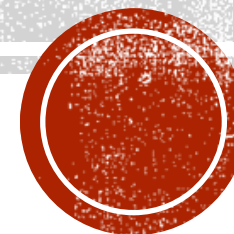
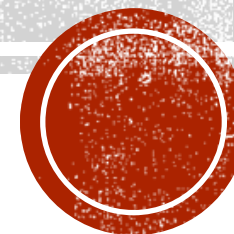


HERENCIA Y CASCADA



HERENCIA



HERENCIA Y CASCADA

- La herencia y la cascada son dos conceptos básicos en CSS, se deben comprender bien para utilizar CSS. Ambos conceptos están relacionados, pero son diferentes. La herencia está relacionada con cómo los elementos del etiquetado de HTML heredan propiedades de sus elementos padres (los que los contienen) y los transmiten a sus hijos, mientras que la cascada tiene que ver con las declaraciones de CSS que se aplican a un documento y cómo las reglas contradictorias se anulan o no entre ellas.



HERENCIA

- La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos. De hecho, se parece mucho a la herencia genética. Si los progenitores tienen los ojos azules, los hijos seguramente también tendrán los ojos azules.



HERENCIA

- No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran. Por ejemplo, los márgenes no se heredan porque es poco probable que un elemento hijo necesite los mismos márgenes que su padre. Normalmente, el sentido común dicta qué propiedades se heredan y cuáles no.



PARA QUÉ SIRVE LA HERENCIA

- ¿Por qué tiene CSS un mecanismo de herencia? Probablemente, la manera más sencilla de responder a esta pregunta sea pensar qué pasaría si no existiera la herencia. Se deberían especificar cuestiones como la familia de fuentes, el tamaño de la fuente y el color del texto individualmente para todos y cada uno de los tipos de elemento.



CÓMO FUNCIONA LA HERENCIA

- Todos los elementos de un documento HTML heredan todas las propiedades heredables de su padre excepto el elemento raíz (html), que no tiene progenitor.
- El hecho de que las propiedades heredadas tengan algún efecto o no depende de otros factores, como veremos más adelante cuando hablemos de la cascada. De la misma manera que una madre de ojos azules puede tener un hijo de ojos marrones si el padre tiene los ojos marrones, las propiedades heredadas en CSS pueden anularse.



UN EJEMPLO DE HERENCIA

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
2.  "http://www.w3.org/TR/html4/strict.dtd">  
3. <html lang="en">  
4.   <head>  
5.     <meta          http-equiv="Content-Type"  
6.     content="text/html; charset=utf-8">  
7.     <title>Herencia</title>  
8.   </head>  
9.   <body>  
10.    <h1>Título</h1>  
11.    <p>Un párrafo de texto.</p>  
12.  </body>  
13. </html>
```



UN EJEMPLO DE HERENCIA

- Se crea un nuevo fichero vacío con el editor de textos, copiad dentro la regla CSS que se muestra a continuación y guardad el fichero como style.css en la misma ubicación que el fichero HTML.

```
html {  
  font: 75% Verdana, sans-serif;  
}  
<link rel="stylesheet" type="text/css"  
media="screen" href="styles.css">
```



FORZAR LA HERENCIA

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

La regla siguiente hace que todos los párrafos hereden todas las propiedades de fondo de sus padres:

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



HTML

```
<ul id="nav">  
  <li><a href="/">Inicio</a></li>  
  <li><a href="/news/">Noticias</a></li>  
  <li><a href="/products/">Productos</a></li>  
  <li><a href="/services/">Servicios</a></li>  
  <li><a href="/about/">Sobre nosotros</a></li>  
</ul>
```



```
#nav {  
    background: blue;  
    color: white;  
    margin: 0;  
    padding: 0;  
}  
#nav li {  
    display: inline;  
    margin: 0;  
    padding: 0 0.5em;  
    border-right: 1px solid;  
}  
#nav li a {  
    color: inherit;  
    text-decoration: none;  
}
```



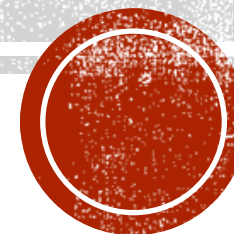
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html;  
charset=iso-8859-1" />  
    <title>Ejemplo de herencia de estilos</title>  
    <style type="text/css">  
      body { color: blue; }  
    </style>  
  </head>  
  
  <body>  
    <h1>Titular de la página</h1>  
    <p>Un párrafo de texto no muy largo.</p>  
  </body>  
</html>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html;  
    charset=iso-8859-1" />  
    <title>Ejemplo de herencia de estilos</title>  
    <style type="text/css">  
      body { font-family: Arial; color: black; }  
      h1 { font-family: Verdana; }  
      p { color: red; }  
    </style>  
  </head>  
  
  <body>  
    <h1>Titular de la página</h1>  
    <p>Un párrafo de texto no muy largo.</p>  
  </body>  
</html>
```



CASCADA



CSS significa *cascading style sheets* (hojas de estilo en cascada) y, por lo tanto, no debería extrañarnos que la cascada sea un concepto importante. 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:

- Importancia.
- Especificidad.
- Orden en las fuentes.



IMPORTANCIA

La importancia es uno de los conceptos más importantes. Si dos declaraciones tienen la misma importancia, la especificidad de las reglas decidirá cuál se debe aplicar. Si las reglas tienen la misma especificidad, el orden de las fuentes controla el resultado.



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 estilos de agente de usuario.
- Declaraciones normales en hojas de estilo de autor.
- Declaraciones normales en hojas de estilo de usuario.
- Declaraciones importantes en hojas de estilo de autor.
- Declaraciones importantes en hojas de estilo de usuario.



ESPECIFICIDAD

- La especificidad es algo que todos los autores de CSS deben comprender y tener en cuenta. Puede considerarse una medida de cuán específico es el selector de una regla.
- Un selector de especificidad baja puede dar como resultado muchos elementos (como *, que da como resultado todos los elementos del documento), mientras que un selector con una especificidad elevada puede que sólo dé como resultado un único elemento de una página (como #nav, que sólo da como resultado el elemento con una id de nav).



- La especificidad de un selector puede calcularse fácilmente, como veremos muy pronto. Si dos o más declaraciones entran en conflicto por un elemento determinado y todas las declaraciones tienen la misma importancia, la de la regla con el selector más específico será la que "gane".



ORDEN EN 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. La declaración que se ve más adelante en las hojas de estilo "ganará" a las anteriores.



- 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. Veamos un ejemplo práctico de cómo funciona esto.

.



COLISIONES DE ESTILOS

- En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen a un mismo elemento HTML. El problema de estas reglas múltiples es que se pueden dar colisiones como la del siguiente ejemplo:
- ***p { color: red; }p { color: blue; }***
- ***<p>...</p>***



1. Determinar todas las declaraciones que se aplican al elemento para el medio CSS seleccionado.
2. Ordenar las declaraciones según su origen (CSS de navegador, de usuario o de diseñador) y su prioridad (palabra clave !important).
3. Ordenar las declaraciones según lo específico que sea el selector. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
4. Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar.



CUAL SE APLICA

- ***p { color: red; }***
- ***p#especial { color: green; }***
- **** { color: blue; } <p id="especial">...</p>***



SELECTORES

- Para crear diseños web profesionales, es imprescindible conocer y dominar los selectores de CSS, una regla de CSS está formada por una parte llamada "selector" y otra parte llamada "declaración".
- La declaración indica *"qué hay que hacer"* y el selector indica *"a quién hay que hacérselo"*. Por lo tanto, los selectores son imprescindibles para aplicar de forma correcta los estilos CSS en una página.



SELECTOR UNIVERSAL

- Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML:
- **** { margin: 0; padding: 0;}***
- El selector universal se indica mediante un asterisco (*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.



SELECTOR DE TIPO O ETIQUETA

- Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:
- ***p { ...}***
- Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.



- El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

- ***h1 { color:red;}***
- ***h2 { color:blue;}***
- ***p { color:black;}***



VARIOS SELECTORES CON UNA SOLA REGLA

- ***`h1, h2, h3 { color: #8A8E27; font-weight: normal; font-family: Arial, Helvetica, sans-serif;}`***



SELECTOR DESCENDIENTE

- Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.
- El selector del siguiente ejemplo selecciona todos los elementos `` de la página que se encuentren dentro de un elemento `<p>`:
- ***`p span { color: red; }`***



- Si el código HTML de la página es el siguiente:
- **`<p> ...`**
 - **`texto1 ...`**
 - **`...`**
 - **`texto2`**
 - **` ...`**
- **`</p>`**
- El selector `p span` selecciona tanto `texto1` como `texto2`. El motivo es que en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.



p span { color:red; }h1 span { color:blue; }

Con las reglas CSS anteriores:

- Los elementos que se encuentran dentro de un elemento <p> se muestran de color rojo.
- Los elementos que se encuentran dentro de un elemento <h1> se muestran de color azul.
- El resto de elementos de la página, se muestran con el color por defecto aplicado por el navegador.



