IOS – Instituto de Oportunidade Social

CSS 11 - FlexBox

Temas abordados

- Compreender a criação de diferentes **layouts** com o uso do **flexbox**;

- Conhecer as diversas **propriedades** do flexbox;

- Aplicar os recursos do flexbox nas folhas de **estilo**.

IOS – Instituto de Oportunidade Social

FlexBox Align

FlexBox Align

CSS Flexible Box Layout

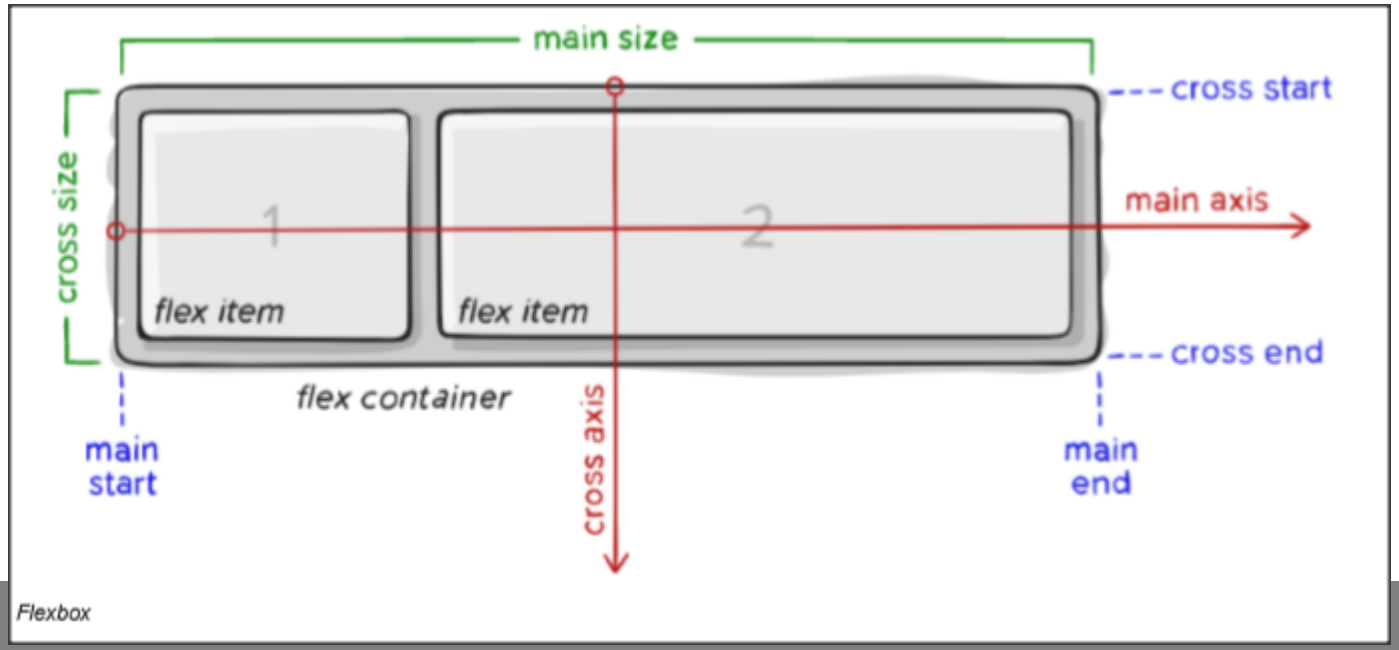
Você tem usado a propriedade **float** do CSS para criar página web com duas ou três colunas, o que é uma técnica muito comum para esse tipo de organização no seu site. Porém, existe uma **técnica lançada em 2017 pela W3C** que oferece uma maneira mais eficiente de **dispor, alinhar e distribuir o espaço** entre os elementos em uma página web, essa técnica de chama **CSS Flexible Box Layout** ou apenas **Flexbox**.

O **objetivo** do Flexbox é propor um **layout flexível**, no qual os elementos estão contidos em um **container flexível** e podem ser configurados em uma dimensão (horizontal ou vertical) de uma maneira bem prática e simples. A ideia principal por trás do layout flexível é dar ao container **flexibilidade** para alterar a **largura e altura** dos itens e ordená-los para melhor encaixá-los no **espaço disponível**.

FlexBox Align

O **container flex** expande os itens para **preencher** o **espaço livre disponível** ou **encolhe-os** para evitar **overflow** (overflow em HTML é o conteúdo **transbordar**, ou seja, ocupar um espaço maior do que o visível, aparecendo as **barras de rolagem**).

O **Flexbox** é um **módulo no CSS** que inclui um conjunto de **propriedades**. Algumas estão atreladas ao **elemento pai (container)** e outras aos **elementos filhos (itens)**. A imagem abaixo mostra a **ideia principal** por trás do flexbox.



FlexBox Align

Os itens serão colocados seguindo o **main axis** (partindo do **main-start** até o **main-end**) ou, também, podem ser colocados seguindo o **cross axis** (partindo do **cross-start** até o **cross-end**). Vamos explicar cada uma dessas nomenclaturas.

