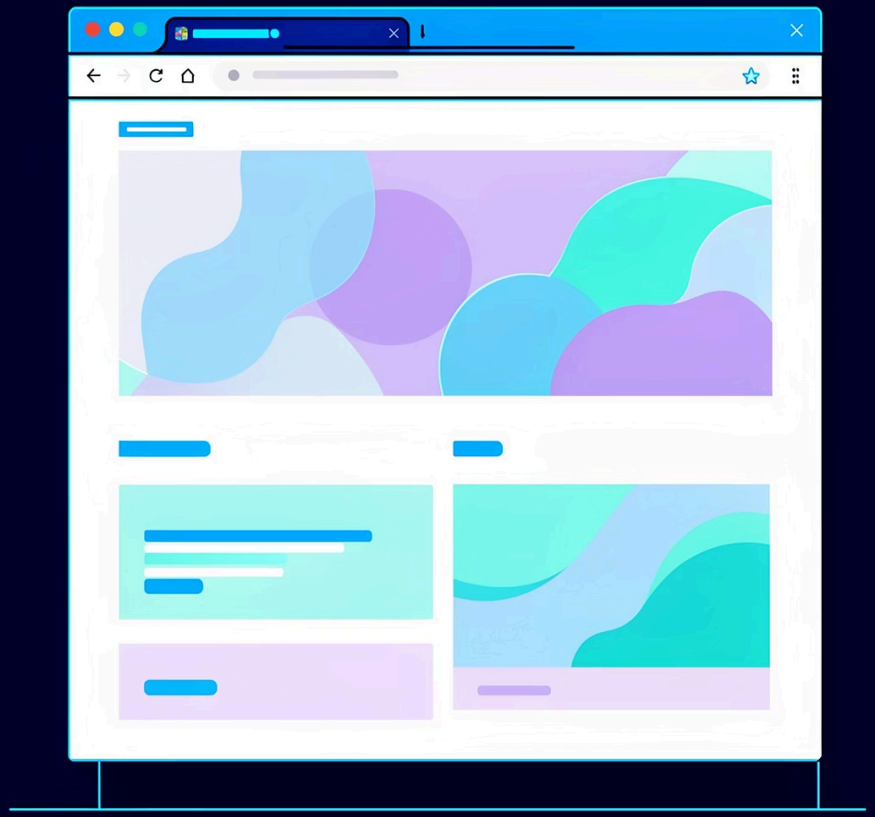


CSS: Cascading Style Sheets

El lenguaje que transforma la estructura HTML en experiencias visuales atractivas y funcionales

¿Qué es CSS?

