

44 CSS: Colores

CSS ofrece múltiples métodos para definir colores, permitiendo precisión y flexibilidad en el diseño. A continuación, se detallan los sistemas y conceptos clave:

1. Colores por Nombres

CSS incluye **nombres predefinidos** para colores comunes (ej: `red`, `blue`, `rebeccapurple`).

<https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>

- **Ventaja:** Sencillos de usar y memorizar.
- **Limitación:** Solo incluyen una paleta limitada (~140 colores).
- **Ejemplo:**

```
.caja {  
  color: coral;      /* Color del texto */  
  background-color: rebeccapurple; /* Color de fondo #663399*/  
}
```

2. Sistema Hexadecimal

Los colores se definen con un **código de 6 dígitos** precedido por `#`, que representa los valores de **rojo, verde y azul (RGB)** en hexadecimal.

- **Formato:** `#RRGGBB` (ej: `#ff0000` es rojo puro).
- **Versión corta:** Si los pares son iguales, se simplifica (ej: `#f00` = `#ff0000`).
- **Opacidad:** Agregando dos dígitos al final (`#RRGGBBAA`, ej: `#ff000080` = rojo al 50% de opacidad).
- **Ejemplo:**

```
.caja {  
  background-color: #1a2b3c;
```

```
border: 2px solid #0f0; /* Verde (#00ff00 simplificado) */
}
```

3. Sistema RGB/RGBA

Define colores mediante los valores de **rojo, verde y azul**, con un rango de **0 a 255**.

- **Sintaxis:** `rgb(red, green, blue)` o `rgba(red, green, blue, alpha)` (alpha: 0-1).
- **Ejemplo:**

```
.caja {
  background-color: rgb(255, 0, 0); /* Rojo */
  color: rgba(0, 255, 0, 0.5); /* Verde al 50% de opacidad */
}
```

4. Sistema HSL/HSLA

Define colores usando **tono (Hue)**, **saturación (Saturation)** y **luminosidad (Lightness)**.

- **Hue:** Ángulo en el círculo cromático (0-360).
 - 0° = Rojo, 120° = Verde, 240° = Azul.
- **Saturación:** 0% (gris) a 100% (color puro).
- **Luminosidad:** 0% (negro) a 100% (blanco).
- **Ejemplo:**

```
.caja {
  background-color: hsl(180, 100%, 50%); /* Cyan */
  color: hsla(300, 100%, 25%, 0.8); /* Morado oscuro al 80% de opacidad */
}
```

5. Valores Especiales

- `transparent`: Hace un elemento completamente transparente.

```
.caja {  
  background-color: transparent;  
}
```

- **currentColor** : Hereda el valor de **color** definido en el elemento o su ancestro.

```
.caja {  
  color: navy;  
  border: 2px solid currentColor; /* Borde del mismo color que el texto */  
}
```

6. Propiedad **opacity**

Controla la **transparencia global** de un elemento (afecta a todo su contenido).

- **Rango**: 0 (transparente) a 1 (opaco).
- **Diferencias con RGBA/HSLA**:
 - **opacity** afecta a todo el elemento (incluyendo hijos).
 - Los canales alfa (ej: **rgba()**) solo afectan al color específico.
- **Ejemplo**:

```
.caja {  
  background-color: rgb(255, 0, 0);  
  opacity: 0.5; /* Elemento y contenido al 50% de opacidad */  
}
```

7. Convertidores de Colores

Existen herramientas y páginas web útiles para convertir entre diferentes sistemas de colores, como el **hexadecimal**, **RGB**, **HSL**, entre otros.

Herramientas útiles para traducir entre sistemas:

- **Google**: Buscar "RGB to HEX converter" o "color picker".
- **Sitios web**: [HTML Color Codes](#), [MDN Color Docs](#).

- **Editores de código:** Visual Studio Code muestra previsualizaciones de colores al escribir.
-

¿Cuándo usar cada sistema?

Sistema	Casos de Uso
Nombres	Prototipado rápido, colores básicos.
Hexadecimal	Ampliamente usado en desarrollo web.
RGB/RGBA	Control preciso de opacidad en elementos.
HSL/HSLA	Ajustes intuitivos de tono, saturación y luz.
