



APRENDERAPROGRAMAR.COM

EFFECTOS CSS. BORDES,
MÁRGENES, FUENTES,
ANIMACIONES... ¿DÓNDE
ESTÁ LA FRONTERA ENTRE
CSS, HTML Y
PROGRAMACIÓN?
(CU01005D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº5 del Tutorial básico: "CSS desde cero".

Autor: César Krall

EFFECTOS CSS. UBICAR LA FRONTERA

En el anterior epígrafe del curso hemos indicado que CSS no es un lenguaje de programación pero que en cierto sentido se entremezcla con éstos. A veces nos encontraremos que se puede lograr un mismo efecto usando HTML, usando CSS ó usando un lenguaje de programación. ¿Por qué tantas formas para hacer una misma cosa? ¿Dónde está la frontera entre cada lenguaje?



Esta pregunta no es de fácil respuesta. Vamos a ver con un ejemplo lo que puede ocurrir para algo tan sencillo como aplicar algunos efectos a un texto. No obstante, ten en cuenta que este ejemplo relativo a texto podría aplicarse a otros conceptos como bordes, márgenes, animaciones, etc.

El lenguaje HTML permite aplicar algunos efectos visuales al texto. Escribe o copia este código y guárdalo en un archivo de nombre ejemplo1.html.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo HTML aprenderaprogramar.com</title>
    <meta name="tipo_contenido" content="text/html;" http-equiv="content-type" charset="utf-8">
  </head>
  <body>
    <p>Negrita: <strong>Quiero aprender a programar</strong></p>
    <p>Itálica: <i>Quiero aprender a programar</i></p>
    <p>Tachado: <del>Quiero aprender a programar</del></p>
    <p>Color fuente: <font color="green">Quiero aprender a programar</font></p>
  </body>
</html>
```

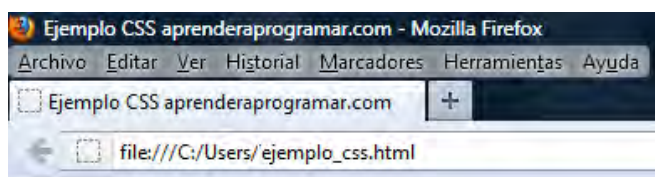
Con Javascript podemos hacer algo parecido. Escribe o copia este código y guárdalo en un archivo de nombre ejemplo2.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo Javascript aprenderaprogramar.com</title>
    <meta name="tipo_contenido" content="text/html;" http-equiv="content-type" charset="utf-8">
  </head>
  <body>
    <script>
      var txt = "Quiero aprender a programar";
      document.write("<p>Negrita: " + txt.bold() + "</p>");
      document.write("<p>Itálica: " + txt italics() + "</p>");
      document.write("<p>Tachado: " + txt.strike() + "</p>");
      document.write("<p>Color fuente: " + txt.fontcolor("green") + "</p>");
    </script>
  </body>
</html>
```

