



El Modelo de Cajas (Box Model)



Propósito de la clase

En esta sesión, exploraremos el modelo de cajas en CSS y cómo afecta la organización y presentación de los elementos en una página web. Además, aprenderemos sobre los distintos tipos de posicionamiento y cuál es su impacto en la disposición del contenido. Finalmente, analizaremos los layouts más utilizados en el diseño web y sus ventajas y desventajas.

Al finalizar esta clase, serás capaz de:

- Comprender el modelo de cajas en CSS, identificando sus propiedades principales y cómo influyen en la estructura de los elementos.
- Diferenciar los tipos de cajas en CSS, reconociendo las diferencias entre elementos de bloque y línea y su impacto en el diseño.
- Inspeccionar y analizar elementos en el navegador, utilizando herramientas de desarrollo para visualizar márgenes, rellenos y bordes.
- Aplicar correctamente propiedades de posicionamiento, incluyendo estático, relativo, absoluto, fijo y flotante, para controlar la disposición de los elementos en una página.
- Definir qué es un layout y sus tipos (fluido, fijo, elástico y absoluto), comprendiendo sus ventajas y desventajas en distintos escenarios de diseño.

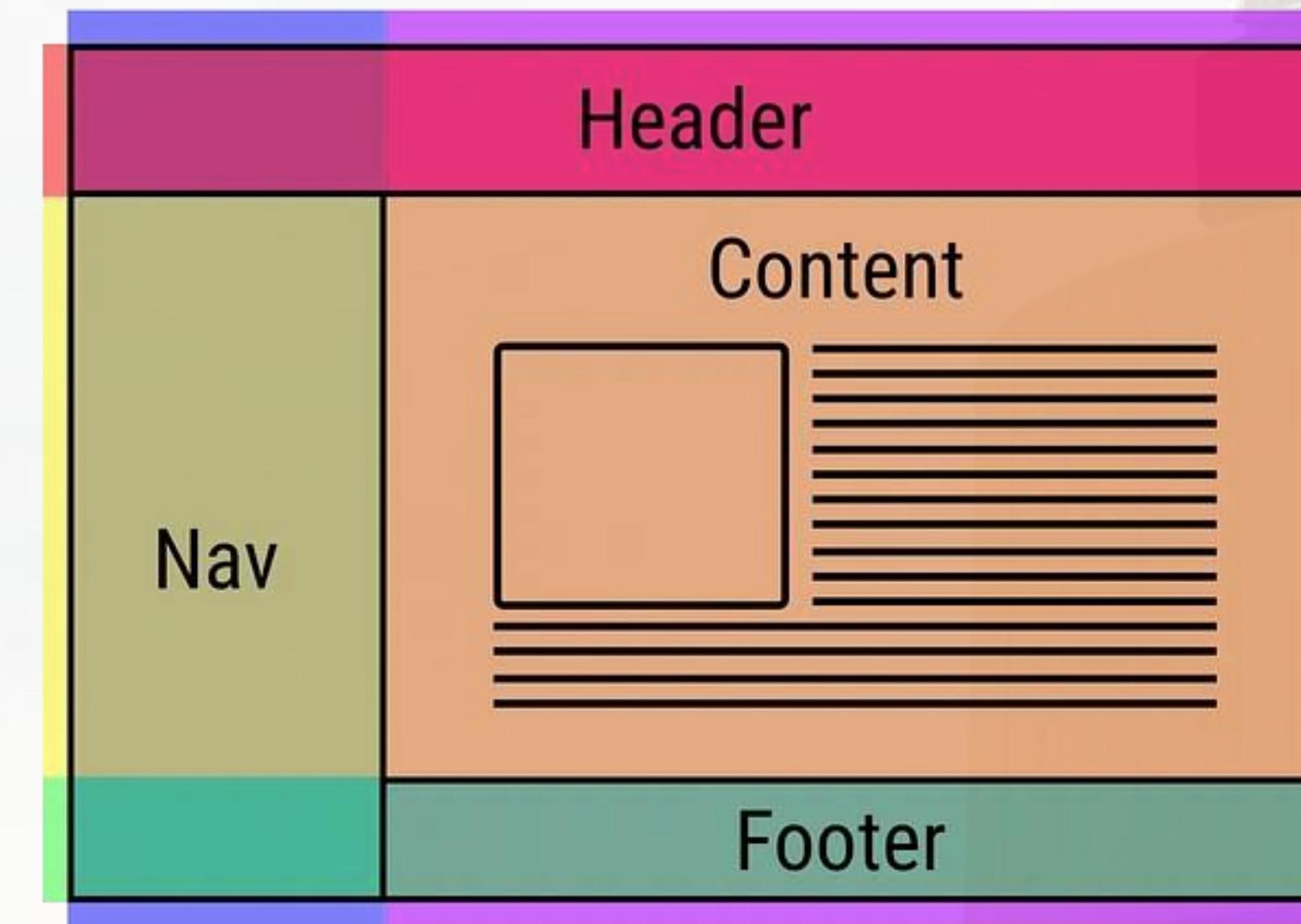
Layouts



📌 ¿Qué es un Layout en CSS?