- **main axis**: o main axis de um container flexível é o **eixo principal** ao longo do qual os itens flexíveis serão dispostos. **Cuidado, não é necessariamente horizontal**; depende da propriedade **flex-direction**.

- **main-star** | **main-end**: os itens flexíveis são colocados dentro do container começando a partir do main-star e indo até o main-end.

- **main-size**: a **largura ou altura** de um item flexível, que estiver na **dimensão principal**, é o main-size do item. O main-size de um item flexível pode ser a propriedade **‘width’** ou **‘height’** do item, pois depende de como foi configurado a propriedade **flex-direction**.

FlexBox Align

Nomenclaturas de Cross:

- **cross-axis**: o **eixo perpendicular** ao eixo principal é chamado cross-axis. Sua direção depende da direção do eixo principal.

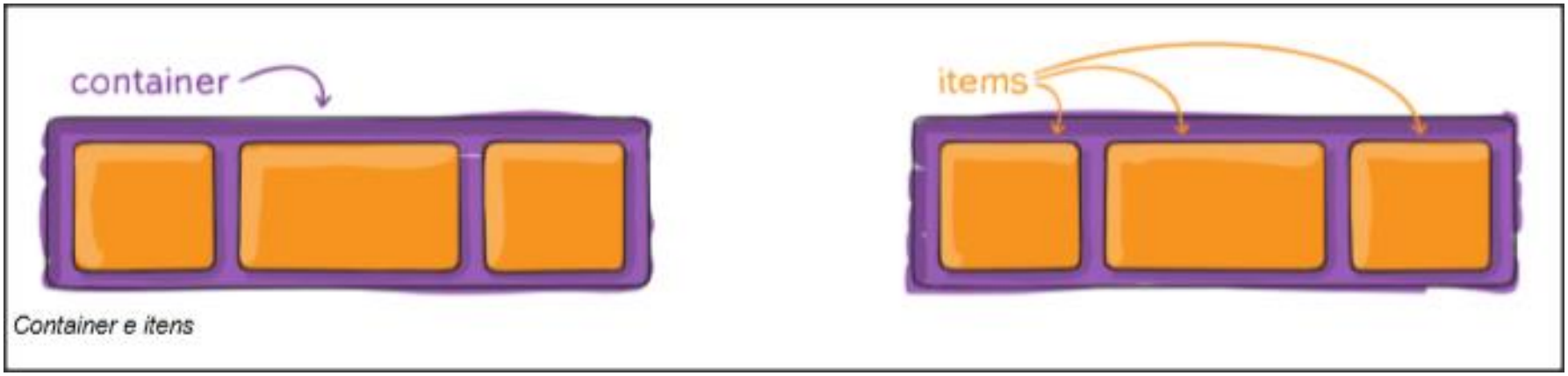
- **cross-start** | **cross-end**: as linhas flexíveis são preenchidas com itens e colocadas no container, começando no cross-start e indo até o cross-end.

- **cross-size**: A **largura ou altura** de um item flexível é o **cross-size**. O cross-size de um item flexível pode ser a propriedade **‘width’** ou **‘height’** do item, pois depende de como foi configurado a propriedade **flex-direction**.

FlexBox Align

Propriedades do flexbox

Temos propriedades relacionadas ao **elemento pai**. Ou seja, **container** e propriedades relacionados aos **elementos filhos**, também chamados de **itens**.



Container e itens

Vamos começar pelas **propriedades do container** do flexbox. Elas são **display, flex-direction, flex-wrap, flex-flow, justify-content, align-items e align-content.**

FlexBox Align

Propriedade display

A propriedade **display** pode definir um container como **flexível, inline ou block** dependendo do valor que é dado. Portanto ela **habilita o contexto flex** para **todos os itens** dentro do **container**.

