**Elementos Block e Inline**

Cada elemento HTML tem um valor de exibição padrão, dependendo do tipo de elemento.

Existem dois valores de exibição: **Block** e **Inline**.

**Elementos block**

Um elemento de nível de bloco sempre começa em uma nova linha e os navegadores adicionam automaticamente algum espaço (uma margem) antes e depois do elemento.

Um elemento de nível de bloco sempre ocupa toda a largura disponível (estende-se para a esquerda e para a direita o máximo possível).

Aqui estão os elementos de nível de **block** em HTML:

<address> <article> <aside> <blockquote> <canvas> <dd> <div> <dl> <dt> <fieldset> <figcaption> <figure> <footer> <form> <h1>-<h6> <header> <hr> <li> <main> <nav> <noscript> <ol> <p> <section> <table> <tfoot> <ul> <video>

**Elementos Inline**

Um elemento **inline** não começa em uma nova linha.

Um elemento **inline** ocupa apenas a largura necessária.

Aqui estão os elementos **inline** em HTML:

<a> <abbr> <acronym> <b> <bdo> <big> <br> <button> <cite> <code> <dfn> <em> <i> <img> <input> <kbd> <label> <map> <object> <output> <q> <samp> <script> <select> <small> <span> <strong> <sub> <sup> <textarea> <time> <tt> <var>

**justify-content**, que alinha itens horizontalmente e aceita os seguintes valores:

* **flex-start**: Itens se alinham à esquerda do container.
* **flex-end**: Itens se alinham à direita do container.
* **center**: Itens se alinham no centro do container.
* **space-between**: Itens se alinham com distância igual entre eles.
* **space-around**: Itens se alinham com distância igual em torno deles.

**align-items**, essa propriedade CSS alinha os itens verticalmente e aceita os seguintes valores:

* **flex-start**: Itens se alinham na parte de cima do container.
* **flex-end**: Itens se alinham na parte de baixo do container.
* **center**: Itens se alinham no centro vertical do container.
* **baseline**: Items se alinham na linha da base do container.
* **stretch**: Itens se esticam para preencher o container.

**flex-direction**, esta propriedade CSS define a direção em que os itens são posicionados no container e aceita os seguintes valores:

* **row**: Itens são posicionados na mesma direção do texto.
* **row-reverse**: Itens são posicionados na direção oposta à do texto.
* **column**: Itens são posicionados de cima para baixo.
* **column-reverse**: Itens são posicionados de baixo para cima.

**Note que quando a direção é em coluna, justify-content muda para a vertical e align-items para a horizontal.**

Às vezes, reverter a ordem de uma coluna ou de um container não é suficiente. Nesses casos, podemos aplicar a propriedade **order** para itens individuais. Por padrão, itens tem um valor de 0, mas nós podemos usar essa propriedade para alterar para um valor inteiro positivo ou negativo.

Outra propriedade que você pode aplicar para itens individuais é **align-self**. Esta propriedade aceita os mesmos valores que **align-items** e seus valores são usados para o item específico

**flex-wrap** define se os itens flexíveis são forçados a ficarem na mesma linha ou se podem ser quebradas em varias linhas, essa propriedade recebe os seguintes valores:

* **nowrap**: Todos os itens são apertados em uma única linha.
* **wrap**: Itens se separam em linhas adicionais.
* **wrap-reverse**: Itens se separam em linhas adicionais em reverso.

Você pode usar **align-content** para definir como múltiplas linhas devem ser espaçadas uma das outras. Essa propriedade recebe os seguintes valores:

* **flex-start**: Linhas são agrupadas no topo do container.
* **flex-end**: Linhas são agrupadas no fundo do container.
* **center**: Linhas são agrupadas no centro vertical do container.
* **space-between**: Linhas são posicionadas com espaço igual entre elas.
* **space-around**: Linhas são posicionadas com espaço igual em torno delas.
* **stretch**: Linhas se esticam para preencher o container.

Isso pode ser confuso, mas **align-content** determina o espaçamento entre linhas, enquanto **align-items** determina como as linhas como um todo são alinhadas dentro do container. Quando há só uma linha, **align-content** não tem nenhum efeito.