

GUÍA COMPLETA DE CSS

Aprende CSS desde cero hasta nivel avanzado

ÍNDICE

1. [¿Qué es CSS?](#)
 2. [Variables CSS \(Custom Properties\)](#)
 3. [Selectores CSS](#)
 4. [Propiedades CSS más importantes](#)
 5. [Layout: Flexbox](#)
 6. [Layout: Grid](#)
 7. [Posicionamiento](#)
 8. [Colores y Fondos](#)
 9. [Tipografía](#)
 10. [Espaciado: Margin y Padding](#)
 11. [Bordes y Sombras](#)
 12. [Transiciones y Animaciones](#)
 13. [Responsive Design](#)
 14. [Mejores Prácticas](#)
 15. [Ejemplos Prácticos](#)
-

¿QUÉ ES CSS?

CSS (Cascading Style Sheets) es el lenguaje que se usa para dar estilo y diseño a las páginas web. Es como el "maquillaje" de HTML.

Funciones principales:

- **Diseño visual:** Colores, fuentes, tamaños
 - **Layout:** Organización de elementos en la página
 - **Responsive:** Adaptación a diferentes dispositivos
 - **Animaciones:** Efectos y transiciones
-

VARIABLES CSS

Las variables CSS (Custom Properties) son como "cajas" donde guardas valores reutilizables.

Sintaxis básica:

```
:root {  
  /* Definir variables */  
  --primary-color: #2563eb;  
  --spacing-md: 1rem;  
}
```

```
--font-family: 'Inter', sans-serif;
}

/* Usar variables */
.button {
  background-color: var(--primary-color);
  padding: var(--spacing-md);
  font-family: var(--font-family);
}
```

Ventajas:

- ☒ **Consistencia:** Cambias un valor y se actualiza en toda la web
- ☒ **Mantenimiento:** Fácil de modificar
- ☒ **Organización:** Todo centralizado

Ejemplo completo:

```
:root {
  /* Paleta de colores */
  --primary-color: #2563eb;
  --secondary-color: #64748b;
  --accent-color: #f59e0b;

  /* Espaciado */
  --spacing-xs: 0.25rem; /* 4px */
  --spacing-sm: 0.5rem; /* 8px */
  --spacing-md: 1rem; /* 16px */
  --spacing-lg: 1.5rem; /* 24px */
  --spacing-xl: 2rem; /* 32px */

  /* Tipografía */
  --font-size-base: 1rem; /* 16px */
  --font-size-lg: 1.125rem; /* 18px */
  --font-size-xl: 1.25rem; /* 20px */

  /* Bordes */
  --border-radius: 8px;
  --border-radius-lg: 12px;

  /* Sombras */
  --shadow-sm: 0 1px 2px 0 rgb(0 0 0 / 0.05);
  --shadow-md: 0 4px 6px -1px rgb(0 0 0 / 0.1);
  --shadow-lg: 0 10px 15px -3px rgb(0 0 0 / 0.1);

  /* Transiciones */
  --transition-fast: 0.15s ease-in-out;
  --transition-normal: 0.3s ease-in-out;
}
```

🔗 SELECTORES CSS

Los selectores son las "direcciones" que le dicen a CSS qué elementos estilizar.

Selectores básicos:

```
/* Por elemento */
h1 { color: blue; }

/* Por clase */
.button { background: red; }

/* Por ID */
#header { height: 60px; }

/* Por atributo */
[type="button"] { border: 1px solid; }

/* Combinación */
button.primary { background: blue; }
```

Selectores avanzados:

```
/* Descendiente */
.container .item { margin: 10px; }

/* Hijo directo */
.container > .item { padding: 10px; }

/* Hermano adyacente */
.item + .item { margin-top: 10px; }

/* Pseudo-clases */
.button:hover { background: darkblue; }
.button:active { transform: scale(0.95); }
.button:focus { outline: 2px solid blue; }

/* Pseudo-elementos */
.text::before { content: "→ "; }
.text::after { content: " ←"; }
```

🔗 PROPIEDADES CSS MÁS IMPORTANTES

1. DISPLAY - Controla el comportamiento del elemento

```
/* Valores principales */
display: block;          /* Elemento ocupa toda la línea */
display: inline;         /* Elemento se comporta como texto */
display: inline-block;   /* Combinación de ambos */
display: flex;           /* Layout flexible */
display: grid;           /* Layout en cuadrícula */
display: none;           /* Oculta el elemento */
display: hidden;         /* Oculta pero mantiene el espacio */
```

Ejemplo práctico:

```
.header {
  display: flex;           /* Los elementos se ponen en línea */
  align-items: center;     /* Los centra verticalmente */
  justify-content: space-between; /* Los distribuye */
}

.card {
  display: block;          /* Cada tarjeta ocupa toda la línea */
  margin-bottom: 1rem;     /* Espacio entre tarjetas */
}
```

2. POSITION - Controla la posición del elemento

```
/* Valores principales */
position: static;         /* Posición normal (por defecto) */
position: relative;       /* Posición relativa a su lugar normal */
position: absolute;       /* Posición absoluta respecto al padre */
position: fixed;          /* Posición fija en la pantalla */
position: sticky;         /* Se pega al hacer scroll */
```

Ejemplo práctico:

```
.header {
  position: sticky;        /* El header se queda fijo al hacer scroll */
  top: 0;                 /* Se pega arriba */
  z-index: 100;           /* Se pone por encima de otros elementos */
}

.modal {
  position: fixed;         /* Modal fijo en la pantalla */
  top: 50%;               /* Centrado vertical */
  left: 50%;              /* Centrado horizontal */
  transform: translate(-50%, -50%); /* Centrado perfecto */
}
```

3. WIDTH Y HEIGHT - Dimensiones

```
/* Valores comunes */
width: 100%;           /* Ancho completo */
width: 50%;            /* Mitad del ancho */
width: 300px;          /* Ancho fijo */
width: auto;           /* Ancho automático */
width: fit-content;    /* Ancho según el contenido */

height: 100vh;         /* Altura completa de la pantalla */
height: 100%;          /* Altura completa del padre */
height: 200px;         /* Altura fija */
height: auto;          /* Altura automática */
```

Ejemplo práctico:

```
.container {
  width: 100%;           /* Ancho completo */
  max-width: 1200px;    /* Ancho máximo */
  margin: 0 auto;       /* Centrado horizontal */
}

.hero {
  height: 100vh;        /* Altura completa de la pantalla */
  display: flex;         /* Para centrar contenido */
  align-items: center;
}
```

LAYOUT: FLEXBOX

Flexbox es perfecto para organizar elementos en línea o columna.

Propiedades del contenedor:

```
.flex-container {
  display: flex;         /* Activa flexbox */
  flex-direction: row;   /* Elementos en línea (por defecto) */
  flex-direction: column; /* Elementos en columna */
  justify-content: center; /* Centra horizontalmente */
  justify-content: space-between; /* Distribuye con espacio */
  justify-content: space-around; /* Distribuye con espacio igual */
  align-items: center;   /* Centra verticalmente */
  align-items: stretch; /* Estira elementos (por defecto) */
  flex-wrap: wrap;       /* Permite saltos de línea */
  gap: 1rem;             /* Espacio entre elementos */
}
```

Propiedades de los elementos hijos:

```
.flex-item {  
  flex: 1; /* Toma espacio disponible */  
  flex: 0 0 200px; /* Ancho fijo de 200px */  
  flex-grow: 1; /* Puede crecer */  
  flex-shrink: 0; /* No puede encogerse */  
  flex-basis: 200px; /* Tamaño base */  
  align-self: center; /* Alineación individual */  
  order: 1; /* Orden de aparición */  
}
```

Ejemplos prácticos:

Navegación horizontal:

```
.nav {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 1rem;  
}  
  
.nav-links {  
  display: flex;  
  gap: 2rem;  
  list-style: none;  
}
```

Centrado perfecto:

```
.centered {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

Botones en línea:

```
.button-group {  
  display: flex;  
  gap: 1rem;  
  justify-content: center;  
}
```

```
.button {  
  padding: 0.5rem 1rem;  
  border: none;  
  border-radius: 4px;  
}
```

LAYOUT: GRID

Grid es perfecto para layouts complejos y bidimensionales.

Propiedades del contenedor:

```
.grid-container {  
  display: grid; /* Activa grid */  
  grid-template-columns: 1fr 1fr; /* Dos columnas del mismo tamaño */  
  grid-template-columns: repeat(3, 1fr); /* Tres columnas iguales */  
  grid-template-columns: 200px 1fr 200px; /* Columnas con tamaños específicos */  
  grid-template-rows: auto; /* Filas automáticas */  
  grid-template-rows: 100px 200px; /* Filas con alturas específicas */  
  gap: 1rem; /* Espacio entre elementos */  
  gap: 1rem 2rem; /* Fila columna */  
  justify-items: center; /* Centra elementos horizontalmente */  
  align-items: center; /* Centra elementos verticalmente */  
}
```

Propiedades de los elementos hijos:

```
.grid-item {  
  grid-column: 1 / 3; /* Ocupa columnas 1 a 3 */  
  grid-row: 1 / 2; /* Ocupa fila 1 */  
  grid-area: header; /* Usa área nombrada */  
  justify-self: center; /* Alineación individual horizontal */  
  align-self: center; /* Alineación individual vertical */  
}
```

Áreas nombradas:

```
.layout {  
  display: grid;  
  grid-template-areas:  
    "header header"  
    "sidebar main"  
    "footer footer";  
  grid-template-columns: 200px 1fr;
```

```
    grid-template-rows: 60px 1fr 60px;
}
```

```
.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
.main { grid-area: main; }
.footer { grid-area: footer; }
```

Ejemplos prácticos:

Layout de portfolio:

```
.hero-content {
  display: grid;
  grid-template-columns: 1fr 1fr; /* Dos columnas iguales */
  gap: 3rem; /* Espacio grande */
  align-items: center; /* Centra verticalmente */
}

.projects-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: 2rem;
}
```

📍 POSICIONAMIENTO

Position Relative:

```
.relative-box {
  position: relative;
  top: 10px; /* Se mueve 10px hacia abajo */
  left: 20px; /* Se mueve 20px hacia la derecha */
  z-index: 1; /* Controla el orden de capas */
}
```

Position Absolute:

```
.parent {
  position: relative; /* El hijo absoluto se posiciona respecto a este */
}

.absolute-box {
  position: absolute;
  top: 0; /* Se pega arriba del padre */
}
```



```
    right: 0;           /* Se pega a la derecha del padre */
}
```

Position Fixed:

```
.fixed-header {
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  background: white;
  z-index: 1000;
}
```

Position Sticky:

```
.sticky-nav {
  position: sticky;
  top: 0;
  background: white;
  z-index: 100;
}
```

COLORES Y FONDOS

Colores:

```
/* Diferentes formatos */
color: #ff0000;           /* Hexadecimal */
color: rgb(255, 0, 0);    /* RGB */
color: rgba(255, 0, 0, 0.5); /* RGB con transparencia */
color: hsl(0, 100%, 50%); /* HSL */
color: hsla(0, 100%, 50%, 0.5); /* HSL con transparencia */
color: red;               /* Nombre del color */
color: var(--primary-color); /* Variable CSS */
```

Fondos:

```
/* Color sólido */
background-color: #f0f0f0;

/* Gradiente lineal */
background: linear-gradient(45deg, #ff6b6b, #4ecdc4);
```

```
/* Gradiente radial */
background: radial-gradient(circle, #ff6b6b, #4ecdc4);

/* Imagen de fondo */
background-image: url('imagen.jpg');
background-size: cover;           /* Cubre todo el contenedor */
background-size: contain;        /* Contiene toda la imagen */
background-position: center;     /* Centra la imagen */
background-repeat: no-repeat;    /* No repite la imagen */

/* Múltiples fondos */
background:
  linear-gradient(rgba(0,0,0,0.5), rgba(0,0,0,0.5)),
  url('imagen.jpg');
```

Ejemplos prácticos:

Gradiente en texto:

```
.gradient-text {
  background: linear-gradient(135deg, #667eea, #764ba2);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-clip: text;
}
```

Fondo con patrón:

```
.pattern-bg {
  background-image:
    linear-gradient(45deg, #f0f0f0 25%, transparent 25%),
    linear-gradient(-45deg, #f0f0f0 25%, transparent 25%),
    linear-gradient(45deg, transparent 75%, #f0f0f0 75%),
    linear-gradient(-45deg, transparent 75%, #f0f0f0 75%);
  background-size: 20px 20px;
  background-position: 0 0, 0 10px, 10px -10px, -10px 0px;
}
```

TIPOGRAFÍA

Propiedades básicas:

```
.text {
  font-family: 'Arial', sans-serif;  /* Familia de fuente */
  font-size: 16px;                  /* Tamaño de fuente */
}
```

```
font-weight: 400;          /* Grosor (100-900) */
font-style: normal;        /* Estilo (normal, italic) */
line-height: 1.5;         /* Altura de línea */
text-align: left;         /* Alineación */
text-decoration: none;     /* Decoración */
text-transform: uppercase; /* Transformación */
letter-spacing: 1px;       /* Espaciado entre letras */
word-spacing: 2px;        /* Espaciado entre palabras */
}
```

Tamaños de fuente:

```
/* Unidades absolutas */
font-size: 16px;    /* Píxeles */
font-size: 12pt;    /* Puntos */

/* Unidades relativas */
font-size: 1rem;    /* Relativo al elemento raíz */
font-size: 1em;     /* Relativo al elemento padre */
font-size: 100%;    /* Porcentaje del padre */

/* Tamaños predefinidos */
font-size: small;   /* Pequeño */
font-size: medium;  /* Mediano */
font-size: large;   /* Grande */
font-size: x-large; /* Extra grande */
```

Ejemplos prácticos:

Jerarquía de títulos:

```
h1 {
  font-size: 2.5rem;
  font-weight: 700;
  line-height: 1.2;
  margin-bottom: 1rem;
}

h2 {
  font-size: 2rem;
  font-weight: 600;
  line-height: 1.3;
  margin-bottom: 0.8rem;
}

h3 {
  font-size: 1.5rem;
  font-weight: 600;
  line-height: 1.4;
}
```

```
margin-bottom: 0.6rem;
}
```

Texto responsivo:

```
.responsive-text {
  font-size: clamp(1rem, 2.5vw, 1.5rem);
  line-height: 1.6;
}
```



ESPACIADO: MARGIN Y PADDING

Margin (espacio exterior):

```
.element {
  margin: 10px;           /* Todos los lados */
  margin: 10px 20px;      /* Arriba/abajo, izquierda/derecha */
  margin: 10px 20px 30px; /* Arriba, izquierda/derecha, abajo */
  margin: 10px 20px 30px 40px; /* Arriba, derecha, abajo, izquierda */

  /* Lados específicos */
  margin-top: 10px;
  margin-right: 20px;
  margin-bottom: 30px;
  margin-left: 40px;

  /* Valores especiales */
  margin: auto;           /* Centrado automático */
  margin: 0;              /* Sin margen */
}
```

Padding (espacio interior):

```
.element {
  padding: 10px;           /* Todos los lados */
  padding: 10px 20px;      /* Arriba/abajo, izquierda/derecha */
  padding: 10px 20px 30px; /* Arriba, izquierda/derecha, abajo */
  padding: 10px 20px 30px 40px; /* Arriba, derecha, abajo, izquierda */

  /* Lados específicos */
  padding-top: 10px;
  padding-right: 20px;
  padding-bottom: 30px;
  padding-left: 40px;
}
```

Box-sizing:

```
.element {  
  box-sizing: content-box; /* Padding y border se suman al width */  
  box-sizing: border-box; /* Padding y border se incluyen en width */  
}  
  
/* Reset global recomendado */  
* {  
  box-sizing: border-box;  
}
```

Ejemplos prácticos:

Card con espaciado:

```
.card {  
  padding: 1.5rem; /* Espacio interno */  
  margin-bottom: 1rem; /* Espacio entre cards */  
  border-radius: 8px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
}
```

Botón con padding:

```
.button {  
  padding: 0.5rem 1rem; /* 8px arriba/abajo, 16px izquierda/derecha */  
  margin: 0.5rem; /* Espacio alrededor */  
  border: none;  
  border-radius: 4px;  
}
```

☐ BORDES Y SOMBRAS

Bordes:

```
.element {  
  /* Borde completo */  
  border: 2px solid #333;  
  
  /* Bordes individuales */  
  border-top: 1px solid #ccc;  
  border-right: 2px solid #999;
```

```
border-bottom: 1px solid #ccc;
border-left: 2px solid #999;

/* Propiedades individuales */
border-width: 2px;
border-style: solid;
border-color: #333;

/* Bordes redondeados */
border-radius: 8px;
border-radius: 50%;           /* Círculo perfecto */
border-radius: 8px 16px;     /* Esquinas diferentes */
}
```

Sombras:

```
.element {
  /* Sombra básica */
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);

  /* Sombra con desplazamiento */
  box-shadow: 2px 4px 8px rgba(0,0,0,0.2);

  /* Múltiples sombras */
  box-shadow:
    0 1px 3px rgba(0,0,0,0.1),
    0 4px 6px rgba(0,0,0,0.1);

  /* Sombra interna */
  box-shadow: inset 0 2px 4px rgba(0,0,0,0.1);

  /* Sombra con color */
  box-shadow: 0 4px 8px rgba(37, 99, 235, 0.3);
}
```

Ejemplos prácticos:

Card con sombra:

```
.card {
  border: 1px solid #e2e8f0;
  border-radius: 8px;
  box-shadow: 0 1px 3px rgba(0,0,0,0.1);
  transition: box-shadow 0.3s ease;
}

.card:hover {
  box-shadow: 0 4px 12px rgba(0,0,0,0.15);
}
```

Botón con borde:

```
.button {  
  border: 2px solid transparent;  
  border-radius: 6px;  
  transition: border-color 0.3s ease;  
}  
  
.button:hover {  
  border-color: #2563eb;  
}
```

⚡ TRANSICIONES Y ANIMACIONES

Transiciones:

```
.element {  
  /* Transición básica */  
  transition: all 0.3s ease;  
  
  /* Transiciones específicas */  
  transition: color 0.3s ease, transform 0.2s ease;  
  
  /* Propiedades individuales */  
  transition-property: color, background-color;  
  transition-duration: 0.3s;  
  transition-timing-function: ease;  
  transition-delay: 0.1s;  
  
  /* Funciones de timing */  
  transition-timing-function: ease;           /* Suave al inicio y final */  
  transition-timing-function: ease-in;        /* Suave al inicio */  
  transition-timing-function: ease-out;       /* Suave al final */  
  transition-timing-function: ease-in-out;    /* Suave al inicio y final */  
  transition-timing-function: linear;         /* Velocidad constante */  
}
```

Animaciones:

```
/* Definir animación */  
@keyframes slideIn {  
  0% {  
    transform: translateX(-100%);  
    opacity: 0;  
  }  
}
```

```
100% {
  transform: translateX(0);
  opacity: 1;
}
}

/* Aplicar animación */
.animated-element {
  animation: slideIn 0.5s ease-out;
  animation-fill-mode: forwards; /* Mantiene el estado final */
  animation-iteration-count: 1; /* Número de repeticiones */
  animation-direction: normal; /* Dirección */
  animation-play-state: running; /* Estado de reproducción */
}

/* Animación infinita */
.pulse {
  animation: pulse 2s infinite;
}

@keyframes pulse {
  0%, 100% { transform: scale(1); }
  50% { transform: scale(1.05); }
}
```

Ejemplos prácticos:

Botón con hover:

```
.button {
  background: #2563eb;
  color: white;
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 4px;
  transition: all 0.3s ease;
}

.button:hover {
  background: #1d4ed8;
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(37, 99, 235, 0.3);
}
```

Loading spinner:

```
.spinner {
  width: 40px;
  height: 40px;
```



```
border: 4px solid #f3f3f3;
border-top: 4px solid #2563eb;
border-radius: 50%;
animation: spin 1s linear infinite;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

RESPONSIVE DESIGN

Media Queries:

```
/* Móviles (hasta 768px) */
@media (max-width: 768px) {
  .container {
    padding: 1rem;
  }

  .hero-title {
    font-size: 2rem;
  }
}

/* Tablets (768px - 1024px) */
@media (min-width: 768px) and (max-width: 1024px) {
  .container {
    padding: 2rem;
  }
}

/* Desktop (más de 1024px) */
@media (min-width: 1024px) {
  .container {
    padding: 3rem;
  }
}

/* Orientación */
@media (orientation: landscape) {
  .hero {
    height: 100vh;
  }
}

@media (orientation: portrait) {
  .hero {
    height: auto;
  }
}
```

```
    min-height: 100vh;
  }
}
```

