

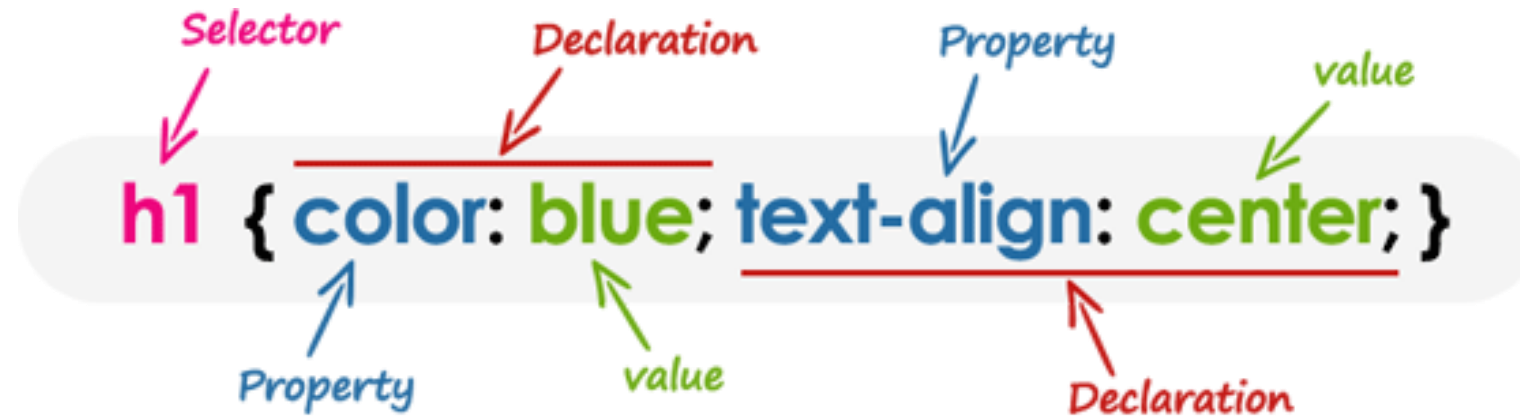
Lenguajes de marcas y sistemas de gestión de la información



UT03 – CSS 4 – Colores

Sintaxis CSS

Reglas CSS: selector y una o más declaraciones



Selector indica a que elemento o elementos del documento HTML se debe aplicar el estilo.

Cada una de las declaraciones tienen la forma “propiedad: valor”. Las declaraciones pueden estar en la misma línea o en varias.

Las declaraciones indican qué estilos queremos aplicar a los elementos seleccionados.

Propiedades más habituales

Las dos propiedades más habituales relacionadas con el color son:

- `color`: indica el color de texto del contenido de un elemento
- `background-color`: indica el color de fondo de un elemento.

Los colores de texto (`color`) se heredan, y los colores de fondo (`background-color`) no se heredan. Esto significa que, si ponemos un color y un color de fondo a un elemento, sus descendientes:

- Tendrán todos (salvo contadas excepciones) el mismo color de texto.
- No tendrán el mismo color de fondo. Esto es más difícil de ver porque si están dentro del elemento, parecerá que tienen ese color de fondo.

Forma de codificar el color

Los colores, en general, no específicamente en HTML y CSS, se pueden codificar de varias formas:

- RGB. Combinación rojo, verde y azul (Red, Green, Blue). Posiblemente la más utilizada en ordenadores (programación, css, ...)
- HLS. Tono, brillo y saturación (Hue, Brightness, Saturation).
<https://www.uifrommars.com/que-es-hsl/>
- HWB: Tono, blancura, oscuridad (Hue, Witeness, Blackness)
<https://dev.to/adrianbenavente/hwb-un-mordisco-del-futuro-de-css-46j8>
- CMYK. Combinación de cian, magenta, amarillo y negro (Cyan, Magenta, Yellow, black). Usada sobre todo en impresión en papel.
- Otros: LAB, LCH, Pantone, etc..

