

3.4. Tipografía y colores



LENGUAJES DE PROGRAMACIÓN NIVEL 3. UNIDAD 3

CSS - ESTILOS PARA LA PRESENTACIÓN VISUAL DE LA PÁGINA WEB

Contenido

| | |
|--|----|
| Introducción | 3 |
| Fundamentos de la tipografía en CSS | 5 |
| Familias de fuentes..... | 5 |
| Tamaño y altura de línea | 6 |
| Estilos y peso de fuente | 7 |
| Importación y uso de fuentes Web..... | 8 |
| Controlando el color en CSS..... | 9 |
| Sintaxis del color..... | 10 |
| Palabras clave: | 10 |
| Hexadecimal: | 10 |
| RGB y RGBA: | 11 |
| HSL y HSLA:..... | 12 |
| ¿Cuándo elegir un sistema u otro?..... | 13 |
| Palabras clave de color | 13 |
| Sistema hexadecimal | 14 |
| Sistema RGB y RGBA | 14 |
| Sistema HSL y HSLA | 14 |
| Accesibilidad y contraste | 15 |
| Importancia del contraste de color | 15 |
| Herramientas para probar el contraste..... | 16 |
| Buenas prácticas..... | 16 |
| Conclusión | 18 |
| Recursos adicionales | 19 |
| Enlaces a documentación y tutoriales | 19 |
| Herramientas de diseño de tipografía y color | 19 |
| Comprobación de accesibilidad y contraste..... | 20 |

Introducción

Bienvenido al módulo sobre "Tipografía y colores", dos elementos esenciales que juegan un papel crucial en el diseño de cualquier página web.

La forma en que presentamos el texto y utilizamos el color no solo define la estética y la identidad visual de un sitio, sino que también influye significativamente en la usabilidad y accesibilidad para los usuarios. A lo largo de este módulo, exploraremos cómo CSS nos permite manipular la tipografía y el color para crear experiencias de usuario atractivas, coherentes y accesibles.

No olvides que la tipografía no se trata solo de elegir fuentes bonitas; es mucho más. La tipografía tiene que ver con cómo el texto se presenta y se organiza para facilitar la lectura, transmitir el tono adecuado y mantener al usuario comprometido. La elección de la tipografía correcta y el manejo apropiado del estilo del texto pueden mejorar enormemente la legibilidad y la comprensión del contenido.

El color, por su parte, va más allá de la estética. Los colores pueden evocar emociones, destacar elementos importantes, guiar la atención del usuario y, lo que es más importante, contribuir a la accesibilidad cuando se utilizan contrastes adecuados. La correcta aplicación del color es fundamental para asegurar que todos los usuarios, incluidos aquellos con discapacidades visuales, puedan navegar e interactuar con el sitio eficientemente.

En este módulo, nos enfocaremos en:

- *Fuentes y estilos de texto: Aprenderás a incorporar y personalizar fuentes web, ajustar el tamaño del texto, el espaciado entre letras y líneas, y aplicar estilos como negrita e itálica para mejorar la tipografía de tu sitio.*
- *Colores en CSS: Exploraremos cómo definir colores para el texto, fondos y bordes, utilizar modelos de color como RGB, HEX y HSL, y aplicar transparencias. Además, discutiremos la importancia de mantener un contraste adecuado para la accesibilidad.*
- *Mejoras en la experiencia del usuario: Veremos cómo la tipografía y el color, cuando se usan estratégicamente, pueden mejorar la experiencia general del usuario, reforzar la identidad de la marca y garantizar la accesibilidad del sitio.*

¡Comencemos!

Fundamentos de la tipografía en CSS



La tipografía es uno de los pilares del diseño web, esencial para comunicar mensajes de manera efectiva y crear experiencias de usuario agradables. Este módulo se centra en los fundamentos de la tipografía en CSS, explorando cómo seleccionar y utilizar fuentes, ajustar el tamaño y la altura de línea, y aplicar estilos y pesos para enriquecer la presentación del texto.