Breakpoints comunes:

```
/* Extra pequeño (móviles) */
@media (max-width: 576px) { }

/* Pequeño (móviles grandes) */
@media (max-width: 768px) { }

/* Mediano (tablets) */
@media (max-width: 992px) { }

/* Grande (desktops) */
@media (max-width: 1200px) { }

/* Extra grande */
@media (min-width: 1200px) { }
```

Ejemplos prácticos:

Grid responsivo:

```
.projects-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: 2rem;
}

@media (max-width: 768px) {
  .projects-grid {
    grid-template-columns: 1fr;
    gap: 1rem;
  }
}
```

Navegación responsiva:

```
.nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

@media (max-width: 768px) {
```

```
.nav {  
  flex-direction: column;  
  gap: 1rem;  
}  
  
.nav-links {  
  flex-direction: column;  
  text-align: center;  
}  
}
```

MEJORES PRÁCTICAS

1. Organización del código:

```
/* 1. Variables CSS */  
:root {  
  --primary-color: #2563eb;  
  --spacing-md: 1rem;  
}  
  
/* 2. Reset/Normalize */  
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
/* 3. Estilos base */  
body {  
  font-family: var(--font-family);  
  line-height: 1.6;  
}  
  
/* 4. Layout */  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 0 1rem;  
}  
  
/* 5. Componentes */  
.button {  
  padding: 0.5rem 1rem;  
  border: none;  
  border-radius: 4px;  
}  
  
/* 6. Utilidades */
```

```
.text-center { text-align: center; }  
.mb-1 { margin-bottom: 1rem; }
```

2. Nomenclatura:

```
/* BEM (Block Element Modifier) */  
.card { } /* Bloque */  
.card__title { } /* Elemento */  
.card--featured { } /* Modificador */  
  
/* O simplemente descriptivo */  
.header { }  
.header-logo { }  
.header-nav { }  
.header-nav-link { }
```

3. Especificidad:

```
/* Evitar especificidad excesiva */  
.header .nav .link { } /* ✗ Muy específico */  
.nav-link { } /* ✓ Mejor */  
  
/* Usar clases en lugar de IDs */  
#header { } /* ✗ Difícil de sobrescribir */  
.header { } /* ✓ Más flexible */
```

4. Performance:

```
/* Usar transform en lugar de position para animaciones */  
.element {  
  transform: translateX(100px); /* ✓ Mejor performance */  
}  
  
.element {  
  left: 100px; /* ✗ Peor performance */  
}  
  
/* Usar will-change para optimizar animaciones */  
.animated {  
  will-change: transform;  
}
```

1. Header moderno:

```
.header {  
  background: white;  
  border-bottom: 1px solid #e2e8f0;  
  position: sticky;  
  top: 0;  
  z-index: 100;  
  box-shadow: 0 1px 3px rgba(0,0,0,0.1);  
}  
  
.header-content {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  padding: 1rem 0;  
}  
  
.header-logo {  
  font-size: 1.5rem;  
  font-weight: 700;  
  color: var(--primary-color);  
}  
  
.header-nav {  
  display: flex;  
  gap: 2rem;  
  list-style: none;  
}  
  
.header-nav a {  
  color: var(--text-secondary);  
  text-decoration: none;  
  transition: color 0.3s ease;  
}  
  
.header-nav a:hover {  
  color: var(--primary-color);  
}
```

2. Card con hover:

```
.card {  
  background: white;  
  border: 1px solid #e2e8f0;  
  border-radius: 8px;  
  padding: 1.5rem;  
  box-shadow: 0 1px 3px rgba(0,0,0,0.1);  
  transition: all 0.3s ease;  
}
```

```
.card:hover {
  transform: translateY(-4px);
  box-shadow: 0 8px 25px rgba(0,0,0,0.15);
}

.card-title {
  font-size: 1.25rem;
  font-weight: 600;
  margin-bottom: 0.5rem;
  color: var(--text-primary);
}

.card-description {
  color: var(--text-secondary);
  line-height: 1.6;
  margin-bottom: 1rem;
}
```