Y por último en vez de aplicar Javascript o simple HTML, podemos usar CSS. Escribe o copia este código y guárdalo en un archivo de nombre ejemplo3.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo CSS aprenderaprogramar.com</title>
    <meta name="tipo_contenido" content="text/html;" http-equiv="content-type" charset="utf-8">
    <style type="text/css">
      #negrita{font-weight:bold;}
      #italica{font-style:italic;}
      #tachado{text-decoration: line-through;}
      #verde{color:green;}
    </style>
  </head>
  <body>
    <p>Negrita: <span id="negrita">Quiero aprender a programar</span></p>
    <p>Italica: <span id="italica">Quiero aprender a programar</span></p>
    <p>Tachado: <span id="tachado">Quiero aprender a programar</span></p>
    <p>Color fuente: <span id="verde">Quiero aprender a programar</span></p>
  </body>
</html>
```

Haz doble click sobre cada uno de los archivos para visualizar el resultado en un navegador. El resultado que obtenemos es algo similar a esto:



Negrita: **Quiero aprender a programar**

Italica: *Quiero aprender a programar*

Tachado: ~~Quiero aprender a programar~~

Color fuente: **Quiero aprender a programar**

El único código que debemos entender por el momento es el correspondiente al ejemplo 1, ya que es código HTML que ya debemos conocer. El código del ejemplo 2 y ejemplo 3 no te preocupes si no lo entiendes ya que el objetivo ahora no es comprender ese código, sino simplemente ver cómo podemos alcanzar un mismo objetivo usando distintos lenguajes como HTML, Javascript ó CSS.

Además si nos fijamos, el código Javascript y el código CSS está dentro de un documento HTML (aunque podrían estar en archivos separados).

Todo esto nos puede llevar a preguntarnos: ¿Por qué se entremezclan unos lenguajes con otros? La respuesta sería histórica y técnica: HTML se convirtió en la forma de crear páginas web, pero tenía

muchas limitaciones. En un momento dado, se consideró que entremezclar (embeber) lenguajes entre sí podía ser una buena opción técnica para resolver problemas o hacer cosas que no era posible o conveniente hacer con HTML. Así, podemos embeber Javascript en HTML ó embeber CSS en HTML, o embeber HTML en PHP, etc. Por ello a veces ocurre que no hay una frontera clara entre lenguajes de programación, HTML y CSS. Esto, que puede resultar un tanto confuso inicialmente, se va convirtiendo en “comprensible” a medida que se trabaja y se aprende más sobre estos lenguajes.

Por otro lado, ¿por qué tantas vías distintas para hacer algo cuando quizás que solo hubiera una manera de poner el texto en negrita, o una sola manera de poner un color de fuente, sería más simple?

Para esto podemos citar varios motivos:

- a) **Motivos históricos:** a veces las cosas se empezaron a hacer de una manera y luego se pensó que era mejor hacerlas de otra. Sin embargo, para evitar que las páginas web existentes dejaran de funcionar, se siguieron permitiendo formas de hacer las cosas “anticuadas”. Por ejemplo la etiqueta `<strike> ... </strike>` en HTML se considera deprecated (obsoleta, de uso no recomendado) en HTML 4.01 y no está admitida en HTML 5. Sin embargo, se sigue usando. Muchas formas de hacer las cosas se admiten aunque no estén recomendadas.
- b) **Motivos de independencia de tecnologías:** HTML es una cosa y Javascript es otra, aunque en la práctica nos encontremos con que Javascript se puede “entremezclar” (embeber) en HTML. Podríamos hacer cosas en Javascript y no querer usar otro lenguaje, es decir, podríamos tratar de hacer cosas independientes sin “entremezclar” tecnologías. Por ello lenguajes como Javascript ó PHP incorporan posibilidades para hacer cosas que ya se pueden hacer de otra manera. De esta forma tienen la potencialidad de ser más independientes.
- c) **Motivos de políticas de desarrollo y estándares:** quizás no te lo hayas preguntado nunca, pero conviene tener al menos una orientación al respecto: ¿Quién define qué es un lenguaje como HTML, CSS, PHP, cómo se debe escribir, qué resultado debe generar cada etiqueta o instrucción, etc.? En general detrás de los lenguajes, aunque algunos fueron creados por personas individuales, hay empresas, comunidades de desarrollo, asociaciones, consorcios internacionales, comités, etc. En ocasiones un grupo de personas no está de acuerdo con la forma en que se está creando una especificación del lenguaje y forman grupos de trabajo alternativos que definen estándares alternativos. A veces triunfa un estándar y el otro se desecha, pero otras veces conviven distintos estándares que permiten hacer las cosas de distintas maneras. Para los creadores de páginas web esto resulta negativo, pero ¡así es la vida!
- d) **Otros motivos:** podríamos abundar en el por qué de que las cosas sean como son, pero con tener una idea general nos basta.

Acostúmbrate a pensar que los desarrollos web no son matemáticas. Las cosas se pueden hacer de muchas maneras, y de hecho muchas veces se hacen “de mala manera” por desconocimiento, por prisas, o por ser más fácil.

Acostúmbrate a pensar que los desarrollos web usan distintos lenguajes que muchas veces se entremezclan entre sí hasta el punto de ser difícil distinguir en qué corresponde a un lenguaje y qué

corresponde a otro. Hay lenguajes comunes en los desarrollos web como HTML, pero por ejemplo en cuanto a lenguajes de programación no todos los programadores usan el mismo.

Acostúmbrate a encontrarte con que a veces las cosas no funcionan como una esperaría que lo hicieran, no debido a que se haya escrito mal el código o usado mal una instrucción, sino debido a que en el mundo de internet existen distintos estándares y distintas versiones. A veces aunque nos esforcemos porque todo se vea como nosotros esperamos en todos los navegadores o dispositivos, es difícil conseguirlo. Es un poco caótico, pero es así.

En este curso más que aprendernos todas las instrucciones, estándares, etc. vamos a tratar de ser capaces de razonar el por qué de las cosas, y a tratar de esforzarnos por saber cómo generar código limpio, bien estructurado y de calidad.

Próxima entrega: CU01006D

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203