- Los selectores descendentes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.
- En el siguiente ejemplo, el selector descendente se compone de cuatro selectores:

p a span em { text-decoration: underline; }



SELECTOR DE CLASE

- Si se considera el siguiente código HTML de ejemplo:
- **<body>**
 - **<p>Lorem ipsum dolor sit amet...</p>**
 - **<p>Nunc sed lacus et est adipiscing accumsan...</p>**
 - **<p>Class aptent taciti sociosqu ad litora...</p>**
- **</body>**
- ¿Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector universal (*) no se puede utilizar porque selecciona todos los elementos de la página. El selector de tipo o etiqueta (p) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (body p) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.



- Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:
- **<body>**
 - **<p class="destacado">Lorem ipsum dolor sit amet...</p>**
 - **<p class="destacado">Nunc sed lacus et est adipiscing accumsan...</p>**
 - **<p>Class aptent taciti sociosqu ad litora...</p>**
- **</body>**



- A continuación, se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) tal y como muestra el siguiente ejemplo:

.destacado { color: red; }



- El selector `.destacado` se interpreta como "*cualquier elemento de la página cuyo atributo class sea igual a destacado*", por lo que solamente el primer párrafo cumple esa condición.
- Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class:
- **`<body>`**
 - `<p class="destacado">Lorem ipsum dolor sit amet...</p>`
 - `<p>Nunc sed lacus et est adipiscing accumsan...</p>`
 - `<p>Class aptent taciti <em class="destacado">sociosqu ad litora...</p>`
 - `</body>`



- A continuación se muestra otro ejemplo de selectores de clase:
- ***.aviso { padding: 0.5em; border: 1px solid #98be10; background: #f6feda; }***
- ***.error { color: #930; font-weight: bold; }***
- ***...***
- ***<div class="aviso">...</div>***



- ¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo class sea igual a destacado? Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

p.destacado { color: red }

- El selector p.destacado se interpreta como *"aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado"*. De la misma forma, el selector a.destacado solamente selecciona los enlaces cuyo atributo class sea igual a destacado.



- Por último, es posible aplicar los estilos de varias clases CSS sobre un mismo elemento. La sintaxis es similar, pero los diferentes valores del atributo class se separan con espacios en blanco. En el siguiente ejemplo:

`<p class="especial destacado error">Párrafo de texto...</p>`

- Al párrafo anterior se le aplican los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:
- **`.error { color: red; }`**
- **`.destacado { font-size: 15px; }`**
- **`.especial { font-weight: bold; }`**

`<p class="especial destacado error">Párrafo de texto...</p>`



- ***.error { color: red; }***
- ***.error.destacado { color: blue; }***
- ***.destacado { font-size: 15px; }***
- ***.especial { font-weight: bold; }***

<p class="especial destacado error">Párrafo de texto...</p>

- En el ejemplo anterior, el color de la letra del texto es azul y no rojo. El motivo es que se ha utilizado un selector de clase múltiple *.error.destacado*, que se interpreta como *"aquellos elementos de la página que dispongan de un atributo class con al menos los valores error y destacado"*.



SELECTORES DE ID

- En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.
- El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.



- La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

`#destacado { color: red; }`

`<p>Primer párrafo</p>`

`<p id="destacado">Segundo párrafo</p>`

`<p>Tercer párrafo</p>`



- La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. Como se sabe, en una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, el atributo class no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.
- De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página HTML.



- de la página HTML.
- Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado:

p#aviso { color: blue; }



COMBINACIÓN DE SELECTORES BÁSICOS

- CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación se muestran algunos ejemplos habituales de combinación de selectores.
- ***.aviso .especial { ... }***
- El anterior selector solamente selecciona aquellos elementos con un `class="especial"` que se encuentren dentro de cualquier elemento con un `class="aviso"`.



- ***div.avisos span.especial { ... }***
- Ahora, el selector solamente selecciona aquellos elementos de tipo con un atributo class="especial" que estén dentro de cualquier elemento de tipo <div> que tenga un atributo class="avisos".



- La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:

ul#menuPrincipal li.destacado a#inicio { ... }

El anterior selector hace referencia al enlace con un atributo id igual a inicio que se encuentra dentro de un elemento de tipo con un atributo class igual a destacado, que forma parte de una lista con un atributo id igual a menuPrincipal.



SELECTORES AVANZADOS

- . Selector de hijos: *p > span { color: blue; }*
- Selector adyacente: *elemento1 + elemento2 { ... }*
- Selector de atributos