Hoja de Estilos en Cascada

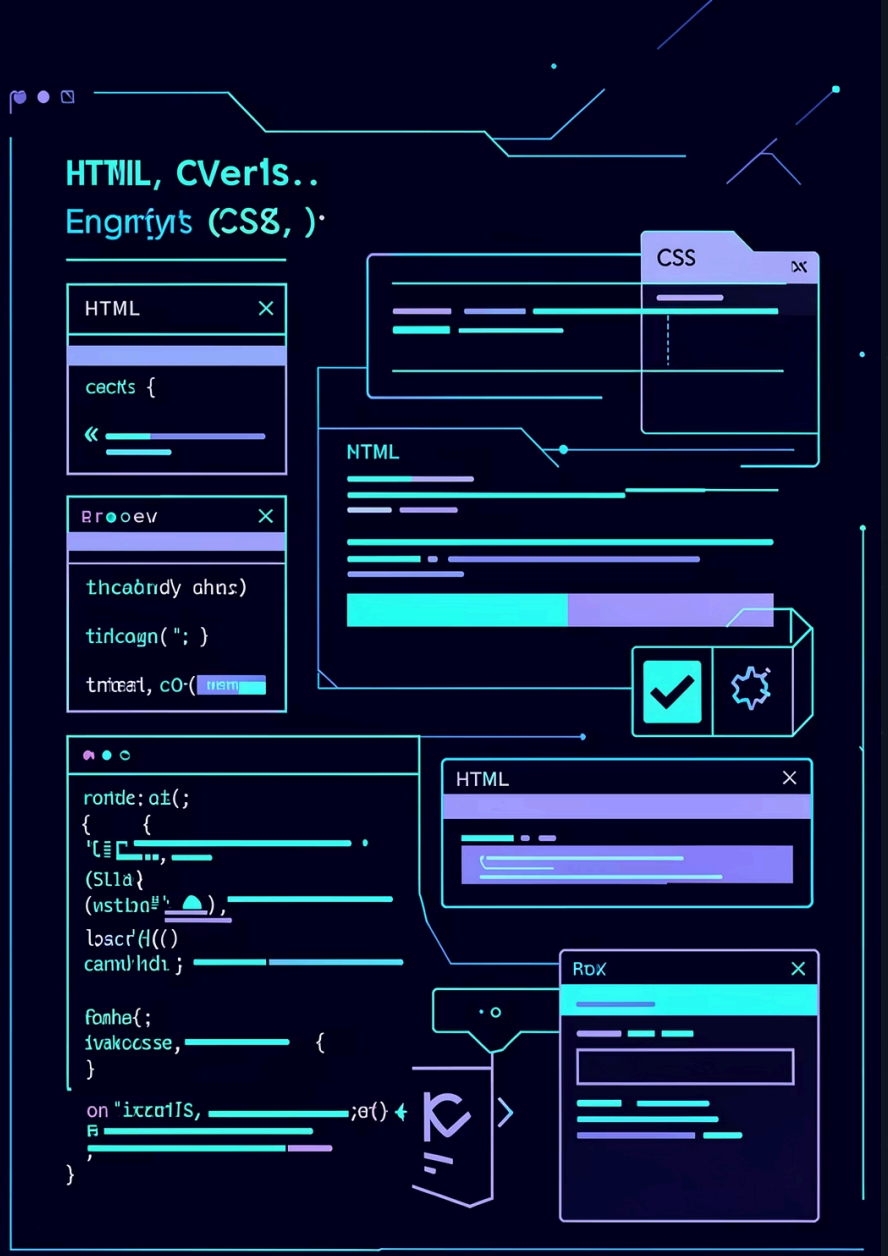
CSS (Cascading Style Sheets) define la presentación visual de documentos HTML y XML

Diseño controlado

Controla colores, fuentes, espaciado, posicionamiento y animaciones para crear interfaces atractivas

Separa forma y contenido

Promueve la separación del contenido estructural (HTML) de su presentación visual



¿Cómo funciona CSS?

HTML + CSS

HTML proporciona la estructura y contenido del documento. CSS aplica estilos a esos elementos estructurales.

El navegador interpreta ambos y renderiza la página final con estilos aplicados.

Conceptos clave

- **Cascada:** Resuelve conflictos de estilos mediante orden de precedencia
- **Herencia:** Los elementos hijo heredan propiedades del padre
- **Especificidad:** Determina qué regla CSS tiene prioridad

