

```
<!DOCTYPE html>
<html lang="pt-br">
<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>justify-items</title>
<style>

    * {
        box-sizing: border-box;
        font-family: sans-serif;
        font-weight: bold;
    }

    body {
        margin: 0;
        padding: 1vw;
        width: 100vw;
        height: 100vh;
        display: flex;
        justify-content: space-between;
        align-items: center;
        flex-direction: column;
    }

    .container {
        width: 100%;
        height: 50%;
        padding: 5px;
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .bloco {
```

```
width: 16%;
height: 100%;
padding: 2px;
margin: 4px;
box-shadow: 10px 10px 10px rgba(0, 0, 0, .2);
display: flex;
justify-content: flex-end;
flex-direction: column;
}

.bloco > div {
  background-color: white;
}

.div1 {
  background-color: aqua;
  width: 30%;
  height: 50%;
  font-size: x-large;
  text-align: center;
}

.div2 {
  background-color: violet;
  width: 30%;
  height: 50%;
  font-size: x-large;
  text-align: center;
}

.div3 {
  background-color: rgb(248, 203, 121);
  width: 30%;
  height: 50%;
  font-size: x-large;
  text-align: center;
}
```

```
}

.div4 {
  background-color: turquoise;
  width: 30%;
  height: 50%;
  font-size: x-large;
  text-align: center;
}

#stretch {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: stretch;
}

#stretch > div {
  width: unset;
}

#flex-start {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: flex-start;
}

#flex-end {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: flex-end;
}
```

```
}

#center {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: center;
}

#baseline {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: baseline;
}
#baseline > div {
  width: unset;
}

#self-start {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: self-start;
}

#self-end {
  width: 100%;
  height: 80%;
  background-color: rgb(252, 248, 213);
  display: grid;
  justify-items: self-end;
}
```

```
#start {  
  width: 100%;  
  height: 80%;  
  background-color: rgb(252, 248, 213);  
  display: grid;  
  justify-items: start;  
}
```

```
#end {  
  width: 100%;  
  height: 80%;  
  background-color: rgb(252, 248, 213);  
  display: grid;  
  justify-items: end;  
}
```

```
#right {  
  width: 100%;  
  height: 80%;  
  background-color: rgb(252, 248, 213);  
  display: grid;  
  justify-items: right;  
}
```

```
#left {  
  width: 100%;  
  height: 80%;  
  background-color: rgb(252, 248, 213);  
  display: grid;  
  justify-items: left;  
}
```

</style>

</head>

<body>

<!--

Propriedade: justify-items (Justificação dos Itens)

Objetivo: Essa propriedade é usada para alinhar horizontalmente os nossos grid-items dentro dos eixos (ou linhas) que vão nos nossos containers grid.

Essa propriedade é equivalente a propriedade "justify-self" usada nos grid-items individualmente, porém, quando usamos "justify-items" nós especificamos um alinhamento para todos os grid-items de uma vez.

Essa propriedade deve ser usada sempre dentro de um container grid, não deve ser usada com containers flex-box!

DETALHE IMPORTANTE!!! Caso um grid-item já tenha um valor "justify-self" declarado ele irá sobrescrever o valor do nosso "justify-items".

- Valores Aceitáveis:

- legacy: Valor ainda não é reconhecido em alguns navegadores;
- legacy-right: Valor ainda não é reconhecido em alguns navegadores;
- legacy-left: Valor ainda não é reconhecido em alguns navegadores;
- legacy-center: Valor ainda não é reconhecido em alguns navegadores;
- normal: Segue o fluxo padrão do grid;
- stretch: Estica os grid-items se eles não tiverem tamanho pré definido onde o esticamento acontece;
- flex-start: Funciona em apenas alguns navegadores que já se adaptaram ao uso do flex como valor para grids (Já funciona no Chrome)!!! Alinha os grid-items no início de um eixo embutido do grid;

