

GUÍA DE REFERENCIA CSS

INTRODUCCIÓN

CSS (Cascading Style Sheets) es el lenguaje utilizado para definir el estilo y la apariencia de una página web. Trabaja junto con HTML (que estructura el contenido) y JavaScript (que añade interactividad). CSS permite controlar colores, fuentes, diseños, animaciones y mucho más, lo que hace que las páginas web sean visualmente atractivas y responsivas.

Índice

CONTENIDO

Introducción	1
1. Selectores 	1
2. Propiedades de Texto 	2
3. Box Model (Modelo de Caja) 	3
4. Display 	4
5. Position 	4
top, right, bottom, left	5
6. Flexbox 	5
7. Grid 	5
8. Background 	6
9. Transiciones (Transitions) 	6
10. Animaciones (Animations) 	6
11. Media Queries 	7
12. Variables CSS (Custom Properties) 	7
13. Pseudo-clases y Pseudo-elementos 	7
14. Transformaciones (Transforms) 	8
scale	8
15. Sombras (Box Shadow y Text Shadow) 	8
Consejos Finales	9
Recursos Adicionales	9

1. SELECTORES

- **Descripción:** Los selectores se utilizan para seleccionar elementos HTML a los que se aplicarán estilos.
- **Tipos comunes:**

- **Selector de elemento:** Aplica estilos a todos los elementos de un tipo específico.

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

- **Selector de clase:** Aplica estilos a todos los elementos con una clase específica.

```
.mi-clase {  
    font-size: 16px;  
}
```

- **Selector de ID:** Aplica estilos a un elemento con un ID específico.

```
#mi-id {  
    background-color: yellow;  
}
```

2. PROPIEDADES DE TEXTO

- **Descripción:** Controlan el estilo del texto, como el color, tamaño, alineación, etc.
- **Propiedades comunes:**
 - **color:** Define el color del texto.
 - **Valores más usados:** red, blue, green, black, white, #hex, rgb(r, g, b), rgba(r, g, b, a).

```
p {  
    color: red;  
}
```

- **font-family:** Define la familia de la fuente.
- **Valores más usados:** Arial, Verdana, Times New Roman, Courier New, sans-serif, serif, monospace.

```
body {  
    font-family: Arial, sans-serif;  
}
```

- **font-size:** Define el tamaño de la fuente.
- **Valores más usados:** 12px, 14px, 16px, 18px, 20px, 24px, 1em, 1.5em, 2em.

```
h1 {  
    font-size: 24px;  
}
```

- **font-weight:** Controla el grosor del texto.
- **Valores más usados:** normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.

```
p {
```

```
font-weight: bold;
}
```

- **text-align:** Alinea el texto dentro de su contenedor.
- Valores más usados: start, end, left, right, center, justify, match-parent.

```
p {
    text-align: center;
}
```

3. BOX MODEL (MODELO DE CAJA)

- **Descripción:** Define el diseño y el espaciado de los elementos (margen, borde, relleno y contenido).
- **Propiedades comunes:**
 - **width:** Define el ancho de un elemento.
 - Valores más usados: 100px, 200px, 50%, auto, inherit.

```
.contenedor {
    width: 50%;
}
```

- **height:** Define la altura de un elemento.
- Valores más usados: 100px, 200px, 50%, auto, inherit.

```
.contenedor {
    height: 200px;
}
```

- **margin:** Define el margen externo de un elemento.
- Valores más usados: 0, auto, 10px, 20px, 1em, 2em.

```
p {
    margin: 20px;
}
```

- **padding:** Define el espacio interno de un elemento.
- Valores más usados: 0, 10px, 20px, 1em, 2em.

```
.contenedor {
    padding: 10px;
}
```

- **border:** Establece un borde alrededor del elemento.
- Valores más usados: 1px solid black, 2px dashed red, 3px dotted blue.

```
div {
```

```
border: 2px solid black;
}
```

4. DISPLAY

- **Descripción:** Controla cómo se muestra un elemento (en bloque, en línea, etc.).
- **Valores comunes:** block, inline, inline-block, none, flex, grid, table.
 - **block:** El elemento ocupa todo el ancho disponible.

```
div {
  display: block;
}
```

- **inline:** El elemento ocupa solo el espacio necesario.

```
span {
  display: inline;
}
```

- **flex:** Convierte el contenedor en un contenedor flexible.

```
.container {
  display: flex;
}
```

5. POSITION

- **Descripción:** Define la posición de un elemento (estático, relativo, absoluto, fijo, etc.).
- **Valores comunes:** static, relative, absolute, fixed, sticky.
 - **relative:** El elemento se posiciona relativamente a su posición normal.

```
.elemento {
  position: relative;
  top: 10px;
}
```

- **absolute:** El elemento se posiciona relativamente al primer ancestro posicionado.

```
.elemento {
  position: absolute;
  top: 0;
  left: 0;
}
```

- **fixed:** El elemento se posiciona relativamente a la ventana del navegador.

```
.menu {
```

```
position: fixed;
top: 0;
}
```

TOP, RIGHT, BOTTOM, LEFT

- **Descripción:** Define la posición de un elemento posicionado.
- **Valores más usados:** 0, 10px, 20px, 50%, auto.

```
.elemento {
  position: relative;
  top: 10px;
  left: 20px;
}
```

6. FLEXBOX

- **Descripción:** Un modelo de diseño para crear diseños flexibles y responsivos.
- **Propiedades comunes:**
 - **justify-content:** Alinea los elementos horizontalmente.

