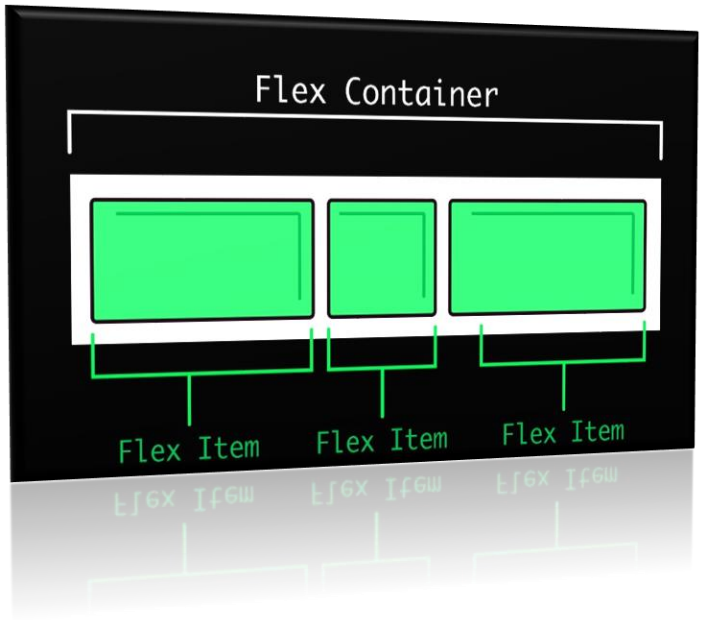
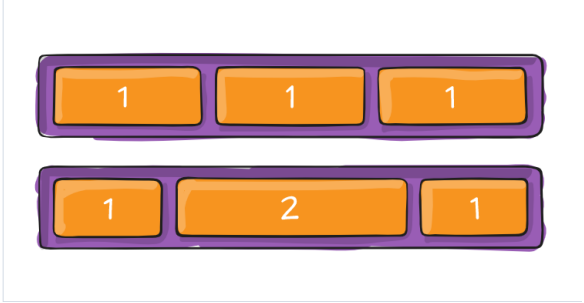
Autor: João vitor Quirino Sarti

GitHub: NULLBYTE-RGH (github.com)

CSS (Display): Flex



Flex-Grow:



A propriedade Grow, faz com que em uma determinada largura de container, determinada por exemplo por width, um dos blocos (Numerados na imagem de 1 a 3) tenham um crescimento maior em relação aos outros. Por padrão a propriedade tem valor 0:

```
flex-grow: 0;
```

Isso significa que todos os blocos estão em igualdade para dividir o espaço do container, porém, ao definir um valor de crescimento, o bloco que receber o valor acima de 0 terá prioridade em relação aos outros.

Se apenas um bloco tiver a prioridade de crescimento Flex-Grow, o valor Máximo que se pode definir é 1, significando que ele ocupara o espaço não ocupado pelos outros blocos. Já se tiver mais de um bloco utilizando a propriedade, aí sim pode-se utilizar valores acima de 1, fazendo

com que o bloco com maior valor tenha maior prioridade e consequentemente ocupe um espaço maior no container flex.

Por exemplo:

- 1° Linha tem-se grow padrão em todos, ou seja 0.
- 2° Linha, apenas o bloco 2 tem grow, 1.
- 3° Linha bloco 2 com Grow 1 e bloco 3 com grow 3

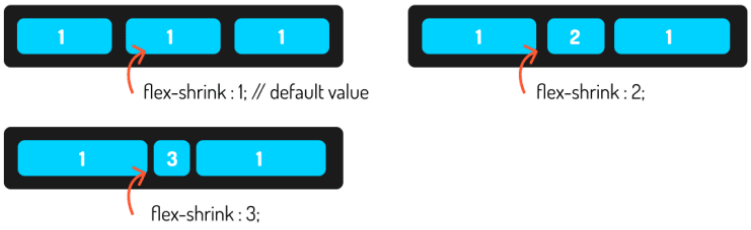
Sendo assim, nota-se claramente o efeito do grow, em blocos concorrentes por espaço.

Código para obter essa saída:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Flex</title>
</head>

<style>
    .container{
        display: flex;
        background-color:blueviolet;
        width:20%;
        justify-content: center;
    }

    .Padrao{
        border: 5px solid coral;
        margin: 6px;
        text-align: center;
    }
    .Titulo{
        display:flex;
        background-color:burlywood;
        justify-content: center;
```