- flex-center: Funciona em apenas alguns navegadores que já se adaptaram ao uso do flex como valor para grids!!! Alinha os grid-items no centro de um eixo embutido do grid;
- flex-end: Funciona em apenas alguns navegadores que já se adaptaram ao uso do flex como valor para grids (Já funciona no Chrome)!!! Alinha os grid-items á partir de onde termina o eixo embutido do grid;
- center: Alinha os grid-items no centro do eixo do grid;
- baseline: Alinha os grid-items ao tamanho da maior letra dentro de um elemento;
- first-baseline: Alinha os grid-items ao tamanho da maior letra de um primeiro parágrafo - porém esse valor ainda é INVÁLIDO em alguns navegadores para esta propriedade;
- last-baseline: Alinha os grid-items ao tamanho da menor letra do último parágrafo - porém esse valor ainda é INVÁLIDO em alguns navegadores para esta propriedade;
- auto: Segue automaticamente o valor padrão ou do elemento pai;
- self-start: Posiciona os grid-items no começo de cada linha grid;
- self-end: Posiciona os grid-items no final de cada linha grid;
- safe center: Desabilitados para os navegadores;
- unsafe center: Desabilitados para os navegadores;
- start: Posiciona os grid-items no começo de cada linha grid;
- end: Posiciona os grid-items no final de cada linha grid;
- right: Ainda não está disponível para todos os navegadores (Já está disponível no Chrome)!!! Posiciona os grid-items á direita de cada linha grid;
- left: Ainda não está disponível para todos os navegadores (Já está disponível no Chrome)!!! Posiciona os grid-items á esquerda de cada linha grid;

- unset: Para desligar a propriedade;
- initial: Volta ao valor inicial;
- inherit: Para herdar o valor do elemento pai;

- Valor Padrão: normal;

- Possui valor herdado? - NÃO;

OBS: ;

-->

```
<div class="container">
```

```
  <div class="bloco" style="justify-content: center;">
```

```
    Veja exemplos de uso da propriedade:<br>justify-items<br>nos nossos grids<br>&rArr;
```

```
  </div>
```

```
  <div class="bloco">
```

```
    justify-items: stretch
```

```
    <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são esticados horizontalmente  
      (para isso acontecer os grid-items não podem ter width pré-definido)...
```

```
    </span>
```

```
    <div id="stretch">
```

```
      <div class="div1"></div>
```

```
      <div class="div2"></div>
```

```
      <div class="div3"></div>
```

```
      <div class="div4"></div>
```

```
    </div>
```

```
    <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados sobre um display: grid;</span>
```

```
  </div>
```

```
  <div class="bloco">
```

```
    justify-items: flex-start
```



```
<span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no início
  de cada linha (eixo)...
</span>
<div id="flex-start">
  <div class="div1"></div>
  <div class="div2"></div>
  <div class="div3"></div>
  <div class="div4"></div>
</div>
<span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados sobre um display: grid;</span>
</div>

<div class="bloco">
  justify-items: flex-end
  <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no final
    de cada linha (eixo)...
  </span>
  <div id="flex-end">
    <div class="div1"></div>
    <div class="div2"></div>
    <div class="div3"></div>
    <div class="div4"></div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados sobre um display: grid;</span>
</div>

<div class="bloco">
  justify-items: center
  <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no centro
    de cada linha (eixo)...
  </span>
  <div id="center">
    <div class="div1"></div>
    <div class="div2"></div>
    <div class="div3"></div>
    <div class="div4"></div>
```



```
<div class="bloco">
  justify-items: self-end
  <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no final de
    cada linha (eixo)...
  </span>
  <div id="self-end">
    <div class="div1"></div>
    <div class="div2"></div>
    <div class="div3"></div>
    <div class="div4"></div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados sobre um display: grid;</span>
</div>
```

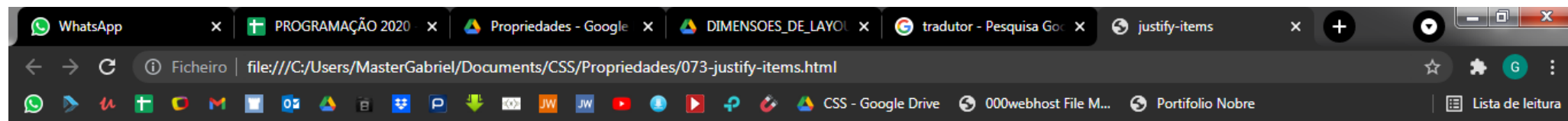
```
<div class="bloco">
  justify-items: start
  <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no início
    de cada linha (eixo)...
  </span>
  <div id="start">
    <div class="div1"></div>
    <div class="div2"></div>
    <div class="div3"></div>
    <div class="div4"></div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados sobre um display: grid;</span>
</div>
```

```
<div class="bloco">
  justify-items: end
  <span style="font-weight: normal;font-size: 10pt;">Perceba como os grid-items são posicionados no final
    de cada linha (eixo)...
  </span>
  <div id="end">
    <div class="div1"></div>
    <div class="div2"></div>
```



```
</body>
</html>
```

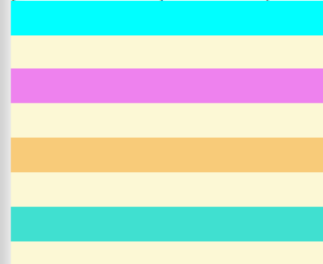
## RESULTADO...



