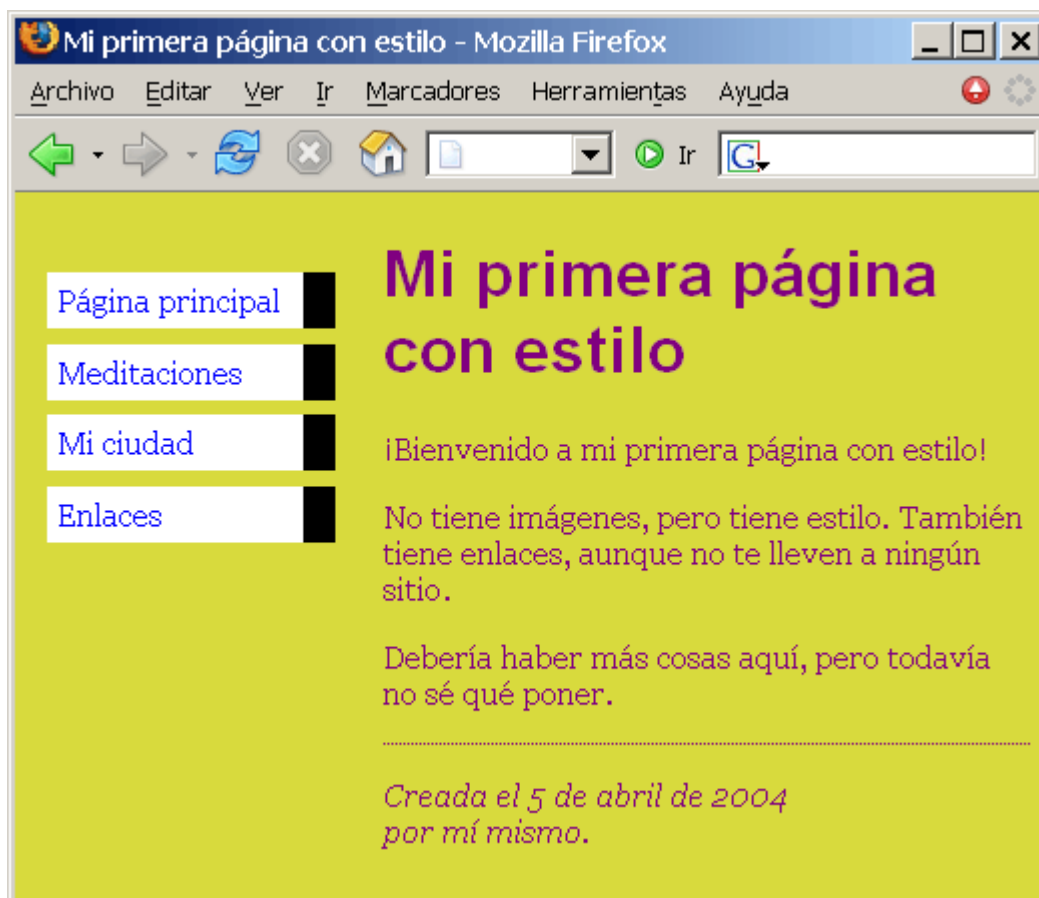
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">  
  
<html>  
  
<head>  
  
    <title>Mi primera página con estilo</title>  
  
    <link rel="stylesheet" href="miestilo.css">  
  
</head>  
  
<body>
```



[etc.]

De esta forma, se le indicará al navegador que la hoja de estilo se encuentra en el archivo llamado "miestilo.css". Al no especificarse ningún directorio, el navegador mirará en el mismo directorio en el que se encuentra el archivo HTML.

Si has guardado el archivo HTML y lo has actualizado en el navegador, no deberías apreciar ningún cambio en el aspecto de la página. Ésta sigue guardando el mismo estilo, pero ahora viene establecido por el archivo externo.



*Resultado final*

El siguiente paso es poner ambos archivos, mipagina.html y miestilo.css, en vuestro sitio Web. (Quizá queráis primero cambiarlos un poco...), pero la forma de realizar esto depende de vuestro proveedor de Internet.

<https://www.w3.org/Style/Examples/011/firstcss.es.html#reading>

<https://www.hazunaweb.com/curso-de-html/listas/>