



Manual de Introducción al CSS

Diseño Web

Dale estilo y vida a tus páginas web

Índice de Contenidos

1. ¿Qué es CSS?	3
2. Cómo enlazar CSS con HTML	4
3. Selectores	5
4. Colores y fondos	7
5. Tipografía	9
6. El modelo de caja (Box Model)	11
7. Posicionamiento y Display	13
8. Flexbox	15
9. Diseño responsive y Media Queries	17
10. Proyecto final integrador	19

1. ¿Qué es CSS?

CSS (Cascading Style Sheets, u Hojas de Estilo en Cascada) es el lenguaje que controla el **aspecto visual** de una página web. Si HTML es el esqueleto, CSS es la piel, la ropa y el maquillaje.

Con CSS puedes controlar:

- Colores, fuentes y tamaños de texto
- Márgenes, rellenos y bordes
- Posición y disposición de los elementos
- Animaciones y transiciones
- Cómo se adapta la web a distintos tamaños de pantalla

¿Cómo funciona la "cascada"?

El nombre «en cascada» hace referencia a cómo CSS resuelve los conflictos: cuando varias reglas afectan al mismo elemento, gana la más **específica** o la que aparece **más abajo** en el archivo.

```
/* Esta regla pone todos los párrafos en azul */
p {
  color: blue;
}

/* Esta es más específica y gana → el párrafo #especial será rojo */
#especial {
  color: red;
}
```

Sintaxis básica de CSS

Una regla CSS tiene dos partes: el selector (a quién afecta) y la declaración (qué hace):

```
selector {
  propiedad: valor;
  propiedad: valor;
}

/* Ejemplo real */
h1 {
  color: navy;
  font-size: 32px;
  text-align: center;
}
```

💡 **¡Recuerda!** Los comentarios en CSS se escriben así: `/* Esto es un comentario */` y son ignorados por el navegador.

2. Cómo Enlazar CSS con HTML

Hay tres formas de añadir CSS a una página HTML:

2.1 Hoja de estilos externa (recomendada)

Se crea un archivo .css separado y se enlaza desde el <head> del HTML. Es el método profesional:

```
<!-- En el archivo index.html -->
<head>
  <link rel="stylesheet" href="css/estilos.css">
</head>
```

```
/* En el archivo css/estilos.css */
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}
```

Ventaja: Un solo archivo CSS puede dar estilo a todas las páginas del sitio.

2.2 Estilos en el <head> (estilo interno)

```
<head>
  <style>
    h1 { color: darkblue; }
    p { font-size: 16px; }
  </style>
</head>
```

2.3 Estilo en línea (inline) — evitar

```
<p style="color: red; font-size: 18px;">Este párrafo es rojo</p>
```

Desventaja: Dificulta el mantenimiento. Solo úsalo en casos muy puntuales.

 **Consejo PRO:** Usa siempre la hoja de estilos externa. Así, si quieres cambiar el color de todos los títulos, solo editas un archivo en vez de tocar cada página.

3. Selectores

Los selectores son la forma de apuntar a los elementos HTML que quieres estilizar. Es una de las partes más importantes de CSS.

3.1 Selector de etiqueta (tipo)

Afecta a todos los elementos de ese tipo:

```
p { color: #333; }
h2 { font-size: 24px; }
```

3.2 Selector de clase

Afecta a los elementos con ese atributo `class`. Se escribe con punto `.`:

```
/* HTML: <p class="destacado">Texto</p> */
.destacado {
  background-color: yellow;
  font-weight: bold;
}
```

💡 **¡Recuerda!** Una clase puede usarse en muchos elementos y un elemento puede tener varias clases: `<p class="destacado grande">`

3.3 Selector de ID

Afecta a un único elemento. Se escribe con `#`:

```
/* HTML: <header id="cabecera-principal"> */
#cabecera-principal {
  background-color: navy;
  color: white;
}
```

3.4 Selector universal

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

3.5 Selectores combinados

```
nav a      { color: white; }      /* <a> dentro de <nav> */
ul > li    { list-style: square; } /* <li> hijo directo de <ul> */
h1, h2, h3 { font-family: serif; } /* Varios a la vez */
p.intro    { font-size: 18px; }   /* <p> con clase intro */
```

