

Guía Básica para Aprender CSS

1. ¿Qué es CSS?

CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para darle estilo a las páginas web. Mientras que HTML define la estructura del contenido, CSS define cómo se ve ese contenido: colores, fuentes, márgenes, alineación, etc.

2. Instalación y Configuración

CSS no necesita instalación, ya que se utiliza directamente en los archivos HTML. Existen dos formas comunes de aplicar CSS:

1. **CSS en línea:** Usado directamente dentro de la etiqueta HTML.
2. **CSS Interno:** Definido dentro de la etiqueta `<style>` dentro del archivo HTML.
3. **CSS Externo:** Usado en un archivo separado con extensión `.css`, y luego vinculado al archivo HTML.

Paso 1: Crear un archivo CSS

1. Crea un archivo con extensión `.css` (por ejemplo, `styles.css`).
2. Vincula el archivo CSS a tu archivo HTML utilizando la etiqueta `<link>` en el encabezado de tu documento HTML:

```
html
Copiar código
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

3. Conceptos Básicos de CSS

3.1 Sintaxis Básica

Un bloque de CSS consta de un **selector** y una **declaración**.

- **Selector:** Es el elemento HTML al que se aplicarán los estilos.
- **Declaración:** Es el conjunto de reglas que definen los estilos, compuesto por **propiedades** y **valores**.

Ejemplo de un bloque CSS:

```
css
Copiar código
h1 {
  color: blue;
  font-size: 24px;
}
```

En este caso, el **selector** es `h1`, y la **declaración** contiene dos reglas: `color` (azul) y `font-size` (24 píxeles).

3.2 Selectores en CSS

- **Selector de etiqueta:** Aplica el estilo a todas las instancias de la etiqueta seleccionada.

```
css
Copiar código
p {
  color: red;
}
```

- **Selector de clase:** Aplica el estilo a los elementos con una clase específica.

```
css
Copiar código
.mi-clase {
```

```
color: green;
}
```

- **Selector de ID:** Aplica el estilo a un solo elemento con un ID específico.

```
css
Copiar código
#mi-id {
  font-size: 18px;
}
```

- **Selector universal:** Aplica el estilo a todos los elementos.

```
css
Copiar código
* {
  margin: 0;
  padding: 0;
}
```

3.3 Propiedades Básicas de CSS

Colores

Puedes usar nombres de colores, valores hexadecimales, RGB, RGBA, HSL, etc.

```
css
Copiar código
h1 {
  color: red; /* Nombre de color */
}

p {
  color: #00ff00; /* Código hexadecimal */
}
```

```
div {  
  color: rgb(255, 0, 0); /* RGB */  
}
```

Fuentes

- **font-family:** Define la fuente del texto.
- **font-size:** Define el tamaño de la fuente.
- **font-weight:** Controla el grosor de la fuente.

```
css  
Copiar código  
p {  
  font-family: Arial, sans-serif;  
  font-size: 18px;  
  font-weight: bold;  
}
```

Fondo

- **background-color:** Define el color de fondo.
- **background-image:** Define una imagen de fondo.

```
css  
Copiar código  
body {  
  background-color: #f0f0f0;  
}  
  
div {  
  background-image: url('fondo.jpg');  
  background-size: cover;  
}
```

Espaciado

- **margin:** Define el espacio exterior del elemento.
- **padding:** Define el espacio interior del elemento.

```
css
Copiar código
div {
  margin: 20px;
  padding: 15px;
}
```

3.4 Modelo de Caja (Box Model)

El modelo de caja es uno de los conceptos fundamentales de CSS. Cada elemento HTML puede verse como una caja que tiene:

1. **Contenido:** El contenido real del elemento.
2. **Padding:** Espacio dentro de la caja, alrededor del contenido.
3. **Border:** El borde de la caja.
4. **Margin:** Espacio fuera de la caja, separando el elemento de otros.

Este modelo te ayuda a comprender cómo se gestionan el tamaño y el espaciado de los elementos en una página web.

3.5 Diseño y Posicionamiento

Flexbox

Flexbox es un sistema de diseño que facilita la creación de layouts. Permite alinear elementos de forma flexible y distribuir espacio entre ellos.

```
css
Copiar código
.container {
  display: flex;
  justify-content: space-between;
}
```

```
}
```

- **justify-content:** Controla la alineación horizontal.
- **align-items:** Controla la alineación vertical.

Grid

CSS Grid es otro sistema de diseño que proporciona un control más preciso sobre el layout.

```
css
Copiar código
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  gap: 10px;
}
```

3.6 Transiciones y Animaciones

Transiciones

Permiten hacer cambios suaves de un estado a otro.

```
css
Copiar código
button {
  background-color: blue;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: red;
}
```

Animaciones

Definen un cambio de estado en un elemento a lo largo del tiempo.

```
css
Copiar código
@keyframes mover {
  from { transform: translateX(0); }
  to { transform: translateX(100px); }
}

div {
  animation: mover 2s infinite;
}
```

4. Ejercicios Básicos de CSS

Ejercicio 1: Estilizar una página web

Crea un archivo HTML con un encabezado (`<h1>`), un párrafo (`<p>`), y una lista (``), luego:

1. Aplica un color de fondo a la página.
2. Estiliza el encabezado para que tenga un color de texto diferente y un tamaño mayor.
3. Cambia el color de los elementos de la lista al pasar el ratón sobre ellos.

Ejercicio 2: Crear una tarjeta de presentación

Crea una tarjeta con un nombre, foto, y descripción, y usa Flexbox para centrarla tanto horizontal como verticalmente en la página.

5. Recursos para Seguir Aprendiendo CSS

1. MDN Web Docs (Mozilla)

[CSS en MDN](#) – Documentación oficial y completa sobre CSS.

2. W3Schools

W3Schools CSS – Tutorial interactivo y ejemplos prácticos.

3. **CSS-Tricks**

CSS-Tricks – Consejos y trucos útiles para diseñadores y desarrolladores CSS.

Sígueme en Instagram para más contenido y actualizaciones:

@joss_0700