

## Flexbox

É uma ferramenta de CSS utilizado para projetar layout, para utilizar o flexbox é necessário ter um container que é uma tag onde as outras tags serão inseridas, e o conteúdo do container chamamos de items.

No exemplo vamos criar um container com uma div e colocar 4 itens que serão os 4 parágrafos

```
<body>
  <div class="container">
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Repellat quis iure ab vitae aliquam autem! Explicabo repellat hic magnam optio aperiam facilis ducimus. Error, tenetur pariatur. Incidunt, consectetur? Perferendis, aliquam?</p>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit beatae ullam voluptatibus nam iure quaerat quibusdam consequatur quis fugit voluptates? Cupiditate a incidunt consectetur debitis perferendis aspernatur nesciunt consequuntur in.</p>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti praesentium blanditiis possimus ex cupiditate ad cumque fuga, aliquid excepturi illum dicta, tempore, placeat soluta quam debitis vitae ab eum perferendis.</p>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Beatae, earum deserunt ipsum quo consectetur voluptatibus, ea magni explicabo temporibus obcaecati, eaque quidem. Error dolorum dolores unde veniam non vero fuga!</p>
  </div>
</body>
```

Para começarmos a utilizar o flexbox, devemos alterar a propriedade display do container para flex.

```
.container{
  display: flex;
}
```

Automaticamente todos os itens ficarão um ao lado do outro.

Seiya de Pégaso, o protagonista destemido e leal. Seiya é conhecido por sua armadura de bronze representando a constelação de Pégaso, e ele é um dos defensores mais fortes da Deusa Atena, sempre disposto a lutar pelo bem e proteger os fracos.

Outro destaque é Shiryu de Dragão, um Cavaleiro que incorpora a virtude da paciência e da sabedoria. Sua armadura de Dragão é uma das mais resistentes e é conhecida por sua invulnerabilidade. Shiryu é mestre na técnica do "Cólera do Dragão", um golpe poderoso que pode superar até mesmo os inimigos mais fortes.

Hyoga de Cisne é outro notável Cavaleiro de Bronze, cujo poder é inspirado na constelação de Cisne. Ele domina técnicas de controle do gelo, uma habilidade que pode congelar qualquer coisa em seu caminho. Hyoga é conhecido por sua personalidade calma e determinação inabalável, e suas técnicas frias e calculistas o tornam um adversário formidável.

Ikki de Fênix, um Cavaleiro que muitas vezes é um anti-herói. Seu espírito indomável e temperamento explosivo são refletidos em suas habilidades ardentes relacionadas à Fênix. Ikki é frequentemente lembrado por seu passado trágico e sua capacidade de superar grandes desafios, tornando-o uma figura complexa e fascinante na história dos Cavaleiros do Zodíaco.

Existe uma série de propriedades que podem ser aplicadas tanto no container quanto nos itens que server para configurar o layout.

# Propriedades para o container

## flex-direction

Com essa propriedade é possível modificar a direção dos itens, o padrão é da esquerda para a direita.

**row(padrão):** Itens serão distribuídos da esquerda para a direita.

**row-reverse:** Itens serão distribuídos da direita para esquerda.

**column:** Distribui o conteúdo de cima para baixo.

**column-reverse:** Distribui o conteúdo de baixo para cima.

## Flex-wrap

Todos os itens, por padrão são distribuídos em uma única linha independente da largura da tela, a propriedade flex-wrap permite quebrar o conteúdo em mais linhas caso a largura não seja o suficiente.

**nowrap(padrão):** Todos os itens flexíveis estarão em uma linha

**wrap:** Os itens flexíveis serão agrupados em várias linhas, de cima para baixo.

**wrap-reverse:** Os itens flexíveis serão agrupados em várias linhas de baixo para cima.

## Flex-flow

Essa propriedade é um “atalho” das propriedades flex-direction e flex wrap, basta colocar os respectivos valores, exemplo.

```
flex-flow: column wrap;
```

## Justify-content

Essa propriedade define como os itens serão distribuídos dentro da linha quando há espaço sobrando entre os itens, por padrão todo conteúdo fica alojado do lado esquerdo.

**flex-start(padão):** Os itens são compactados no lado esquerdo do container.

**flex-end:** Os itens são compactados no lado direito do container.

**center:** Os itens são centralizados ao longo da linha

**space-between:** Os itens são distribuídos uniformemente na linha; o primeiro item está na linha inicial, o último item na linha final.

**space-around:** Os itens são distribuídos uniformemente na linha com espaço igual ao seu redor.

**space-evenly:** Os itens são distribuídos de forma que o espaçamento entre dois itens quaisquer (e o espaço até as bordas) seja igual.

### **Align-items**

Essa propriedade irá alinhar o conteúdo na vertical, para isso a altura dos itens deve ser diferente para fazer sentido o seu uso.

**stretch(padão):** Esticar para preencher o contêiner (ainda respeitar largura mínima/largura máxima)

**flex-start/ start/ self-start:** Os itens são colocados no topo do container.

**flex-end/ end/ self-end:** Os itens são colocados no final do container.

**center:** Os itens são centralizados na vertical.

**baseline:** Os itens são alinhados, assim como suas linhas de base se alinham.

### **Align-content**

Essa propriedade tem a função de determinar como os itens na vertical serão distribuídos e só funciona se a propriedade flex-wrap estiver com o valor wrap e a altura do container deve ser maior que a altura do conteúdo, para possibilitar mais de uma linha.

**normal(padão):** Os itens são empacotados em sua posição padrão como se nenhum valor tivesse sido definido.

**flex-start/ start:** Itens alojados na parte superior do container

**flex-end/ end:** Itens alojados na parte de baixo do container container.

**center:** Itens centralizados no contêiner

**space-between:** Itens distribuídos uniformemente; a primeira linha está no início do contêiner enquanto a última está no final

**space-around:** Itens distribuídos uniformemente com espaço igual em torno de cada linha

**space-evenly:** Os itens são distribuídos uniformemente com espaço igual ao seu redor.

**stretch:** As linhas se esticam para ocupar o espaço restante.

## Propriedades para os itens do contêiner

Essas propriedades devem ser usadas no seletor dos itens que estão dentro do container.

### Order

Por padrão os itens são distribuídos pela sequência que está o código, o valor do order padrão é zero para todos os itens, para utilizar coloque em qual posição se deseja colocar o item.

### Flex-grow

É a habilidade de um flex item aumentar de tamanho caso tenha espaço sobrando dentro do container. Por padrão o valor é zero o que indica que a largura do item vai de acordo com o seu conteúdo, se todos os itens tiverem o valor 1 todos os elementos irão ocupar a largura total do container distribuindo a “sobra” da largura, quanto maior o valor do flex-grow, mais espaço aquele item terá acesso.

### Flex-basis

Mesmo utilizando o flex-grow, a distribuição do espaço entre os itens é desigual, pois o item com mais conteúdo será maior mesmo que todos tenham o mesmo flex-grow, para resolver este problema usamos a propriedade flex-basis.

O valor padrão para essa propriedade é o auto, ele faz com que a largura da base seja igual para todos os itens, ou seja, o item que tiver mais conteúdo ficará maior.

Para resolver esse problema colocamos o valor zero, isso fará que todos os elementos tenham a mesma largura, normalmente usamos o basis para isso, mas também é possível definir outros valores

### **flex-shrink**

É a capacidade de determinar se um item pode ser reduzido para caber dentro de um container o valor padrão é 1 onde indica que o item pode ser diminuído para caber no container.

se o valor for 0 isso irá indicar que o item não irá reduzir de tamanho, e caso coloque outro valor é o número de vezes que o item irá reduzir para encaixar no container.

### **Flex**

É uma abreviação para as propriedades flex-grow flex-shrink e flex-basis nesta mesma ordem.

Geralmente você verá a propriedade flex nos flex itens ao invés de cada um dos valores separados.