# PRACTICA DE DISEÑO DE INTERFACES WEB

**Práctica de CSS: Márgenes, bordes, padding y posicionamiento**

**Objetivos generales:**

* Comprender y aplicar márgenes, padding y bordes con precisión.
* Dominar los tipos de posicionamiento (static, relative, absolute, fixed, sticky).
* Crear maquetaciones básicas sin flexbox, usando inline-block y float.
* Observar cómo interactúan los distintos comportamientos de contenedores.

⏳ **Duración estimada:** 2–3 horas

**🧪 Ejercicio 1 — Márgenes externos y centrado horizontal**

Crea un div principal:

* Ancho: 300px, alto: 200px, color de fondo visible.
* Centra horizontalmente la caja con margin.
* Añade diferentes márgenes superior e inferior para separar del resto del contenido.
* Añade otra caja igual debajo y ajusta los márgenes para que no colapsen.

👉 *Objetivo: trabajar con márgenes horizontales y verticales, entender colapso de márgenes.*

**🧪 Ejercicio 2 — Padding y efecto sobre el tamaño**

Crea un contenedor con texto:

* Ancho: 300px, fondo distinto.
* Añade padding en cada lado con valores diferentes.
* Usa la herramienta de inspección para ver cómo **el tamaño total crece**.
* Luego, aplica box-sizing: border-box y compara.

👉 *Objetivo: comprender cómo padding afecta el box model y cómo cambia con border-box.*

**🧪 Ejercicio 3 — Bordes avanzados**

Crea una caja y experimenta:

* Diferentes estilos de borde (solid, dashed, dotted, double).
* Bordes con distintos grosores y colores en cada lado.
* Bordes redondeados (border-radius), incluyendo círculos perfectos.

👉 *Objetivo: dominar las propiedades de borde y personalizarlas por lado.*

**🧪 Ejercicio 4 — Box model completo**

Crea tres cajas una debajo de otra:

* Aplica a cada una una combinación distinta de margin, border y padding.
* Añade texto dentro para ver cómo se adapta el contenido.
* Usa la herramienta de desarrollo para **visualizar el modelo de caja completo**.
* Documenta en comentarios la diferencia entre cada caja.

👉 *Objetivo: visualizar y comparar modelos de caja.*

**🧪 Ejercicio 5 — Posicionamiento relative**

Crea un contenedor padre y un elemento hijo:

* Al padre, position: relative.
* Al hijo, position: relative también, y desplázalo con top, left, right.
* Observa cómo el desplazamiento no saca al elemento de su flujo original.

👉 *Objetivo: entender cómo se comporta la posición relativa dentro del flujo.*

**🧪 Ejercicio 6 — Posicionamiento absolute**

Usa el mismo contenedor:

* Cambia el hijo a position: absolute.
* Mueve el hijo con top, right, bottom, left.
* Quita position: relative al padre y observa cómo cambia la referencia (deja de estar limitado al contenedor).

👉 *Objetivo: diferenciar entre posicionamiento relativo y absoluto.*

**🧪 Ejercicio 7 — Fixed y sticky**

Crea un header en la parte superior:

* Usa position: fixed; top: 0; width: 100%.
* Añade contenido largo para poder hacer scroll.
* Verifica que el header queda fijo.
* Crea otro header debajo y hazlo sticky con top: 0 para ver la diferencia.

👉 *Objetivo: comprender cuándo conviene usar fixed o sticky.*

**🧪 Ejercicio 8 — Display: block, inline e inline-block**

Crea 4 pequeñas cajas con fondo distinto:

* Aplica display: block y observa el comportamiento en vertical.
* Cambia a inline y luego a inline-block.
* Usa width, height, margin y padding para ver qué efectos son ignorados o aplicados según el display.

👉 *Objetivo: controlar el flujo de elementos con diferentes valores de display.*

**🧪 Ejercicio 9 — Float y clear**

Crea una imagen y un párrafo de texto:

* Usa float: left para alinear la imagen a la izquierda con texto fluyendo alrededor.
* Usa float: right para alinear a la derecha.
* Añade un nuevo elemento con clear: both para forzar que empiece debajo de los flotantes.
* Usa overflow: auto en un contenedor padre para contener los floats.

👉 *Objetivo: practicar maquetación tradicional con float y clear.*

**🧪 Ejercicio 10 — Maquetación clásica sin flexbox**

Construye un **layout de 3 columnas** con un **header** y **footer**, usando solo:

* display: inline-block o float.
* Márgenes, bordes y padding para separar columnas y dar estructura.
* Asegúrate de que el footer queda debajo de las columnas.
* Usa box-sizing: border-box para facilitar el cálculo de anchos.