Un layout es la estructura o distribución de los elementos en una página web. Piensa en ello como el plano de una casa 🏠, donde decides dónde colocar cada mueble (texto, imágenes, botones, etc.).

En CSS, hay varios tipos de layout, y cada uno tiene sus propias ventajas y desventajas. Vamos a explicarlos con lujo de detalles para que quede clarísimo.





Tipos de Layout en CSS

Layout Fijo (Fixed Layout)

En un layout fijo, el ancho de la página se define con valores exactos en píxeles (px). No importa si la pantalla es grande o pequeña, el tamaño del contenido siempre será el mismo.

```
.container {  
    width: 1200px; /* Tamaño fijo en pixeles */  
    margin: 0 auto; /* Centra el contenido */  
    background: lightgray;  
}
```

```
<div class="container">Este es un layout fijo</div>
```

✓ Ventajas:

- ✓ Es fácil de diseñar.
- ✓ No se deforma en diferentes pantallas.
- ✓ Se mantiene la misma apariencia en todos los dispositivos.

✗ Desventajas:

- ⚠ En pantallas pequeñas, se ve mal porque no se ajusta.
- ⚠ En móviles, puede desbordarse y necesitar scroll horizontal.

📌 Cuándo usarlo:

- Si el diseño debe ser exactamente igual en todas las pantallas.
- Si los usuarios van a usar pantallas grandes (PCs).

Layout Fluido (Liquid Layout)

Un layout fluido usa porcentajes (%) en lugar de píxeles (px). Así, el tamaño del contenido se ajusta automáticamente al tamaño de la pantalla.

```
.container {  
    width: 80%; /* El ancho será el 80% de la pantalla */  
    margin: 0 auto;  
    background: lightblue;  
}
```

```
<div class="container">Este es un layout fluido</div>
```

✓ Ventajas:

- ✓ Se adapta a diferentes tamaños de pantalla.
- ✓ No necesita diseño separado para móviles.
- ✓ Ahorra tiempo y esfuerzo.

✗ Desventajas:

- ⚠ En pantallas muy grandes, puede verse **demasiado ancho**.
- ⚠ En pantallas muy pequeñas, puede volverse **inusable**.

📌 Cuándo usarlo:

- Cuando quieres que la web **se ajuste a cualquier pantalla**.
- Para evitar hacer un diseño diferente para móviles.

Layout Elástico (Elastic Layout)

Un layout elástico usa em o rem en lugar de píxeles (px) o porcentajes (%). Esto significa que el tamaño depende del tamaño del texto en la página.

```
.container {  
    width: 60em; /* Se ajusta según el tamaño del texto */  
    margin: 0 auto;  
    background: lightgreen;  
    font-size: 1em; /* Base */  
}
```

```
<div class="container">Este es un layout elástico</div>
```

✓ Ventajas:

- ✓ Se ajusta cuando el usuario **cambia el tamaño del texto**.
- ✓ Es más accesible para personas con problemas de visión.
- ✓ Funciona bien en distintas pantallas.

✗ Desventajas:

- ⚠ Puede romperse si el usuario cambia el tamaño del texto mucho.
- ⚠ No es tan predecible como un layout fijo.

📌 Cuándo usarlo:

- Cuando el contenido es **principalmente texto**.
- Para sitios que necesitan ser **muy accesibles**.

Layout Absoluto (Absolute Layout)

Un layout absoluto usa position: absolute; para colocar los elementos en posiciones fijas, sin importar el flujo normal de la página.

```
.container {  
    width: 400px;  
    height: 300px;  
    position: absolute;  
    top: 100px;  
    left: 200px;  
    background: orange;  
}
```

```
<div class="container">Este es un layout absoluto</div>
```

✓ Ventajas:

- ✓ Permite hacer **diseños precisos**.
- ✓ Se puede colocar contenido **exactamente donde lo necesitas**.
- ✓ Útil para **ventanas emergentes y overlays**.

✗ Desventajas:

- ⚠ Puede hacer que la página sea **difícil de mantener**.
- ⚠ No se adapta bien a diferentes pantallas.
- ⚠ Puede **romper el diseño** si no se usa bien.

📌 Cuándo usarlo:

- Para **elementos flotantes** como ventanas emergentes.
- Para elementos que **no deben moverse** con el contenido.