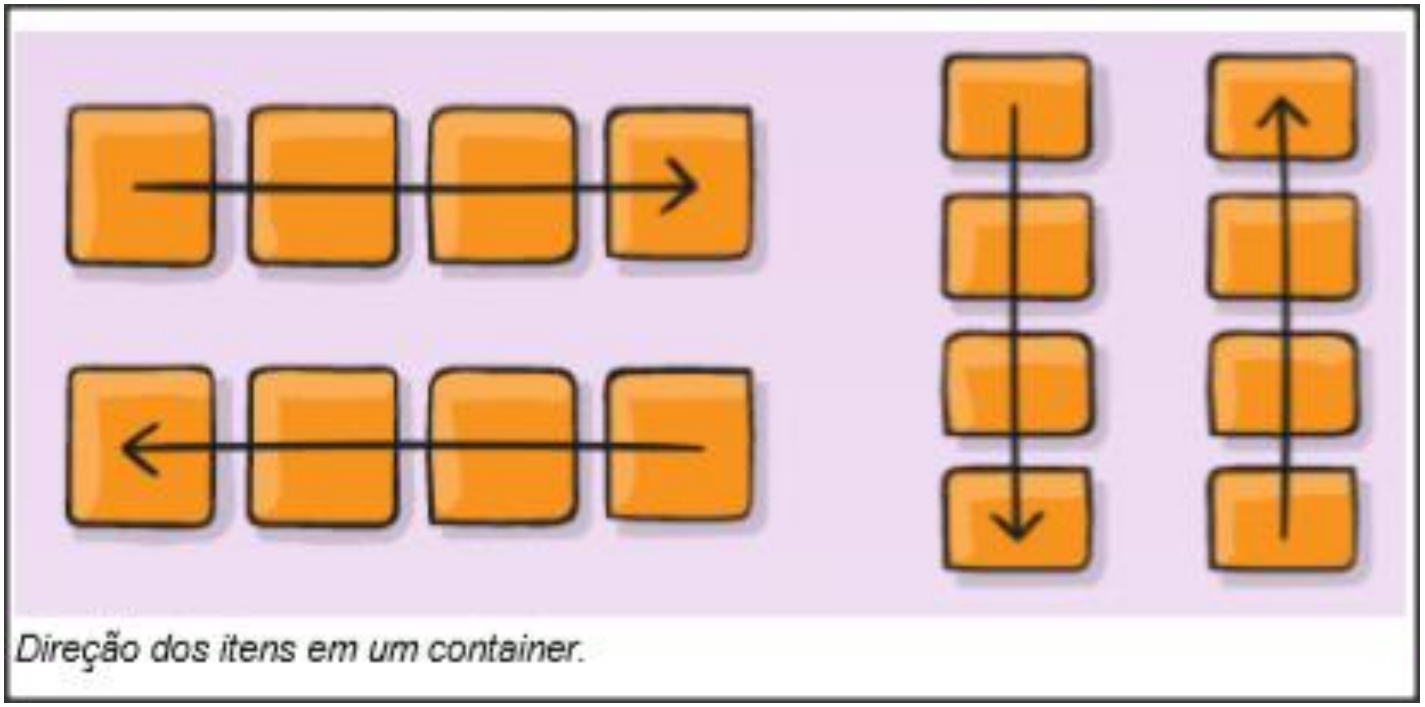
```
@charset "utf-8";

.container {
        display: flex;
}
```

FlexBox Align

Propriedade flex-direction

A propriedade **flex-direction** configura a **direção dos itens** flexíveis, ou seja, ela estabelece o **main-axis** do **container**.

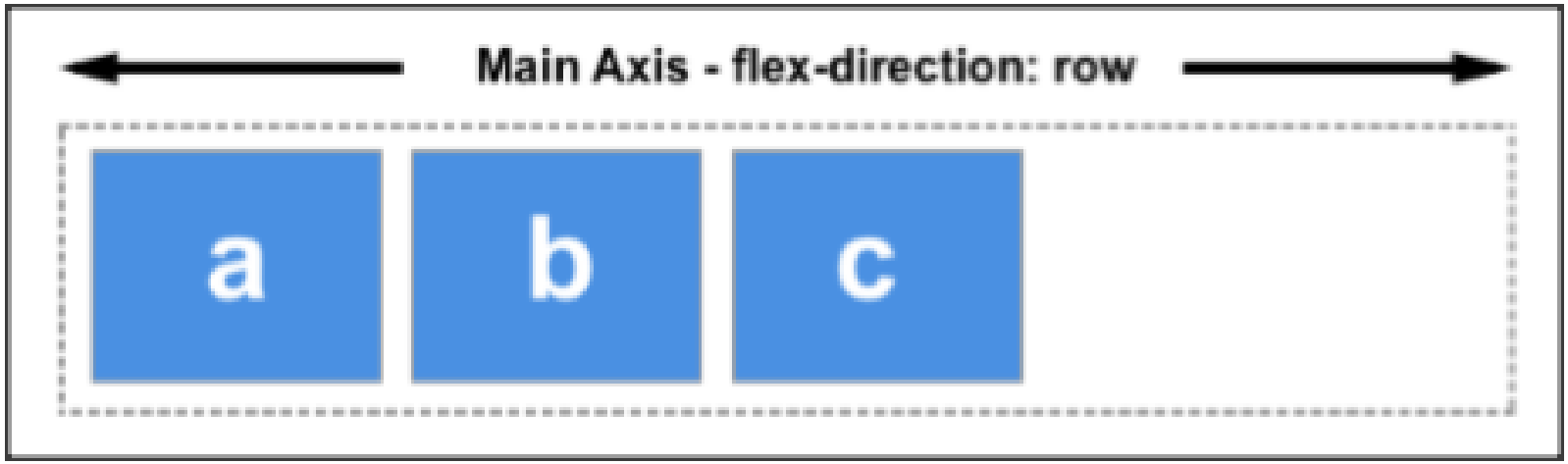


Direção dos itens em um container.

