* ¿Qué orden prevalece para asignar un estilo? ¿el último? ¿el primero?

Como hemos mencionado, si varios selectores competidores tienen la misma importancia y especificidad, el tercer factor que interviene para decidir qué regla vence es el orden del código. as últimas reglas prevalecen sobre las primeras.

* ¿Qué significa que un selector es más específico que otro? Dame un ejemplo

La **Especificidad** es en sí una medida de cómo de específico es un selector — cuantos elementos puede seleccionar. los selectores de elementos son poco específicos. Los selectores de clase son más específicos, por lo que prevalecen sobre los selectores de elementos. Los selectores ID son todavía más específicos, por lo tanto ganan frente a los de clase. La única forma de vencer a los selectores ID es usando la declaración !important.

En el siguiente ejemplo, la regla CSS que prevalece se decide por lo específico que es cada selector:

p { color: red; }

p#especial { color: green; }

\* { color: blue; }

<p id="especial">...</p>

Al elemento <p> se le aplican las tres declaraciones. Como su origen y su importancia es la misma, decide la especificidad del selector. El selector \* es el menos específico, ya que se refiere a *"todos los elementos de la página"*. El selector p es poco específico porque se refiere a *"todos los párrafos de la página"*. Por último, el selector p#especial sólo hace referencia a *"el párrafo de la página cuyo atributo id sea igual a especial"*. Como el selector p#especial es el más específico, su declaración es la que se tiene en cuenta y por tanto el párrafo se muestra de color verde.

* ¿Se aplica el estilo con mayor o menor especificidad?

Con mayor especificidad

* ¿Cómo se calcula la especificidad?

La especificidad que tiene un selector se mide mediante 4 valores (o componentes) diferentes, podemos pensar en ellos como en 4 columnas de unidades de millar, centenas, decenas y unidades:

1. Unidades de millar: Puntúa 1 en esta columna si la declaración está dentro de un atributo [style](https://developer.mozilla.org/es/docs/Web/HTML/Global_attributes" \l "attr-style) (como las declaraciones que no tienen selectores, que su especificidad es siempre 1000). Sino puntúa 0.
2. Centenas: Puntúa 1 en esta columna por cada selector ID contenido en el selector.
3. Decenas: Puntúa 1 en esta columna para cada selector de clase, selector de atributo o de pseudo-clase contenidos en el selector.
4. Unidades: Puntúa 1 en esta columna por cada selector de elemento o pseudo-elemento contenidos en el selector.

**Nota**: El selector universal (\*), las combinaciones (+, >, ~, ' ') y la pseudo-clase de negación (:not) no afectan a la especificidad

El que tiene mas puntaje gana

* ¿Qué significa !important? ¿Por qué **NO** debemos usarlo a menos que sea estrictamente necesario?

En CSS, hay un trozo de sintaxis que podemos usar para asegurarnos que una determinada regla siempre "gane" sobre todas las demás: !important

Es útil saber que !important existe para poder reconocerlo en el código escrito por otros, **PERO**, no debemos usarlo a menos que sea estrictamente necesario. Una de estas ocasiones la encontramos cuando estamos trabajando un CMS y no podemos editar los módulos principales de CSS, y queremos sobrescribir un estilo que no puede sobrescribirse de ningún otro modo. NO lo usaremos si lo podemos evitar. Pues !important cambia la forma de trabajo de la cascada, y puede causar problemas de dificil solución en el depurado de CSS, especialmente en CSS grandes.

* ¿Todos los estilos en CSS se heredan? ¿Por qué?

Las propiedades que se heredan por defecto y las que no([margin](https://developer.mozilla.org/es/docs/Web/CSS/margin" \o "La propiedad CSS margin establece el margen para los cuatro lados. Es una abreviación para evitar tener que establecer cada lado por separado con las otras propiedades de margen:  margin-top, margin-right, margin-bottom y margin-left.), [padding](https://developer.mozilla.org/es/docs/Web/CSS/padding" \o "La propiedad CSS padding establece el espacio de relleno requerido por todos los lados de un elemento. El área de padding es el espacio entre el contenido del elemento y su borde (border) no se permiten valores negativos.), [border](https://developer.mozilla.org/es/docs/Web/CSS/border" \o "La documentación acerca de este tema no ha sido escrita todavía . ¡Por favor  considera contribuir !), y [background-image](https://developer.mozilla.org/es/docs/Web/CSS/background-image)), viene marcado en gran medida por el sentido común. Pero para estar seguros podemos consultar la [Referencia CSS](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference) — cada propiedad viene en una página que comienza con una tabla resumen que incluye diversos detalles sobre cada elemento, incluyendo si se hereda o no.