Evolución de CSS: del origen al presente

1

1996

Primera especificación oficial de CSS1 publicada por W3C

2

1998

CSS2 introduce posicionamiento absoluto y relativo, z-index

3

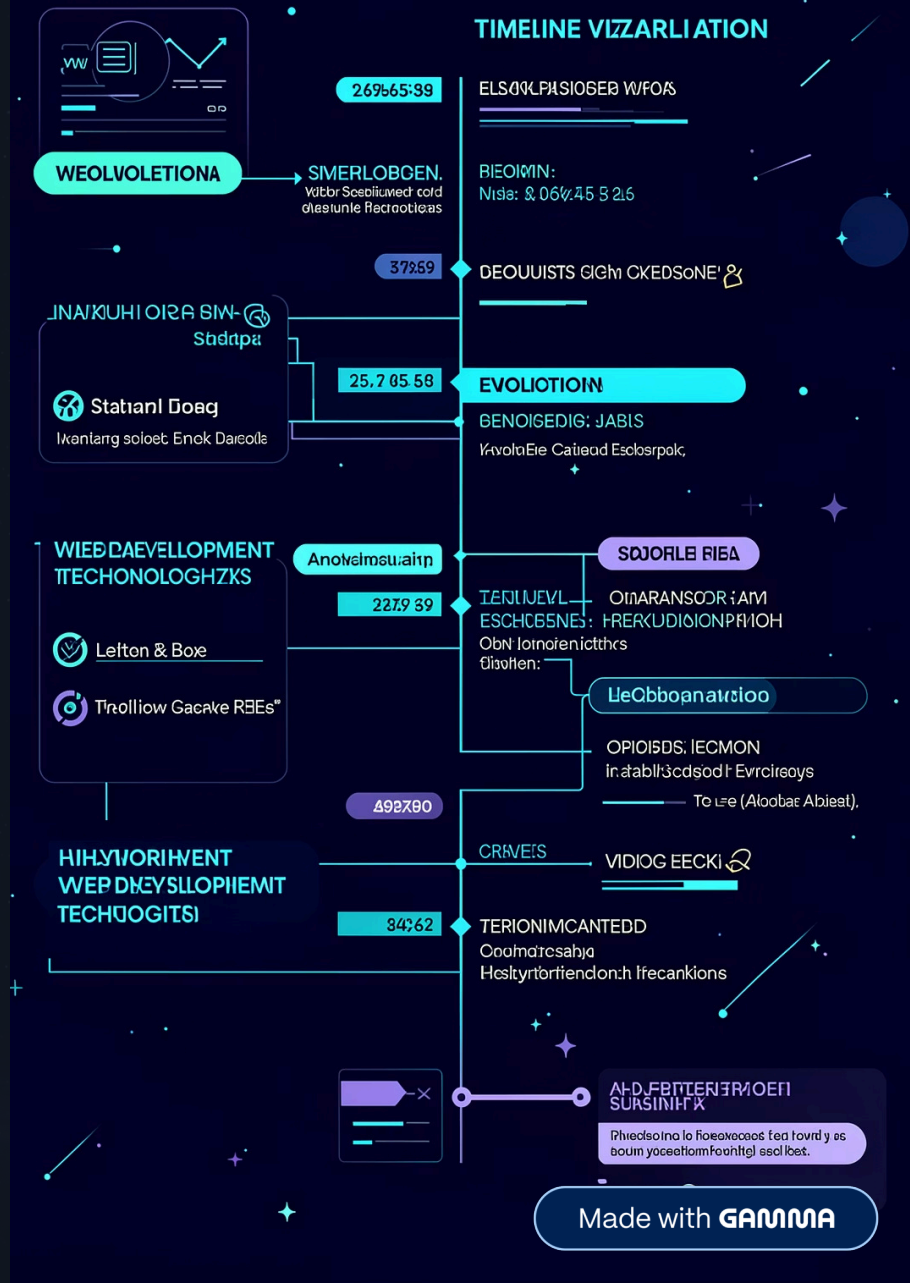
2011

CSS3 se divide en módulos independientes para desarrollo paralelo

4

Hoy

Módulos modernos: Flexbox, Grid, Custom Properties, Animaciones



¿Para qué sirve CSS?

