**📦 display: block**

**✔️ Comportamento:**

* O elemento **ocupa toda a largura disponível** da página (100% da linha).
* **Cria uma nova linha** no fluxo do documento (ou seja, empurra os próximos elementos para baixo).
* Exemplo de elementos com display block por padrão: <div>, <p>, <section>, <h1>, etc.

**🧠 Quando usar?**

Use block quando quiser que o elemento:

* Fique **em sua própria linha**.
* **Empurre os outros elementos para baixo**.
* **Organize conteúdo verticalmente**.

**📐 display: flex**

**✔️ Comportamento:**

* Transforma o elemento em um **"flex container"**.
* Permite controlar o layout dos **filhos (flex items)** com **alinhamento horizontal ou vertical**, espaçamento, ordenação etc.
* Os filhos **não criam novas linhas automaticamente** (a menos que flex-wrap: wrap esteja ativado).

**🧠 Quando usar?**

Use flex quando quiser:

* **Alinhar elementos lado a lado** (horizontalmente ou verticalmente).
* **Distribuir espaço automaticamente** entre os filhos.
* Criar **layouts responsivos** de forma fácil.

**🆚 Comparação Visual**

| **Atributo** | **display: block** | **display: flex** |
| --- | --- | --- |
| Alinhamento interno | Vertical, padrão | Totalmente controlável (horizontal e vertical) |
| Ocupa largura inteira? | Sim | Não necessariamente |
| Fluxo dos filhos | Um embaixo do outro | Um ao lado do outro (por padrão) |
| Uso ideal | Estrutura básica de página | Layouts dinâmicos e alinhamentos complexos |

Excelente dúvida! A diferença entre **position: relative** e **position: absolute** é fundamental para o controle do layout em CSS. Vamos quebrar as diferenças para você entender quando usar cada uma delas:

**🏷️ position: relative**

**✔️ Comportamento:**

* O elemento é posicionado **relativamente à sua posição original** no fluxo do documento.
* Ele **ainda ocupa o espaço** no layout, mas você pode **ajustar sua posição** usando as propriedades top, right, bottom, e left, **sem afetar o layout dos outros elementos**.

**🧠 Quando usar:**

* **Mover um elemento ligeiramente** sem removê-lo do fluxo normal.
* Criar um "referência" para outros elementos com **position: absolute** dentro dele.

**🏷️ position: absolute**

**✔️ Comportamento:**

* O elemento é posicionado **em relação ao seu contêiner pai mais próximo** com position: relative, absolute ou fixed. Se não houver um, ele será posicionado em relação ao **elemento <html> (ou seja, a página inteira)**.
* Ele **sai do fluxo normal do layout**, ou seja, não ocupa mais espaço e não afeta o posicionamento dos outros elementos.

**🧠 Quando usar:**

* **Posicionar um elemento de forma independente** dentro de um contêiner, por exemplo, para fazer um pop-up ou menu suspenso.
* Quando você quer **precisar que o elemento seja sobreposto** a outros.

**🆚 Diferença Visual e Comportamental**

| **Propriedade** | **position: relative** | **position: absolute** |
| --- | --- | --- |
| Fluxo do Layout | O elemento **ainda ocupa espaço no fluxo** | **Remove-se do fluxo** (não ocupa espaço) |
| Deslocamento | O deslocamento é feito em relação à **posição original** | O deslocamento é feito em relação ao **pai mais próximo posicionado** |
| Dependência | **Depende do seu local original** no layout | **Independente** (fica em um ponto específico no contêiner) |
| Exemplo comum | Usado para mover levemente um elemento dentro de seu contexto | Usado para **pop-ups**, **menus suspensos**, **tooltips** |

**🧠 Quando usar cada um?**

* **Use relative** quando você precisa:
  + Mover um elemento **sem alterar o fluxo de layout** (ex: ajustar a posição de um botão).
  + Criar um **ponto de referência** para elementos absolute dentro do mesmo contêiner.
* **Use absolute** quando você precisa:
  + Posicionar um elemento de forma **independente** e **fora do fluxo normal**.
  + **Sobrepor ou flutuar** um elemento em cima de outros (ex: criar um pop-up, um tooltip, ou um menu dropdown).