**FlexBox**

- Objetivo dele é oferecer distribuição de espaço entre itens em uma interface

**Flex Container**

- Ele tem filhos, ele envolve os itens. Nela tem o display:flex

**Propriedades relacionadas:**

* **Display** - Inicializados
* **Flex-direction** - Direcionamento em linha ou coluna
* **Flex-wrap** - Quebra de linha ou nao
* **Flex-flow** - Abreviação de direction e wrap
* **justify-content** - Ajusta de acordo com sua direção
* **align-items** - Ajusta de acordo com seu eixo
* **align-content** - Alinhar as linhas

**Flex Item**

- Filho do Container, se você colocar um display ele tmb é um flex container

**Propriedades relacionadas:**

* **Flex-grow** - Define o fator de crescimento
* **Flex-Basis** - Tamanho inicial
* **Flex-shrink** - Capacidade de redução
* **order** - Ordem de distribuição
* **align self**- Alinhamento de item especifico do container

**Display Flex**

Torna a tag um elemento do tipo flex, assim todos seus filhos viram flex tmb.

**Flex Direction**

Estabelece o eixo principal do container, definindo assim a direção que os flex itens são colocados no flex container

**Eixos**:

**Row** (Padrão) em linha e da esquerda pra direita

**Row-Reverse** sentido oposto, da direita pra esquerda

**Column** em coluna, de cima pra baixo

**Column-Reverse** sentido oposto, de baixo pra cima

**Flex Wrap**

Define se os itens devem ou não quebrar a linha

* **NoWrap** - Padrão, não permite a quebra de linha
* **Wrap** - Permite a quebra de linha assim que um dos flex itens não puder ser mais compactado
* **Wrap-reverse** - Permite a quebra mas no sentido contrario, a quebra ocorre pra cima ao invés de pra baixo

**Flex Flow**

Atalho para as propriedades Direction e Wrap, o uso dele não é tão comum.

**Justify Content**

Vai se encarregar de alinhar os itens dentro do container de acordo com a direção pretendida e tratar da distribuição dos itens

* **Flex-start** - Início do Container
* **Flex-end** - Final do Container
* **Center** - Todos ao centro
* **Space-between** - Cria um espaço igual entre os elementos
* **Space-around** - Os espaçamento do meio são duas vezes maiores que o inicial e o final

**Align Items**

Trata do alinhamento dos flex itens de acordo com o eixo do container.

Diferença entre **Align items e Justify Content** é que o Justify sempre pega o tamanho do container pra se basear

**Tipos de alinhamento:**

* **Center** - Trás ao centro
* **Stretch** - Padrão, cresçam igualmente. De acordo com o maior conteudo
* **Flex-start** - Alinhamento dos itens no inicio
* **Flex-end** - Alinhamento dos itens no final
* **Baseline** - Alinhamento de acordo com a linha base da tipografia dos itens

**Align Content**

Responsável pelo tratamento do eixo vertical, trata o alinhamento do container

Container precisa utilizar quebra de linhas e a altura do container seja maior que a soma das linhas do itens

**Tipos de alinhamento:**

* **Center** - Trás ao centro
* **Stretch** - Padrão, cresçam igualmente. De acordo com o maior conteúdo
* **Flex-start** - Alinhamento dos itens no início
* **Flex-end** - Alinhamento dos itens no final
* **Space Between** - Cria um espaçamento igual entre os elementos
* **Space Around** - Espaçamentos do meio são duas vezes maiores que o inicio e final

**Flex-Grow**

Define a proporcionalidade de crescimento dos itens. Não ira funcionar caso tenhamos adicionado justify-content ao nosso flex container

**Flex-basis**

Estabelece o tamanho inicial dos itens antes da distribuição de espaço restante dentro dele

Valores possíveis

* **auto** - caso o item não tenha tamanho pré definido
* **px, %, em, ...** - valores exatos previamente definidos
* **0** - terá relação com a definição do flex-grow

**Flex-Shrink**

Estabelece a capacidade de redução ou compressão do tamanho de um item

Na quebra de linha eu perco meu grow

**Flex**

Atalho ou abreviação do Grow, Shrink e basis

**Align self**

Propriedade que estabelece o alinhamento de modo individual sobre determinado item

Valores possíveis

* **Auto** - Valor padrão, irá respeita a definição de align items
* **Start** - início do container
* **End** - final do container
* **stretch** - ocupa todo o espaço relativo
* **baseline** - utiliza a linha base da tipografia
* **Center** - Relativo ao centro de acordo com o eixo