

UD 4: Herencia y cascada

<http://mosaic.uoc.edu/ac/le/es/m6/ud2/index.html>

1er curso de DAM

- La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos. No todas las propiedades CSS son heredadas.
- Si no existiese la herencia, se debería especificar las propiedades CSS de todos y cada uno de los elementos de un documento HTML.
- Todos los elementos de un documento HTML heredan todas las propiedades heredables de su padre excepto el elemento raíz (html), que no tiene progenitor.
- Al inspeccionar un elemento en un navegador, se puede apreciar qué propiedades hereda (realizar prueba con Inspeccionar elemento, del navegador, y observar propiedades heredadas).

EJEMPLO [REVISAR VALORES COMPUTADOS EN NAVEGADOR]

Forzar herencia

Mediante la palabra clave inherit (heredar) puede forzarse la herencia incluso para propiedades que no se heredan normalmente.

```
p {  
  background: inherit;  
}
```

Forzar la herencia no es algo que haya que hacer a menudo. Puede ser útil para "deshacer" una declaración de una regla conflictiva o para no tener que introducir los datos de determinados valores directamente en el código fuente.

EJEMPLO

- CSS significa cascading style sheets (hojas de estilo en cascada). Es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.
- Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:
 1. Importancia.
 2. Especificidad.
 3. Orden en las fuentes.

Importancia

La importancia de una declaración de CSS depende de dónde se ha especificado. Las declaraciones contrapuestas se aplicarán en el orden siguiente: las nuevas anularán a las más antiguas.

Hoja de estilo de agente de usuario vs hoja de estilos de usuario

Las declaraciones importantes son las declaraciones que van seguidas de una directiva !important, y tienen prioridad sobre todas las demás.

```
* {  
  font-family: "Comic Sans MS" !important;  
}
```

En este caso, independientemente de lo que haya especificado el diseñador, e independientemente de aquello que se haya establecido como familia de fuentes predeterminada del navegador, todo se mostrará con Comic Sans MS.

La directiva !important se ha de utilizar en **ocasiones excepcionales**.

La especificidad determina qué regla CSS aplicar en caso de conflicto. Tiene cuatro componentes:

- "a" es bastante sencillo: es 1 para una declaración en un atributo style; si no, es 0.
- "b" es el número de selectores de id en el selector (los que empiezan con #).
- "c" es el número de selectores de atributo, incluidos los selectores de clase y pseudoclases.
- "d" es el número de tipo de elementos v pseudoelementos del selector.

Selector	a	b	c	d	Especificidad
h1	0	0	0	1	0,0,0,1
.foo	0	0	1	0	0,0,1,0
#bar	0	1	0	0	0,1,0,0
html >head+body ul#nav *.home a:link	0	1	2	5	0,1,2,5

Ejercicio

Orden de las fuentes

Si dos declaraciones afectan al mismo elemento, tienen la misma importancia y la misma especificidad, la señal distintiva final es el orden en las fuentes.

Si tenéis una única hoja de estilo externa, las declaraciones al final del fichero anularán a las que sucedan antes al fichero en caso de conflicto. Las declaraciones contrapuestas también pueden suceder en diferentes hojas de estilo. En este caso, el orden en el que se enlazan, se incluyen o se importan las hojas de estilo determina qué declaración se aplica, de manera que si se tienen dos hojas de estilo enlazadas en el head a un documento, la enlazada al último anulará a la enlazada al primero.

Ejercicio