Familias de fuentes

La propiedad `font-family` en CSS te permite especificar la familia tipográfica que quieres elegir para el texto de tus elementos.

```
p {  
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

1

¹ Accesibilidad:

```
p {  
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

Buenas prácticas a tener en cuenta son las siguientes:

- Es recomendable proporcionar varias fuentes como respaldo, en caso de que la primera elección no esté disponible en el sistema del usuario.
- Incluye siempre al final de esa lista al menos una familia genérica (serif, sans-serif, monospace, cursive, fantasy) por si las fuentes específicas no están disponibles.

Al elegir fuentes, considera la legibilidad, el tono del contenido y la coherencia con la identidad visual de la marca o el proyecto. En cualquier caso, las fuentes sans-serif como "Helvetica" o "Arial" suelen ser una elección popular para el texto del cuerpo por su legibilidad en pantallas

Tamaño y altura de línea

El font-size determina el tamaño de la tipografía, es decir, qué tan grande o pequeño se muestra el texto. Esta propiedad puede ser especificada en unidades relativas como em o rem, que son más flexibles y adaptativas, o en unidades absolutas como px..

Elegir el tamaño de fuente adecuado es fundamental para garantizar que el texto sea fácil de leer y que se ajuste al diseño general de la página.

```
body {  
  font-size: 16px; /* Un buen tamaño base para legibilidad */  
}
```

2

Por otro lado, el line-height, o altura de línea, especifica la altura de una línea de texto. No se limita solo al espacio ocupado por los caracteres (que sería determinado por el font-size), sino que incluye también el espacio adicional por encima y por debajo de los caracteres que conforman esa línea.

² Accesibilidad:

```
body {  
  font-size: 16px; /* Un buen tamaño base para la legibilidad */  
}
```

Esta propiedad es esencial para la legibilidad, ya que un line-height demasiado ajustado puede hacer que el texto parezca apretado y difícil de leer, mientras que un line-height excesivamente amplio puede dispersar demasiado el texto y romper la armonía visual.

```
p {
  line-height: 1.5;
}
```

3

Estilos y peso de fuente

El font-style permite aplicar estilos como la cursiva, mientras que el font-weight controla el grosor de la fuente, permitiendo crear jerarquías visuales y destacar partes importantes del contenido.

```
em {
  font-style: italic; /* Aplica cursiva al texto enfatizado */
}

strong {
  font-weight: bold; /* Hace que el texto importante sea más grueso y destacado */
}
```

4

³ Accesibilidad:

```
p {
  line-height: 1.5;
}
```

⁴ Accesibilidad:

```
em {
  font-style: italic; /* Aplica cursiva al texto enfatizado */
}

strong {
  font-weight: bold; /* Hace que el texto sea más grueso y destacado */
}
```

Importación y uso de fuentes Web



La elección de la tipografía es un aspecto crucial en el diseño web que va más allá de la simple estética; afecta la legibilidad, la accesibilidad y la percepción general de un sitio web. En este sentido, y si bien los navegadores web vienen con un conjunto predeterminado de fuentes "seguras para la web", estas opciones pueden ser limitadas y no siempre se alinean con la identidad visual o las necesidades específicas de un proyecto. Para superar estas limitaciones y enriquecer la experiencia del usuario, quizá en algún momento necesites importar fuentes web. Este proceso nos permite acceder a un espectro más amplio de opciones tipográficas, desde fuentes modernas y elegantes hasta estilos más específicos y únicos, proporcionando así una mayor libertad creativa y coherencia de marca.

