

```
<!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>place-self</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: 33%;
height: 100%;
padding: 2px;
margin: 4px;
box-shadow: 10px 10px 10px rgba(0, 0, 0, .2);
display: flex;
justify-content: center;
flex-direction: column;
}

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

.div1 {
  background-color: aqua;
  font-size: x-large;
  text-align: center;
}

.div2 {
  background-color: violet;
  font-size: x-large;
  text-align: center;
}

.div3 {
  background-color: rgb(248, 203, 121);
  font-size: x-large;
  text-align: center;
}

.div4 {
  background-color: turquoise;
  font-size: x-large;
  text-align: center;
}
```

```
}

#auto-grid {
  display: grid;
  height: 80%;
}
#auto-grid > div {
  width: 25%;
  height: 30%;
  display: flex;
  justify-content: center;
  align-items: center;
}
#auto-grid > .div2 {
  height: unset;
  place-self: auto center;
}

#stretch-grid {
  display: grid;
  height: 80%;
}
#stretch-grid > div {
  width: 25%;
  height: 30%;
  display: flex;
  justify-content: center;
  align-items: center;
}
#stretch-grid > .div2 {
  width: unset;
  height: unset;
  place-self: stretch;
}
}
```

```
#center-grid {
  display: grid;
  height: 80%;
}
#center-grid > div {
  width: 25%;
  height: 30%;
  display: flex;
  justify-content: center;
  align-items: center;
}
#center-grid > .div2 {
  height: unset;
  place-self: center;
}

#flex-start-grid {
  display: grid;
  height: 80%;
}
#flex-start-grid > div {
  width: 25%;
  height: 30%;
  display: flex;
  justify-content: center;
  align-items: center;
}
#flex-start-grid > .div2 {
  height: unset;
  place-self: flex-start flex-end;
}

#flex-end-grid {
```

```
    display: grid;
    height: 80%;
  }
  #flex-end-grid > div {
    width: 25%;
    height: 30%;
    display: flex;
    justify-content: center;
    align-items: center;
  }
  #flex-end-grid > .div2 {
    height: unset;
    place-self: flex-end flex-start;
  }

  #baseline-grid {
    display: grid;
    height: 80%;
  }
  #baseline-grid > div {
    width: 25%;
    height: 30%;
    display: flex;
    justify-content: center;
    align-items: center;
  }
  #baseline-grid > .div2 {
    height: unset;
    place-self: baseline center;
  }
}
```

</style>

</head>

<body>

<!--

Propriedade: place-self (Alocação de Si Mesmo)

Objetivo: Essa propriedade é uma abreviação das propriedades:

- align-self;
- justify-self;

Isso significa que "place-self" tem a missão de trazer um alinhamento tanto horizontal quanto vertical sobre um único item. Para isso ela recebe 2 valores, onde o 1º será aplicado sobre o "align-self" e o 2º será aplicado sobre o "justify-self", se apenas um valor for declarado ele vai valer para as 2 propriedades.

ATENÇÃO!!! Essa propriedade pode ser aplicada tanto a grid-items quanto a flex-items, porém, como a propriedade "justify-self" só pode ser aplicada sobre grid-items, ela fica inutilizada quando a aplicamos em flex-items. Por essa razão essa propriedade não é muito utilizada para alinhar flex-items, muito embora ela possa fazer o alinhamento vertical deles.

DETALHE IMPORTANTE!!! Sempre deve ser usada em flex-items ou grid-items, nunca deve ser utilizada sobre um elemento container.

- Valores Aceitáveis:

- auto: Vai pegar o valor automático do conteúdo;
- stretch: O item vai ser esticado;
- center: O item vai ser centralizado;
- flex-start: O item vai ser posicionado no início do eixo;
- flex-end: O item vai ser posicionado no final do eixo;

- baseline: O item vai se alinhar ao tamanho do conteúdo textual;
- unset: Para desligar a propriedade;
- initial: Volta ao valor inicial;
- inherit: Para herdar o valor do elemento pai;

- Valor Padrão: auto;

- Possui valor herdado? - NÃO;

OBS: ;

