

CONFECCION Y PUBLICACION DE PÁGINAS WEB

**HOJAS DE ESTILO.
HERENCIA , CASCADA Y SELECTORES,**

CONCEPTO DE HERENCIA

- La herencia de propiedades de estilo permite que una propiedad llegue a la declaración de una etiqueta cuando no se ha definido explícitamente para ella.
- Esta característica se basa en el **principio de herencia**:
 - Una propiedad de estilo no definida para un elemento es heredada del elemento padre que lo contiene no habiéndose definido explícitamente para ella.
 - La mayor parte de las propiedades de estilo se heredan, pero **no todas** (**Por ejemplo no hereda el color de los vínculos, ni el tamaño de los H.... Ni es heredable por ejemplo la propiedad background-color....**)
- **Ejemplo:** definir el color de la fuente en la etiqueta <body> hace que se herede en todo el documento.

```
<!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: blue; }
    h1 { font-family: Verdana; }
    p { color: red; }
</style>
</head>
<body>
    <h1>Titular de la pagina</h1>
    <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

EJERCICIO HERENCIA

HERENCIA EJEMPLO 1.

EXPLICACIÓN

- En el ejemplo anterior, se ha indicado que la etiqueta `<body>` tiene asignado un tipo de letra Arial y un color de letra azul. Así, todos los elementos de la página (salvo que se indique lo contrario) se muestran de color azul y con la fuente Arial.
- La segunda regla indica que los elementos `<h1>` se muestran con otra tipografía diferente a la heredada.
- La tercera regla indica que los elementos `<p>` varían su color respecto del color que han heredado.

CÓMO FORZAR LA HERENCIA

- Por medio de **inherit**
 - Para forzar la herencia en aquellas propiedades que no son heredables
 - O en etiquetas como a, h(x)...que tienen un formato predeterminado por el navegador

Ejemplo

a{color:inherit;} En caso de que body pusiera el texto en rojo, los enlaces también estarían en rojo

Colisiones de estilos (I)

Cuando se aplican más de una declaración CSS a un elemento HTML diremos que estamos ante una **colisión de estilos**.

```
ul li{color: red}  
  
li{color: green}
```

Para resolver las colisiones de estilos se sigue un orden de prioridad según el concepto de cascada

Colisiones de estilos (II)

- Se aplican primero aquellos que al lado del valor de atributo se añade la palabra clave **!important**

```
li { color: red !important }
```

- Cuanto más específico (el último que se introduzca) sea el selector más prioridad tiene.

```
ul li{ color: green }
```

```
li{ color: red }
```

- Si existen varias reglas con la misma prioridad se aplica el que se ha declarado en último lugar.

```
p { color: red; }
```

```
p { color: green; }
```

-hay que tener en cuenta el tipo de selector

id > clase > elemento > heredadas

BLOQUEAR UNA PROPIEDAD

- Cuando queramos impedir que el valor de una propiedad sea sobrescrita por una regla más específica, usaremos el modificador **!important**.
- Ejemplo: `div#introduccion { color: black; !important }`
Si no hay reglas **!important**, la **última regla que se procesa**: las reglas se procesan en el orden de declaración.

SELECTORES BASICOS Y AVANZADOS

- El selector es el encargado de especificar el o los elementos de un documento HTML al que se aplicará una regla CSS.
- Encontramos dos grandes grupos de

SELECTORES BASICOS:

- a) **De etiqueta**
- b) **Descendente**
- c) **De clase**
- d) **De ID**
- e) **Selector universal**
- f) **Combinar selectores**

SELECTORES AVANZADOS

- a) **De atributos**
- b) **De hijos**
- c) **Adyacente**

Selectores básicos (I)

Universal: Se utiliza para seleccionar todos los elementos de una página o descendientes.(Poco utilizado)

```
*{ margin: 0; padding: 0 }
```

Etiqueta: Selecciona todos los elementos cuya etiqueta coincida con el valor del selector.(Excepto las que forman parte de la cabecera del documento ya que no tienen implicación en el contenido de la página).

```
p{ font-family: sans-serif; color:  
#333 }
```

Selectores básicos (II)

SELECTOR ETIQUETA

Se pueden declarar

De forma individual

```
p{  
  color: red;  
  font-size:10px;  
}
```

**o combinada (en el caso de
que varios elementos
compartan estilos.
SOLUCION OPTIMA)**

```
p, h1, div{  
  color: red;  
  font-size:10px;  
}
```

EJ

```
Body, p, div{  
  color:black;  
  font-size:12px;  
  padding:2px;  
}  
div{  
  Font-style:bold;  
}  
P{  
  Font-style:italic;  
}
```

OJO!. No hacer copia/pega .
Problemas con comillas

No hacer crear en el body un
parrafo y un div y aplicarle el
siguiente estilo (Habitualmente
se aplican las propiedades
comunes a un grupo, y
posteriormente reglas más
concretas para ciertos
elementos)

EJERCICIO COLISION DE ELEMENTOS. CASCADA

Selectores básicos (III)

Clase: Selecciona todos los elementos de una clase.

