



APRENDERAPROGRAMAR.COM

CONCEPTO DE HERENCIA
EN HOJAS DE ESTILO EN
CASCADA CSS. ¿QUÉ ES?
FORZAR HERENCIA CON
INHERIT. EJEMPLOS.
(CU01013D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº13 del Tutorial básico: "CSS desde cero".

Autor: César Krall

HERENCIA EN CSS

Hemos visto cómo aplicar estilos CSS a todas las etiquetas HTML de un tipo, o bien a una parte del documento HTML con un nombre único (id), o bien en distintas partes de un documento o proyecto web usando class. Una característica interesante de CSS es cómo nos permite definir estilos en clases superiores o clases padre que se van transmitiendo hacia las subclases o clases hijas.



DEFINIR CLASES QUE SOLO SEAN APLICABLES A ETIQUETAS ESPECÍFICAS

Partimos del código que hemos utilizado en la anterior entrega del curso. En concreto habíamos definido una clase CSS cuyo nombre era "destacado" y cuyo código era:

```
/* Curso CSS estilos aprenderaprogramar.com */  
.destacado { background-color: orange; font-weight:bold; font-size:100%; }
```

Podemos definir que una clase solo sea aplicable a una etiqueta específica. Por ejemplo si modificamos el código que define la clase destacado y escribimos:

```
/* Curso CSS estilos aprenderaprogramar.com */  
p.destacado{ background-color: orange; font-weight:bold; font-size:100%; }
```

Ahora el estilo destacado sólo surte efecto cuando va en una etiqueta p, en cambio en otras etiquetas como div, span, a, etc. no tendría efecto.

Esto nos permitiría usar el mismo nombre de clase y obtener distintos efectos según en qué etiqueta aplicáramos la clase. Por ejemplo:

```
/* Curso CSS estilos aprenderaprogramar.com */  
p.destacado { background-color: orange; font-weight:bold; font-size:100%; }  
div.destacado {background-color: blue; font-weight:bold; font-size:100%; }
```

Ahora si definimos un párrafo con el atributo class = "destacado" su fondo será naranja. En cambio si es un divisor div con el atributo class = "destacado" su fondo será azul. Prueba a ejecutar código comprobando estas diferencias.

DEFINIR SUBESTILOS DENTRO DE UNA CLASE

Podemos definir que determinadas cajas dentro de otra principal tengan estilos diferenciados. Por ejemplo, podemos querer que los párrafos de clase "destacado" tengan fondo naranja, pero que si

dentro del párrafo hay un link (etiquetas <a> ...) dicho link tenga un estilo específico, por ejemplo que tenga fondo amarillo y un tamaño un 20% superior a lo normal. Para ello escribiríamos:

```
/* Curso CSS estilos aprenderaprogramar.com*/
p.destacado{
background-color: orange;
font-weight:bold;
font-size:100%; }
p.destacado a { background-color: yellow;
font-size:120%;
}
```

Con la declaración p.destacado a { ... } estamos indicando que los links dentro de párrafos con estilo destacado estarán en negrita (**por herencia** del estilo general aplicable a la clase destacado) y tendrán fondo amarillo y tamaño de letra un 20% superior a lo normal (ya que hemos sobrescrito las propiedades de color de fondo y tamaño de letra de la clase padre).

Si quisiéramos definir un estilo destacado para los párrafos, otro para los div, pero el mismo subestilo para los links que estén tanto dentro de párrafos como dentro de divs escribiríamos lo siguiente:

```
/* Curso CSS estilos aprenderaprogramar.com*/
p.destacado{ background-color: orange; font-weight:bold; font-size:100%; }
div.destacado { background-color: blue; font-weight:bold; font-size:100%; }
p.destacado a, div.destacado a { background-color: yellow; font-size:120%; }
```

Con la declaración p.destacado a, div.destacado a { ... } aplicamos el mismo estilo a los link dentro de párrafos o div que lleven el atributo class = "destacado".

Se observa una cosa: podemos definir estilos agrupando el código de distintas maneras. Podemos repetir código pero también podemos crear clases padre que agrupen las características comunes y definir en clases hijo las características específicas. Veamos el siguiente código:

```
/* Curso CSS estilos aprenderaprogramar.com*/
.destacado {font-weight:bold; font-size:100%;}
p.destacado {background-color: orange;}
div.destacado { background-color: blue; }
p.destacado a, div.destacado a {background-color: yellow; font-size:120%; font-weight:lighter; }
```

Este código define: una clase padre o superclase <<destacado>> según la cual la letra será negrita y el tamaño de letra normal (100 %). Las clases hijas <<p.destacado>> y <<div.destacado>> aportan las características de la clase padre (por herencia) más algunas otras características adicionales (añadidas) o distintas (sobrescritura de propiedades de la clase padre), y las clases hijas de las hijas <<p.destacado a>> y <<div.destacado a>> tienen las características de la clase padre y de la clase abuela con algunas características adicionales o distintas.

Observamos aquí una propiedad importante de CSS: la herencia o transmisión de características de clases antecesoras a clases sucesoras o de clases padre a clases hija.

CONCEPTO DE HERENCIA CSS

La herencia CSS permite evitar la repetición de código en todas las subclases que derivan a partir de una clase superior. La herencia se aplica tanto a estilos de etiquetas HTML (por ejemplo todas las etiquetas son subetiquetas de la etiqueta padre body, de modo que un estilo aplicado a body será heredable por el resto de etiquetas HTML) como a selectores id o class.

Esto nos da alguna pista sobre prácticas CSS que son habituales entre programadores o diseñadores CSS experimentados: si aplicamos algunos estilos básicos como un tipo de letra (propiedad CSS font-family) a la etiqueta body y todas las otras etiquetas son herederas del estilo de body, habremos conseguido definir el tipo de letra en toda la página web en una simple línea, sin necesidad de tener que estar repitiendo el estilo en múltiples ocasiones para párrafos, div, span, forms, etc.

Ejemplo de declaración de este tipo:

```
body {font-family: Arial, sans-serif; }
```

En un documento HTML todo tiene ascendiente excepto el elemento html. Así body hereda de html, div hereda de body y <<div a>> (link dentro de un contenedor div) hereda de div. En nuestro caso el estilo <<p.destacado a>> hereda de <<p.destacado>> y <<p.destacado>> hereda de <<destacado>>.

Sin embargo no todas las propiedades se heredan. Por ejemplo la propiedad font-family es heredable, pero los márgenes (propiedad margin) no se heredan excepto si se indica explícitamente que deben heredarse. ¿Por qué unas propiedades se heredan y otras no? Los creadores de CSS pensaron que sería útil que algunas propiedades se heredaran porque suelen mantenerse constantes dentro de una caja. Por ejemplo lo más normal es usar un tipo de letra en un contenedor y no mezclar distintos tipos de letra. En cambio, otras propiedades era poco lógico que se heredaran, como los márgenes, ya que por ejemplo las fotografías dentro de un artículo normalmente tenían un margen distinto al que tenía el propio artículo respecto a su contenedor o los párrafos. Nos encontramos así con que la herencia de propiedades se rige en cierta manera por el "sentido común". No obstante, en la especificación oficial de CSS encontraremos que cuando se define una propiedad se establece un atributo denominado "inherit" que puede estar establecido en "yes" o en "no" según la propiedad de que se trate. Así font-family es una propiedad cuyo valor inherit es "yes": esta propiedad se hereda. En cambio la propiedad margin tiene un valor inherit "no", lo cual significa que no se hereda. Más adelante abordaremos las propiedades CSS, explicaremos cómo consultar la documentación oficial CSS y estudiaremos esta circunstancia con más detalle.

Hay algunas curiosidades en torno a la herencia. Por ejemplo no todos los navegadores responden exactamente igual en cuanto a la herencia. Podrías encontrarte algún navegador donde no se genera herencia de una propiedad mientras que en otro sí.