-->

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

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

```
    place-self: auto center<br>
```

```
    <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" pega o tamanho automático  
      do grid, que é preencher todo o eixo verticalmente. E além disso, ocupa o alinhamento central do eixo...
```

```
  </span>
```

```
  <div id="auto-grid">
```

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

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

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

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

```
  </div>
```

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

```
  </span>
```

```
</div>
```

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

```
  place-self: stretch<br>
```

```
  <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" estica até ter o tamanho
```

vertical completo no eixo do grid. E além disso, estica sobre alinhamento horizontal do eixo - para isso acontecer o elemento não pode ter largura e altura pré-definidos...

```
</span>
<div id="stretch-grid">
  <div class="div1">A</div>
  <div class="div2">B</div>
  <div class="div3">C</div>
  <div class="div4">D</div>
</div>
<span style="font-weight: bold;font-size: 8pt;">Atenção: aplicado sobre um display-grid;
</span>
</div>
```

```
<div class="bloco">
  place-self: center<br>
  <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" pega o centro vertical no eixo
    do grid. E além disso, ocupa o alinhamento central do eixo...
</span>
<div id="center-grid">
  <div class="div1">A</div>
  <div class="div2">B</div>
  <div class="div3">C</div>
  <div class="div4">D</div>
</div>
<span style="font-weight: bold;font-size: 8pt;">Atenção: aplicado sobre um display-grid;
</span>
</div>
```

```
</div>
```

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

```
  <div class="bloco">
    place-self: flex-start flex-end<br>
    <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" pega a parte de cima na
      vertical do eixo do grid. E além disso, ocupa o final do eixo primário...
```



```
</span>
<div id="flex-start-grid">
  <div class="div1">A</div>
  <div class="div2">B</div>
  <div class="div3">C</div>
  <div class="div4">D</div>
</div>
<span style="font-weight: bold;font-size: 8pt;">Atenção: aplicado sobre um display-grid;
</span>
</div>

<div class="bloco">
  place-self: flex-end flex-start<br>
  <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" pega a parte de baixo na
    vertical do eixo do grid. E além disso, ocupa o início do eixo primário...
  </span>
  <div id="flex-end-grid">
    <div class="div1">A</div>
    <div class="div2">B</div>
    <div class="div3">C</div>
    <div class="div4">D</div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: aplicado sobre um display-grid;
  </span>
</div>

<div class="bloco">
  place-self: baseline center<br>
  <span style="font-weight: normal;font-size: 10pt;">Perceba que somente a div "B" pega o tamanho do texto
    menor verticalmente. E além disso, ocupa o alinhamento central do eixo...
  </span>
  <div id="baseline-grid">
    <div class="div1">A</div>
    <div class="div2"><span style="font-size: 10pt">B</span></div>
    <div class="div3">C</div>
    <div class="div4">D</div>
```

```

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

</div>

</body>
</html>

```

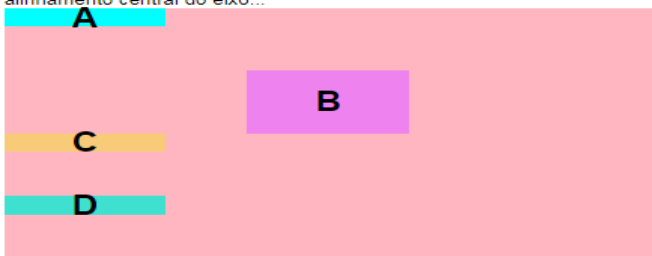
## RESULTADO...

WhatsApp x PROGRAMAÇÃO 2020 x Propriedades - Google x DIMENSOES\_DE\_LAYOUT x Dominando CSS 3 com x place-self x

Ficheiro | file:///C:/Users/MasterGabriel/Documents/CSS/Propriedades/082-place-self.html

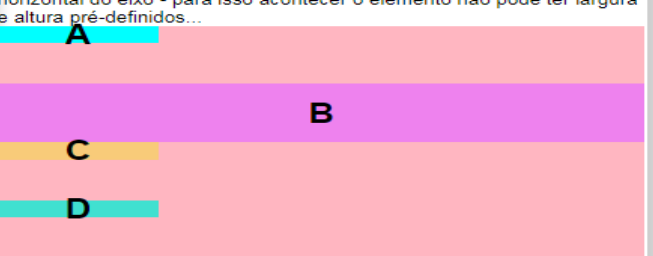
WhatsApp, Fich, G, M, W, O, A, J, J, Y, T, P, CSS - Google Drive, 000webhost File M..., Portfolio Nobre, Lista de leitura

**place-self: auto center**  
Perceba que somente a div "B" pega o tamanho automático do grid, que é preencher todo o eixo verticalmente. E além disso, ocupa o alinhamento central do eixo...



