

# Jerarquías

Una parte vital de los estilos es entender que jugamos con ciertas jerarquías al momento de trabajar. Es claro que un mismo elemento puede verse afectado de diferentes maneras y es importante saber esas maneras como lo afectan y se desempeñan al momento de tener más prioridad que otras.

La realidad es que podríamos insumir tiempo en ejemplos y demostraciones que sólo aumentan los casos y la confusión

Por eso les traemos ejemplos sencillos que les van a ayudar a entender mejor cómo funciona cada elemento de manera jerárquica.

Si por ejemplo trabajo con un elemento y luego vuelvo en algún otro tipo de selector compuesto supongamos un selector grupal con ese mismo elemento el orden dirá todo,

```
<style>
p, h2 { color: red; font-style:italic; font-size: 12px}
p,h1 { color: blue; }

</style>
```

La pregunta es, por qué en nuestra propia hoja de estilo vendríamos a utilizar el mismo selector más de una vez.

Bueno, es que en el primer caso lo hicimos para aprovechar un selector grupal, donde nuestros párrafos compartían varias propiedades con los h2. Luego en realidad, compartían todas las propiedades salvo el color, por esa razón utilizamos este conocimiento de saber que el mismo selector , con la misma propiedad diferente valor obtiene su jerarquía en base al orden y de ahí que decidimos hacer lo anterior.

Qué pasa si de pronto tengo dos clases sobre un mismo elemento:

```
<style>

.clase1 { color: blue; font-variant: small-caps; background: red;}
.clase2 { color: lightBlue;}

</style>
```

En el ejemplo anterior podríamos entender este detalle debido a que seguramente la primera clase que fue aplicada a varios elementos le permitía aprovechar ese espacio porque compartía ciertas características pero luego, también, sucede que queríamos sobrescribir una propiedad, en este caso el color que a su vez compartía con otros elementos que poseían otra clase.

Como en sí el selector al ser una clase en ambos casos, tiene en sí o por sí la misma jerarquía acá su orden también define quien gana la pelea.

Luego también tenemos que entender que los elementos de nuestro HTML, también tienen estilos que pueden ser adquiridos mediante una hoja de estilo que está siendo trabajada por ejemplo de parte de un plugin

Para comprenderlo mejor, yo pude haber usado un plugin para realizar una galería de imágenes, y ese plugin por supuesto tiene sus propios estilos, estos estilos entonces también afectan mis elementos, por tanto el saber las jerarquías me puede ayudar a solucionar inconvenientes, donde quizás yo quiero todas las funcionalidades del plugin pero sin quizás tomar por completo todos sus estilos.

Vamos a suponer que un elemento tiene una clase asignada y también tiene asignado un selector de elemento,

```
<style>

p { color: blue;}
.parrafo { color: white;}

</style>
```

En este caso siempre la pelea la gana lo más específico, el orden acá no importa, si algo tiene una clase y también tiene la misma propiedad asignada por un selector de elemento o etiqueta siempre gana la clase por ser más puntual sobre el elemento.

Si por ejemplo tengo un ID y una clase que pasa?

```
<style>

#parrafo { color: lightblue;}
.parrafos { color: blue;}

</style>
```

Quedará en el caso anterior el ID, ya que este tiene más jerarquía que la clase.

En el siguiente ejemplo, el que tendrá más jerarquía es p, por qué?

Porque si bien elemento es uno y id el otro, el id está aplicado sobre el contenedor no sobre el elemento por tanto, lo más específico gana en este caso,

Ejemplo para copiar y pegar en el HTML:

[code]

```
<style>
#contenedor { color: red;}
p { color: blue;}
</style>

<body>

<div id="contenedor">
<p> lorem    </p> </div>
</body>
```

[/code]

Ahora que pasa con los elementos como estilos en línea, estilos externos e internos.

Bueno, si por caso yo en una hoja externa tengo lo siguiente:

```
<style>

p { color: blue;}

</style>
```

Y luego en mi estilo interno tengo:

```
<!doctype html>
<title> Jerarquías </title>

<link href="estilos.css" rel="stylesheet">

<style>
p { color: white}

</style>
```

Dependerá del orden, no hay mayor jerarquía porque sea interno o externo sino que depende del orden.

Ustedes preguntarán por qué yo habría de hacer esto de esta forma, claro la respuesta es en el trabajo con hojas de plugins o estilos que nos vienen dados de otras fuentes, puede que en esos estilos se esté trabajando con el mismo selector y quizás el porqué de que no tomen nuestros propios estilos se debe al orden. En el caso anterior, queda el interno pero no por ser interno, sino por el orden.

Si tengo dos hojas externas pasa lo mismo, si trabajan por caso sobre el mismo selector , misma propiedad , diferente valor , siempre queda el último:

```
<!doctype html>
<title> Jerarquías </title>

<link href="estilos1.css" rel="stylesheet">
<link href="estilos2.css" rel="stylesheet">
```

Recordemos que lo que no se pisa, se suma a los estilos, es decir, el decir que queda un estilo u otro es cuando trabajamos sobre misma propiedad mismo selector, diferente valor, el resto de las propiedades se van sumando a los estilos

Luego también, podemos trabajar con estilos en línea, aquellos que se utilizan mayormente para trabajar con newsletters como hemos visto en anteriores clases.

En ese caso el estilo en línea tiene más prioridad que el resto de los estilos, pero qué pasa si utilizamos la palabra !important

Bueno en ese caso, !important hace que el que tenían menor valor jerárquico tome el primer lugar por ejemplo, para copiar y pegar en el HTML:

```
[code]
<style>
p { color: blue !important;}
p {color: red;}

</style>
[/code]
```

En el caso anterior según lo visto anteriormente quedaría el último estilo, pero como el primero tiene !important, ahora ese es el que queda

Luego si por el contrario ambos terminan teniendo !important, ambos vuelven al orden anterior establecido y quedaría el último, ejemplo para copiar y pegar en el HTML:

```
[code]

<style>
p { color: blue !important;}
p {color: red !important;}

[/code]
```

</style>

[/code]

El **!important**, se debe aplicar luego del valor de cada propiedad que quiero remarcar, es decir si hubiese dos propiedades que quiero que tengan importancia debería hacerlo con ambas

```
<!doctype html>
<title> Jerarquías </title>

<style>
p { color: blue !important; background-color: green !important;}
p {color: red !important;}

</style>
```

Los condicionales son elementos o comentarios que nos permitían generar que un estilo tanto interno como externo se direcciona para ciertas versiones de **Internet Explorer**.

En las versiones actuales de Internet Explorer esto ya no sirve más ( en la versión 10 dejó de funcionar) pero como desarrolladores debemos conocerlos y reconocerlos al momento de trabajar.

```
<!doctype html>
<title> >Condicionales </title>

<!--[if IE]>
| Este navegador es cualquier versión de Internet Explorer
<![endif]>

<!--[if IE 5.5]>
| Este navegador es Internet Explorer 5.5
<![endif]>

<!--[if gte IE 8]>
| Este navegador es Internet Explorer 8 o cualquier versión más reciente
<![endif]>

<!--[if lte IE 6]>
| Este navegador es Internet Explorer 6 o cualquier versión anterior
<![endif]>
```