Colores en CSS

Hay varias formas de especificar un color en CSS:

- Colores por nombre. red, green , teal, slategray, etc. Muchos valores, pero irremediablemente limitados. No podemos usar cualquier color del espectro.
- Codificación RGB. Va precedida de la almohadilla (#):
 - Hexadecimal: #1CB2E4. Cada par de dígitos hexadecimales es un valor de 0 a 255 para el color. Las dos primeras R, las dos siguientes G, las dos últimas B.
 - Hexadecimal abreviado: #3B5. Cuando se repiten los dígitos hexadecimales en cada componente R, G y B. #3B5 = #33BB55

Colores en CSS

- RGB con canal alfa.
 - El canal alfa se añade a un color para indicar su índice de transparencia.
 - Por defecto, los colores son opacos, sin transparencia, con un alfa o alpha de valor 1 o 100%. Si ponemos un alfa de 0%, o simplemente 0, tendríamos un color transparente. Un alfa de 50% o 0.5 (con .) sería semitransparente.
 - En el formato hexadecimal, añadiríamos dos dígitos hexadecimales al código RGB, de modo que 00 sería transparente y FF (255) sería opaco. Ejemplo: #ff000080 sería el rojo a un 50% de transparencia.

Función rgb

Permite calcular un color en función de la cantidad de sus componentes.

Tiene cuatro parámetros:

- Componente R. Cantidad de rojo en el color.
- Componente G. Cantidad de verde en el color.
- Componente B. Cantidad de azul en el color.
- Canal alfa: nivel de transparencia. Es opcional. Si no aparece el color será opaco.

Puede recibir:

- Un valor de 0 a 255 o un porcentaje de 0% a 100% para los colores.
- Un valor de 0 a 1 o un porcentaje de 0% a 100% para el canal alfa.

Función rgb – Ejemplos

Algunos ejemplos de colores con la función rgb:

```
color: rgb(255 0 0);  
color: rgb(100% 0% 0%);  
background-color: rgb(230 150 45 / 20%);  
background-color: rgb(90% 59% 18% / 20%);  
background-color: rgb(90% 59% 18% 20%); /* barra opcional */
```

Esta es la notación moderna. Es muy habitual ver la notación clásica con comas:

```
color: rgb(255, 0, 0);  
color: rgb(100%, 0%, 0%);  
background-color: rgb(230, 150, 45, 20%);  
background-color: rgb(90%, 59%, 18%, 20%);
```

En la notación clásica no se puede usar la barra (/).

No todas las notaciones están soportadas en todos los navegadores.

Función rgba

La función rgba es una versión antigua de rgb, que se usaba para poder especificar un canal alfa.

Desde la actualización de la función RGB para que use canales alfa, no deberíamos usar rgba, pero la seguiremos encontrando en mucho código "legacy".

Colores especiales

Hay dos valores especiales para los colores:

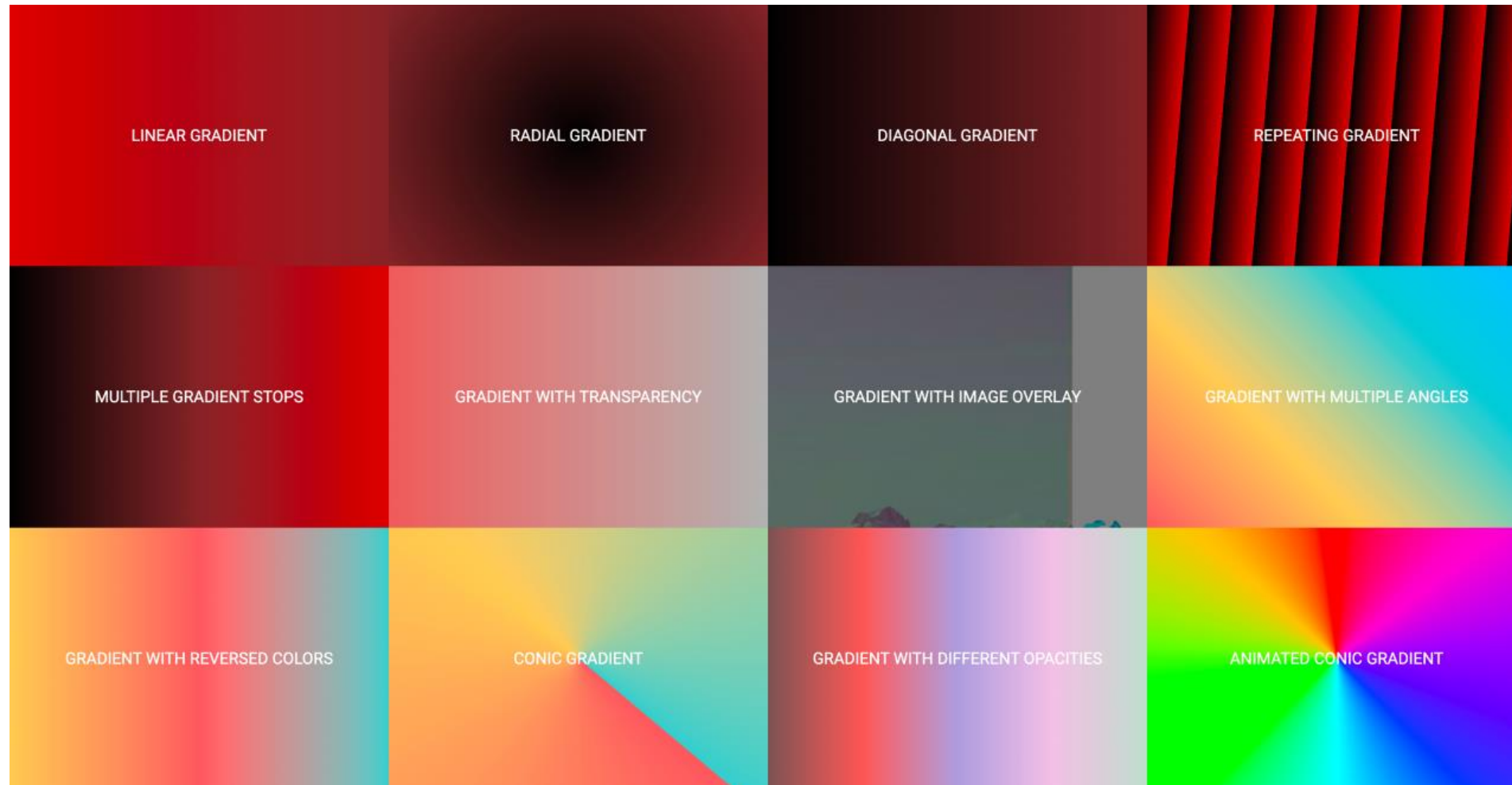
- transparent: transparente. Es el equivalente a una transparencia total, o canal alfa cero. Es el valor por defecto para los fondos (background-color)
- currentColor: sirve para referirse al color actual del texto. Sería la propiedad color del elemento o de un elemento ancestro.

También tenemos "constantes" para colores del sistema, como:

- linktext: color de enlaces no visitados
- field: color de fondo de un campo de texto <input>
- highlight: color de fondo de textos seleccionados
- ...

Gradientes (degradados)

Permiten crear colores no uniformes, que varían de un color a otro.



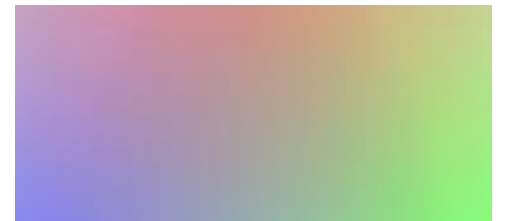
Fuente de la imagen:

<https://www.linkedin.com/pulse/12-examples-css-gradients-elevate-your-web-design-abhishek-garg-/>

Gradientes

Los gradientes se definen con funciones.

- Lineales: varían de un color a otro, creando bandas que progresan a lo largo de una dirección, siguiendo una línea recta. Se usa la función "linear-gradient".
- Radiales: El color inicial está en el centro, y progresa hacia el exterior, siguiendo los radios imaginarios de una circunferencia o elipse. Se usa "radial-gradient".
- Cónicos: la transición se produce rodeando un punto central. Se usa "conic-gradient". El ejemplo combina varios radiales con distintos alfa.



De cada uno de estos tipos hay variantes para poder repetir el degradado, y hay algunos otros más avanzados. Sólo veremos el lineal.

Gradientes lineales

Importante: no funcionan con la propiedad "background-color". Se debe usar "**background-image**" o el atajo "**background**".

Se definen con la función linear-gradient.

Esta función está sobrecargada, es decir, tiene varias versiones que permiten personalizar la forma en la que se hace el degradado.

En su forma más básica admite dos colores, y la transición se hace del primero al segundo, y de arriba hacia abajo:

```
background: linear-gradient(red, yellow);
```

Pero se pueden especificar más de dos colores, cuánto ocupa cada color, degradar a transparencia, superponer colores con canal alfa, hacer patrones repetitivos, etc. Ver los ejemplos en el repositorio en GitHub (UT03).