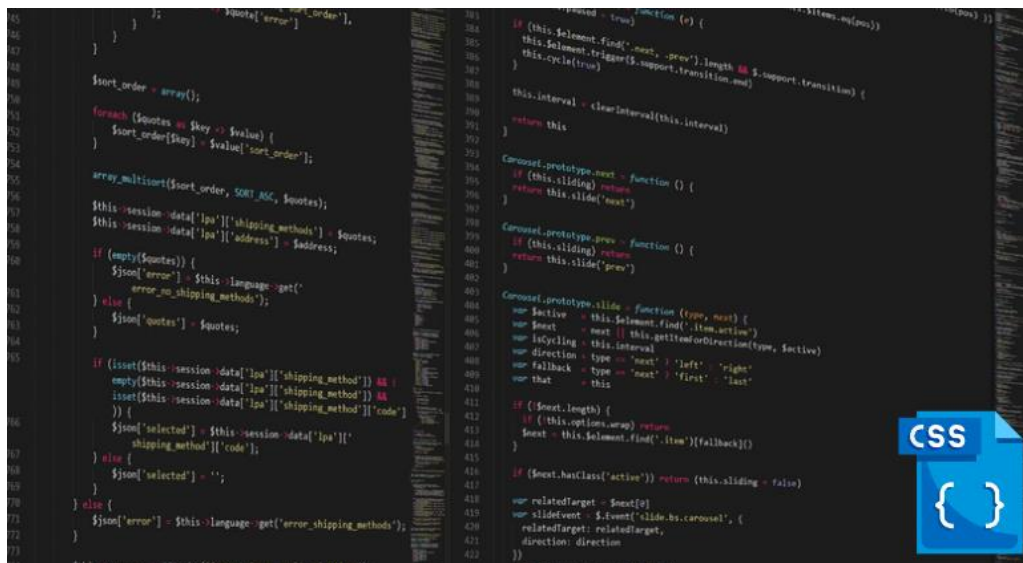
Pero ¿de dónde las vas a importar a tu diseño?

Algunas de las opciones más relevantes son:

- Google Fonts es un directorio ampliamente utilizado que ofrece una gran variedad de fuentes web gratuitas.
- Font Squirrel ofrece una selección de fuentes gratuitas para uso comercial. También proporciona un generador de @font-face que te permite convertir cualquier fuente a los formatos web adecuados.

Hay otras opciones que no son gratuitas como Adobe Fonts.

Controlando el color en CSS



El color es una herramienta poderosa en el diseño web, capaz de influir en las emociones, dirigir la atención del usuario y reforzar la identidad de marca. En este módulo, exploraremos cómo CSS nos permite manipular el color de manera precisa y efectiva en nuestros proyectos web, desde la sintaxis básica hasta la aplicación de teorías del color y la creación de paletas coherentes.

Sintaxis del color

CSS ofrece varias maneras de especificar colores, cada una con sus propias ventajas:

Palabras clave:

CSS define un conjunto de palabras clave para colores, como red, blue, green, etc. Son sencillas y fáciles de recordar, pero limitadas en número.

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

5

Hexadecimal:

El sistema hexadecimal de color en CSS es una forma popular de especificar colores en diseño web. Este sistema utiliza una combinación de seis dígitos y letras que representan los valores de los componentes rojo, verde y azul (RGB) de un color.

Los colores hexadecimales comienzan con # seguido de 6 caracteres (0-9 y A-F) que como hemos dicho representan los valores de rojo, verde y azul (RGB). También hay versiones cortas con 3 caracteres.

Cada par de dígitos/letras en un color hexadecimal puede tener valores desde 00 hasta FF. En el sistema decimal, esto equivale a un rango de 0 a 255. Por lo tanto, 00 representa la ausencia total de ese color, mientras que FF representa la máxima intensidad de ese color.

Combinando diferentes valores para los componentes rojo, verde y azul, puedes crear más de 16 millones de colores posibles (256 x 256 x 256).

⁵ Accesibilidad:

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

```
p {
  color: #ff0000; /* Rojo */
}
```

6

Ejemplos:

- Negro: #000000 (sin rojo, verde ni azul).
- Blanco: #FFFFFF (máxima intensidad de rojo, verde y azul).
- Rojo Puro: #FF0000 (máxima intensidad de rojo, sin verde ni azul).
- Verde Puro: #00FF00 (máxima intensidad de verde, sin rojo ni azul).
- Azul Puro: #0000FF (máxima intensidad de azul, sin rojo ni verde).

RGB y RGBA:

El sistema RGB (Red, Green, Blue) es un modelo de color ampliamente utilizado en el diseño digital, incluido el diseño web, que se basa en la mezcla de luz roja, verde y azul para crear una amplia gama de colores. En CSS, los colores RGB se especifican utilizando la función `rgb()`, donde se proporcionan tres valores numéricos que representan la intensidad de cada uno de los componentes de color rojo, verde y azul, respectivamente con valores de 0 a 255.