```
        width:20%;
    }

    .Grow{
        flex-grow: 1;
    }

</style>
<body>


    <h2 class="Titulo">Grow</h2>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao">Bloco
1</div>
        <div style = "background-color:khaki;" class="Padrao">Bloco
2</div>
        <div style = "background-color:aquamarine;"
class="Padrao">Bloco 3</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao">Bloco
1</div>
        <div style = "background-color:khaki;" class="Padrao
Grow">Bloco 2 com Grow</div>
        <div style = "background-color:aquamarine;"
class="Padrao">Bloco 3</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue; flex-grow:0;"
class="Padrao">Bloco 1</div>
        <div style = "background-color:khaki;" class="Padrao
Grow">Bloco 2 Grow 1</div>
        <div style = "background-color:aquamarine; flex-grow:3;"
class="Padrao">Bloco 3 Grow 2</div>
    </div>

</body>
</html>
```

Flex-shrink:



O Shrink funciona de forma com que o valor padrão é 1:

```
flex-shrink : 1;
```

Isso significando que, todos os blocos tendem a se encolher a fim de permitir com que o tamanho do container seja respeitado e todos os blocos ocupem um tamanho igual dentro do container:

- Blocos de tamanho Iguais
- Respeitar o tamanho do container

Ao se mudar esse valor para 0

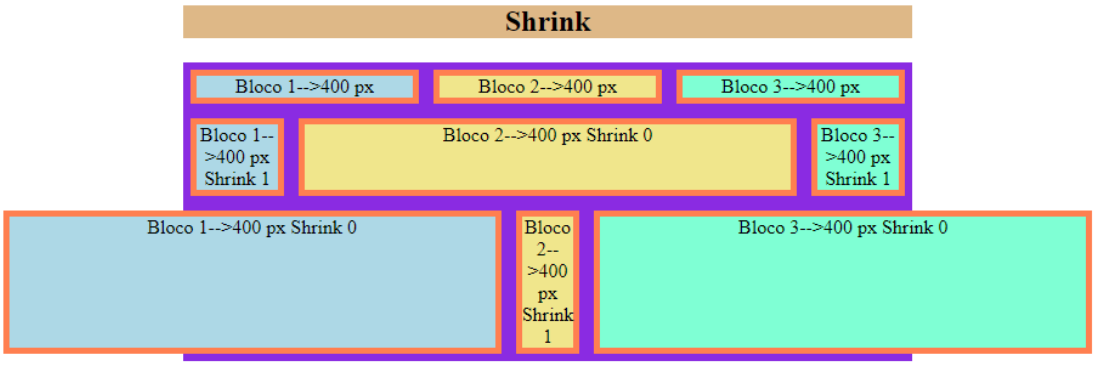
```
flex-shrink : 0;
```

O bloco que estiver com essa propriedade vai utilizar apenas a largura definida:

```
width:600px;
```

E não ligara para os blocos ao seu arredor.

Caso o valor do Shrink seja maior do que 1, cada vez mais o bloco com essa propriedade ira encolher para que outros blocos maiores possam ocupar mais espaço, na tentativa de atingir o espaço definido na largura original, WIDTH.



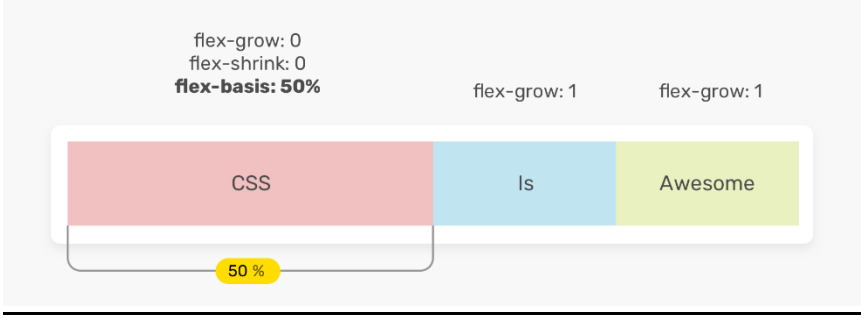
Nesse caso os blocos se estenderam tanto que até passaram o tamanho do container, isso seria resolvido com o Flex-Wrap adicionado ao container.

Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Flex</title>
</head>

<style>
    .container{
        display: flex;
        background-color:blueviolet;
        width:600px;
        justify-content: center;
        margin-left:500px;
    }

    .Padrao{
        border: 5px solid coral;
        margin: 6px;
        text-align: center;
        width: 400px;
        flex-shrink: 1;
    }
    .Titulo{
        display:flex;
        background-color:burlywood;
        justify-content: center;
        width:600px;
        margin-left:500px;
    }

    .Shrink{
        flex-shrink : 0;
    }


</style>
<body>


    <h2 class="Titulo">Shrink</h2>

    <div class="container">
```

```
        <div style = "background-color:lightblue" class="Padrao">Bloco
1-->400 px</div>
        <div style = "background-color:khaki;" class="Padrao">Bloco 2-
->400 px</div>
        <div style = "background-color:aquamarine;"
class="Padrao">Bloco 3-->400 px</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao">Bloco
1-->400 px Shrink 1</div>
        <div style = "background-color:khaki;" class="Padrao
Shrink">Bloco 2-->400 px Shrink 0</div>
        <div style = "background-color:aquamarine;"
class="Padrao">Bloco 3-->400 px Shrink 1</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
Shrink">Bloco 1-->400 px Shrink 0</div>
        <div style = "background-color:khaki;" class="Padrao">Bloco 2-
->400 px Shrink 1</div>
        <div style = "background-color:aquamarine;" class="Padrao
Shrink">Bloco 3-->400 px Shrink 0</div>
    </div>

</body>
</html>
```

Flex-basis:



O Basis tem como intuito de permitir ou não o uso das propriedades do bloco definidas como:

```
width:600px;
margin-left:500px;
padding:30px;
```

Essas propriedades fazem com que o uso do display Flex, não cumpra seu papel caso os blocos cresçam e precisem ser redimensionados, para isso a propriedade:

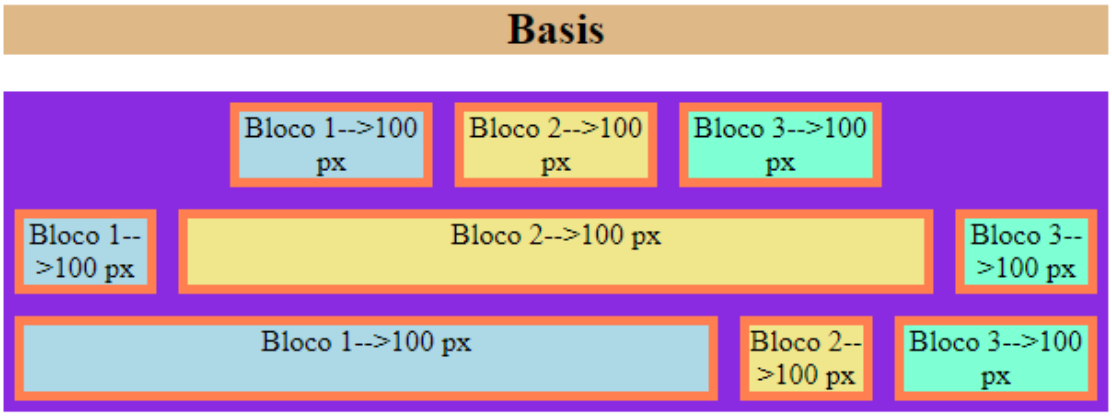
```
flex-basis: auto;
```

Permaneces por padrão em automático. Isso significa que ela tenta usar as propriedades do BoxModel, porém caso precise redimensionar ela assim o fará.

Agora, caso deseje que o bloco tenha uma largura inicial diferente da salva no WIDTH, o comando a ser usado é:

```
flex-basis: 600px;
```

Sendo assim a largura agora passa a ser 600px, e isso faz com que todas as outras caixas que têm o Basis em AUTO, respeitem essa medida da nova caixa.

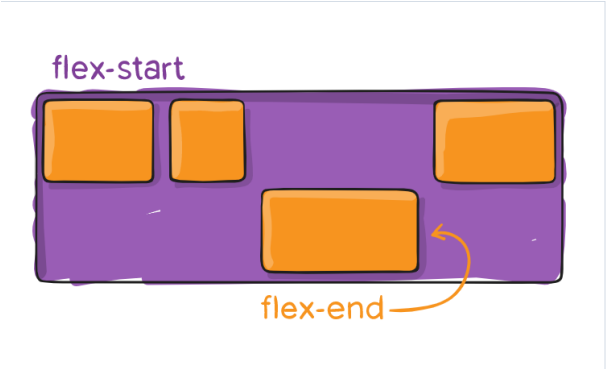


Codigo:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Flex</title>
</head>

<style>
    .container{
        display: flex;
        background-color:blueviolet;
        width:600px;
        justify-content: center;
        margin-left:500px;
    }

    .Padrao{
        border: 5px solid coral;
        margin: 6px;
        text-align: center;
        width: 100px;
        flex-shrink: 1;
    }
    .Titulo{
        display:flex;
        background-color:burlywood;
        justify-content: center;
        width:600px;
        margin-left:500px;
    }

    .Basis{
        flex-basis : 600px;
    }

    .BasisPadrao{
        flex-basis: auto;
    }

    .Shrink{
        flex-shrink: 0;
    }

</style>
```

```
<body>


    <h2 class="Titulo">Basis</h2>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
BasisPadrao">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao
BasisPadrao">Bloco 2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
BasisPadrao">Bloco 3-->100 px</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
BasisPadrao">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao Basis">Bloco
2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
BasisPadrao">Bloco 3-->100 px</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
Basis">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao
BasisPadrao">Bloco 2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
Shrink">Bloco 3-->100 px</div>
    </div>