FlexBox Align

Os **valores possíveis** para configurar essa propriedade são:

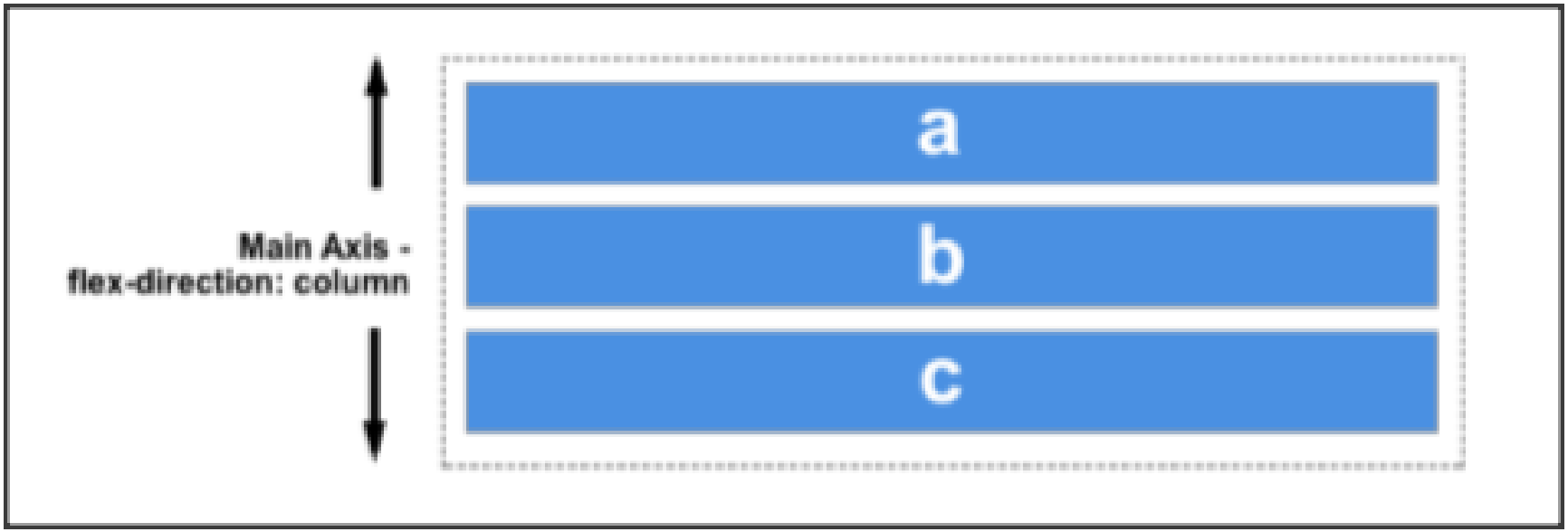
- **row** (default): da **esquerda para a direita** se a propriedade **direction** estiver com o **valor padrão ltr** (**left-to-right**, é o padrão do HTML). Ou da **direita para a esquerda** se a propriedade direction estiver com o valor **rtl** (**right-to-left**).



- **row-reverse**: da **direita para a esquerda** se a propriedade **direction** estiver com o **valor padrão ltr** (**left-to-right**, é o padrão do HTML). Ou da **esquerda para a direita** se a propriedade direction estiver configura com o valor **rtl** (**right-to-left**)

FlexBox Align

- **column**: de **cima para baixo** se a propriedade **direction** estiver com o valor **padrão ltr** (**left-to-right**, é o padrão dos documentos HTML). Ou de **baixo para cima** se a propriedade direction estiver configura com o valor **rtl** (**right-to-left**).

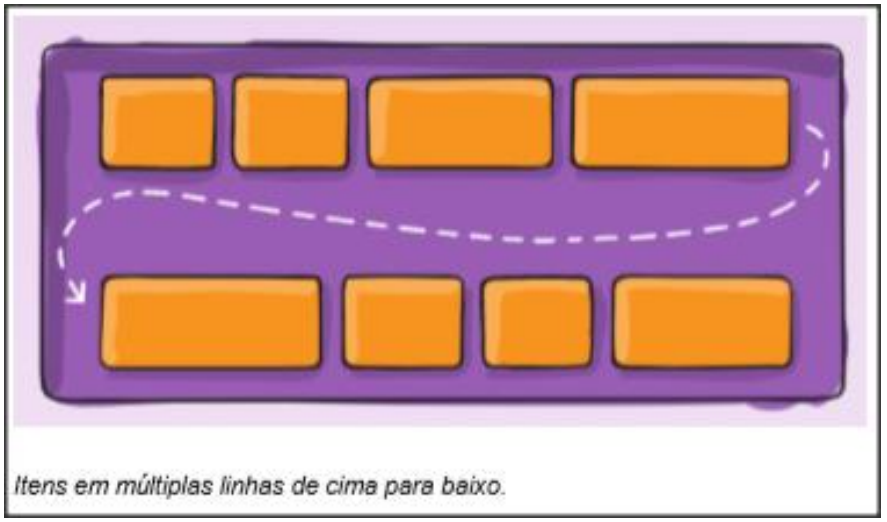


- **column-reverse**: de **baixo para cima** se a propriedade **direction** estiver com o valor padrão **ltr** (**left-to-right**, é o padrão do HTML). Ou de **cima para baixo** se a propriedade direction estiver configura com o valor **rtl** (**right-to-left**)

FlexBox Align

Propriedade flex-wrap

A propriedade **flex-wrap** configura se os itens serão exibidos em **múltiplas linhas**. Por **padrão**, os itens irão tentar encaixar em uma **única linha**, pois o valor **default** é **nowrap**. Você precisa modificar essa propriedade, caso precise exibir os itens em mais de uma linha.



