

Tutorial Completo de CSS

Asistencia sobre aplicaciones Especificas - Profe: Agustin Gatica



Índice Completo

1. [Introducción a CSS](#)
2. [Configuración en VSCode](#)
3. [Sintaxis Básica de CSS](#)
4. [Selectores CSS](#)
5. [Propiedades de Texto](#)
6. [Colores y Fondos](#)
7. [Modelo de Caja \(Box Model\)](#)
8. [Posicionamiento](#)
9. [Estilos Avanzados](#)
10. [FLEXBOX](#)
11. [GRID](#)
12. [Tips de Expertos](#)



Introducción a CSS

¿Qué es CSS?

CSS (Cascading Style Sheets) es un lenguaje de estilos que controla la apariencia de los elementos HTML. Define colores, tamaños, espacios, animaciones y más.

¿Por qué CSS?

- Separa el contenido (HTML) de la presentación (CSS)
- Permite reutilizar estilos
- Facilita el mantenimiento del código
- Crea diseños responsivos



Configuración en VSCode

Paso 1: Instala Extensiones Recomendadas

1. Abre VSCode
2. Ve a **Extensiones** (Ctrl + Shift + X)
3. Instala estas extensiones:
 - o **Live Server** - Para ver cambios en tiempo real
 - o **CSS Peek** - Para navegar entre CSS e HTML
 - o **Prettier** - Para formatear código automáticamente
 - o **Thunder Client** o **REST Client** - Opcional para testing

Paso 2: Configura tu Espacio de Trabajo

```
# Crea una carpeta para tu proyecto
mkdir mi-proyecto-css
cd mi-proyecto-css
```

```
# Crea los archivos necesarios
touch index.html
touch style.css
```

Paso 3: Estructura Básica

index.html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Proyecto CSS</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>¡Bienvenido a CSS!</h1>
  <p>Este es mi primer proyecto.</p>
</body>
</html>
```

Tip de Experto: Usa la extensión Live Server para actualizar automáticamente. Click derecho en index.html → "Open with Live Server"



Estructura de una Regla CSS

```
/* Selector { propiedad: valor; } */

p {
  color: blue;           /* Propiedad: valor */
  font-size: 16px;       /* Cada propiedad termina con ; */
  text-align: center;
}
```

Los Tres Métodos para Aplicar CSS

1. CSS Inline (NO RECOMENDADO)

```
<p style="color: red; font-size: 18px;">Texto en rojo</p>
```

2. CSS Interno (RECOMENDADO PARA APRENDER)

```
<head>
  <style>
    p { color: blue; }
  </style>
</head>
```

3. CSS Externo (LA MEJOR PRÁCTICA)

```
<head>
  <link rel="stylesheet" href="style.css">
</head>
```

Selectores CSS

1. Selector de Elemento (Tag)

```
/* Aplica a TODOS los <p> */
p {
  color: darkblue;
  margin: 10px;
}
```

2. Selector de Clase (.)

```
.destacado {  
  background-color: yellow;  
  font-weight: bold;  
}
```

```
<p class="destacado">Este párrafo es destacado</p>  
<p class="destacado">Este también</p>
```

3. Selector de ID (#)

```
#encabezado {  
  color: white;  
  background-color: navy;  
  padding: 20px;  
}
```

```
<h1 id="encabezado">Título Principal</h1>
```

⚠ **Nota:** Un ID debe ser único por página. Las clases se pueden reutilizar.

4. Selector Universal (*)

```
* {  
  margin: 0;           /* Elimina márgenes por defecto */  
  padding: 0;  
  box-sizing: border-box; /* Muy importante (verás después) */  
}
```

5. Selectores Combinados

```
/* Descendientes - Cualquier <a> dentro de <nav> */  
nav a {  
  color: white;  
}
```

```
/* Hijo directo - Solo <li> que son hijos directos de <ul> */  
ul > li {  
  list-style: none;  
}
```

```
/* Hermanos adyacentes */  
h2 + p {  
  margin-top: 0;  
}
```

```
}

/* Todos los hermanos */
h2 ~ p {
    line-height: 1.8;
}
```

6. Selectores por Atributo

```
/* Elementos con atributo específico */
input[type="text"] {
    border: 2px solid blue;
    padding: 8px;
}

a[href^="https"] {
    color: green; /* URLs que comienzan con https */
}

a[href$=".pdf"] {
    color: red; /* URLs que terminan en .pdf */
}
```

7. Pseudo-clases

```
/* Estados del elemento */
a:hover {
    text-decoration: underline;
    color: orange;
}

a:active {
    color: red;
}

a:visited {
    color: purple;
}

input:focus {
    outline: 2px solid blue;
    background-color: lightyellow;
}

/* Primer y último hijo */
li:first-child {
    font-weight: bold;
}
```

```
li:last-child {
    border-bottom: none;
}

/* Elementos pares e impares */
tr:nth-child(even) {
    background-color: #f0f0f0;
}

tr:nth-child(odd) {
    background-color: white;
}
```

8. Pseudo-elementos

```
/* ::before - Insertar contenido antes */
p::before {
    content: "▶ ";
    color: red;
}

/* ::after - Insertar contenido después */
p::after {
    content: "◀";
    color: blue;
}

/* ::first-letter - Primera letra */
p::first-letter {
    font-size: 2em;
    font-weight: bold;
}

/* ::first-line - Primera línea */
p::first-line {
    text-transform: uppercase;
}

/* ::selection - Texto seleccionado */
::selection {
    background-color: yellow;
    color: black;
}
```

Propiedades de Texto

Fuentes

```

/* Especificar familia de fuentes */
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

/* Tamaño de fuente */
h1 { font-size: 32px; }      /* Píxeles (fijo) */
p { font-size: 1.2em; }     /* Em (relativo al padre) */
h2 { font-size: 2rem; }     /* Rem (relativo a HTML) */

/* Peso de fuente */
.normal { font-weight: 400; } /* normal */
.bold   { font-weight: 700; } /* bold */
.light  { font-weight: 300; }

/* Estilo */
em      { font-style: italic; }
strong  { font-style: normal; }

/* Variante */
.mayusculas { font-variant: small-caps; }

/* Propiedad corta */
body {
    font: italic bold 16px 'Arial', sans-serif;
}

```

Importar Fuentes de Google Fonts

En index.html:

```
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&family=Playfair+Displ
```



En style.css:

```

body {
    font-family: 'Roboto', sans-serif;
}

h1 {
    font-family: 'Playfair Display', serif;
}

```

Alineación y Espaciado de Texto

```

p {
    text-align: left;           /* izquierda */
    text-align: center;        /* centro */
    text-align: right;         /* derecha */
    text-align: justify;       /* justificado */
}

/* Decoración */
a {
    text-decoration: none;     /* Quita subrayado */
    text-decoration: underline;
    text-decoration: overline;
    text-decoration: line-through;
}

/* Transformación */
.mayusculas { text-transform: uppercase; }
.minusculas { text-transform: lowercase; }
.capitalize { text-transform: capitalize; }

/* Espaciado entre letras */
h1 { letter-spacing: 3px; }

/* Espaciado entre palabras */
p { word-spacing: 5px; }

/* Alto de línea */
p { line-height: 1.6; }      /* Recomendado para lectura */
p { line-height: 24px; }

```

Sombras de Texto

```

h1 {
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
    /* desplazamiento-x desplazamiento-y difuminado color */
}

/* Múltiples sombras */
h2 {
    text-shadow:
        2px 2px 4px red,
        -2px -2px 4px blue;
}

```



Colores y Fondos

Formas de Especificar Colores

```
p {  
    /* Nombre */  
    color: red;  
  
    /* Hexadecimal */  
    color: #FF0000;  
    color: #f00; /* Forma corta */  
  
    /* RGB */  
    color: rgb(255, 0, 0);  
  
    /* RGBA (con transparencia) */  
    color: rgba(255, 0, 0, 0.5); /* 0.5 = 50% opacidad */  
  
    /* HSL (Hue, Saturation, Lightness) */  
    color: hsl(0, 100%, 50%);  
  
    /* HSLA (con transparencia) */  
    color: hsla(0, 100%, 50%, 0.5);  
}
```

Propiedades de Fondo

```
div {  
    /* Color de fondo */  
    background-color: #f0f0f0;  
    background-color: rgba(255, 0, 0, 0.2);  
  
    /* Imagen de fondo */  
    background-image: url('imagen.jpg');  
  
    /* Repetición */  
    background-repeat: no-repeat; /* No repetir */  
    background-repeat: repeat-x; /* Solo horizontalmente */  
    background-repeat: repeat-y; /* Solo verticalmente */  
    background-repeat: repeat; /* Por defecto */  
  
    /* Posición */  
    background-position: top left;  
    background-position: center;  
    background-position: 50% 50%;  
    background-position: 10px 20px;  
  
    /* Tamaño */  
    background-size: cover; /* Cubre todo */  
    background-size: contain; /* Cabe dentro */  
    background-size: 100px 100px;  
    background-size: 50% 50%;
```

```

/* Anclaje */
background-attachment: fixed; /* La imagen no se mueve */
background-attachment: scroll; /* La imagen se mueve */
}

/* Propiedad corta */
div {
  background: url('bg.jpg') center/cover fixed no-repeat #333;
}

```

Gradientes

```

/* Gradiente lineal */
div {
  background: linear-gradient(to right, red, blue);
  background: linear-gradient(45deg, red, blue);
  background: linear-gradient(to right, red 0%, yellow 50%, blue 100%);
}

/* Gradiente radial */
div {
  background: radial-gradient(circle, red, blue);
  background: radial-gradient(ellipse, red 0%, blue 100%);
}

