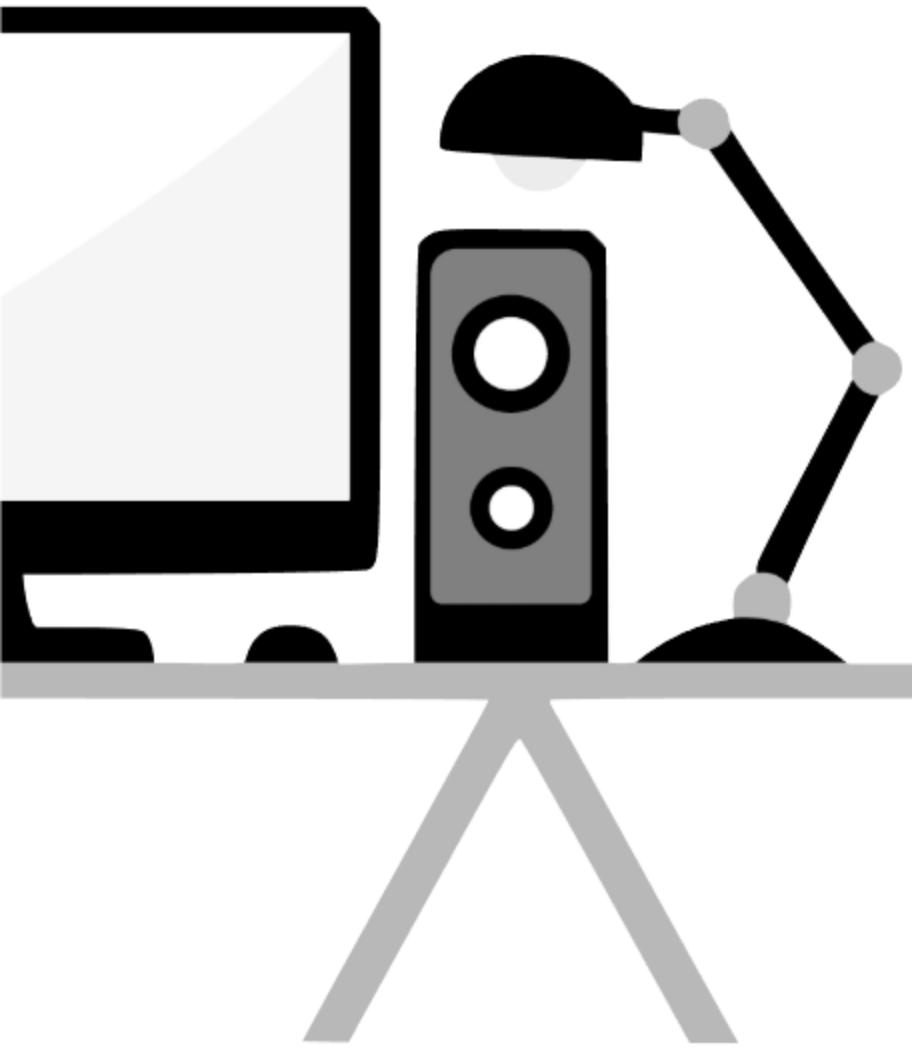




# DESARROLLO DE LA INTERFAZ DE USUARIO WEB

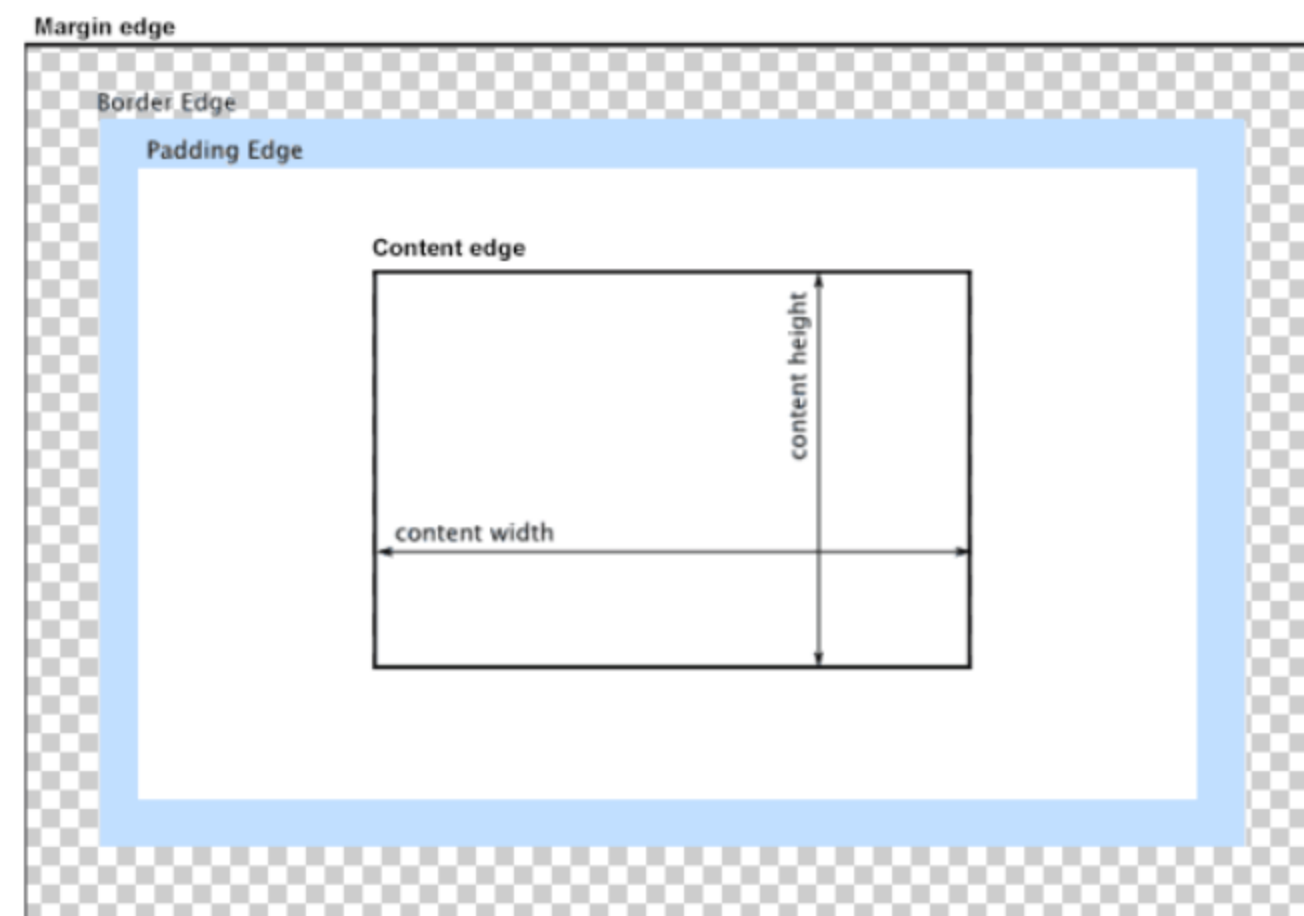
## Lección 04



1. El modelo de cajas y el Layout Modelo de cajas
2. Qué es el modelo de cajas.
3. Propiedades que componen el modelo de cajas. Tipos de cajas.
4. ¿Existen otros modelos?
5. Diferencias entre elementos de bloque y elementos de línea.
6. Inspeccionando elementos con navegador para identificar las cajas.
7. Posicionamiento de elementos y visualización
8. Conceptos básicos de propiedades de posicionamiento.
9. Tipo de Posicionamiento (normal, relativo, absoluto, fijo, flotante)
10. Visualización
11. Qué es un layout
12. Tipos de layout (fluido, fijo, elástico, absoluto)

Al maquetar un documento, el motor de renderizado del navegador representa cada elemento como una caja rectangular, conforme al estándar del **modelo de caja básico de CSS**. CSS determina el tamaño, la posición y las propiedades (color, fondo, tamaño del borde, etc.) de estas cajas.