3. Botón con gradiente:

```
.btn-gradient {
  background: linear-gradient(135deg, var(--primary-color), var(--accent-color));
  color: white;
  border: none;
  padding: 0.75rem 1.5rem;
  border-radius: 6px;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s ease;
  box-shadow: 0 2px 4px rgba(37, 99, 235, 0.3);
}

.btn-gradient:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 12px rgba(37, 99, 235, 0.4);
}

.btn-gradient:active {
  transform: translateY(0);
}
```

4. Hero section:

```
.hero {
  background: linear-gradient(135deg, var(--bg-primary), var(--bg-secondary));
  min-height: 100vh;
  display: flex;
  align-items: center;
```

```
    position: relative;
  }

.hero-content {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 3rem;
  align-items: center;
}

.hero-title {
  font-size: 3rem;
  font-weight: 700;
  line-height: 1.1;
  margin-bottom: 1rem;
  color: var(--text-primary);
}

.hero-description {
  font-size: 1.125rem;
  line-height: 1.7;
  color: var(--text-secondary);
  margin-bottom: 2rem;
}

.hero-buttons {
  display: flex;
  gap: 1rem;
}

@media (max-width: 768px) {
  .hero-content {
    grid-template-columns: 1fr;
    text-align: center;
  }

  .hero-title {
    font-size: 2rem;
  }
}
```

5. Grid de proyectos:

```
.projects-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: 2rem;
  padding: 2rem 0;
}

.project-card {
```

```
    background: white;
    border-radius: 12px;
    overflow: hidden;
    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
    transition: transform 0.3s ease;
}

.project-card:hover {
    transform: translateY(-8px);
}

.project-image {
    width: 100%;
    height: 200px;
    background: linear-gradient(135deg, var(--primary-color), var(--accent-color));
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-size: 2rem;
    font-weight: 700;
}

.project-content {
    padding: 1.5rem;
}

.project-title {
    font-size: 1.25rem;
    font-weight: 600;
    margin-bottom: 0.5rem;
    color: var(--text-primary);
}

.project-description {
    color: var(--text-secondary);
    line-height: 1.6;
    margin-bottom: 1rem;
}

.project-tech {
    display: flex;
    flex-wrap: wrap;
    gap: 0.5rem;
}

.tech-tag {
    background: var(--bg-secondary);
    color: var(--text-secondary);
    padding: 0.25rem 0.75rem;
    border-radius: 20px;
    font-size: 0.875rem;
    font-weight: 500;
}
```

CONSEJOS FINALES

1. Empieza simple:

- Comienza con layouts básicos
- Usa Flexbox para la mayoría de casos
- Grid para layouts complejos

2. Usa variables CSS:

- Centraliza colores y espaciados
- Facilita cambios globales
- Mantiene consistencia

3. Piensa en mobile-first:

- Diseña primero para móviles
- Usa media queries para pantallas grandes
- Prueba en diferentes dispositivos

4. Mantén el código limpio:

- Usa nombres descriptivos
- Organiza por secciones
- Comenta código complejo

5. Aprende las herramientas:

- DevTools del navegador
- Flexbox y Grid visualizadores
- CSS validators

RECURSOS ADICIONALES

Herramientas:

- **CSS Grid Generator:** <https://cssgrid-generator.netlify.app/>
- **Flexbox Froggy:** <https://flexboxfroggy.com/>
- **CSS Grid Garden:** <https://cssgridgarden.com/>
- **CSS Validator:** <https://jigsaw.w3.org/css-validator/>

Referencias:

- **MDN CSS:** <https://developer.mozilla.org/en-US/docs/Web/CSS>
- **CSS-Tricks:** <https://css-tricks.com/>
- **Can I Use:** <https://caniuse.com/>

Práctica:

- **Frontend Mentor:** <https://www.frontendmentor.io/>
 - **CSS Battle:** <https://cssbattle.dev/>
 - **CodePen:** <https://codepen.io/>
-

¡Ahora tienes una base sólida de CSS! Practica, experimenta y no tengas miedo de probar cosas nuevas. El CSS es muy visual, así que la mejor manera de aprender es viendo los resultados de tu código.

¡Buena suerte con tu portfolio! 🍀