/* Múltiples colores */
div {
  background: linear-gradient(
    90deg,
    #ff0000 0%,
    #00ff00 25%,
    #0000ff 50%,
    #ffff00 75%,
    #ff00ff 100%
  );
}

```

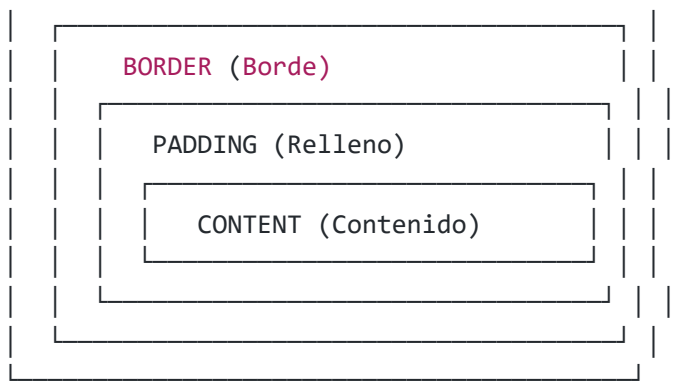


Modelo de Caja (Box Model)

Este es uno de los conceptos más importantes de CSS.

Estructura de la Caja





Margen (margin)

```
div {  
  /* Todos los lados */  
  margin: 20px;  
  
  /* Superior/Inferior, Izquierda/Derecha */  
  margin: 10px 20px;  
  
  /* Superior, Derecha, Inferior, Izquierda (TRLD) */  
  margin: 10px 20px 30px 40px;  
  
  /* Lados individuales */  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
  
  /* Centrar horizontalmente */  
  margin: 0 auto;  
}
```

Relleno (padding)

```
div {  
  /* Igual que margin */  
  padding: 20px;  
  padding: 10px 20px;  
  padding: 10px 20px 30px 40px;  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 40px;  
}
```

Borde (border)

```
div {
  /* Ancho, estilo, color */
  border: 2px solid black;

  /* Borde completo */
  border: 3px dashed red;

  /* Estilos: solid, dashed, dotted, double, groove, ridge */
  border-style: solid;
  border-width: 2px;
  border-color: black;

  /* Bordas individuales */
  border-top: 2px solid red;
  border-right: 3px dashed blue;
  border-bottom: 2px dotted green;
  border-left: 4px double orange;

  /* Radio de bordes (esquinas redondeadas) */
  border-radius: 10px; /* Todas las esquinas */
  border-radius: 10px 20px; /* Arriba-abajo, Derecha-izquierda */
  border-radius: 10px 20px 30px 40px; /* Arriba-der, Arriba-izq, Abajo-izq, Abajo-der */
  border-radius: 50%; /* Círculo perfecto */
}
```

Sombras (box-shadow)

```
div {
  /* desplazamiento-x desplazamiento-y difuminado propagación color */
  box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.3);

  /* Sombra interna */
  box-shadow: inset 5px 5px 10px rgba(0, 0, 0, 0.3);

  /* Múltiples sombras */
  box-shadow:
    5px 5px 10px rgba(0, 0, 0, 0.2),
    -5px -5px 10px rgba(255, 255, 255, 0.5);
}
```

box-sizing (Muy Importante ★)

```
/* Por defecto: width no incluye padding y border */
div {
  width: 100px;
  padding: 10px;
  border: 2px solid;
  /* Ancho TOTAL = 100 + 10*2 + 2*2 = 124px ✗ */
}
```

```
}

/* Solución: incluir padding y border en el ancho */
* {
  box-sizing: border-box; /* ✅ SIEMPRE USAR ESTO */
}

div {
  width: 100px;
  padding: 10px;
  border: 2px solid;
  /* Ancho TOTAL = 100px (como esperado) ✅ */
}
```

Posicionamiento

static (Por defecto)

```
div {
  position: static; /* No puede ser movido */
}
```

relative (Relativo)

```
div {
  position: relative;
  top: 10px; /* Mueve 10px hacia abajo */
  left: 20px; /* Mueve 20px hacia la derecha */
  right: 0;
  bottom: 0;
  /* Ocupa su espacio original, pero se desplaza visualmente */
}
```

absolute (Absoluto)

```
.contenedor {
  position: relative; /* Contenedor padre debe ser relativo */
}

.elemento {
  position: absolute;
  top: 50px;
  left: 50px;
```

```
    /* Se posiciona relativo a su padre más cercano posicionado */  
}
```

fixed (Fijo)

```
.navbar {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    background: navy;  
    color: white;  
    /* Se mantiene fijo incluso al desplazar la página */  
}
```

sticky (Pegajoso)

```
.encabezado {  
    position: sticky;  
    top: 0;  
    background: white;  
    /* Se comporta como static, pero se fija cuando alcanza la posición especificada */  
}
```

z-index (Capas)

```
.atras {  
    position: absolute;  
    z-index: 1;    /* Capa 1 (más atrás) */  
}  
  