Itens em múltiplas linhas de cima para baixo.

As configurações dessa propriedade são:

- **nowrap** (default): todos os itens serão dispostos em uma única linha.
- **wrap**: os itens flex serão dispostos em múltiplas linhas de cima para baixo.
- **wrap-reverse**: os itens serão dispostos em múltiplas linhas de baixo para cima.

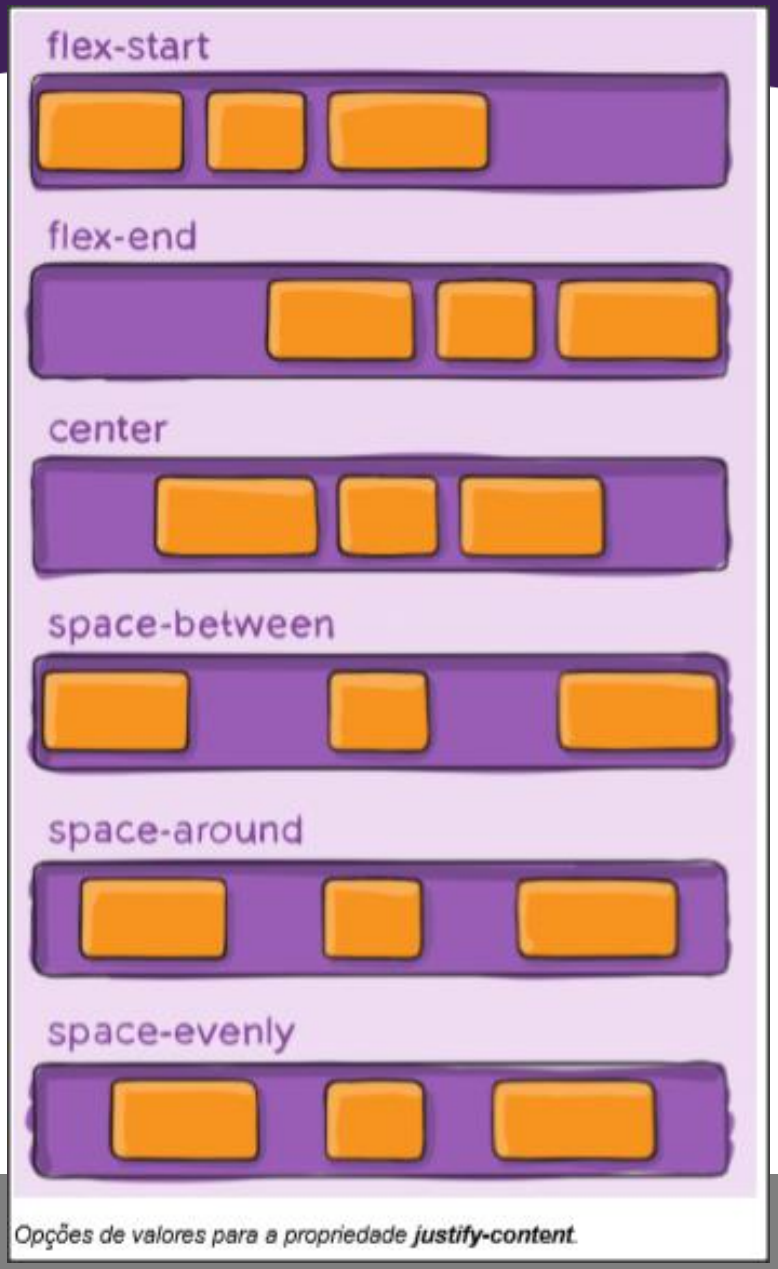
FlexBox Align

Propriedade flex-flow

Essa é uma abreviação das propriedades **flex-direction** e **flex-wrap**. Com uma única instrução você pode configurar a direção e a múltiplas linhas de uma vez.

```
.container {
        flex-flow: column wrap;
}
```

FlexBox Align



*Opções de valores para a propriedade **justify-content**.*

Propriedade justify-content

A propriedade **justify-content** configura como o navegador irá exibir qualquer **espaço extra** que possa existir no container, ou seja, ela define o alinhamento ao longo do **main-axis** do container.

FlexBox Align

Os valores possíveis para configurar essa propriedade são:

- **flex-start** (default): os itens são exibidos alinhados no **início** do flex-direction.
- **flex-end**: os itens são exibidos alinhados no **final** do flex-direction.
- **start**: os itens são exibidos alinhados no **início** da direção do **modo de escrita**.
- **end**: os itens são exibidos alinhados no **final** da direção do **modo de escrita**.
- **left**: itens são exibidos alinhados em direção da **borda esquerda do container***
- **right**: itens são exibidos alinhados em direção da **borda direita do container***

* (a menos que isso não faça sentido com o **flex-direction**, então se comporta como o valor **start**).