Cada caja está compuesta de cuatro partes (o áreas), definidas por sus respectivos límites: *límite de contenido (content edge)* , *límite de relleno (padding edge)* , *límite de borde (border edge)* y *límite de margen (margin edge)* .



## Área de contenido (content area)

El área de contenido o content area, delimitada por el límite de contenido (Content edge en la imagen), contiene el contenido "real" del elemento, como texto, una imagen o un reproductor de video. Sus dimensiones son "content width" o ("content-box width") que hace referencia al ancho del contenido, y "content height" o ("content-box height") que hace referencia al alto.

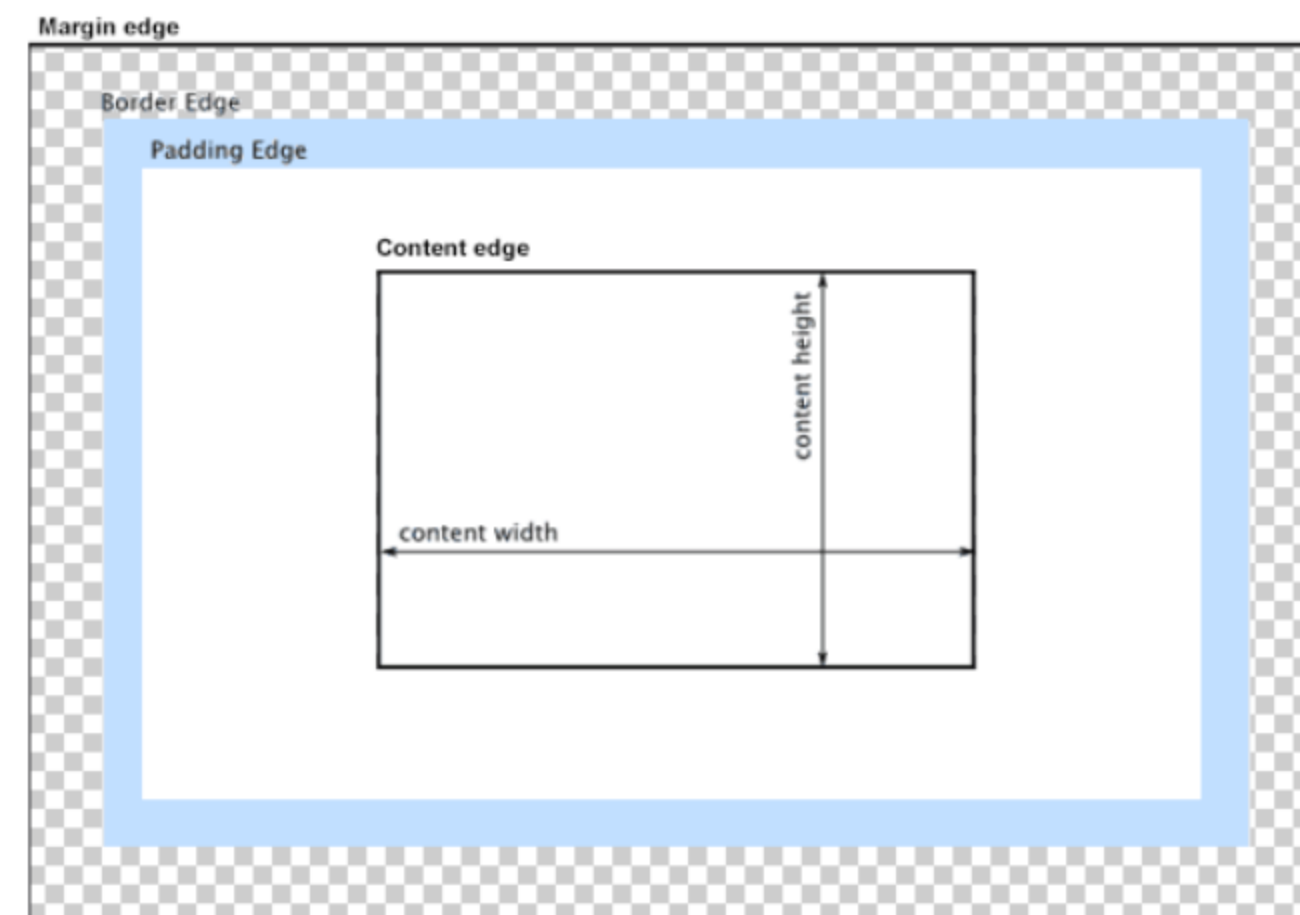
A menudo, tiene un color o una imagen de fondo.

Si la propiedad box-sizing está definida como content-box (que es el valor por defecto) y el elemento es un "elemento de bloque", el tamaño del área de contenido puede definirse explícitamente con las propiedades width , min-width , max-width, height , min-height y max-height .

## Área de relleno (padding area)

El área de relleno o padding area, delimitada por el límite de relleno (Padding Edge en la imagen), extiende el área de contenido para incluir el relleno del elemento. Sus dimensiones son "padding-box width" para el ancho y "padding-box height" para el alto.

El tamaño del relleno viene determinado por las propiedades padding-top , padding-right (en-US) , padding-bottom , padding-left (en-US) y la propiedad resumida padding .



## Área de margen (margin area)

El área de margen o margin area, delimitada por el límite de margen (Margin edge en la imagen), extiende el área de borde para incluir un área vacía, utilizada para separar al elemento del resto de sus vecinos. Sus dimensiones son "margin-box width" para el ancho y "margin-box height" para el alto.

El tamaño del área de margen está determinado por las propiedades margin-top , margin-right , margin-bottom , margin-left y la propiedad resumida margin . Cuando se produce un colapso de margen, el área de margen no está claramente definida ya que los márgenes se comparten entre cajas.

Por último, tenga en cuenta que para los elementos en línea no modificados, la cantidad de espacio ocupado (la contribución a la altura de la línea) está determinada por la propiedad line-height , aunque los bordes y el relleno sigan apareciendo alrededor del contenido.



# ¿Otros modelos de cajas?

Es importante destacar que el desarrollo de css a través del tiempo ha cambiado mucho. Estos cambios también han existido respecto al modelo de cajas, aunque a estas alturas del desarrollo web, se considera que el modelo de cajas tradicional es el modelo que impera y por tanto el que debe aplicarse.

Otra cosa a considerar es que aunque el modelo de cajas es la base y piedra angular del diseño de elementos web, existen otros métodos de diseño a considerar y que son importantes, nos referimos acá al sistema de flex y al sistema grid.

Estos sistemas de diseño de layout auxiliares ayudan a poder manejar de forma más fácil el diseño responsivo, sin tener que acudir tanto a media queries para a mano cambiar cosas en el diseño de cajas tradicional y sus elementos.

# Diferencias entre elementos de bloque y elementos de línea.

En CSS, en general, hay dos tipos de cajas: cajas en bloque y cajas en línea.

Estas características se refieren al modo como se comporta la caja en términos de flujo de página y en relación con otras cajas de la página.

Si una caja se define como un bloque, se comportará de las maneras siguientes:

- 1- La caja fuerza un salto de línea al llegar al final de la línea.
- 2.- La caja se extenderá en la dirección de la línea para llenar todo el espacio disponible que haya en su contenedor. En la mayoría de los casos, esto significa que la caja será tan ancha como su contenedor, y llenará el 100% del espacio disponible.
- 3.- Se respetan las propiedades width y height.
- 4.- El relleno, el margen y el borde mantienen a los otros elementos alejados de la caja.

**A menos que decidamos cambiar el tipo de visualización a en línea, elementos como los encabezados (por ejemplo, <h1>) y todos los elementos <p> usan por defecto block como tipo de visualización externa.**



# Diferencias entre elementos de bloque y elementos de línea.

Si una caja tiene una visualización externa de tipo inline, entonces:

- 1.- La caja no fuerza ningún salto de línea al llegar al final de la línea.
- 2.- Las propiedades width y height no se aplican.
- 3.- Se aplican relleno, margen y bordes verticales, pero no mantienen alejadas otras cajas en línea.
- 4.- Se aplican relleno, margen y bordes horizontales, y mantienen alejadas otras cajas en línea.