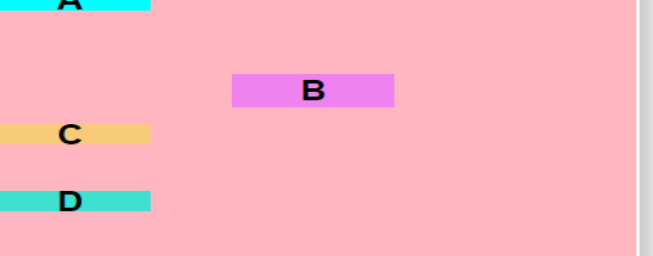
Atenção: aplicado sobre um display-grid;

**place-self: stretch**  
Perceba que somente a div "B" estica até ter o tamanho vertical completo no eixo do grid. E além disso, estica sobre alinhamento horizontal do eixo - para isso acontecer o elemento não pode ter largura e altura pré-definidos...



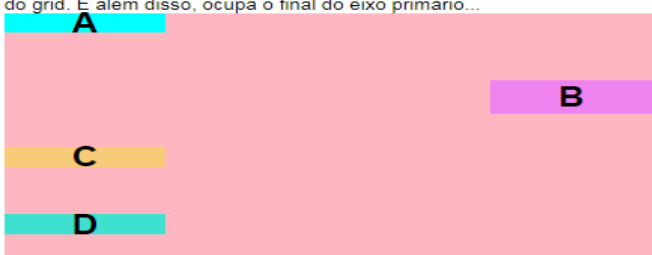
Atenção: aplicado sobre um display-grid;

**place-self: center**  
Perceba que somente a div "B" pega o centro vertical no eixo do grid. E além disso, ocupa o alinhamento central do eixo...



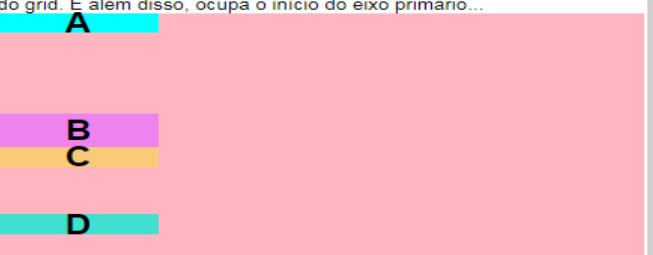
Atenção: aplicado sobre um display-grid;

**place-self: flex-start flex-end**  
Perceba que somente a div "B" pega a parte de cima na vertical do eixo do grid. E além disso, ocupa o final do eixo primário...



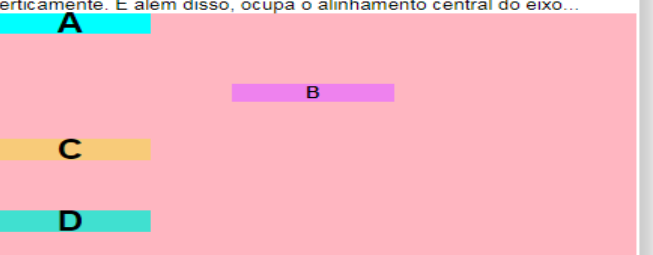
Atenção: aplicado sobre um display-grid;

**place-self: flex-end flex-start**  
Perceba que somente a div "B" pega a parte de baixo na vertical do eixo do grid. E além disso, ocupa o início do eixo primário...



Atenção: aplicado sobre um display-grid;

**place-self: baseline center**  
Perceba que somente a div "B" pega o tamanho do texto menor verticalmente. E além disso, ocupa o alinhamento central do eixo...



Atenção: aplicado sobre um display-grid;

Windows, VS Code, PC, 01/07/2021, Google, Excel, Word, PDF, 11:04 17/07/2021