</body>
</html>
```

Align-self:



A propriedade Align-Self:

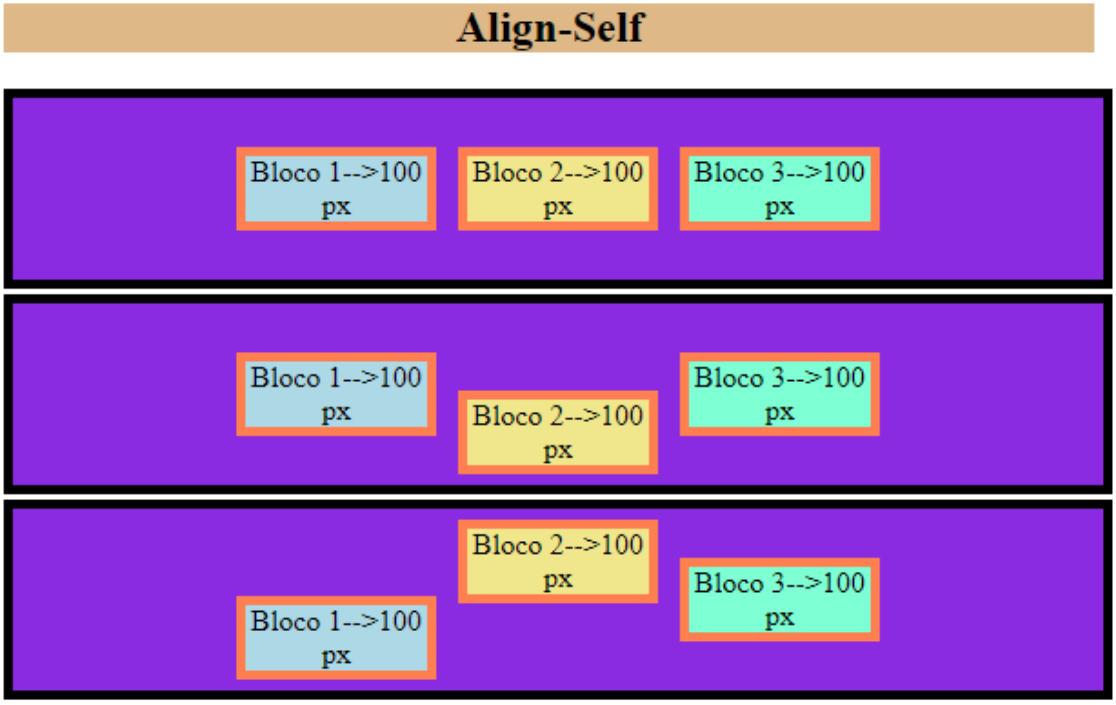
```
Align-Self: stretch;
```

Tem como função aplicar propriedades como:

```
align-items: stretch | flex-start | flex-end | center | baseline;
```

```
align-content: flex-start | flex-end | center | space-between | space-around | stretch;
```

Só que ao invés de aplicar ao contêiner, oque acarreta que todos os blocos vão se comportar da mesma maneira, ele aplica as mesmas propriedades a blocos individuais.



Codigo:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Flex</title>
</head>

<style>
    .container{
        display: flex;
        background-color:blueviolet;
        width:600px;
        height:100px;
        justify-content: center;
        margin-left:500px;
        border:5px solid black;
        margin-bottom :3px;
        align-items:center;
    }

    .Padrao{
        border: 5px solid coral;
        margin: 6px;
        text-align: center;
        width: 100px;
        flex-shrink: 1;
    }
    .Titulo{
        display:flex;
        background-color:burlywood;
        justify-content: center;
        width:600px;
        margin-left:500px;
    }

    .Align-Self1{
        Align-Self:flex-end;
    }

    .Align-Self1:hover{
        Align-Self:flex-start;
    }

    .Align-Self2{
```

```
        Align-Self:flex-start;
    }

    .Align-Self2:hover{
        Align-Self: stretch;
    }

</style>
<body>


    <h2 class="Titulo">Align-Self</h2>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao ">Bloco
2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
">Bloco 3-->100 px</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao
">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao  Align-
Self1">Bloco 2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
">Bloco 3-->100 px</div>
    </div>

    <div class="container">
        <div style = "background-color:lightblue" class="Padrao Align-
Self1">Bloco 1-->100 px</div>
        <div style = "background-color:khaki;" class="Padrao Align-
Self2">Bloco 2-->100 px</div>
        <div style = "background-color:aquamarine;" class="Padrao
">Bloco 3-->100 px</div>
    </div>

</body>
</html>
```

Sites recomendados:

flex-basis - CSS: Cascading Style Sheets | MDN (mozilla.org)

box-sizing - CSS: Cascading Style Sheets | MDN (mozilla.org)

Flexbox CSS: guia completo, elementos e exemplos | Alura