El elemento <a>, que se utiliza para los enlaces, y los elementos <span>, <em> y <strong> son ejemplos de elementos que se muestran en línea por defecto.

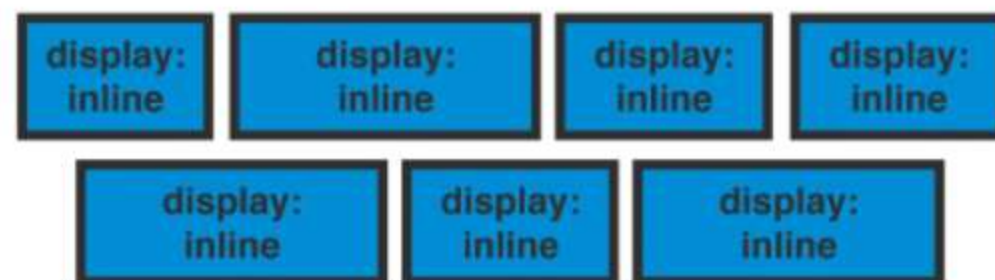
# Diferencias entre elementos de bloque y elementos de línea.

El tipo de caja que se aplica a un elemento está definido por los valores de propiedad display, como block y inline, y se relaciona con el valor **externo** (*outer*) de visualización (display).

## display: block

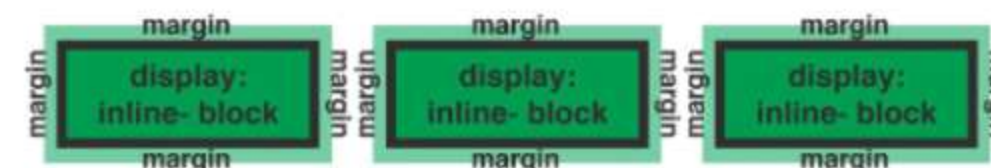
display: block	abarca 100% del width
display: block	abarca 100% del width
display: block	abarca 100% del width

## display: inline



- Permite a sus hermanos abarcar espacios vacíos.
- No nos permite manipular width ni height, tampoco agregar margin ni padding en ninguna dirección

## display: inline-block



- Permite a sus hermanos abarcar espacios vacíos.
- Si nos permite manipular width y height, también agregar margin y padding.

# Visualización interna de los elementos

Las cajas también tienen un tipo de visualización interna, que determina cómo se disponen los elementos dentro de esa caja.