Comparación de Tipos de Layout

Tipo de Layout	Se adapta a la pantalla 📱💻	Fácil de diseñar 🔧	Recomendado para ✂️
Fijo	✗ No	✓ Sí	Sitios con diseño exacto
Fluido	✓ Sí	🟡 Medio	Sitios flexibles
Elástico	🟡 Sí (depende del texto)	🟡 Medio	Sitios con mucho texto
Absolute	✗ No	✗ No	Ventanas emergentes

🎯 ¿Cuál elegir?

- 📌 Si necesitas que el sitio se vea igual en todas las pantallas → Usa un layout Fijo.
- 📌 Si quieres que el sitio se adapte automáticamente → Usa un layout Fluido.
- 📌 Si el contenido es texto y quieres accesibilidad → Usa un layout Elástico.
- 📌 Si necesitas colocar elementos en posiciones específicas → Usa un layout Absoluto.

Ejercicio



Ejercicio Práctico para el Facilitador: Construcción de una Landing Page Simple con Posicionamiento.

Objetivo:

El facilitador demostrará en vivo cómo construir una landing page simple sin utilizar flexbox, float, ni position: fixed. Esto permitirá comprender mejor la disposición de los elementos en CSS sin depender de estas propiedades avanzadas.

Descripción del Ejercicio:

El facilitador deberá desarrollar una landing page básica utilizando solo position: absolute y relative para organizar los elementos. La página debe incluir:

- Encabezado fijo en la parte superior.
- Una imagen de fondo con un área de contenido sobrepuerta.
- Texto con descripción y un botón centrado.
- Pie de página al final de la página. Los estudiantes podrán observar cómo se comportan los elementos en la disposición y cómo se pueden lograr diseños atractivos sin recurrir a flexbox o float.

Modelo del Sitio

Simple Landing Page



Welcome to Our Page

We create stunning visuals and provide innovative solutions to make your workflow easier and more fun. Explore our services and get in touch with us for a better digital experience.

[Learn More](#)

© 2024 All Rights Reserved

Demostración



Lluvia de Ideas



Lluvia de Ideas sobre el Modelo de Cajas y el Layout en CSS

Hemos explorado diferentes conceptos clave sobre **el modelo de cajas en CSS** y su impacto en la disposición de los elementos en una página web. Comprender este modelo es fundamental para estructurar correctamente cualquier diseño.

Ahora, queremos reflexionar sobre lo aprendido y compartir nuestras opiniones para reforzar los conceptos.

Instrucciones:

- Responde de manera breve y honesta.
- No hay respuestas incorrectas, la idea es generar discusión y reflexión.
- Comparte tu opinión y escucha la de tus compañeros para enriquecer el aprendizaje.



Preguntas para la Lluvia de Ideas

- ◆ ¿Qué concepto del modelo de cajas te pareció más claro o fácil de entender?
- ◆ ¿Cómo crees que el conocimiento sobre las propiedades del modelo de cajas te ayuda a estructurar mejor un sitio web?
- ◆ ¿Qué tipo de posicionamiento crees que es más útil en un diseño web y por qué?
- ◆ ¿Cómo influye la elección del tipo de layout en la experiencia del usuario?
- ◆ ¿Cuál de los tipos de layout (fluido, fijo, elástico, absoluto) crees que se adapta mejor a los proyectos web modernos?

📌 Comparte tus ideas y reflexionemos juntos sobre cómo el modelo de cajas y el layout en CSS nos ayudan a crear sitios web organizados, accesibles y eficientes. 🚀

Conclusión





Conclusión: Reflexionando sobre el Modelo de Cajas y el Layout en CSS

Después de explorar y trabajar con el modelo de cajas en CSS, hemos visto cómo este concepto es fundamental para estructurar y organizar los elementos en una página web. Cada elemento en HTML se comporta como una caja, y entender sus propiedades nos permite diseñar interfaces más flexibles y funcionales.

También hemos aprendido sobre los diferentes tipos de cajas y cómo los elementos de bloque y línea afectan la distribución del contenido en la página. A través de la inspección de elementos en el navegador, podemos identificar estos comportamientos y realizar ajustes precisos.

Además, hemos analizado el posicionamiento en CSS, diferenciando entre estático, relativo, absoluto, fijo y flotante, lo que nos permite colocar los elementos en la pantalla de manera estratégica y optimizar la experiencia del usuario.

Finalmente, abordamos los layouts en CSS, explorando diseños fluidos, fijos, elásticos y absolutos, entendiendo sus ventajas y desventajas para aplicarlos correctamente en distintos escenarios de diseño.



Preguntas :

¿Cómo ha cambiado tu forma de entender la disposición de los elementos en CSS después de aprender sobre el modelo de cajas, el posicionamiento y los distintos tipos de layouts?

¿Cómo podrías aplicar este conocimiento en futuros proyectos para mejorar la estructura de tus diseños web?



IT Academy

by KIBERNUM