3.6 Pseudoclases

Las pseudoclases seleccionan elementos en un estado especial:

```
a:hover      { color: orange; }      /* Al pasar el ratón */
a:visited    { color: purple; }      /* Enlace visitado */
li:first-child { font-weight: bold; } /* Primer hijo */
li:last-child { border-bottom: none; } /* Último hijo */
input:focus  { border-color: blue; } /* Campo enfocado */
```

3.7 Tabla resumen de selectores

Selector	Ejemplo	Selecciona
Etiqueta	p	Todos los <p>
Clase	.menu	Elementos con class="menu"
ID	#logo	El elemento con id="logo"
Universal	*	Todos los elementos
Descendiente	nav a	Los <a> dentro de <nav>
Hijo directo	ul > li	Los hijos de
Pseudoclase	a:hover	El <a> al pasar el ratón

4. Colores y Fondos

4.1 Formas de escribir colores en CSS

```
color: red;                /* Nombre del color */
color: #FF5733;            /* Hexadecimal (la más usada) */
color: #333;               /* Hex corto: equivale a #333333 */
color: rgb(255, 87, 51);   /* RGB */
color: rgba(255, 87, 51, 0.5); /* RGB con transparencia (0-1) */
color: hsl(11, 100%, 60%); /* Tono, Saturación, Luminosidad */
```

4.2 Color de texto y fondo

```
body {
  color: #333333;          /* Color del texto */
  background-color: #f9f9f9; /* Color de fondo */
}
h1 {
  color: white;
  background-color: #2C3E50;
}
```

4.3 Imágenes y gradientes de fondo

```
body {
  background-image: url("img/fondo.jpg");
  background-size: cover;          /* Cubre todo el área */
  background-position: center;     /* Centrada */
  background-repeat: no-repeat;    /* Sin repetición */
}

/* Gradiente lineal */
header {
  background: linear-gradient(135deg, #667eea, #764ba2);
}

/* Gradiente radial */
.hero {
  background: radial-gradient(circle, #f093fb, #f5576c);
}
```

4.4 Opacidad

```
.foto-overlay {
  opacity: 0.7; /* 0 = transparente, 1 = sólido */
}
```

💡 **Herramienta útil:** Usa colors.co para generar paletas de colores armoniosas para tu web. ¡Es gratis!

5. Tipografía

5.1 Familia tipográfica (font-family)

```
body {  
  font-family: Arial, Helvetica, sans-serif;  
  /* Si no hay Arial → Helvetica → cualquier sans-serif */  
}  
  
code {  
  font-family: "Courier New", Courier, monospace;  
}
```

5.2 Google Fonts (fuentes gratuitas)

```
<!-- En el <head> del HTML -->  
<link  
  href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"  
  rel="stylesheet">
```

```
/* En el CSS */  
body { font-family: 'Roboto', sans-serif; }
```

5.3 Propiedades de texto principales

```
p {  
  font-size: 16px;           /* Tamaño */  
  font-weight: bold;         /* Grosor: normal, bold, 100-900 */  
  font-style: italic;        /* Estilo: normal, italic */  
  line-height: 1.6;          /* Altura de línea */  
  letter-spacing: 1px;        /* Espacio entre letras */  
  text-align: center;         /* left, right, center, justify */  
  text-decoration: none;      /* none, underline, line-through */  
  text-transform: uppercase; /* uppercase, lowercase, capitalize */  
  text-shadow: 2px 2px 4px rgba(0,0,0,0.3);  
}
```

5.4 Unidades de medida en CSS

Unidad	Descripción
px	Píxeles fijos. Absoluta e independiente del contexto
em	Relativa al font-size del elemento padre
rem	Relativa al font-size de la raíz <html>

%	Porcentaje del elemento contenedor
vw	1% del ancho de la ventana del navegador
vh	1% del alto de la ventana del navegador