Veja exemplos de uso da  
propriedade:  
**justify-items**  
nos nossos grids  
⇒

### **justify-items: stretch**

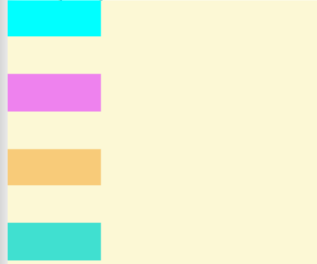
Perceba como os grid-items são esticados horizontalmente (para isso acontecer os grid-items não podem ter width pré-definido)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: flex-start**

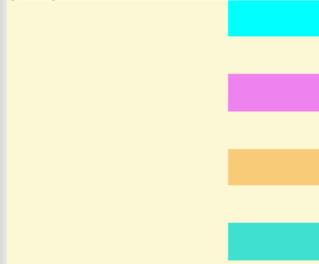
Perceba como os grid-items são posicionados no início de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: flex-end**

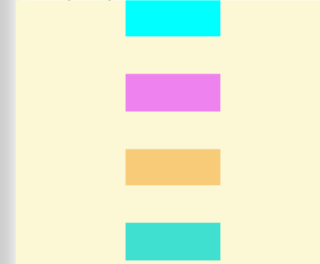
Perceba como os grid-items são posicionados no final de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: center**

Perceba como os grid-items são posicionados no centro de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: baseline**

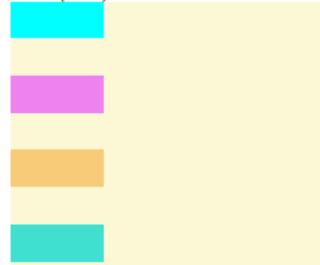
Perceba como os grid-items são em relação ao tamanho da maior letra de cada elemento (Para isso a largura do elemento não poderá estar definida)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: self-start**

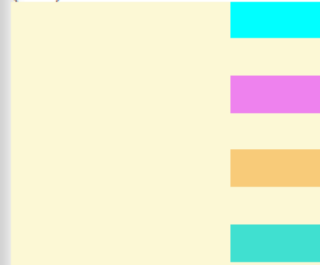
Perceba como os grid-items são posicionados no início de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: self-end**

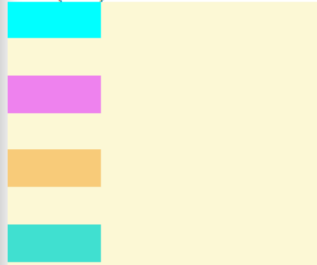
Perceba como os grid-items são posicionados no final de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: start**

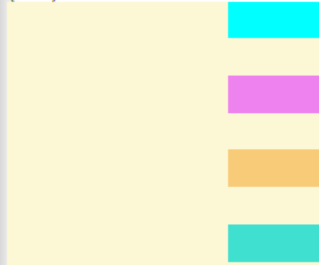
Perceba como os grid-items são posicionados no início de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: end**

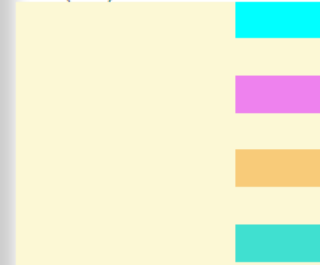
Perceba como os grid-items são posicionados no final de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: right**

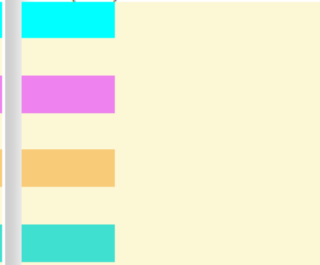
Perceba como os grid-items são posicionados à direita de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

### **justify-items: left**

Perceba como os grid-items são posicionados à esquerda de cada linha (eixo)...



Atenção: Valores aplicados sobre um display: grid;