```
.destacado { color: red }
```

```
.color{color:red}
```

podemos dar color rojo a cualquier elemento de html

```
div.color{color:red}
```

solo se aplicaría a las etiquetas div que tuvieran definido el atributo `class="color"`

Ojo !! Tener cuidado con los espacios en la sintaxis. **Cualquier espacio** cambiaría por completo el sentido de nuestro selector (pasaríamos de un selector de clase a un selector descendente o viceversa)

```
div.color{color:red} define color rojo a las etiquetas div que tengan atributo class ="color"
```

```
div .color{color:red} define color rojo solo a los elementos con atributo class="color" que estén encerrados dentro de div
```

SELECTOR CLASE, EJERCICIO

```
<!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=UTF-8" />
<title>Selectores descendentes en CSS</title>
<style type="text/css">
p.color {color:red;}
.color{color:blue;}
div h3 a.link{ color:green;}
</style>
</head>

<body>
<div>
<p class="color">parrafo 1</p>
<div class="color">parrafo 2</div>
<h3><a href="" class="link">Enlace 1</a></h3>
</div>
</body>
</html>
```

Parrafo 1 en rojo
Parrafo 2 en azul
Enlace 1 en verde

p.color define un color rojo para los elementos p con atributo class="color"

.color define un color azul para cualquier etiqueta del documento que tenga un atributo class="color"

div h3 a .uno define un color verde en cualquier elemento a con atributo class"link" que esté encerrado dentro del elemento h3 y que a su vez se encuentre dentro de un div

Selectores básicos (IV)

Id: Selecciona el elemento con un ID determinado.

```
#logo { background-color: #00f }
```

En un documento HTML no pueden existir dos elementos con el mismo atributo ID, por eso es el selector más específico que existe.

Selectores básicos (V)

Descendente: Selecciona elementos dentro de elementos.

div h1{font-size:10px;}

únicamente los elementos h1 que se encuentren encerrados dentro de una capa(div) tendrán un tamaño de letra de 10px

Un elemento no tiene que ser obligatoriamente descendiente directo de otro elemento. Bastará simplemente con que se encuentre encerrado en el elemento en cuestión

Selectores básicos (V)

Descendente: Selecciona elementos dentro de elementos.

Ej **div** **h1{font** **-size:10px;}**
únicamente los elementos h1 que se encuentren encerrados dentro de una capa(div) tendrán un tamaño de letra de 10px

Un elemento no tiene que ser obligatoriamente descendiente directo de otro elemento. Bastará simplemente con que se encuentre encerrado en el elemento en cuestión

Ejercicio selector descendente

```
<style type="text/css">
  div h2{font-size:20px;}
  div p a{ color:red;}
</style>
<body>
  <div>
    <h2>encabezado 1</h2>
    <p>parrafo 1</p>
    <p>
      <a href="  ">enlace 2</a>
    </p>
  </div>
  <h2>encabezado 2</h2>
  <a href="">enlace 2</a>
</body>
```

¡ojo: NO confundir el selector descendente con un selector de etiquetas combinadas
div a{ no es lo mismo que **div, a{**

Resultado:

encabezado 1 tendrá tamaño 20 px y
enlace 1 será de color rojo

Selectores básicos (V)

Combinaciones: Todos estos selectores se pueden combinar entre sí.

```
div.footer strong { color: red }
```

Imprescindible combinar los diferentes selectores para poder diseñar hojas de estilo eficaces y bien codificadas

```
<style type="text/css">
  div p { color: green; }
  div, p { color: green; }
  a{ font-size:20px; }
  a.link{ font-size:30px; }
  div div.capa1, div p.texto { color: red; }
</style>
</head>
<body>
<div>
  <p class="texto">Párrafo 1</p>
  <div class="texto">párrafo 2.</div>
  <h3><a class="link" href="">enlace 2</a>
  <div class="capa1">parrafo 3</div>
  <div >párrafo 4</div>
</div>
</body>
</html>
```

Ojo! Ir aplicando este código de forma individual (dejar una declaración cada vez para ver cómo varia de una a otra).
Estilo a estilo para ver el resultado

EJERCICIO ELEMENTOS COMBINADOS

div p { color: green; } color verde para los elementos p definidos dentro de div. Resultado parrafo 1 verde

div, p { color: green; } aplica color verde a todos div y p, resultado párrafo 1, 2, 3 y 4 verdes

a{ font-size:20px; } tamaño 20 para todos los vínculos. Resultado enlace 2 y 2 con tamaño 20

a.tamano{ font-size:30px; } tamaño 30 solo a los vínculos con atributo class tamano. Resultado solo enlace 1

div div.capa1, divp.texto { color: red; } color rojo a los elementos div con atributo class capa1 que estén definidos dentro de un div y a los elementos p con atributo class texto que se encuentre encerrados en un div.

Resultado parrafo 1 y parrafo 3 de color rojo

Explicación ejercicio anterior

Selectores avanzados (I)

Hijos directos: Selecciona elementos que son hijos directos. Utiliza el operador **>**