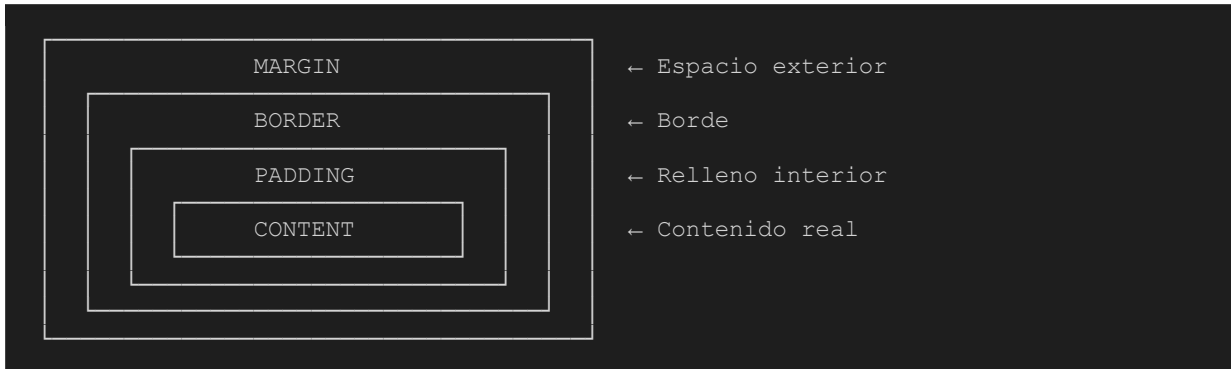
```
html { font-size: 16px; } /* 1rem = 16px */
h1   { font-size: 2rem; } /* = 32px */
h2   { font-size: 1.5rem; }/* = 24px */

.hero {
  height: 100vh; /* Toda la altura de la pantalla */
}
```

 **Consejo de accesibilidad:** Usa rem en lugar de px para los tamaños de fuente. Así la web respeta la configuración de tamaño de letra del usuario en el navegador.

6. El Modelo de Caja (Box Model)

El **Box Model** es el concepto más importante de **CSS**. Todo elemento HTML es una caja rectangular compuesta por cuatro capas:



6.1 Padding, Border y Margin

```
.caja {
  /* PADDING - espacio entre contenido y borde */
  padding: 20px;           /* Los 4 lados iguales */
  padding: 10px 20px;      /* Arriba/abajo | Izq/der */
  padding: 10px 20px 15px 5px; /* Arriba | Der | Abajo | Izq */
  padding-top: 10px;       /* Solo arriba */

  /* BORDER - el borde de la caja */
  border: 2px solid #333;
  border-radius: 8px;      /* Esquinas redondeadas */
  border-top: 3px dashed red;

  /* MARGIN - espacio exterior entre cajas */
  margin: 20px;
  margin: 0 auto;          /* Centrar horizontalmente */
  margin-bottom: 30px;
}
```

6.2 Width y Height

```
.contenedor {
  width: 800px;
  max-width: 1200px; /* Ancho máximo (útil para responsive) */
  min-width: 300px;
  height: 400px;
  min-height: 100vh;
}
```

6.3 box-sizing: border-box (¡muy importante!)

Por defecto, `width` solo mide el contenido. Con `border-box` el ancho incluye también el padding y el borde:

```
/* Pon esto SIEMPRE al inicio de tu CSS */
* {
  box-sizing: border-box;
}

/* Ahora width: 300px = 300px reales, siempre ✅ */
.caja {
  width: 300px;
  padding: 20px;
  border: 2px solid black;
  /* Sin border-box: 300 + 40 + 4 = 344px real ❌ */
  /* Con border-box: siempre 300px ✅ */
}
```

6.4 Overflow

```
.caja {
  overflow: hidden; /* Oculta lo que sobra */
  overflow: scroll; /* Barra de desplazamiento siempre */
  overflow: auto; /* Barra solo si hace falta (recomendado) */
}
```

7. Posicionamiento y Display

7.1 La propiedad display

```
display: block;      /* Ocupa todo el ancho, genera salto de línea */
display: inline;     /* Solo ocupa lo necesario, sin salto */
display: inline-block; /* Como inline pero acepta width y height */
display: none;       /* Oculta el elemento (no ocupa espacio) */
display: flex;       /* Activa Flexbox (ver sección 8) */
```

7.2 La propiedad position

```
/* static: posición normal en el flujo (por defecto) */
p { position: static; }

/* relative: se desplaza respecto a su posición normal */
.desplazado {
  position: relative;
  top: 10px;
  left: 20px;
}

/* absolute: se posiciona respecto al antepasado posicionado */
.etiqueta {
  position: absolute;
  top: 0;
  right: 0;
}

/* fixed: fijo en la pantalla aunque se haga scroll */
.navbar {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  z-index: 100;
}

/* sticky: se pega al llegar al límite al hacer scroll */
.cabecera-tabla {
  position: sticky;
  top: 0;
}
```

7.3 z-index — qué está encima

```
/* Mayor número = más arriba visualmente */
.fondo { z-index: 1; }
.contenido { z-index: 2; }
.modal { z-index: 100; }
```

8. Flexbox

Flexbox es el sistema de maquetación más útil para organizar elementos en una fila o columna. Con unas pocas propiedades consigues diseños que antes requerían trucos complicados.