Otra curiosidad es algo que hemos comentado: hemos dicho que la propiedad background-color se transmite por herencia. Sin embargo la definición del estándar CSS dice que background-color tiene un valor inherit "no". ¿Cómo se explica esto? Lo veremos más adelante cuando hablemos de las propiedades CSS, pero baste decir que el color de fondo por defecto para todos los elementos es el transparente. Al tener el padre un color y superponer el hijo transparente el efecto visual es que se hereda el color del padre, aunque formalmente no es así.

FORZAR LA HERENCIA CON INHERIT

Hay una manera explícita para indicar que la propiedad que se debe aplicar es la heredada del estilo padre. Considera el código con que venimos trabajando en el que tenemos un menú. Supón que el menú lo hemos identificado con `<div id="menu1">`:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!--Código ejemplo para el curso -->
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title>
<meta charset="utf-8">
<meta name="description" content="Portal web aprenderaprogramar.com">
<meta name="keywords" content="aprender, programar, cursos, libros">
<link rel="stylesheet" type="text/css" href="estilos.css">
</head>
</head>
<!-- Contenido de la página web -->
<body>
<!-- Cabecera de la página web -->
<div>
<h1>Portal web aprenderaprogramar.com</h1>
<h2>Didáctica y divulgación de la programación</h2>
</div>
<!-- Fin de la cabecera de la página web -->
<br />
<!-- Contenedor para la parte central -->
<div>
<!-- menu -->
<div id="menu1">
<div>Menú</div>
<hr />
<ul>
<li><a href="#">Inicio</a></li>
<li> <a href="libros.html">Libros de programación</a> </li>
<li> <a href="cursos.html">Cursos de programación</a> </li>
<li> <a href="humor.html">Humor informático</a> </li>
</ul>
</div>
<!-- fin menu -->
<!--Aquí el resto del código del ejemplo -->

</html>
```

Escribe esta definición en el archivo estilos.css y visualiza el resultado:

```
/* Curso CSS estilos aprenderaprogramar.com */
body {font-family: Arial, sans-serif;}
#menu1{ color: red; text-decoration:none;}
```

Se obtiene algo similar a esto:

Portal web aprenderaprogramar.com

Didáctica y divulgación de la programación

Menú

- [Inicio](#)
- [Libros de programación](#)
- [Cursos de programación](#)
- [Humor informático](#)

Aprender a programar es un objetivo que se plantea mucha gente y que no todos alcanzan.

Vemos que el color rojo no está siendo aplicado por la etiqueta a (link) dentro del elemento con id menu1, y esto a pesar de que la propiedad color es heredable (su valor inherit = yes). La propiedad text-decoration tampoco está siendo aplicada. En este caso text-decoration no es heredable (inherit = no). En realidad que sea heredable o no no está teniendo efecto aquí ya que las propiedades por defecto del elemento <<a>> sobrescriben a las propiedades que hubiéramos definido en nuestro estilo menu1. Ya sabemos que podemos sobrescribir la propiedad de color y no subrayado del elemento <<a>> dentro de menu1 por ejemplo escribiendo este código (ejemplo A):

```
/* Curso CSS estilos aprenderaprogramar.com */
body {font-family: Arial, sans-serif;}
#menu1 { color: red; text-decoration:none;}
#menu1 a { color: red; text-decoration:none;}
```

Que más sintéticamente podemos escribir así (ejemplo B):

```
/* Curso CSS estilos aprenderaprogramar.com */
body {font-family: Arial, sans-serif;}
#menu1, #menu1 a { color: red; text-decoration:none;}
```

Conociendo la palabra clave inherit también podríamos obtener el mismo efecto así (ejemplo C):

