

```
<!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-content</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: 20%;
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: 30%;
  font-size: x-large;
  text-align: center;
}

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

.div3 {
  background-color: wheat;
  width: 30%;
  height: 30%;
  font-size: x-large;
  text-align: center;
}
```

```
}

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

#normal {
  width: 100%;
  height: 80%;
  background-color: thistle;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  place-content: normal;
}

#stretch {
  width: 100%;
  height: 80%;
  background-color: thistle;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  place-content: stretch center;
}

#stretch > div {
  height: unset;
}

#flex-start {
  width: 100%;
```

```
height: 80%;
background-color: thistle;
display: flex;
flex-direction: row;
flex-wrap: wrap;
place-content: flex-start;
}

#flex-end {
width: 100%;
height: 80%;
background-color: thistle;
display: flex;
flex-direction: row;
flex-wrap: wrap;
place-content: flex-end;
}

#center {
width: 100%;
height: 80%;
background-color: thistle;
display: flex;
flex-direction: row;
flex-wrap: wrap;
place-content: center;
}

#baseline {
height: 80%;
background-color: thistle;
display: flex;
flex-direction: row;
flex-wrap: wrap;
place-content: baseline space-between;
}
```

```
#baseline > div {
  width: unset;
  height: unset;
}

#space-between {
  width: 100%;
  height: 80%;
  background-color: thistle;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  place-content: space-between;
}

#space-evenly {
  width: 100%;
  height: 80%;
  background-color: thistle;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  place-content: space-evenly;
}

#space-around {
  width: 100%;
  height: 80%;
  background-color: thistle;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  place-content: space-around;
}
```

```
</style>
</head>
<body>

<!--
```

Propriedade: place-content (Lugar do Conteúdo)

Objetivo: Essa propriedade é uma abreviação das seguintes propriedades:

- align-content;
- justify-content;

Onde poderemos especificar 2 valores em uma única linha para as 2 propriedades, se especificarmos 1 valor só, ele será aplicado sobre as 2 propriedades;

Lembrando que os valores do "align-content" só pode ser atribuído sobre um elemento container flex-box que estiver usando "flex-wrap", sem o "flex-wrap" ele não funciona.

DETALHE IMPORTANTE!!! O 1º valor sempre é aplicado ao "align-content" e o 2º valor é aplicado sobre o "justify-content"!

- Valores Aceitáveis:

- normal: Para Flex: Segue o fluxo padrão do justify-content;
- stretch: Para Flex e Grid: Estica os flex-items se eles não tiverem tamanho pré definido onde o esticamento acontece; Estica os elementos grid dentro do espaço da linha do grid;
- flex-start: Para Flex: Alinha os flex-items á partir de onde começa o eixo principal do container;
- flex-end: Para Flex: Alinha os flex-items á partir de onde termina o eixo principal do container;
- center: Para Flex e Grid: Alinha os flex-items no centro do container; Posiciona os elementos grid

no centro de cada linha grid;

- baseline: Para Flex: Alinha os flex-items ao tamanho da maior letra dentro de um elemento;
 - first-baseline: Para Flex: Alinha os flex-items ao tamanho da maior letra de um primeiro parágrafo - porém esse valor ainda é INVÁLIDO em alguns navegadores para esta propriedade;
 - last-baseline: Para Flex: Alinha os flex-items ao tamanho da menor letra do último parágrafo - porém esse valor ainda é INVÁLIDO em alguns navegadores para esta propriedade;
 - space-between: Para Flex: Alinha os flex-items nas laterais do container distribuindo o espaço em vazio no meio deles;
 - space-around: Para Flex: Alinha os flex-items distribuindo o espaço em vazio entre eles por igual, mas o espaço entre os flex-items e as laterais tem espaçamento menor;
 - space-evenly: Para Flex: Distribui o espaço em vazio por igual, tanto entre os flex-items, quanto entre eles e as laterais;
 - safe center: Desabilitados para os navegadores;
 - unsafe center: Desabilitados para os navegadores;
 - start: Para Grid: Posiciona os elementos grid no começo de cada linha grid;
 - end: Para Grid: Posiciona os elementos grid no fim 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: "place-content" pode ser aplicado sobre grid-layout, mas não faz muito sentido aplicar os valores de "justify-content" já que eles só são aplicados sobre flex-box, então, acaba sendo melhor usar a propriedade "align-content" sobre os grid-layout;

-->

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

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

```
    Exemplos do uso de: <br>"place-content"<br> em elementos flex-box...<br>&rArr;
```

```
  </div>
```

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

```
    place-content: normal
```

```
    <span style="font-weight: normal;font-size: 10pt;">O valor é aplicado só sobre o align-content, o  
      justify-content se mantém padrão...
```

```
    </span>
```

```
    <div id="normal">
```

```
      <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: Valores aplicados a um flex-direction: row;</span>
```

```
  </div>
```

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

```
    place-content: stretch center
```

```
    <span style="font-weight: normal;font-size: 10pt;">O align-content é esticado, quanto o justify-content  
      centraliza os elementos ao meio do container...
```

```
    </span>
```