`rgba()` añade un cuarto valor para la opacidad, de 0 (completamente transparente) a 1 (completamente opaco).

```
p {
  color: rgba(255, 0, 0, 0.5); /* Rojo semi-transparente */
}
```

7

⁶ Accesibilidad:

```
p {
  color: #ff0000; /* Aplica al párrafo el color rojo */
}
```

⁷ Accesibilidad:

```
p {
  color: rgba(255, 0, 0, 0.5); /* Aplica al párrafo un color rojo semi-transparente */
}
```

Cada uno de los tres valores en `rgb()` define la intensidad del color correspondiente, donde 0 representa ninguna intensidad (apagado) y 255 representa la máxima intensidad (completamente encendido).

Combinando diferentes valores de rojo, verde y azul, se pueden generar más de 16 millones de colores posibles (256 x 256 x 256).

Ejemplos:

- Negro: `rgb(0, 0, 0)` (ninguna intensidad de rojo, verde o azul).
- Blanco: `rgb(255, 255, 255)` (máxima intensidad de rojo, verde y azul).
- Rojo Puro: `rgb(255, 0, 0)` (máxima intensidad de rojo, sin verde ni azul).
- Verde Puro: `rgb(0, 255, 0)` (máxima intensidad de verde, sin rojo ni azul).
- Azul Puro: `rgb(0, 0, 255)` (máxima intensidad de azul, sin rojo ni verde).

HSL y HSLA:

El sistema HSL (Hue, Saturation, Lightness - Matiz, Saturación, Luminosidad) es un modelo de color utilizado en el diseño web que ofrece una forma intuitiva de especificar colores basándose en sus cualidades perceptuales. A diferencia de los modelos RGB y hexadecimal que se centran en cómo se mezcla la luz para crear colores, HSL se centra en cómo las personas perciben el color.

`hsl()` se basa en matiz (hue), saturación (saturation) y luminosidad (lightness). `hsla()` añade la opacidad. El matiz se mide en grados (de 0 a 360), mientras que la saturación y la luminosidad se miden en porcentajes.

```
p {  
  color: hsla(120, 100%, 50%, 0.3); /* Verde claro semi-transparente */  
}
```

8

⁸ Accesibilidad:

```
p {  
  color: hsla(120, 100%, 50%, 0.3); /* Aplica al párrafo un color verde claro semi-transparente */  
}
```

Ejemplos

- Rojo puro: `hsl(0, 100%, 50%)` - Rojo con máxima saturación y luminosidad media.
- Verde claro: `hsl(120, 100%, 75%)` - Verde con máxima saturación y alta luminosidad.
- Azul suave: `hsl(240, 100%, 25%)` - Azul con máxima saturación y baja luminosidad.

¿Cuándo elegir un sistema u otro?

La elección entre los sistemas de color RGB, HSL o hexadecimal en CSS depende de varios factores, incluyendo la preferencia personal, la facilidad de uso en un contexto específico, y las necesidades del proyecto. Aquí te doy algunas pautas para ayudarte a decidir cuándo elegir un sistema u otro:

Palabras clave de color

Las palabras clave son fáciles de recordar y escribir, lo que las hace convenientes para colores comunes y cuando se busca rapidez en el desarrollo sin necesidad de precisión en la paleta de colores.

Cuando usar palabras clave:

- Prototipado rápido y pruebas: Cuando estás experimentando con el diseño o construyendo prototipos rápidos, las palabras clave permiten aplicar cambios de color de manera ágil y sin complicaciones.
- Proyectos educativos o tutoriales: En contextos educativos, las palabras clave son excelentes para enseñar los conceptos básicos del color en CSS sin abrumar a los principiantes con sistemas de color más complejos.
- Colores comunes sin variaciones específicas: Para colores estándar donde no se requiere una tonalidad específica, las palabras clave pueden ser suficientes y evitan la necesidad de memorizar o buscar códigos de color.

No obstante, hay otros aspectos a considerar que no las hace tan aconsejables:

- Limitación en la variedad: Aunque la lista de palabras clave de color en CSS es bastante extensa, sigue siendo limitada en comparación con los

millones de colores que se pueden especificar mediante sistemas como hexadecimal, RGB o HSL.