.adelante {  
    position: absolute;  
    z-index: 999;  /* Capa 999 (más adelante) */  
}
```

✨ Estilos Avanzados

Opacidad (Transparency)

```
img {  
    opacity: 1;    /* Totalmente opaco (por defecto) */  
}
```

```

    opacity: 0.5;      /* 50% de opacidad */
    opacity: 0;        /* Totalmente transparente (invisible) */
}

img:hover {
    opacity: 0.8;      /* Al pasar el mouse */
}

```

Filtros

```

img {
    /* Escala de grises */
    filter: grayscale(100%);

    /* Brillo */
    filter: brightness(150%);

    /* Contraste */
    filter: contrast(150%);

    /* Saturación */
    filter: saturate(150%);

    /* Blur (difuminado) */
    filter: blur(5px);

    /* Sombra */
    filter: drop-shadow(5px 5px 10px rgba(0, 0, 0, 0.3));

    /* Invertir colores */
    filter: invert(100%);

    /* Múltiples filtros */
    filter: grayscale(100%) brightness(150%) blur(2px);
}

```

Transformaciones (Transformations)

```

div {
    /* Escala */
    transform: scale(1.5);           /* 150% del tamaño original */
    transform: scaleX(2);            /* Doble en el eje X */
    transform: scaleY(0.5);          /* Mitad en el eje Y */

    /* Rotación */
    transform: rotate(45deg);         /* Rotar 45 grados */

    /* Traslación (movimiento) */
    transform: translate(50px, 100px); /* Mover 50px derecha, 100px abajo */
}

```

```

transform: translateX(50px);
transform: translateY(100px);

/* Sesgo (skew) */
transform: skew(20deg);           /* Sesgar en X */
transform: skewX(20deg);
transform: skewY(20deg);

/* Perspectiva 3D */
transform: rotateX(45deg);
transform: rotateY(45deg);
transform: rotateZ(45deg);

/* Múltiples transformaciones */
transform: rotate(45deg) scale(1.5) translate(50px, 100px);
}

```

Transiciones (Smooth Animations)

```

button {
  background-color: blue;
  color: white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;

  /* Transición de todas las propiedades en 0.3 segundos */
  transition: all 0.3s ease;
}

button:hover {
  background-color: darkblue;
  transform: scale(1.1);
}

/* Transiciones específicas */
div {
  transition: background-color 0.3s ease,
             color 0.5s ease-in,
             transform 0.2s linear;
}

/* Propiedad, duración, función de tiempo */
p {
  transition: font-size 0.5s ease-in-out;
}

/* Tiempos de función:
- ease (por defecto)
- linear
- ease-in

```



```
- ease-out
- ease-in-out
- cubic-bezier(valores)

*/
```

Animaciones (Keyframes)

```
/* Definir animación */
@keyframes deslizar {
  0% {
    left: 0;
    opacity: 0;
  }
  50% {
    opacity: 1;
  }
  100% {
    left: 100px;
    opacity: 1;
  }
}

/* Aplicar animación */
div {
  animation: deslizar 2s ease-in-out infinite;

  /* Propiedades individuales */
  animation-name: deslizar;
  animation-duration: 2s;
  animation-timing-function: ease-in-out;
  animation-iteration-count: infinite; /* 1, 2, infinite */
  animation-direction: normal; /* normal, reverse, alternate */
  animation-delay: 0s; /* Esperar antes de comenzar */
  animation-fill-mode: forwards; /* Mantener estado final */
}

/* Animación de carga */
@keyframes girar {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

.spinner {
  animation: girar 2s linear infinite;
}

/* Animación de pulso */
```

```

@keyframes pulsar {
  0%, 100% {
    opacity: 1;
  }
  50% {
    opacity: 0.5;
  }
}

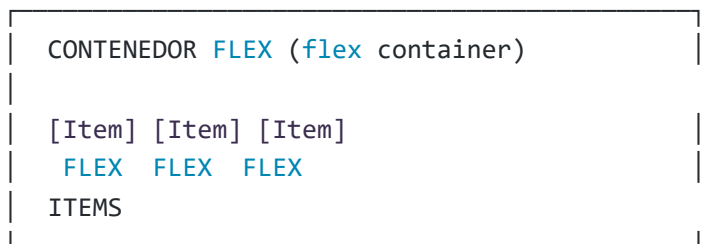
.boton {
  animation: pulsar 1s ease-in-out infinite;
}

```

FLEXBOX

Flexbox es un sistema de diseño moderno que facilita crear layouts flexibles.

Conceptos Básicos



Activar Flexbox

```

.contenedor {
  display: flex; /*  Activa Flexbox */
}

/* También existe inline-flex */
.contenedor {
  display: inline-flex;
}

```

Propiedades del Contenedor

1. flex-direction (Dirección)