8.1 Activar Flexbox

Se activa en el **contenedor padre** con `display: flex`:

```
.contenedor {
  display: flex;
}
/* Todos los hijos directos se colocan en fila automáticamente */
```

8.2 Dirección y ajuste

```
.contenedor {
  flex-direction: row;           /* fila (por defecto) */
  flex-direction: column;       /* columna */
  flex-direction: row-reverse;  /* fila al revés */
  flex-wrap: nowrap;           /* Todo en una línea (por defecto) */
  flex-wrap: wrap;             /* Salta si no cabe */
}
```

8.3 Alineación — el superpoder de Flexbox

```
.contenedor {
  /* Eje principal (horizontal con flex-direction: row) */
  justify-content: flex-start; /* Al inicio */
  justify-content: flex-end;   /* Al final */
  justify-content: center;     /* Centrados */
  justify-content: space-between; /* Espacio entre ellos */
  justify-content: space-around; /* Espacio alrededor */
  justify-content: space-evenly; /* Espacio igual */

  /* Eje secundario (vertical con flex-direction: row) */
  align-items: stretch;       /* Se estiran (por defecto) */
  align-items: flex-start;     /* Arriba */
  align-items: flex-end;       /* Abajo */
  align-items: center;         /* Centrados verticalmente */
}
```

8.4 El truco del centrado perfecto

Centrar algo perfectamente (horizontal y vertical) con Flexbox es muy sencillo:

```
.centrar-todo {
  display: flex;
  justify-content: center; /* Centro horizontal */
  align-items: center;     /* Centro vertical */
}
```

```
height: 100vh;
}
```

8.5 Propiedades de los hijos (flex items)

```
.hijo {
  flex: 1;           /* Ocupa el espacio disponible */
  flex-grow: 2;      /* Crece el doble que sus hermanos */
  flex-shrink: 0;     /* No se encoge */
  flex-basis: 200px;  /* Tamaño base */
  align-self: center; /* Alineación individual */
  order: -1;          /* Cambia el orden visual */
}
```

8.6 Ejemplo práctico: Menú de navegación

```
nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #2C3E50;
  padding: 0 20px;
}
nav ul {
  display: flex;
  list-style: none;
  gap: 20px;
}
nav ul li a {
  color: white;
  text-decoration: none;
  padding: 20px 10px;
  display: block;
}
nav ul li a:hover {
  background-color: #E74C3C;
}
```

💡 **gap:** La propiedad gap en Flexbox añade espacio entre los hijos sin necesidad de márgenes. ¡Es mucho más cómodo que usar margin en cada hijo!

9. Diseño Responsive y Media Queries

El diseño responsive hace que una web se vea bien en cualquier dispositivo: móvil, tablet, ordenador...

9.1 El meta viewport (obligatorio)

Sin esta línea en el <head>, el diseño responsive no funciona en móviles:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

9.2 ¿Qué son las Media Queries?

Las Media Queries aplican estilos solo cuando se cumplen ciertas condiciones:

```
@media (condición) {  
  /* estilos que solo se aplican si se cumple la condición */  
}
```

9.3 Breakpoints más comunes

```
/* Móvil: hasta 768px */  
@media (max-width: 768px) {  
  .columnas { flex-direction: column; }  
  nav ul    { display: none; }  
}  
  
/* Tablet: entre 768px y 1024px */  
@media (min-width: 768px) and (max-width: 1024px) {  
  .tarjeta { flex: 1 1 calc(50% - 10px); }  
}  
  
/* Escritorio: más de 1024px */  
@media (min-width: 1024px) {  
  .contenedor {  
    max-width: 1200px;  
    margin: 0 auto;  
  }  
}
```

9.4 Estrategia Mobile First

La mejor práctica es diseñar primero para móvil y luego ampliar para pantallas mayores:

```
/* Base: móvil */  
.tarjetas {  
  display: flex;
```

```
    flex-direction: column;
    gap: 16px;
  }

  /* Tablet */
  @media (min-width: 768px) {
    .tarjeta { flex: 1 1 calc(50% - 8px); } /* 2 columnas */
  }

  /* Escritorio */
  @media (min-width: 1024px) {
    .tarjeta { flex: 1 1 calc(33% - 11px); } /* 3 columnas */
  }
```

9.5 Imágenes responsive

```
img {
  max-width: 100%;
  height: auto;
  display: block;
}
```

💡 **¡Actividad práctica!** Redimensiona la ventana de tu navegador mientras ves una web. Observa cómo el diseño cambia al hacerla más estrecha. ¡Eso es el diseño responsive en acción!