- Precisión y consistencia: Para una marca o diseño que requiere colores precisos y consistentes, especialmente en diferentes medios y dispositivos, es recomendable usar sistemas de color más específicos como hexadecimal, RGB(A) o HSL(A).

Sistema hexadecimal

Es aconsejable usarlo si buscas:

- Simplicidad y brevedad: Si buscas una forma concisa de definir colores y estás cómodo con el sistema hexadecimal, este puede ser tu mejor opción. Es ampliamente utilizado y reconocido en el diseño web.
- Compatibilidad: Dado que el sistema hexadecimal ha sido un estándar en el diseño web desde los inicios, es una apuesta segura en términos de compatibilidad con todos los navegadores y dispositivos.

Sistema RGB y RGBA

Es aconsejable usarlo si buscas:

- Familiaridad con colores digitales: Si tienes experiencia en diseño gráfico o estás familiarizado con sistemas de color basados en la luz (como en monitores y televisores), el sistema RGB puede resultarte más intuitivo.
- Necesidad de transparencia: Cuando necesitas controlar la transparencia de un color, `rgba()` es una opción excelente, ya que te permite definir el color y su opacidad en una sola función.

Sistema HSL y HSLA

Es aconsejable usarlo si buscas:

- Intuitividad y ajustes finos: Si prefieres trabajar con colores de una manera que refleje más directamente la percepción humana del color, o necesitas hacer ajustes finos en términos de saturación y luminosidad, HSL es una excelente elección.
- Creación de paletas y variaciones de color: HSL es particularmente útil para crear paletas de colores coherentes y para ajustar colores de manera que mantengan el mismo tono pero varíen en luminosidad y saturación. Esto es útil para diseños que requieren variaciones sutiles de un mismo color.

Accesibilidad y contraste



La accesibilidad en el diseño web se refiere a la creación de contenido y interfaces que pueden ser utilizadas por todas las personas, incluidas aquellas con discapacidades. Un aspecto crucial de la accesibilidad es el contraste de color, que juega un papel fundamental en la legibilidad del texto y la facilidad de interacción con los elementos de la página.

Importancia del contraste de color

Un contraste adecuado entre el texto (o los elementos de la interfaz) y su fondo no solo mejora la legibilidad para todos los usuarios, sino que es especialmente crucial para aquellos con discapacidades visuales, como daltonismo o baja visión. Las pautas de accesibilidad web, como las WCAG (Pautas de Accesibilidad para el Contenido Web), establecen estándares mínimos de contraste que ayudan a garantizar que el contenido sea accesible.

En este sentido, las WCAG recomiendan un ratio de contraste mínimo de 4.5:1 para texto normal y 3:1 para texto grande o elementos gráficos de la interfaz de usuario.

Herramientas para probar el contraste

Varias herramientas en línea y plugins de navegador pueden ayudarte a evaluar y mejorar el contraste de color en tus diseños web:

- WebAIM Contrast Checker: Una herramienta fácil de usar que permite verificar rápidamente el contraste de colores y su cumplimiento con las WCAG. (<https://webaim.org/resources/contrastchecker/>)
- ColorZilla: Una extensión de navegador que, entre otras características, incluye un selector de color y un analizador de contraste.
(<https://chromewebstore.google.com/detail/colorzilla/bhlhnicpbhignbdhedgjhgdcmhohmnp?hl=es-419>)

Buenas prácticas

Las buenas prácticas de accesibilidad en relación con el color y el contraste son esenciales para asegurar que tu contenido web sea accesible para todos los usuarios, incluidos aquellos con discapacidades visuales. Aquí te detallo algunas de las mejores prácticas:

- Cumple con las pautas WCAG
Contraste mínimo: Asegúrate de que el texto tenga un contraste de color suficiente respecto a su fondo. Como hemos visto, las Pautas de Accesibilidad para el Contenido Web (WCAG) recomiendan un ratio de contraste de al menos 4.5:1 para el texto normal y 3:1 para el texto grande y los componentes de la interfaz de usuario.
- Usa colores con cuidado
No confíes solo en el color: No utilices el color como el único medio para transmitir información. Asegúrate de que la información que depende del color esté disponible también a través de otros medios, como el texto o los marcadores visuales.
- Considera diferentes tipos de discapacidades visuales
Daltonismo: Diseña tus interfaces teniendo en cuenta a los usuarios con daltonismo. Evita combinaciones de colores problemáticas, como rojo/verde o azul/púrpura, que son comúnmente difíciles de distinguir.