FlexBox Align

- **center**: os itens são **centralizados** ao longo da linha.

- **space-between**: os itens são **distribuídos uniformemente na linha**. Ou seja, o **primeiro item** está no **início** da linha e o **último** está no **final** da linha.

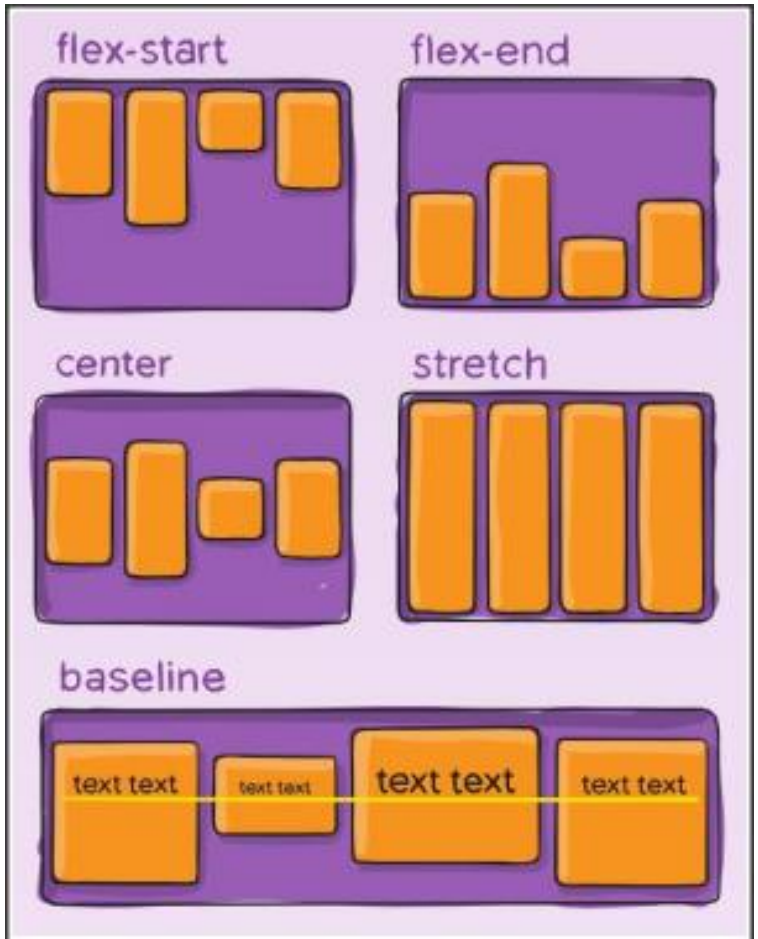
- **space-around**: os itens são **distribuídos uniformemente na linha** com tamanhos de **espaços iguais ao redor do item** (na **direita** e **esquerda**). Note que **visualmente os espaços não são iguais**, uma vez que **todos os itens têm espaço igual em ambos os lados**. O **primeiro item** terá **uma unidade de espaço** contra a borda do contêiner, mas **duas unidades** de espaço entre o **próximo item**, porque esse **próximo item** tem seu **próprio espaçamento** que se aplica.

- **space-evenly**: os itens são distribuídos de forma que o **espaçamento entre quaisquer dois itens seja igual**.

FlexBox Align

Propriedade align-itens

A propriedade **align-itens** define o comportamento padrão de como os itens flex estão dispostos ao longo do **cross-axis**. Ele faz a mesma função para o cross-axis, quanto o **justify-content** para o **main-axis**.



Propriedade align-items

FlexBox Align

Os valores possíveis para configurar essa propriedade são:

stretch (default): esticar os itens para preencher todo o container (respeitando ainda a minwidth/max-width)

flex-start / start / self-start: os itens são colocados no início do cross-axis. A diferença entre eles é sutil e diz respeito às regras do flex-direction e do writing-mode.

flex-end / end / self-end: os itens são colocados no final do cross-axis. A diferença entre eles é sutil e diz respeito às regras do flex-direction e do writing-mode.

center: os itens são centralizados no cross-axis.

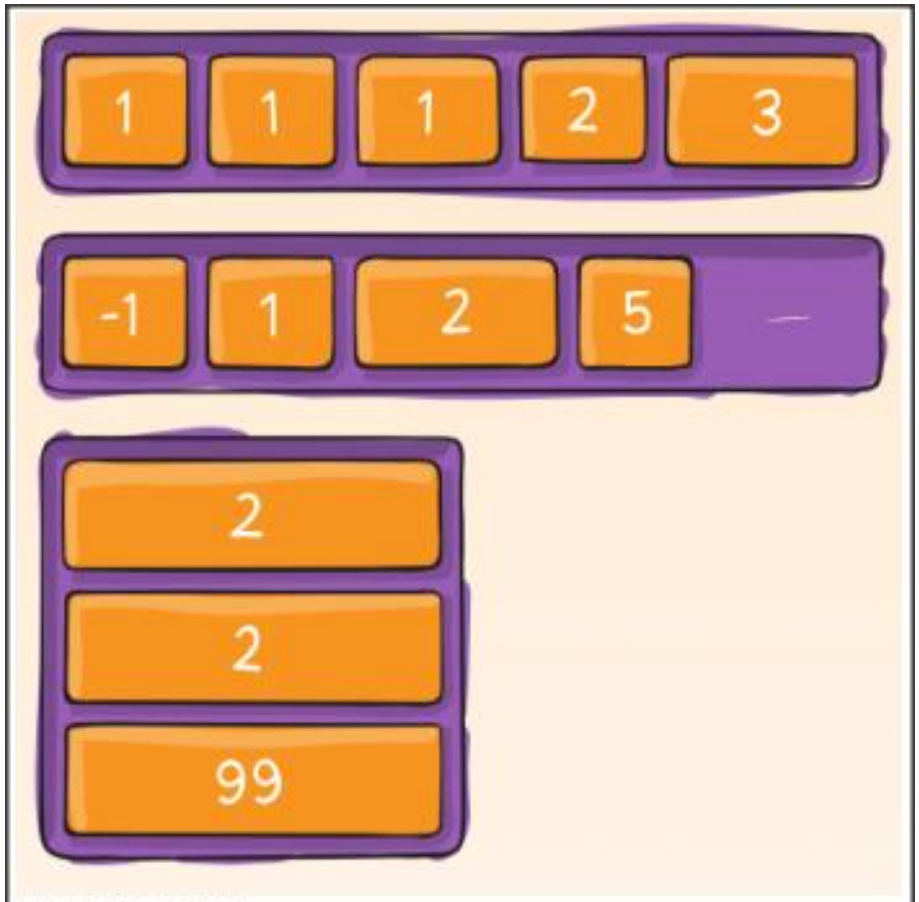
baseline: os itens são alinhados assim como suas linhas base são alinhadas.

FlexBox Align

Propriedade order

A propriedade **order** controla a ordem onde cada item deverá aparecer no container. Você pode por exemplo alterar a ordem de um item por meio de uma classe.

```
.item {
        order: 5;
}
```



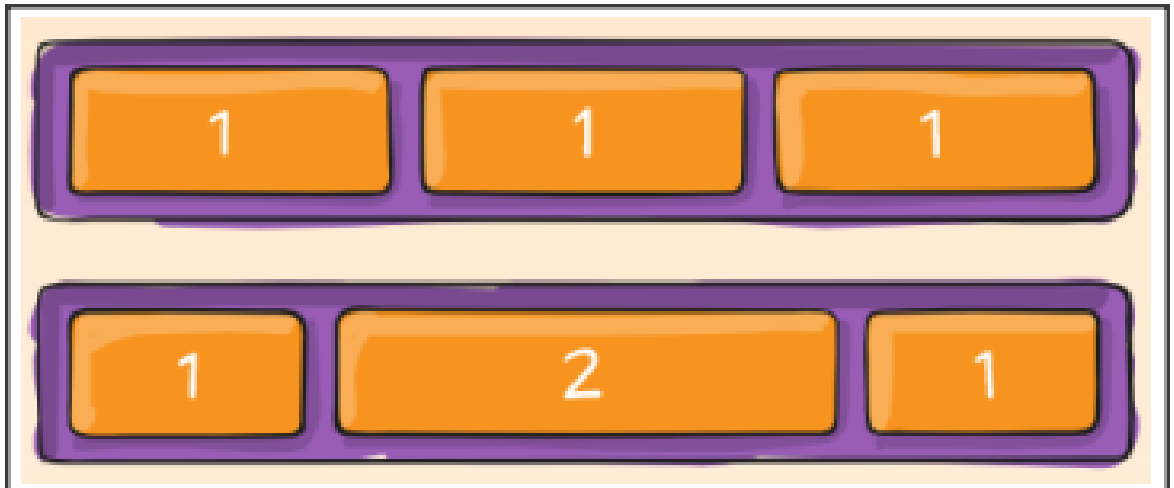
Propriedade order.

FlexBox Align

Propriedade flex-grow

A propriedade **flex-grow** define a capacidade de cada item flex. Ela determina a quantidade de espaço do container que o item deve ocupar, de acordo com o valor total. Ex: Se todos os itens tiverem definido como 1, o espaço no container será distribuído igualmente para todos os filhos. Se um dos filhos tiver o valor 2, o espaço restante ocupará o dobro do espaço dos outros.

```
.item {
        flex-grow: 4;
        /* default 0 */
}
```



Propriedade flex-grow.