```

.contenedor {
  display: flex;
}

```

```
flex-direction: row;           /* Horizontal (por defecto) */
flex-direction: row-reverse;   /* Horizontal inverso */
flex-direction: column;        /* Vertical */
flex-direction: column-reverse; /* Vertical inverso */
}
```

2. justify-content (Alineación Horizontal)

```
.contenedor {
  display: flex;
  justify-content: flex-start; /* Al inicio (por defecto) */
  justify-content: flex-end;   /* Al final */
  justify-content: center;     /* Centrado */
  justify-content: space-between; /* Espacio entre items */
  justify-content: space-around; /* Espacio alrededor */
  justify-content: space-evenly; /* Espacio igual */
}
```

3. align-items (Alineación Vertical)

```
.contenedor {
  display: flex;
  height: 300px;

  align-items: flex-start; /* Arriba */
  align-items: flex-end;   /* Abajo */
  align-items: center;     /* Centro */
  align-items: stretch;   /* Estirar (por defecto) */
  align-items: baseline;   /* Línea base del texto */
}
```

4. flex-wrap (Envoltura)

```
.contenedor {
  display: flex;

  flex-wrap: nowrap; /* Todo en una línea (por defecto) */
  flex-wrap: wrap;   /* Envuelve a la siguiente línea */
  flex-wrap: wrap-reverse; /* Envuelve al revés */
}
```

5. align-content (Alineación de Líneas)

```
.contenedor {
  display: flex;
```

```
flex-wrap: wrap;
height: 400px;

align-content: flex-start;
align-content: flex-end;
align-content: center;
align-content: space-between;
align-content: space-around;
align-content: stretch;
}
```

6. gap (Espacio entre Items)

```
.contenedor {
  display: flex;

  gap: 20px;           /* Espacio igual en ambas direcciones */
  gap: 20px 30px;      /* Vertical, Horizontal */
  column-gap: 20px;    /* Solo horizontal */
  row-gap: 30px;       /* Solo vertical */
}
```

Propiedades de Items (Flex Items)

1. flex-grow (Crecer)

```
.item {
  flex-grow: 0; /* No crece (por defecto) */
  flex-grow: 1; /* Crece igualmente con otros */
  flex-grow: 2; /* Crece el doble que flex-grow: 1 */
}
```

2. flex-shrink (Encoger)

```
.item {
  flex-shrink: 1; /* Se encoge (por defecto) */
  flex-shrink: 0; /* No se encoge */
}
```

3. flex-basis (Tamaño Base)

```
.item {
  flex-basis: auto; /* Tamaño automático */
  flex-basis: 200px; /* Ancho fijo */
}
```

```
    flex-basis: 50%;          /* Porcentaje */  
}
```

4. Propiedad Corta: flex

```
.item {  
    /* flex-grow flex-shrink flex-basis */  
    flex: 1 1 200px;  
  
    flex: 1;          /* Equivalente a: 1 1 0 */  
    flex: auto;       /* Equivalente a: 1 1 auto */  
    flex: none;       /* Equivalente a: 0 0 auto */  
}
```

5. align-self (Alineación Individual)

```
.item {  
    align-self: auto;          /* Hereda del contenedor */  
    align-self: flex-start;  
    align-self: flex-end;  
    align-self: center;  
    align-self: stretch;  
}
```

Ejemplo Completo: Barra de Navegación

```
/* HTML:  
<nav class="navbar">  
  <a href="#">Logo</a>  
  <a href="#">Inicio</a>  
  <a href="#">Sobre</a>  
  <a href="#">Contacto</a>  
</nav>  
*/  
  
.navbar {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    background-color: #333;  
    color: white;  
    padding: 20px;  
    gap: 20px;  
}  
  
.navbar a {  
    color: white;
```

```

    text-decoration: none;
    padding: 10px 15px;
    transition: 0.3s;
}

.navbar a:hover {
    background-color: #555;
}

```

Ejemplo: Tarjetas Flexibles

```

/* HTML:
<div class="contenedor-tarjetas">
  <div class="tarjeta">Tarjeta 1</div>
  <div class="tarjeta">Tarjeta 2</div>
  <div class="tarjeta">Tarjeta 3</div>
</div>
*/

.contenedor-tarjetas {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: center;
}

.tarjeta {
    flex: 1 1 300px; /* Crece, se encoge, base 300px */
    background-color: #f0f0f0;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    transition: 0.3s;
}

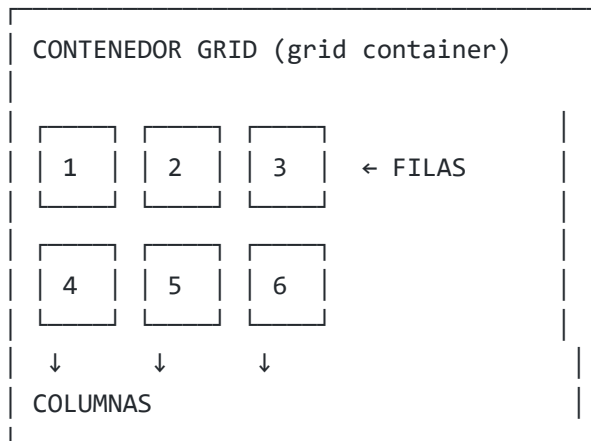
.tarjeta:hover {
    transform: translateY(-10px);
    box-shadow: 0 5px 20px rgba(0, 0, 0, 0.2);
}

```

GRID

CSS Grid es un sistema de diseño bidimensional (filas y columnas).

Conceptos Básicos



Activar Grid

```
.contenedor {  
  display: grid; /* ✅ Activa Grid */  
}  
  
/* También existe inline-grid */  
.contenedor {  
  display: inline-grid;  
}
```

Definir Columnas y Filas

1. grid-template-columns

```
.contenedor {  
  display: grid;  
  
  /* 3 columnas iguales */  
  grid-template-columns: 1fr 1fr 1fr;  
  
  /* Formas alternativas */  
  grid-template-columns: repeat(3, 1fr);  
  
  /* Columnas con tamaños específicos */  
  grid-template-columns: 200px 1fr 200px; /* Dos fijas, una flexible */  
  
  /* Columnas con diferentes unidades */  
  grid-template-columns: 300px auto 1fr;  
  
  /* Rango mínimo-máximo */  
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
  
  /* Etiquetas (nombres de líneas) */  
}
```

```
    grid-template-columns: [inicio] 1fr [medio] 1fr [fin];
}
```

2. grid-template-rows

```
.contenedor {
    display: grid;
    grid-template-rows: 100px 1fr 100px; /* Arriba, Contenido, Abajo */

    /* Auto ajusta según contenido */
    grid-template-rows: auto auto auto;

    /* Altura mínima */
    grid-template-rows: repeat(3, minmax(100px, auto));
}
```

3. gap (Espacio entre Celdas)

```
.contenedor {
    display: grid;

    gap: 20px; /* Vertical y Horizontal */
    gap: 20px 30px; /* Vertical, Horizontal */
    column-gap: 30px; /* Solo horizontal */
    row-gap: 20px; /* Solo vertical */
}
```

Posicionar Items en Grid

1. grid-column y grid-row

```
.item-1 {
    /* Comienza en columna 1, termina en columna 3 (ocupa 2 columnas) */
    grid-column: 1 / 3;

    /* Equivalente a */
    grid-column-start: 1;
    grid-column-end: 3;

    /* Otra forma: contar espacios */
    grid-column: 1 / span 2; /* Comienza en 1, ocupa 2 columnas */
}

.item-2 {
    /* Ocupa filas 2 a 4 */
    grid-row: 2 / 4;
}
```



```
    grid-row: 2 / span 2; /* Igual efecto */
}
```

2. grid-column-start, grid-column-end, etc.

```
.item {
    grid-column-start: 1;
    grid-column-end: 3;
    grid-row-start: 1;
    grid-row-end: 2;
}
```

Propiedades de Alineación

1. justify-items (Alineación Horizontal)

```
.contenedor {
    display: grid;

    justify-items: start; /* Izquierda */
    justify-items: end;   /* Derecha */
    justify-items: center; /* Centro */
    justify-items: stretch; /* Estirar (por defecto) */
}
```

2. align-items (Alineación Vertical)

```
.contenedor {
    display: grid;

    align-items: start; /* Arriba */
    align-items: end;   /* Abajo */
    align-items: center; /* Centro */
    align-items: stretch; /* Estirar */
}
```

3. justify-content y align-content

```
.contenedor {
    display: grid;
    width: 100%;

    /* Alineación de la grilla completa */
    justify-content: center;
    align-content: center;
}
```

```
    justify-content: space-around;
    align-content: space-between;
}
```

4. place-items (Atajo)

```
.contenedor {
    display: grid;

    /* Equivalente a: align-items: center; justify-items: center; */
    place-items: center;
}
```

Propiedades de Items

1. justify-self

```
.item {
    justify-self: start;
    justify-self: end;
    justify-self: center;
    justify-self: stretch;
}
```

2. align-self

```
.item {
    align-self: start;
    align-self: end;
    align-self: center;
    align-self: stretch;
}
```

Ejemplo Completo: Página Responsiva

```
/* HTML:
<div class="layout">
    <header>Encabezado</header>
    <nav>Navegación</nav>
    <main>Contenido Principal</main>
    <aside>Barra Lateral</aside>
    <footer>Pie de Página</footer>
</div>
*/
```

```

.layout {
  display: grid;
  grid-template-columns: 1fr 300px;
  grid-template-rows: auto auto 1fr auto;
  gap: 20px;
  min-height: 100vh;
}

header {
  grid-column: 1 / -1; /* Ocupa todas las columnas */
  background-color: #333;
  color: white;
  padding: 20px;
}

nav {
  grid-column: 1 / -1;
  background-color: #555;
  color: white;
  padding: 10px;
}

main {
  background-color: #f9f9f9;
  padding: 20px;
}

aside {
  background-color: #e0e0e0;
  padding: 20px;
}

footer {
  grid-column: 1 / -1;
  background-color: #333;
  color: white;
  padding: 20px;
  text-align: center;
}