- Proporciona suficiente contraste entre los elementos interactivos botones e hipervínculos: Asegúrate de que los botones, enlaces y otros elementos interactivos no solo se distingan por el color. Proporciona indicadores adicionales, como subrayados o cambios de forma, especialmente cuando se pasan por encima con el cursor o se seleccionan.

Conclusión

Al explorar la importancia de la accesibilidad y el contraste en el diseño web, hemos abierto una ventana a una práctica de diseño inclusiva que no solo mejora la experiencia del usuario para una audiencia más amplia, sino que también eleva la calidad general de nuestros proyectos web. La comprensión y aplicación de principios de contraste adecuados no solo es una cuestión de cumplimiento normativo, sino también una expresión de empatía y consideración hacia la diversidad de usuarios que interactúan con nuestro contenido digital.

Este módulo marca solo el comienzo de tu exploración de la accesibilidad en el diseño web. Te animo a seguir aprendiendo, probando y aplicando los principios de accesibilidad en todos tus proyectos. Las herramientas y recursos discutidos aquí son excelentes puntos de partida, pero la verdadera maestría proviene de la aplicación consciente y la reflexión sobre tu trabajo.

Nos vemos en la siguiente lección.

Recursos adicionales

Para concluir nuestra serie sobre tipografía y color en CSS, aquí te proporciono una selección de recursos adicionales que te ayudarán a profundizar en estos temas y a experimentar con diseños innovadores.

Enlaces a documentación y tutoriales

- MDN Web Docs (Mozilla Developer Network): Este recurso ofrece una documentación exhaustiva sobre CSS, incluyendo secciones dedicadas a la tipografía y al color.
 - MDN - CSS: Texto y Fuentes
(https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Fundamentals)
 - MDN - CSS: Color
(https://developer.mozilla.org/es/docs/Web/CSS/color_value)
- CSS-Tricks: Un sitio lleno de consejos, trucos y guías sobre todos los aspectos de CSS, incluyendo la tipografía y el color.
 - CSS-Tricks – Tipografía
(<https://css-tricks.com/almanac/properties/f/font/>)
 - CSS-Tricks – Color
(<https://css-tricks.com/almanac/properties/c/color/>)

Herramientas de diseño de tipografía y color

- Google Fonts: Una biblioteca de fuentes gratuitas optimizadas para la web, que te permite explorar y utilizar una amplia gama de estilos tipográficos en tus proyectos. (<https://fonts.google.com/>)
- Font Squirrel: Un recurso para descargar fuentes gratuitas y utilizar su herramienta Webfont Generator para convertir cualquier fuente en un formato web utilizable. (<https://www.fontsquirrel.com/>)
- Coolers: Un generador de esquemas de color rápido que te permite crear, compartir y explorar miles de combinaciones de colores, perfecto para encontrar la paleta perfecta para tu proyecto. (<https://coolers.co/>)
- Canva Color Wheel: Una herramienta útil para entender la teoría del color y generar esquemas de color armoniosos. (<https://www.canva.com/colors/color-wheel/>)

Comprobación de accesibilidad y contraste

- WebAIM Contrast Checker: Una herramienta sencilla que te permite comprobar rápidamente el contraste de colores en relación con las directrices WCAG. (<https://webaim.org/resources/contrastchecker/>)
- ColorZilla: Una extensión de navegador muy útil que incluye un selector de color, un cuentagotas y un analizador de gradiente, facilitando el trabajo con colores en la web.
 - ColorZilla for Chrome
(<https://chromewebstore.google.com/detail/colorzilla/bhlhnicpbhignbdhedgjhgdncnmhomnp?hl=es-419>)
 - ColorZilla for Firefox
(<https://addons.mozilla.org/es/firefox/addon/colorzilla/>)