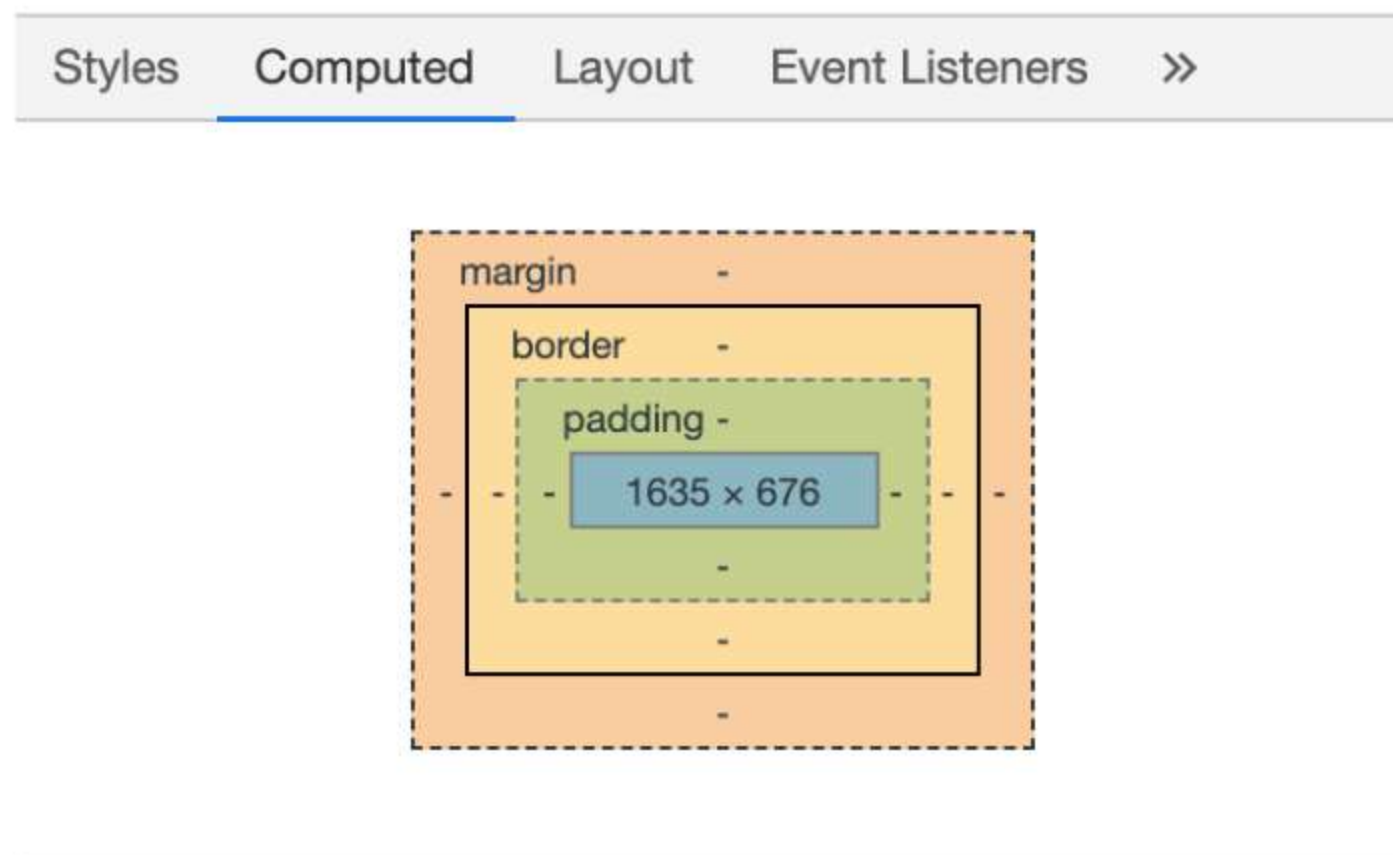
De forma predeterminada, los elementos dentro de una caja se presentan en flujo normal, lo que significa que se comportan como otros elementos de tipo en bloque o en línea (como se explicó anteriormente).

Sin embargo, podemos cambiar el tipo de visualización interna utilizando valores de display, como flex o grid. Si en un elemento establecemos `display: flex;`, el tipo de visualización externa es de tipo bloque (block), pero el tipo de visualización interna cambia a flexible (flex). Cualquier elemento que sea hijo directo de esta caja pasará a comportarse como un elemento de tipo flex, de acuerdo con las reglas que se establecen en la especificación de Flexbox, tema que veremos más adelante.