/* Responsivo */
@media (max-width: 768px) {
  .layout {
    grid-template-columns: 1fr; /* Una columna */
  }
}

```

Ejemplo: Galería con Grid

```

/* HTML:
<div class="galeria">

```

```






</div>
*/

.galeria {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 20px;
  padding: 20px;
}

.galeria img {
  width: 100%;
  height: 250px;
  object-fit: cover;
  border-radius: 10px;
  cursor: pointer;
  transition: 0.3s;
}

.galeria img:hover {
  transform: scale(1.05);
  box-shadow: 0 5px 20px rgba(0, 0, 0, 0.2);
}
```

Grid vs Flexbox: ¿Cuándo usar qué?

Característica	Flexbox	Grid
Dimensiones	1D (fila o columna)	2D (filas y columnas)
Mejor para	Barras de navegación, botones	Layouts de página, galerías
Alineación	Elementos en una dirección	Elementos en dos direcciones
Complejidad	Menos complejo	Más poderoso

Tips de Expertos

1. Mobile-First (Comienza desde Móvil)

```

/* Primero: estilos para móvil */
body {
    font-size: 14px;
}

.contenedor {
    display: flex;
    flex-direction: column;
}

/* Luego: enhancements para pantallas mayores */
@media (min-width: 768px) {
    body {
        font-size: 16px;
    }

    .contenedor {
        flex-direction: row;
    }
}

@media (min-width: 1024px) {
    body {
        font-size: 18px;
    }
}

```

2. CSS Reset Profesional

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

html {
    scroll-behavior: smooth;
}

body {
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial;
    line-height: 1.6;
    color: #333;
}

img, picture, video, canvas, svg {
    display: block;
    max-width: 100%;
}

```

```
input, button, textarea, select {
    font: inherit;
}

p, h1, h2, h3, h4, h5, h6 {
    overflow-wrap: break-word;
}
```

3. Variables CSS (CSS Custom Properties)

```
:root {
    /* Colores */
    --color-primario: #007bff;
    --color-secundario: #6c757d;
    --color-peligro: #dc3545;
    --color-fondo: #f8f9fa;

    /* Fuentes */
    --fuente-base: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    --fuente-titulos: 'Playfair Display', serif;

    /* Espaciado */
    --espaciado-xs: 5px;
    --espaciado-sm: 10px;
    --espaciado-md: 20px;
    --espaciado-lg: 40px;

    /* Sombras */
    --sombra-sm: 0 1px 3px rgba(0, 0, 0, 0.1);
    --sombra-md: 0 4px 6px rgba(0, 0, 0, 0.1);
}

body {
    font-family: var(--fuente-base);
    background-color: var(--color-fondo);
}

.boton-primario {
    background-color: var(--color-primario);
    padding: var(--espaciado-md);
    box-shadow: var(--sombra-md);
}

/* Usar variable en media query */
@media (max-width: 768px) {
    :root {
        --espaciado-md: 15px;
    }
}
```

4. Naming Convention (BEM)

```
/* BEM = Block Element Modifier */

/* Block: componente independiente */
.tarjeta { }

/* Element: parte del block */
.tarjeta__titulo { }
.tarjeta__contenido { }
.tarjeta__imagen { }

/* Modifier: variante del block */
.tarjeta--destacada { }
.tarjeta--pequeña { }
.tarjeta__titulo--importante { }
```

Ejemplo:

```
<div class="tarjeta tarjeta--destacada">
  <h2 class="tarjeta__titulo">Título</h2>
  
  <p class="tarjeta__contenido">Contenido</p>
</div>
```

5. Especificidad CSS (Evita !important)

```
/* ❌ NO HACER */
.importante {
  color: red !important; /* Difícil de sobrescribir */
}

/* ✅ HACER - Usar selectores más específicos */
.seccion .tarjeta .titulo {
  color: red;
}

/* Especificidad: (0, 3, 0) vs (0, 0, 1) de una clase */
```

Guía de Especificidad:

- Element: `p` → (0, 0, 1)
- Clase: `.titulo` → (0, 1, 0)
- ID: `#header` → (1, 0, 0)
- Inline: `style="color: red"` → (1, 0, 0, 0)

6. Pseudo-clases Útiles

```
/* Validación de formularios */
input:valid {
    border-color: green;
}

input:invalid {
    border-color: red;
}

/* Elemento deshabilitado */
button:disabled {
    opacity: 0.5;
    cursor: not-allowed;
}

/* Elemento no disponible */
.contenido:not(.activo) {
    display: none;
}

/* Primer/último de su tipo */
p:first-of-type {
    margin-top: 0;
}

p:last-of-type {
    margin-bottom: 0;
}

/* Solo hijo */
li:only-child {
    list-style: none;
}
```