Diseño Visual

Crea interfaces atractivas con colores, tipografía, espaciado y efectos visuales

Diseño Responsive

Adapta la presentación a diferentes tamaños de pantalla y dispositivos

Separación de Concerns

Distancia contenido (HTML) de presentación (CSS), facilitando mantenimiento

Interactividad

Agrega transiciones y animaciones que mejoran experiencia de usuario

Formas de aplicar CSS

1

CSS en línea

Atributo `style` dentro de etiquetas HTML

Rápido pero no recomendado

2

CSS interno

Bloque `<style>` en el `<head>`

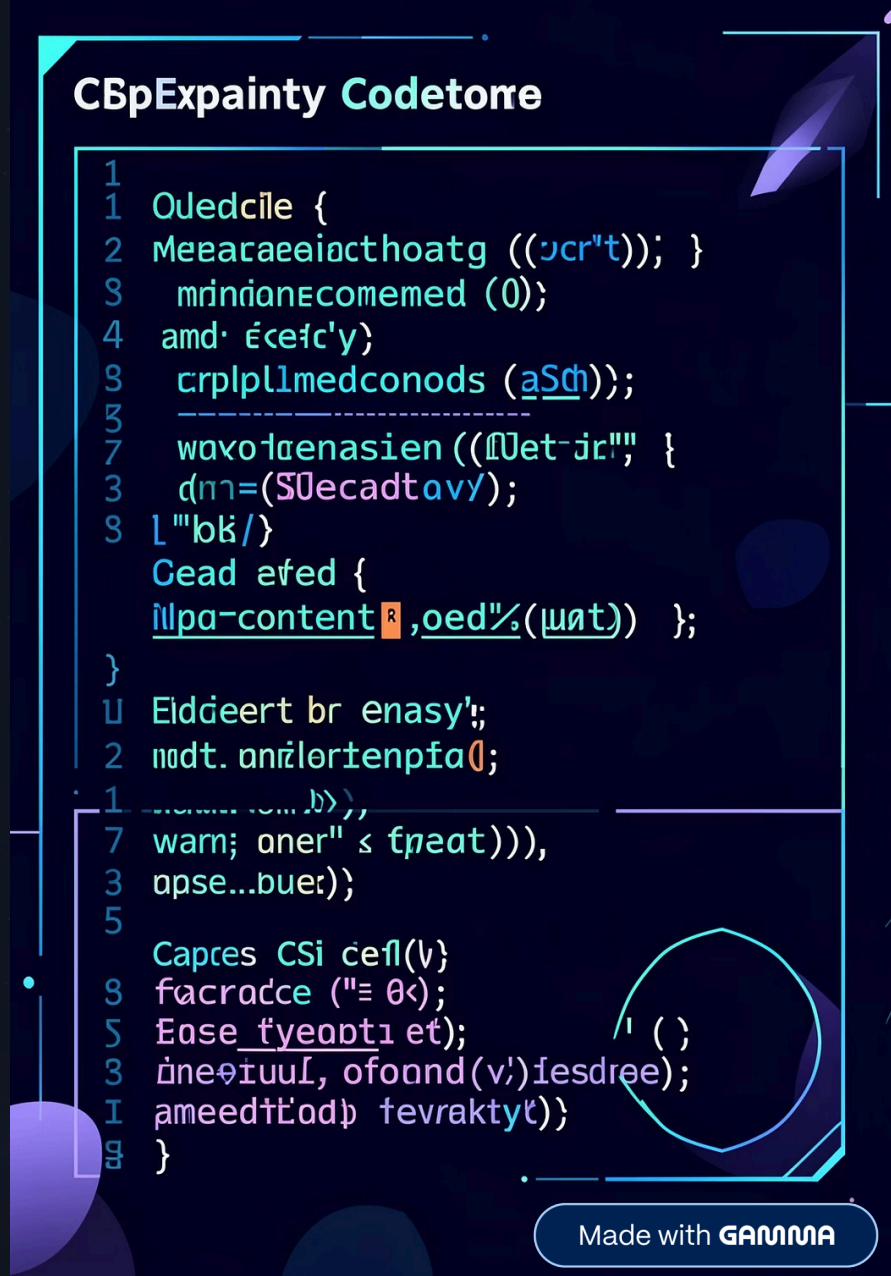
Una página específica

3

CSS externo

Archivo `.css` vinculado con `<link>`

Mejor práctica general



Selectores y propiedades esenciales

Selectores básicos

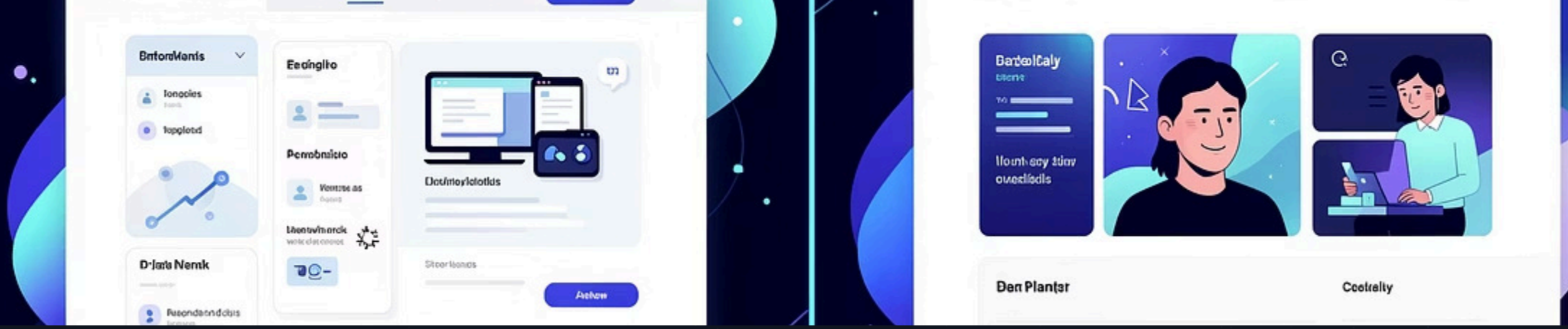
```
/* Por etiqueta */
p { color: blue; }

/* Por clase */
.card { margin: 1rem; }

/* Por ID */
#header { padding: 2rem; }
```