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

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

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

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



```
<div class="bloco">
  place-content: center
  <span style="font-weight: normal;font-size: 10pt;">Tanto o align-content quando o justify-content são
    posicionados no centro dos eixos principal e secundário...
  </span>
  <div id="center">
    <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: Valores aplicados a um flex-direction: row;</span>
</div>
```

```
<div class="bloco">
  place-content: baseline space-between
  <span style="font-weight: normal;font-size: 10pt;">O align-content faz com que os elementos tenham o tamanho
    da letra, enquanto o justify-content faz um espaçamento justificado entre eles, colando os elementos dos
    cantos nas extremidades...
  </span>
  <div id="baseline">
    <div class="div1"><span style="font-size: 20pt">A</span></div>
    <div class="div2"><span style="font-size: 7pt">B</span></div>
    <div class="div3"><span style="font-size: 32pt">C</span></div>
    <div class="div4"><span style="font-size: 72pt">D</span></div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados a um flex-direction: row;</span>
</div>
```

```
<div class="bloco">
  place-content: space-between
  <span style="font-weight: normal;font-size: 10pt;">Tanto o align-content quando o justify-content vai gerar
    um espaçamento entre os elementos por igual, colando os elementos dos cantos nas suas respectivas
    extremidades...
  </span>
```

```
<div id="space-between">
  <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: Valores aplicados a um flex-direction: row;</span>
</div>
```

```
<div class="bloco">
  place-content: space-evenly
  <span style="font-weight: normal;font-size: 10pt;">Tanto o align-content quando o justify-content vai gerar
    um espaçamento entre os elementos e as suas extremidades por igual...
  </span>
  <div id="space-evenly">
    <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: Valores aplicados a um flex-direction: row;</span>
</div>
```

```
<div class="bloco">
  place-content: space-around
  <span style="font-weight: normal;font-size: 10pt;">Tanto o align-content quando o justify-content vai gerar
    um espaçamento entre os elementos por igual, criando um pequeno espaçamento entre os elementos dos
    cantos e suas respectivas extremidades...
  </span>
  <div id="space-around">
    <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: Valores aplicados a um flex-direction: row;</span>
```

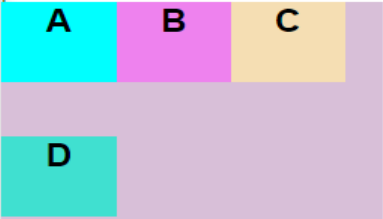
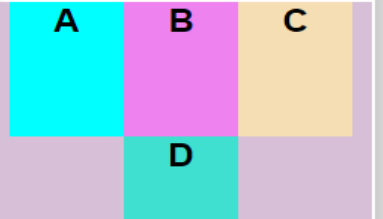


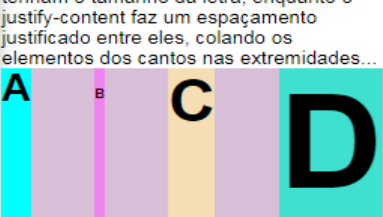
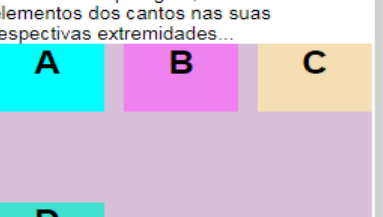
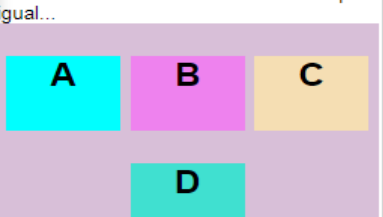
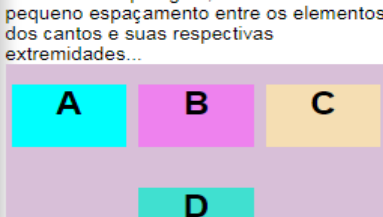
```
</div>

</div>

</body>
</html>
```

RESULTADO...

Exemplos do uso de:
"place-content"
em elementos flex-box...
⇒

<p>place-content: normal</p> <p>O valor é aplicado só sobre o align-content, o justify-content se mantém padrão...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: stretch center</p> <p>O align-content é esticado, quanto o justify-content centraliza os elementos ao meio do container...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: flex-start</p> <p>Tanto o align-content quando o justify-content são posicionados no começo dos eixos principal e secundário...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: flex-end</p> <p>Tanto o align-content quando o justify-content são posicionados no fim dos eixos principal e secundário...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>
<p>place-content: center</p> <p>Tanto o align-content quando o justify-content são posicionados no centro dos eixos principal e secundário...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: baseline space-between</p> <p>O align-content faz com que os elementos tenham o tamanho da letra, enquanto o justify-content faz um espaçamento justificado entre eles, colando os elementos dos cantos nas extremidades...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: space-between</p> <p>Tanto o align-content quando o justify-content vai gerar um espaçamento entre os elementos por igual, colando os elementos dos cantos nas suas respectivas extremidades...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>	<p>place-content: space-evenly</p> <p>Tanto o align-content quando o justify-content vai gerar um espaçamento entre os elementos e as suas extremidades por igual...</p>  <p>Atenção: Valores aplicados a um flex-direction: row;</p>

09:33
14/07/2021