```
p > a { color: blue; }
```

Se aplica:

```
<p><a href="#">Texto1</a></p>
```

No se aplica:

```
<p><span><a href="#">Texto2</a></span></p>
```

Selectores avanzados (II)

Hermanos adyacentes: Selecciona elementos que están al mismo nivel (hermanos) y están seguidos en el código. Utiliza el operador **+**

```
p + p { color: red; }
```

NO se aplica

```
<div><p>Lorem ipsum dolor sit amet...</p></div>
```

NO se aplica

```
<p>Lorem ipsum dolor sit amet...</p>
```

```
<h1></h1>
```

Se aplica

```
<p>Lorem ipsum dolor sit amet...</p>
```

Selectores avanzados (III)

Hermanos generales: Permite seleccionar todos los elementos que están al mismo nivel (hermanos). Utiliza el operador ~

```
h2 ~ p { color: red; }
```

NO se aplica

```
<h2>Mi encabezado</h2>
```

NO se aplica

```
<h3>Mi subencabezado</h3>
```

Se aplica

```
<p>Mi párrafo</p>
```

```
h2 ~ p { color: red; }
```

```
<h2>Mi encabezado</h2>
```

```
<h3>Mi subencabezado</h3>
```

```
<p>Mi párrafo</p>
```



Selectores avanzados (y IV)

Selector de atributos: Selecciona elementos que contengan un atributo y/o valor.

- **Posee el atributo:** `input[placeholder] {}`
- **Valor concreto de atributo:** `border[size="3px"] {}`
- **Comienza con:** `a[href^="http:"] {}`
- **Finaliza con:** `a[src$=".jpg"] {}`
- **Contiene:** `a[href*="heraldo"] {}`
- **Elementos cuyo valor son palabras separadas por guiones y comienzan con el valor:** `a[lang|=es] {}`

Pseudo-clases (I)

Se utilizan para aplicar diferentes estilos en función a la interacción del usuario, posición o idioma:

Se aplican a enlaces:

`:link` → Enlaces no visitados

`:visited` → Enlaces visitados

`:active` → Enlace activo (click)

`:hover` → Enlace activo (mouse over)

Pseudo-clases (II)

En función del idioma.

`:lang` → Se aplica a todos los elementos que estén definidos como perteneciente a un idioma.
No tiene nada que ver con el selector `[lang]`

Pseudo-clases (III)

Según su posición y hermanos/siblings:

`:first-child` → Primer hijo

`:last-child` → Último hijo

`:nth-child(#num)`

`:first-of-type` → Primero de un tipo

`:last-of-type` → Último de un tipo

`:nth-of-type(n)`

`:not(selector)` → Invierte la selección

`:empty` → Elemento vacío (sin hijos)

Pseudo-clases (III)

Ejemplo de nth:child.

Se trata de asignar un color distinto a cada uno de los elementos de una barra de navegación

```
UI li:nth-child(1){background:#FFFFFFF;}
```

```
UI li:nth:child(2){background: #CCCCCCC;}
```

```
UI li:nth-child(3){background: #999999;}
```

Pseudo-clases (III)

Ejemplo de **PSEUDO CLASE :NOT**

Permite dejar a algún elemento de una clase o id fuera de un selector,

:nth-of-type permite, entre otras cosas, aplicar estilos diferentes para los elementos pares e impares de una lista

```
header:not(#cabeceraPaginas){
```

```
Background-color:rgba(255,255,255,0.3);}
```

Pone en el color de fondo a todos los header excepto a los que tenga el id #cabeceraPaginas

Pseudo-clases (III)

Ejemplo de **:nth-of-type**

permite, entre otras cosas, aplicar estilos diferentes para los elementos pares e impares de una lista
odd impares

Even pares

```
article p:nth-of-type(odd){  
background-color-rgba(250,250,250, .7);}
```

```
article p:nth-of-type(even){  
background-color-rgba(250,250,250, .5);}
```

Los párrafos pares tendrán un color y los impares otros

Pseudo-clases (y IV)

Se aplican a elementos de formulario:

`:focus` → Se aplica a elementos que tienen el “foco”

`:enabled` → Selecciona elementos activos

`:disabled` → Selecciona elementos no-activados

`:checked` → Selecciona elementos con el estado checked

Pseudo-elementos (I)

Se distinguen de las pseudo-clases porque hacen referencia a parte del contenido del elemento seleccionado y no a todo.

Su sintaxis ha cambiado en CSS3 y ahora usa el doble `::` aunque mantiene la compatibilidad

```
p::first-letter { font-size: 2em }
```

Pseudo-elementos (y II)

`::first-letter` → Permite seleccionar el primer carácter del elemento

`::first-line` → Permite seleccionar la primera línea del elemento

`::before` → Permite añadir contenido al principio del elemento

`::after` → Permite añadir contenido al final del elemento

`::selection` → Permite dar estilo a la selección del usuario