Propiedades comunes

color
background(-color, -image)
font-size, font-family
margin, padding
border(-radius, -color)
display (block, flex, grid)



Ejemplo práctico paso a paso

HTML Base

```
<article class="card">
  <h2>Título</h2>
  <p>Contenido del artículo</p>
</article>
```

CSS Aplicado

```
.card {
  background: #6296ff;
  padding: 1rem;
  border-radius: 8px;
  margin: 1rem 0;
}
```

El resultado es un card estilizado con fondo azul, esquinas redondeadas y espaciado coherente

Buenas prácticas de CSS

Organización del código

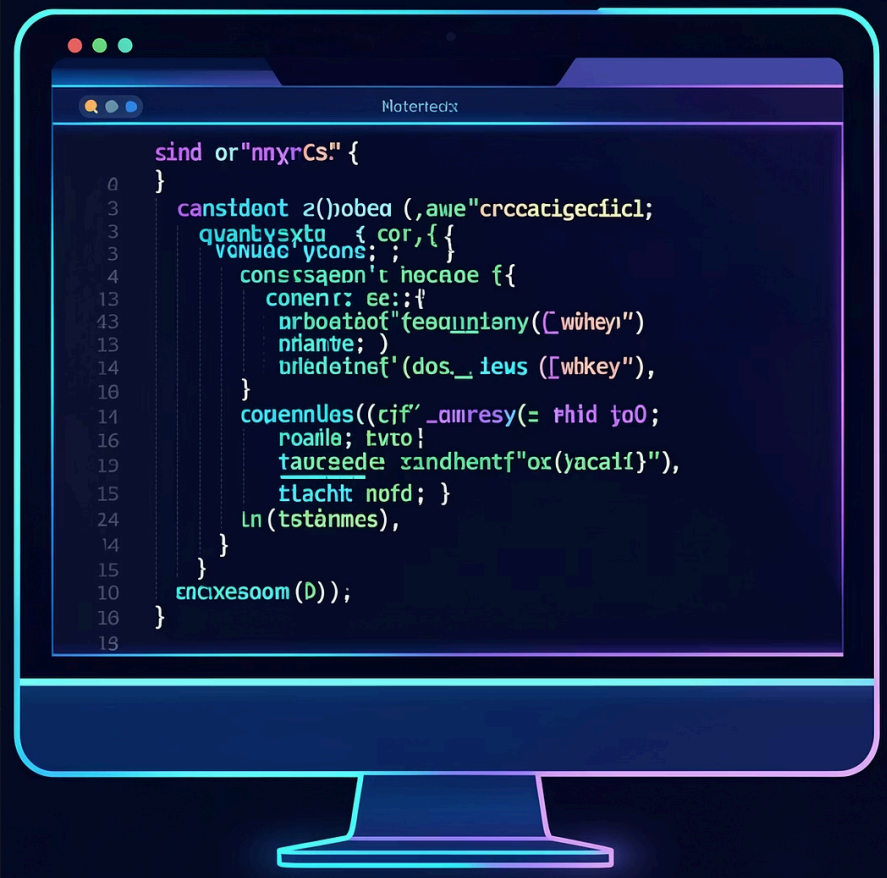
- Usa archivos CSS externos separados
- Comenta secciones importantes
- Mantén consistencia en formato

Selectores semánticos

- Evita exceso de IDs (son específicos)
- Usa clases para reutilizar estilos
- Sigue metodologías como BEM

Arquitectura moderna

- Preprocesadores (Sass, Less)
- Variables CSS para temas
- Modularización con CSS Custom Properties



Errores comunes y cómo evitarlos

!important abusivo

Evita depender de `!important`. Mejora especificidad y estructura de CSS en su lugar.

Estilos en línea

Limita CSS en línea a casos muy específicos (emails, componentes dinámicos). Prioriza archivos externos.

Estilos muy específicos

Evita selectores IDs y exceso de anidamiento. Prioriza clases reutilizables y modularidad.

No usar reset CSS

Implementa Normalize.css o CSS reset básico para consistencia entre navegadores.

Consejo: Usa herramientas como CSSLint o integración con ESLint para detectar problemas automáticamente