Aunque la herencia de estilos se aplica automáticamente, se puede anular su efecto estableciendo de forma explícita otro valor para la propiedad que se hereda

* ¿Qué es y para qué sirve inherit?

Este valor establece el valor de la propiedad de un elemento seleccionado en el mismo que su elemento padre. El valor inherit es el más interesante — nos permite, de forma explícita, hacer que un elemento herede de su padre el valor de una propiedad.

VALORES Y UNIDADES

* Dame un ejemplo de una undidad absoluta y una relativa

Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado. completamente definida, ya que su valor no depende de otro valor de referencia

Nos referimos a píxeles (px) como **unidades absolutas** pues siempre tienen el mismo tamaño independientemente de cualquier otra medida. Otras unidades absolutas:

* mm, cm, in: Milímetros, centímetros, o pulgadas.
* pt, pc: Puntos (1/72 de una pulgada) or picas (12 puntos.)ejm\_

/\* El cuerpo de la página debe mostrar un margen de media pulgada \*/

body { margin: 0.5in; }

/\* Los elementos <h1> deben mostrar un interlineado de 2 centímetros \*/

h1 { line-height: 2cm; }

También disponemos de unidades relativas, que lo son respecto al tamaño de fuente font-size o a la ventana actual ([viewport](https://developer.mozilla.org/en-US/docs/Glossary/viewport" \o "viewport: A viewport represents a polygonal (normally rectangular) area in computer graphics that is currently being viewed. In web browser terms, it refers to the part of the document you're viewing which is currently visible in its window. Content outside the viewport is currently not displayed onscreen.)). **ems son las unidades relativas más usadas en el desarrollo web.** Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes mediosjm : “porcentaje”:

* em, (no confundir con la etiqueta <em> de HTML) relativa respecto del tamaño de letra del elemento.
* ex, relativa respecto de la altura de la letra x (*"equis minúscula"*) del tipo y tamaño de letra del elemento.
* px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML
* ¿Para qué sirve line-height?

[line-height](https://developer.mozilla.org/es/docs/Web/CSS/line-height), que establece la altura de cada línea de texto en un elemento. Podemos usar unidades para establecer la altura de la línea, pero normalmente es más fácil usar simplemente un valor que actúe como factor multiplicador.

p {

line-height: 1.5;

}

El (tamaño de fuente) font-size son 16px; la altura de linea 1.5 veces esta, o sea 24px.

* ¿Para qué sirven valores en procentajes? Dame un ejemplo

También podemos usar valores porcentuales para expresar la mayoría de cosas que requieran de valores numéricos, lo que nos permite crear, por ejemplo, cajas cuya anchura siempre cambie según el ancho del contenedor padre.

Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:

body { font-size: 1em; }

h1 { font-size: 200%; }

h2 { font-size: 150%; }

Los porcentajes también se utilizan para establecer la anchura de los elementos:

div#contenido { width: 600px; }

div.principal { width: 80%; }

<div id="contenido">

<div class="principal">

...

</div>

</div>

* El tamaño de fuente por defecto que los navegadores usan antes de aplicar CSS es de ... El tamaño de fuente por defecto que los navegadores usan antes de aplicar CSS es de 16 píxeles,  1em equivale a un margen de anchura 16px.
* ¿Qué unidad es #ff0000? ¿Qué color representa? rojo
* ¿RGB es una función? El tercer sistema al que vamos a hacer referencia es RGB. Un valor RGB es una función — rgb() — a la que pasamos tres parámetros que representas los colores primarios **rojo, verde y azul**,
* ¿Qué color representa rgb(0,0,255)?azul
* ¿Cómo le damos opacidad al color?

especificar la transparencia vía CSS — la propiedad [opacity](https://developer.mozilla.org/es/docs/Web/CSS/opacity" \o "La propiedad CSS opacity define la transparéncia de un elemento, esto es, en que grado se superpone el fondo al elemento.). En lugar de fijar la transparencia de un determinado color, aquí fijamos la transparencia de todos los elementos seleccionados y sus hijos.

la opacidad es útil cuando queremos crear un efecto animado en que todo el elemento UI va, desde completamente visible a invisible.

* ¿Cuándo se utilizan valores sin unidades?

Imaginemos que queremos eliminar el margen o el relleno de un elemento, simplemente usaremos el 0, ya que 0 es 0 cualesquiera que fueran las unidades anteriores. Numero de veces que ocurrirá una animacion

* ¿Cuáles son las recomendaciones generales para el uso de unidades?