7. Optimización de Rendimiento

```
/* ❌ Evitar cambios frecuentes que fuerzan redibujado */
.item {
    width: calc(100% - 10px); /* Muy caro */
}

/* ✅ Usar transform en su lugar */
.item {
    transform: translateX(-10px); /* Mejor rendimiento */
}

/* ❌ Evitar cambios frecuentes de tamaño en animations */
@keyframes crecer {
```



```

    0% { width: 0; }
    100% { width: 100%; }
}

/* ✅ Usar transform */
@keyframes crecer {
    0% { transform: scaleX(0); }
    100% { transform: scaleX(1); }
}

/* ✅ Usar will-change para animaciones complejas */
.animado {
    will-change: transform;
}

```

8. Accesibilidad (a11y)

```

/* Asegurar suficiente contraste */
body {
    background-color: white;
    color: #333; /* Contraste de al menos 4.5:1 */
}

/* Foco visible para navegación con teclado */
button:focus {
    outline: 3px solid #0066ff;
    outline-offset: 2px;
}

/* No esconder foco */
*:focus {
    outline: 2px solid #007bff;
}

/* Texto legible */
p {
    font-size: 16px; /* Mínimo 14px */
    line-height: 1.5;
    letter-spacing: 0.02em;
}

/* Evitar texto solo en color */
.error {
    color: red;
    padding-left: 20px;
}

.error::before {
    content: "⚠ ";
}

```

9. Compatibilidad con Navegadores

```
/* Usar prefijos para propiedades nuevas */
.gradient {
  background: -webkit-gradient(linear, left top, right top, from(red), to(blue)); /* Safari
  background: -moz-linear-gradient(to right, red, blue); /* Firef
  background: linear-gradient(to right, red, blue); /* Están
}

/* Verificar soporte */
@supports (display: grid) {
  .contenedor {
    display: grid;
  }
}

@supports not (display: grid) {
  .contenedor {
    display: flex; /* Fallback */
  }
}
```



10. Debugging en DevTools

```
/* Tip: Usa esto temporalmente para ver box models */
* {
  outline: 1px solid red;
}

/* Ver estructura de grid */
.contenedor {
  display: grid;
}

/* En DevTools (Inspector) → haz click en la grilla para ver líneas */

/* Para debugging de flexbox, igual */
.contenedor {
  display: flex;
  outline: 1px solid blue;
}
```

11. Organización de Archivos CSS Profesional

```
proyecto/
├── css/
│   ├── variables.css /* Variables y temas */
```

```

|   |— reset.css           /* Reset/Normalize */
|   |— typography.css      /* Tipografía */
|   |— components/
|   |   |— botones.css
|   |   |— tarjetas.css
|   |   |— formularios.css
|   |   |— navbar.css
|   |— layout/
|   |   |— grid.css
|   |   |— flexbox.css
|   |   |— responsive.css
|   |— main.css            /* Archivo principal (importa todo) */
|— index.html
|— ...

```

main.css:

```

@import url('variables.css');
@import url('reset.css');
@import url('typography.css');
@import url('components/botones.css');
@import url('components/tarjetas.css');
@import url('layout/grid.css');
@import url('layout/responsive.css');

```

12. Tema Oscuro Fácil

```

:root {
  --bg-color: white;
  --text-color: #333;
  --border-color: #ddd;
}

@media (prefers-color-scheme: dark) {
  :root {
    --bg-color: #1a1a1a;
    --text-color: #fff;
    --border-color: #333;
  }
}

body {
  background-color: var(--bg-color);
  color: var(--text-color);
}

input {

```

```
border: 1px solid var(--border-color);
}
```

13. Responsive Units (Unidades Responsivas)

```
/* ❌ NO RECOMENDADO */
```

```
p { font-size: 16px; }
```

```
/* ✅ RECOMENDADO */
```

```
p { font-size: 1rem; } /* Relativo a HTML (16px por defecto) */
```

```
h1 { font-size: 2rem; } /* 32px */
```

```
h2 { font-size: 1.5rem; } /* 24px */
```

```
/* Em - Relativo al padre */
```

```
.contenedor { font-size: 14px; }
```

```
.contenedor p { font-size: 1.2em; } /* 16.8px */
```

```
/* Viewport Units */
```

```
.hero { height: 100vh; } /* 100% altura de viewport */
```

```
.ancho { width: 100vw; } /* 100% ancho de viewport */
```



Resumen Rápido

Concepto	Uso
Flexbox	Layouts 1D, alineación, componentes
Grid	Layouts 2D, páginas complejas, galerías
Variables CSS	Reutilizar valores, temas
Media Queries	Diseño responsivo
Transform	Animaciones eficientes
Transitions	Cambios suaves
Keyframes	Animaciones personalizadas