FlexBox Align

Propriedade flex-shrink

A propriedade **flex-shrink** define a capacidade de um item flexível encolher, se necessário.

```
.item {
        flex-shrink: 3;
        /* default 1 */
}
```

FlexBox Align

Propriedade flex-basis

A propriedade **flex-basis** define o tamanho padrão de um elemento antes do espaço restante seja distribuído. Pode ser um tamanho (ex. 20%, 5rem, etc) ou uma keyword. A keyword auto significa “verifique a minha propriedade **width** e **height**”. A keyword **content** significa “dimensionar com base no conteúdo do item”. Existem outras Keywords, mas nesse momento iremos abordar apenas as **principais**.

```
.item {
        flex-basis: auto;
        /* default auto */
}
```

FlexBox Align

Propriedade flex

A propriedade **flex** uma abreviação das propriedades **flex-grow**, **flex-shrink** e **flex-basis**, ou seja, com uma única instrução você pode configurar a capacidade de um item flex crescer ou diminuir e o tamanho padrão de um elemento.

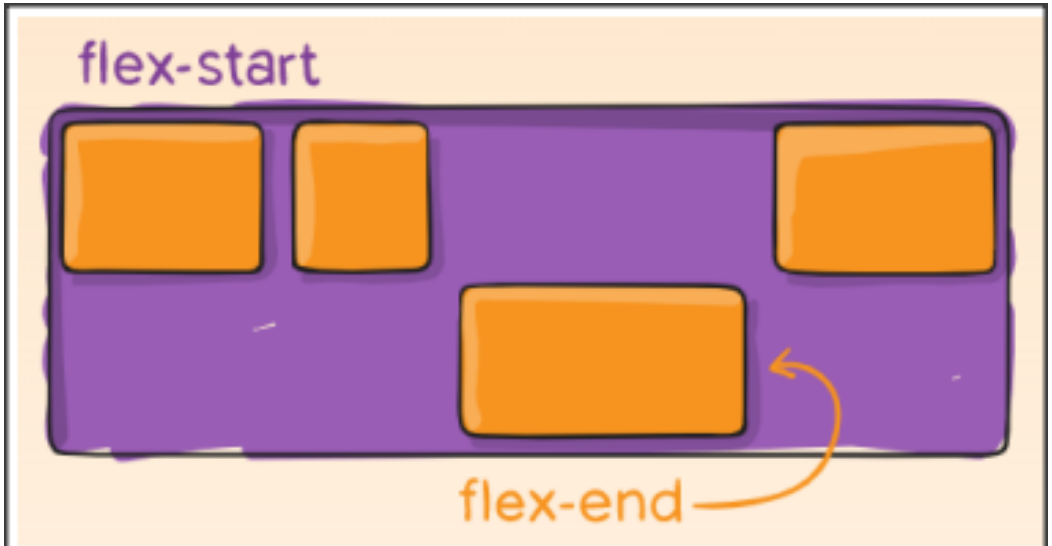
```
.item {
        flex: [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'>];
}
```

FlexBox Align

Propriedade align-self

A propriedade **align-self** permite que o alinhamento padrão (ou aquele especificado por **align-itens**) seja substituído nos itens.

```
.item {
 align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```



Propriedade align-self.

FlexBox Align

```
<section class="container">
<div class="item-box-1">
        <h3>Box um</h3>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus voluptates
        dolore voluptatem, rerum eum omnis consequuntur iste libero fugit beatae sequi
        voluptas id deleniti quibusdam eius, ad recusandae.</p>
</div>
<div class="item-box-2">
        <h3>Box dois</h3>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus voluptates dolore
        voluptatem, rerum eum omnis consequuntur iste libero fugit beatae sequi voluptas id
        deleniti quibusdam eius, ad recusandae.</p>
</div>
<div class="item-box-3">
        <h3>Box três</h3>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint velit minus voluptates dolore
        voluptatem, rerum eum omnis consequuntur iste libero fugit beatae sequi voluptas id
        deleniti quibusdam eius, ad recusandae.</p>
 </div>
</section>
```

FlexBox Align

Página sem a formatação:

FlexBox Align

Aplicando display flex e bordas:

```
@charset "utf-8";

.container {
        display: flex;
}

.container div {
        border: 1px #ccc solid;
        padding: 10px;
}
```

FlexBox Align

Página com display flex e bordas:

FlexBox Align

Configurando propriedades flex e order:

```
.item-box-1 {
        flex: 2;
        order: 2; }

.item-box-2 {
        flex: 1;
        order: 1; }

.item-box-3 {
        flex: 1;
        order: 3; }
```

FlexBox Align

Página após as configurações flex e order:

IOS – Instituto de Oportunidade Social

Vamos Praticar

Vamos Praticar

Apostila de CSS:

- Flexbox e suas Propriedades

Páginas 132 a 136

OBS: Acompanhar o passo a passo com o instrutor

IOS – Instituto de Oportunidade Social

Exercícios

Exercícios

Montar uma página HTML com nome **flexboxAlign.html** com seu par **CSS** atendendo os seguintes critérios:

- Possuir 4 divs com **tamanhos** de **conteúdos diferentes**
- Possuir um contêiner (section) envolvendo as 4 divs
- Utilizar as propriedades **display: flex** e escolher um **align-items** para a section
- Definir propriedades **flex** e **order** com valores diferentes entre as divs
- Utilizar a propriedade **align-self** diferente da section para uma das divs
- Publicar no GitHub e enviar no Moodle