# Inspeccionando elementos con navegador.

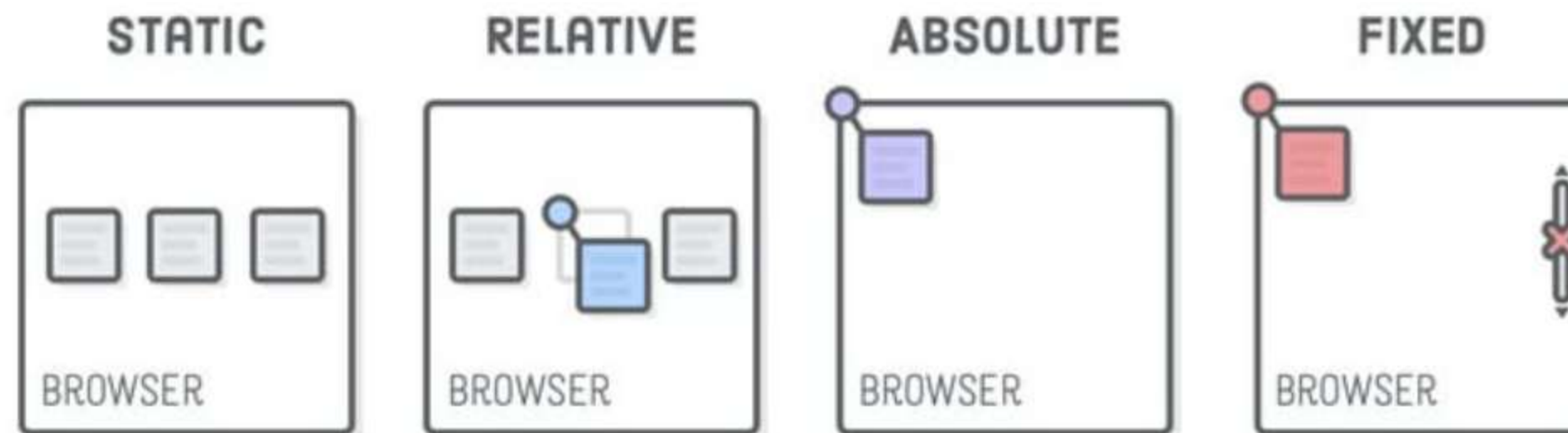
La mayoría de los navegadores cuentan con un inspector web en el cual se puede visualizar el modelo de cajas de un elemento web seleccionado, en el caso de chrome y su motor v8, esto se puede hacer accediendo al inspector web de Chrome.



# Conceptos básicos de propiedades de posicionamiento

El **posicionamiento** en CSS nos permite controlar cómo se ubican los elementos en la página, más allá del flujo normal del documento. Esto se logra principalmente a través de la propiedad *position* y sus valores asociados.

La propiedad *position* define el **método de posicionamiento** utilizado para un elemento HTML. Según el valor que se use, el elemento se comportará de forma distinta en relación con su contenedor y con el resto del contenido.





# Tipos de posicionamiento

Tipo	Descripción	Comportamiento
<b>static</b>	Valor por defecto. El elemento se posiciona según el flujo normal del documento.	No se aplican top, left, right ni bottom.
<b>relative</b>	Se posiciona en relación a su <b>posición original</b> .	Puedes moverlo usando top, left, etc., sin sacarlo del flujo del documento.
<b>absolute</b>	Se posiciona respecto a su <b>contenedor con posición relativa (o al &lt;body&gt; si no hay otro)</b> .	Elimina el elemento del flujo, y lo posiciona exactamente con top, left, right, bottom.
<b>fixed</b>	Se posiciona respecto al <b>viewport (pantalla visible)</b> y no se mueve al hacer scroll.	Ideal para menús o botones flotantes.
<b>sticky</b>	Comienza como relative, pero cuando se alcanza una posición definida (como top: 0), se “pega”.	Muy útil para headers que se fijan al llegar a cierto punto del scroll.