10. Proyecto Final Integrador

Llega el momento de estilizar la web personal que creaste con HTML. Vas a transformarla completamente con CSS.

10.1 Estructura de archivos

```
mi-web-personal/
├── index.html
├── sobre-mi.html
├── hobbies.html
├── contacto.html
├── css/
│   └── estilos.css      ← Todo tu CSS aquí
├── img/
│   └── ...
```

10.2 Plantilla de estilos base

```
/* === RESET Y BASE === */
* { box-sizing: border-box; margin: 0; padding: 0; }
body {
  font-family: 'Segoe UI', Arial, sans-serif;
  font-size: 16px; line-height: 1.6; color: #333;
}
img { max-width: 100%; height: auto; }
a { color: #E8500A; text-decoration: none; }
a:hover { text-decoration: underline; }

/* === CONTENEDOR === */
.contenedor { max-width: 1100px; margin: 0 auto; padding: 0 20px; }

/* === CABECERA === */
header {
  background: linear-gradient(135deg, #1A5276, #2E86C1);
  color: white; padding: 60px 20px; text-align: center;
}
header h1 { font-size: 2.5rem; }

/* === NAVEGACIÓN (FLEXBOX) === */
nav { background: #2C3E50; position: sticky; top: 0; z-index: 100; }
nav ul { display: flex; list-style: none; justify-content: center; }
nav ul li a { display: block; color: white; padding: 16px 24px; }
nav ul li a:hover { background-color: #E8500A; }

/* === TARJETAS (FLEXBOX) === */
.tarjetas { display: flex; flex-wrap: wrap; gap: 20px; }
.tarjeta {
  flex: 1 1 280px; background: white;
  border-radius: 8px; padding: 24px;
  box-shadow: 0 2px 8px rgba(0,0,0,0.1);
  transition: transform 0.2s;
}
.tarjeta:hover { transform: translateY(-4px); }

/* === FORMULARIO === */
input, textarea, select {
  width: 100%; padding: 10px 14px;
```

```
border: 2px solid #ddd; border-radius: 6px;
font-size: 1rem; margin-bottom: 20px;
}
input:focus, textarea:focus { border-color: #2E86C1; outline: none; }
button[type="submit"] {
  background: #E8500A; color: white; padding: 12px 32px;
  border: none; border-radius: 6px; cursor: pointer;
}

/* === RESPONSIVE === */
@media (max-width: 768px) {
  header h1 { font-size: 1.8rem; }
  nav ul { flex-direction: column; }
  .tarjetas { flex-direction: column; }
}
```

10.3 Requisitos del proyecto

Estilos generales

- Reset con box-sizing: border-box
- Tipografía coherente con font-family y tamaños con rem
- Paleta de colores definida y consistente en toda la web

Navegación

- Menú estilizado con Flexbox
- Efecto hover visible en los enlaces

Contenido

- Cabecera con fondo de color o gradiente
- Al menos una sección con tarjetas usando Flexbox
- Imágenes con max-width: 100%

Formulario

- Campos estilizados con foco visible (:focus)
- Botón con efecto hover

Responsive

- La web debe verse bien en móvil (pantalla < 768px)
- Al menos una Media Query que cambie el layout

10.4 Criterios de evaluación

Criterio	Puntuación
Reset y estilos base correctos	1 punto
Selectores variados (etiqueta, clase, ID, pseudoclase)	2 puntos
Colores y tipografía coherentes	1.5 puntos
Box model bien aplicado (padding, margin, border)	1.5 puntos
Flexbox para maquetación	2 puntos
Al menos una Media Query funcional	1.5 puntos
Creatividad y acabado visual	0.5 puntos

¡Enhorabuena! Ya conoces los fundamentos del CSS 🎨

El siguiente paso: CSS Grid y Animaciones CSS 🚀