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

- **align-items:** Alinea los elementos verticalmente.

```
.container {
  align-items: center;
}
```

7. GRID

- **Descripción:** Un sistema de diseño basado en cuadrículas para crear layouts complejos.
- **Propiedades comunes:**
 - **grid-template-columns:** Define las columnas de la cuadrícula.

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
}
```

- **gap:** Define el espacio entre las celdas de la cuadrícula.

```
.grid-container {  
  gap: 10px;  
}
```

8. BACKGROUND

- **Descripción:** Controla el fondo de un elemento (color, imagen, posición, etc.).
- **Propiedades comunes:**
 - **background-color:** Define el color de fondo.

```
body {  
  background-color: lightblue;  
}
```

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

```
body {  
  background-image: url('fondo.jpg');  
}
```

- **background-size:** Ajusta el tamaño de la imagen de fondo.

- **Valores más usados:** auto, cover, contain, 100%, 50%.

```
body {  
  background-size: cover;  
}
```

9. TRANSICIONES (TRANSITIONS)

- **Descripción:** Permite crear efectos de transición suaves entre estados.
- **Valores más usados:** all 0.3s ease, color 0.5s linear, background-color 0.3s ease-in-out.
- **Ejemplo:**

```
button {  
  transition: background-color 0.3s ease;  
}  
button:hover {  
  background-color: #ff6347;  
}
```

10. ANIMACIONES (ANIMATIONS)

- **Descripción:** Permite crear animaciones complejas usando keyframes.

- **Valores más usados:** name, duration, timing-function, delay, iteration-count, direction, fill-mode.
- **Ejemplo:**

```
@keyframes slide {  
  from { transform: translateX(0); }  
  to { transform: translateX(100px); }  
}  
.box {  
  animation: slide 2s infinite;  
}
```

11. MEDIA QUERIES

- **Descripción:** Permite aplicar estilos condicionales basados en el tamaño de la pantalla o dispositivo.
- **Valores más usados:** max-width, min-width, max-height, min-height, orientation.
- **Ejemplo:**

```
@media (max-width: 768px) {  
  body {  
    font-size: 14px;  
  }  
}
```

12. VARIABLES CSS (CUSTOM PROPERTIES)

- **Descripción:** Permite definir valores reutilizables en todo el CSS.
- **Ejemplo:**

```
:root {  
  --main-color: #3498db;  
}  
.header {  
  background-color: var(--main-color);  
}
```

13. PSEUDO-CLASES Y PSEUDO-ELEMENTOS

- **Descripción:** Permite aplicar estilos a elementos en estados específicos o partes de elementos.

- **Valores más usados:** :hover, :focus, :active, :first-child, :last-child, :nth-child, ::before, ::after, ::first-line, ::first-letter.
- **Ejemplo:**

```
a:hover {  
    color: red;  
}  
p::first-line {  
    font-weight: bold;  
}
```

14. TRANSFORMACIONES (TRANSFORMS)

- **Descripción:** Permite modificar la forma, tamaño y posición de un elemento.
- **Valores más usados:** rotate(45deg), scale(1.5), translate(10px, 20px), skew(20deg, 10deg).
- **Ejemplo:**

```
.rotate {  
    transform: rotate(45deg);  
}
```

SCALE

- **Descripción:** Aplica una escala a un elemento.
- **Valores más usados:** scale(1.5), scaleX(1.5), scaleY(1.5).
- **Ejemplo:**

```
.elemento {  
    transform: scale(1.5);  
}
```

15. SOMBRAS (BOX SHADOW Y TEXT SHADOW)

- **Descripción:** Añade sombras a elementos o texto.
- **Ejemplo:**

```
.card {  
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);  
}  
h1 {  
    text-shadow: 1px 1px 2px black;  
}
```

CONSEJOS FINALES

- **Organización:** Usa comentarios en tu CSS para mantener el código organizado.

```
/* Estilos para el header */  
.header { ... }
```

- **Responsive Design:** Siempre prueba tus diseños en diferentes dispositivos y tamaños de pantalla.
- **Herramientas:** Usa herramientas como **Chrome DevTools** para inspeccionar y depurar tu CSS.

RECURSOS ADICIONALES

- [MDN Web Docs: CSS](#)
- [CSS-Tricks](#)