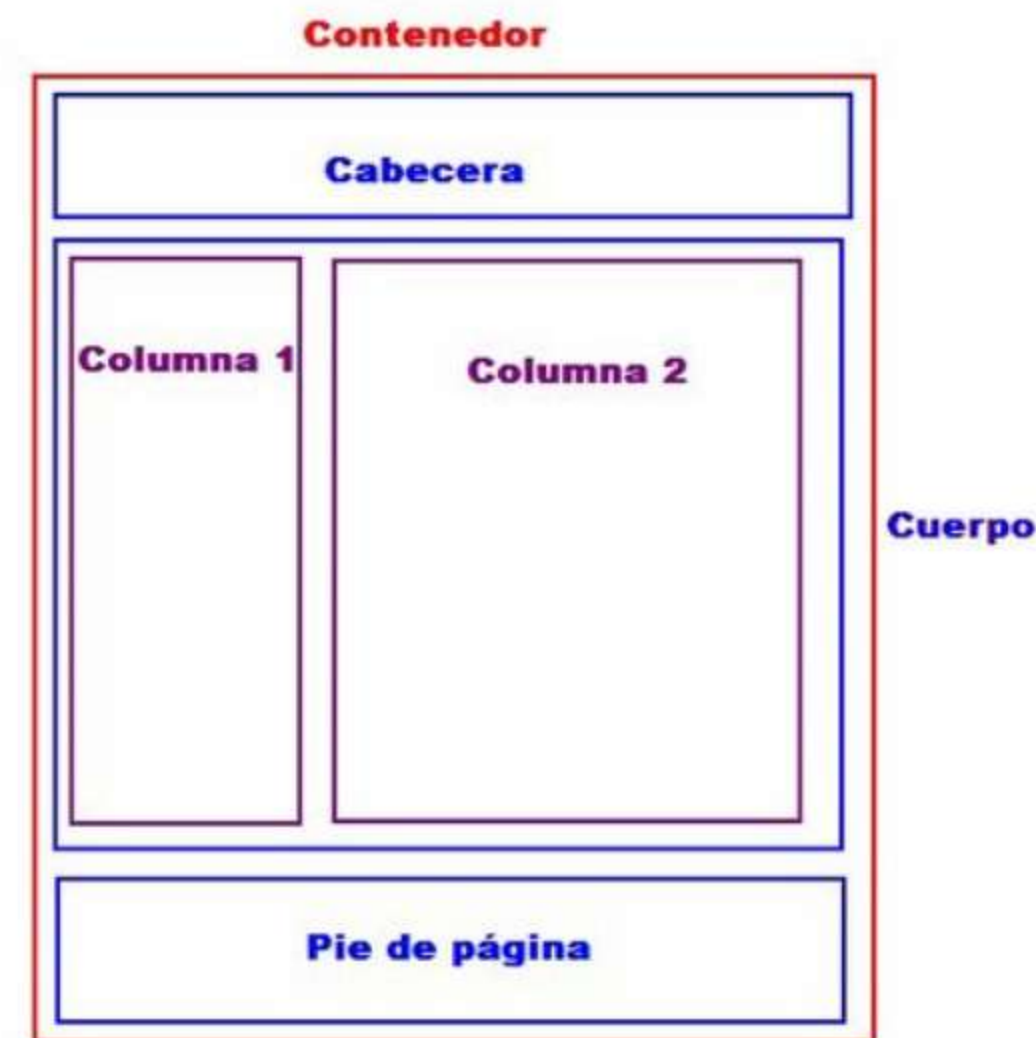


# Qué es un layout

Un **layout** es la forma en que se **organizan y distribuyen los elementos visuales** dentro de una página web. Define **cómo se posicionan** textos, imágenes, menús y otros contenidos en el espacio disponible, adaptándose a diferentes pantallas y dispositivos.

Es un componente clave del diseño web, ya que afecta directamente la **experiencia del usuario**, la **legibilidad** y la **navegación**.

Sirve para **estructurar visualmente** el contenido de una página de manera **coherente, funcional y atractiva**, facilitando que el usuario comprenda la información, navegue con mayor facilidad y disfrute de una mejor experiencia.



# Tipos de layout

Tipo de Layout	Descripción	Ventajas	Desventajas
<b>Fixed</b>	Usa unidades absolutas como px. El ancho del contenedor se mantiene constante, sin importar el tamaño de la pantalla.	Fácil de diseñar y controlar.	No se adapta bien a pantallas pequeñas. Puede generar scroll horizontal.
<b>Fluid</b>	Usa unidades relativas como %. El ancho del contenido se ajusta proporcionalmente al tamaño del viewport.	Se adapta a distintos tamaños de pantalla.	Puede deformarse en resoluciones extremas si no se controla bien.
<b>Elastic</b>	Usa unidades relativas al texto (em, rem). El layout se adapta al tamaño de la fuente base.	Mejora la accesibilidad, ya que se ajusta al zoom del navegador.	Puede ser impredecible si se usan fuentes muy grandes o muy pequeñas.
<b>Absolute</b>	Coloca elementos en posiciones fijas sin responder al cambio de pantalla, generalmente con position: absolute.	Control total sobre la ubicación de los elementos.	No es responsivo. Dificulta la adaptabilidad y mantenimiento.



*Instantiva*

CON • SENTIDO