```
/* Curso CSS estilos aprenderaprogramar.com */
body {font-family: Arial, sans-serif;}
#menu1 { color: red; text-decoration:none;}
#menu1 a { color: inherit; text-decoration:inherit; }
```

En este caso hemos indicado que los links dentro de menu1, los hijos <<menu1 a>>, deben tener un color y un text-decoration tal como se haya definido en el padre, <<menu1>>.

El resultado con (A), (B) o (C) será algo similar a esto:

Portal web aprenderaprogramar.com

Didáctica y divulgación de la programación

Menú

- Inicio
- Libros de programación
- Cursos de programación
- Humor informático

Aprender a programar es un objetivo que se plantea mucha gente y que no todos alcanzan.

El uso de inherit puede tener algunas desventajas:

- a) Algunos navegadores (en especial más antiguos) puede que no reconozcan la palabra clave inherit, con lo cual seguirían mostrando los links dentro del menú en color azul y subrayados.
- b) Puede hacer el código menos claro o menos fácil de entender, en especial si tenemos que recorrer varios niveles de herencia hasta comprobar desde dónde viene una propiedad. En este caso la sobreescritura puede resultar más clara.
- c) Podemos caer en la tentación, como se ha hecho en el ejemplo anterior, de aplicar inherit a una propiedad que en CSS tiene valor inherit = no. Es decir, aplicar herencia CSS a una propiedad que de acuerdo con la documentación oficial CSS no tiene herencia. Esto puede generar confusiones y hacer el código difícil de comprender y mantener. Sobreescribiendo la propiedad no habría lugar a dudas.

Por ello en general no usaremos la invocación de estilos de elementos padre con inherit excepto en circunstancias excepcionales. Esto no quita que debemos conocer su significado pues podemos enfrentarnos a tener que analizar o modificar hojas de estilo que hayan sido generadas por otras personas.

RESUMEN

Las clases CSS pueden ser aplicadas a etiquetas específicas (por ejemplo que la clase solo sea aplicable a `<p> ... </p>`) o a etiquetas de cualquier tipo, según hagamos la definición de reglas CSS.

También pueden definirse los estilos de modo que un mismo nombre de clase dé lugar a la aplicación de distintas reglas en función de en qué etiqueta se aplique, por ejemplo que una clase dé lugar a un fondo naranja si se aplica en una etiqueta `<p>` y a un fondo azul si se aplica en una etiqueta `<div>`.

Las etiquetas HTML o estilos CSS definidos presentan herencia de estilos para aquellas propiedades que tienen un valor inherit "yes". La herencia puede no producirse si existe sobreescritura del estilo por parte de algún elemento, o si una propiedad tiene valor inherit no.

Puede forzarse la herencia desde un elemento padre escribiendo en el código CSS del elemento hijo nombreDeLaPropiedad: inherit, aunque en general será preferible sobreescribir la propiedad en lugar de indicar herencia con inherit por ser más claro.

EJERCICIO

Analiza el siguiente código HTML. En él encontrarás tres etiquetas div. Haz lo siguiente:

- a) Establece atributos class para cada una de ellas con valores `<<principal>>` para la primera, y `<<secundario>>` para la segunda y la tercera.
- b) Usando CSS establece como color de texto el rojo (red) y tamaño de fuente el 130% respecto de lo normal para los elementos h1 que se encuentren dentro de un elemento cuyo valor class sea `<<principal>>`.

- c) Usando CSS establece como color de texto el verde y tamaño de fuente el 110% respecto de lo normal para los elementos h3 que se encuentren dentro de un elemento cuyo valor class sea <<secundario>>.
- d) Usando CSS establece como color de fondo el amarillo (yellow) para los elementos span que se encuentren dentro de elementos h3 que se encuentren dentro de elementos cuyo atributo class sea <<secundario>>.
- e) Establece como tipo de fuente para todo el documento HTML el tipo Arial.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title>
<meta name="description" content="Portal web aprenderaprogramar.com">
<meta name="keywords" content="aprender, programar, cursos, libros">
<meta charset="utf-8">
</head>
<body>
<div>
<h1>Novedades</h1>
<p>Aquí presentamos las novedades del sitio.</p>
</div>
<div>
<h3>Lanzamos el producto <span>X-FASHION</span></h3>
<p>Este producto permite estirar la piel hasta dejarla como la de un bebé.</p>
<p></p>
</div>
<div>
<h3>Mejoramos el producto T-MOTION</h3>
<p>Hemos lanzado una nueva versión del producto <span>T-MOTION</span></p>
<p></p>
</div>
</body>
</html>
```

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01014D

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