



## Paso 5 - ¡A practicar!



### ¡Bienvenido a tu siguiente paso!

En este paso vamos a ver cómo funciona el **modelo de cajas** y entender cómo se comportan los **elementos en bloque vs los elementos en línea**.

Podemos mencionar como elementos en bloque

- `<div>`: Es un contenedor genérico que se utiliza para agrupar otros elementos y aplicarles estilos CSS o realizar manipulaciones con JavaScript.
- `<p>`: Representa un párrafo de texto.
- `<h1>`, `<h2>`, ..., `<h6>`: Son los encabezados, siendo `<h1>` el de mayor jerarquía y `<h6>` el de menor.
- `<ul>`, `<ol>`, `<li>`: Se utilizan para crear listas, ya sean desordenadas (`<ul>`) o ordenadas (`<ol>`), y cada elemento de la lista se representa con `<li>`.
- `<table>`, `<tr>`, `<td>`, `<th>`: Se utilizan para crear tablas.

Podemos mencionar como elementos en línea

- `<span>`: Similar a `<div>`, pero es un contenedor en línea. Se utiliza para aplicar estilos o JavaScript a una porción de texto.
- `<a>`: Define un enlace (hipervínculo) a otra página o recurso.
- `<img>`: Se utiliza para incrustar imágenes.
- `<strong>` y `<b>`: Se utilizan para dar énfasis al texto, haciéndolo negrita. Aunque visualmente son similares, `<strong>` tiene un significado semántico de mayor importancia o seriedad.
- `<em>` y `<i>`: Se utilizan para dar énfasis al texto, haciéndolo cursiva. Al igual que con `<strong>` y `<b>`, `<em>` tiene un significado semántico de énfasis, mientras que `<i>` es simplemente una representación visual.

## ✍ Actividad

### Actividad - Practicando con el Modelo de Cajas

El objetivo de esta actividad es practicar el modelo de cajas

Para ello, sigue estos pasos en tu computadora, y luego comparte con tus compañeros:

1. Dentro de la estructura de carpetas donde vienes trabajando, crea una nueva carpeta dentro de PWDC que se llame 📁 **css-modelo-de-cajas**. (Esta no se encuentra dentro **mi\_primera\_web**, sino que se encuentra a la misma altura.)

2. Dentro de ella, crea un archivo HTML llamado **cajas.html** y coloca el siguiente código

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title>Modelo de cajas</title>
    <link rel="stylesheet" href="box-styles.css" />
  </head>
  <body>
    <h1>Los encabezados son elementos de bloque</h1>

    <p>
      Los párrafos son también elementos de bloque <em>sin mebargo</em>,
      &lt;em>> y &lt;strong>> no lo son. Ellos se comportan como elementos
      <strong>en línea</strong>
    </p>

    <div>
      Los elementos en bloque ocupan todo el ancho del elemento que los
      contiene. Los <span>span, son elemenos en línea. </span>
    </div>
  </body>
</html>
```

3. Crea el archivo **box-styles.css** en la misma carpeta que **cajas.html** (en el HTML que te brindamos ya se encuentra vinculado si mantienes los nombres de los archivos tal como están escritos).

4. Coloca el siguiente código css en el archivo **box-styles.css**

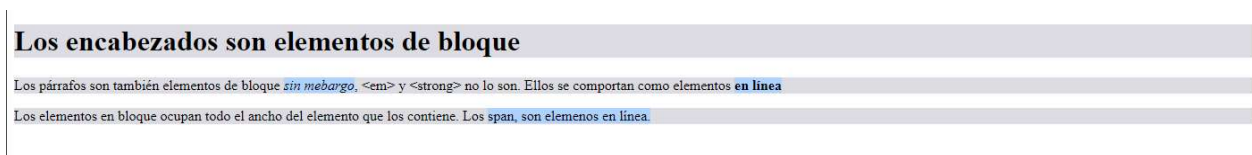
```
h1,
p,
div {
  background-color: #dde0e3;
}

em,
strong,
span {
  background-color: #b2d6ff;
}
```

Aquí lo que estamos haciendo es aplicar un color de fondo para los elementos HTML que se comportan como bloque y otro para los elementos que se comportan como en línea.

💡 **Tip:** Si más de un elemento contiene los mismos estilos de CSS, podemos separar a los selectores con una coma, y luego escribir una sola vez las propiedades y valores. Ejemplo: `elemento1, elemento2, elemento3 {propiedad:valor;}`

Al final deberían obtener una página similar a la siguiente:



5. Al elemento **h1**, aplicarle un **padding** de **50px**, **margin** de **50px** y **border** **5px solid black**.

💡 **Tip:** Para la propiedad **border**, lo que hicimos es darle un ancho en pixels al borde (**5px**), indicarle que deberá ser **sólido (solid)**, y de color **negro (black)**.

6. Al elemento **div**, aplicarle el siguiente **padding: 10px 5px 15px 3px;** (De esta manera, al asignar "10px 5px 15px 3px" se está indicando un margen superior de 10px, uno derecho de 5px, un margen inferior de 15px e izquierdo de 3px)

7. Agrega **border**, **margin** y **padding** al resto de los elementos y visualiza cómo se comportan.

📌 **Importante:** Un concepto clave en **html** es el de "indentar" el código. Esto se refiere a agregar sangrías para hacerlo más legible y estructurado. Aunque el navegador no interpreta los espacios adicionales o tabulaciones en el código **HTML**, indentar es una buena

*práctica para los desarrolladores, ya que ayuda a visualizar la estructura del documento y la relación entre los elementos.*

Una extensión de **Visual Studio Code**, que ayuda en esta organización, es [Prettier](#) 🙌

¿**Cómo logro esto?** Si tienes dudas sobre indentar el código, pueden ver el siguiente video:

