```html
<!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>align-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: 33%;
        padding: 5px;
        display: flex;
        justify-content: space-between;
        align-items: center;
        box-shadow: 10px 10px 10px rgba(0, 0, 0, .2);
    }
```

```css
.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;
    align-content: normal;
}

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

#stretch > div {
    height: unset;
}

#flex-start {
```

```
        width: 100%;
        height: 80%;
        background-color: thistle;
        display: flex;
        flex-direction: row;
        flex-wrap: wrap;
        align-content: flex-start;
    }

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

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

    #baseline {
        height: 80%;
        background-color: thistle;
        display: flex;
        flex-direction: row;
        flex-wrap: wrap;
        align-content: baseline;
```

```
}

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

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

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

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

```css
#stretch-grid {
    width: 100%;
    height: 80%;
    background-color: thistle;
    display: grid;
    align-content: stretch;
}

#stretch-grid > div {
    height: unset;
}

#center-grid {
    width: 100%;
    height: 80%;
    background-color: thistle;
    display: grid;
    align-content: center;
}

#start-grid {
    width: 100%;
    height: 80%;
    background-color: thistle;
    display: grid;
    align-content: start;
}

#end-grid {
    width: 100%;
    height: 80%;
    background-color: thistle;
    display: grid;
    align-content: end;
}
```

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

<!--

        Propriedade: align-content (Alinhamento de Conteúdo)


    Objetivo: Essa propriedade define como ficarão alinhadas as linhas quebradas de um "flex-wrap".

                Quando atribuímos um "flex-wrap" sobre os flex-items de um container, o "flex-wrap" vai quebrar as
                linhas segundo o seu padrão, que é "stretch", isso faz com que as linhas sejam separadas
                por igual, deixando grandes espaçamentos entre elas. Mas essa pode não ser a nossa vontade, para isso
                foi criada a propriedade "align-content", que alinha o espaçamento das linhas quebradas de um
                "flex-wrap" segundo a nossa vontade.

                O "align-content" só pode ser atribuído sobre um elemento container. Ele sempre deve ser usado para
                arrumar um "flex-wrap", sem o "flex-wrap" ele não funciona.

                DETALHE IMPORTANTE!!! O "align-content" vai trabalhar sobre os eixos segundários do nosso flex-box,
                enquanto o "justify-content" vai trabalhar sobre os eixos primários. Portanto, o comportamento do nosso
                "align-content" vai depender do nosso "flex-direction".



    - 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 distribuíndo o espaço em vazio no
                        meio deles;

    - space-around:     Para Flex: Alinha os flex-items distribuíndo 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: Distribuí 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: flex-start;
```

```
    - Possuí valor herdado? - NÃO;

    OBS: O "align-content" pode aceitar valores especiais, como os: space-between, space-evenly e space-around;

        ATENÇÃO: para usar o valor stretch tanto em flex-box quanto em grid-layout temos que desligar o valor da altura
                    ou da largura do eixo secundário ao qual desejamos aplicar o nosso stretch;

-->

    <div class="container">

        <div class="bloco" style="justify-content: center;">
            Ao lado temos detalhes de como usar a propriedade "align-content" em container do tipo flex-box...<br>&rAarr;
        </div>

        <div class="bloco">
            align-content: normal
            <span style="font-weight: normal;font-size: 10pt;">Perceba que segue o padrão do flex-wrap...
            </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">
            align-content: stretch
            <span style="font-weight: normal;font-size: 10pt;">Perceba que os elementos foram esticados para ocupar todo
                o espaçamento
            </span>
            <div id="stretch">
                <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">
        align-content: flex-start
        <span style="font-weight: normal;font-size: 10pt;">Perceba que os itens foram alinhados ao topo...
        </span>
        <div id="flex-start">
            <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">
        align-content: flex-end
        <span style="font-weight: normal;font-size: 10pt;">Perceba que os itens foram alinhados a parte de baixo...
        </span>
        <div id="flex-end">
            <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>

<div class="container">
```

```
<div class="bloco">
    align-content: center
    <span style="font-weight: normal;font-size: 10pt;">Perceba que os itens foram alinhados ao centro...
    </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">
    align-content: baseline
    <span style="font-weight: normal;font-size: 10pt;">Perceba que os elementos vão pegar o tanho da maior letra
        dentro dele...
    </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">
    align-content: space-between
    <span style="font-weight: normal;font-size: 10pt;">Os elementos fazem o wrap sendo jogados para as
        extremidades e tendo o espaço restante como o seu meio...
    </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">
        align-content: space-evenly
        <span style="font-weight: normal;font-size: 10pt;">Os elementos tem o espaçamento dividido por igual entre
            eles e as extremidades...
        </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">
        align-content: space-around
        <span style="font-weight: normal;font-size: 10pt;">Os elementos tem o espaçamento dividido por igual entre
            eles e um espaçamento curto entre eles e as 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>
```

```
<div class="container">

    <div class="bloco" style="justify-content: center;">
        Estes são exemplos usando apenas grid-layout<br>&rAarr;
    </div>

    <div class="bloco">
        align-content: stretch
        <span style="font-weight: normal;font-size: 10pt;">Perceba que os elementos são esticados para pegar o
            tamanho total da linha ou coluna grid...
        </span>
        <div id="stretch-grid">
            <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 a um display: grid;</span>
    </div>

    <div class="bloco">
        align-content: center
        <span style="font-weight: normal;font-size: 10pt;">Perceba que as linhas grid estão centralizadas dentro do
            nosso container...
        </span>
        <div id="center-grid">
            <div class="div1">A</div>
            <div class="div2">B</div>
            <div class="div3">C</div>
        </div>
        <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados a um display: grid;</span>
    </div>

    <div class="bloco">
        align-content: start
        <span style="font-weight: normal;font-size: 10pt;">Perceba que as linhas grid estão posicionadas no topo do
```

```
            nosso container...
        </span>
        <div id="start-grid">
            <div class="div1">A</div>
            <div class="div2">B</div>
            <div class="div3">C</div>
        </div>
        <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados a um display: grid;</span>
    </div>

    <div class="bloco">
        align-content: end
        <span style="font-weight: normal;font-size: 10pt;">Perceba que as linhas grid estão posicionadas na parte
            inferior do nosso container...
        </span>
        <div id="end-grid">
            <div class="div1">A</div>
            <div class="div2">B</div>
            <div class="div3">C</div>
        </div>
        <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados a um display: grid;</span>
    </div>

</div>



</body>
</html>
```

RESULTADO LOGO ABAIXO...

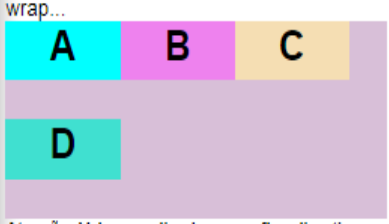
WhatsApp | PROGRAMAÇÃO 2020 | Propriedades - Google | DIMENSOES_DE_LAYOU | como usar baseline, firs | align-content

Ficheiro | file:///C:/Users/MasterGabriel/Documents/CSS/Propriedades/071-align-content.html

CSS - Google Drive | 000webhost File M... | Portifolio Nobre | Lista de leitura

Ao lado temos detalhes de como usar a propriedade "align-content" em container do tipo flex-box...
⇛

align-content: normal
Perceba que segue o padrão do flex-wrap...

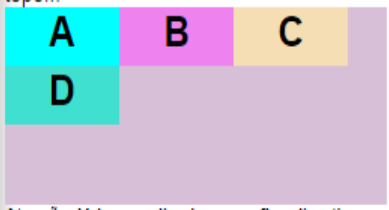


Atenção: Valores aplicados a um flex-direction: row;

align-content: stretch
Perceba que os elementos foram esticados para ocupar todo o espaçamento

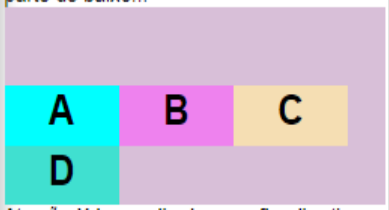
Atenção: Valores aplicados a um flex-direction: row;

align-content: flex-start
Perceba que os itens foram alinhados ao topo...



Atenção: Valores aplicados a um flex-direction: row;

align-content: flex-end
Perceba que os itens foram alinhados a parte de baixo...



Atenção: Valores aplicados a um flex-direction: row;

align-content: center
Perceba que os itens foram alinhados ao centro...

Atenção: Valores aplicados a um flex-direction: row;

align-content: baseline
Perceba que os elementos vão pegar o tanho da maior letra dentro dele...

Atenção: Valores aplicados a um flex-direction: row;

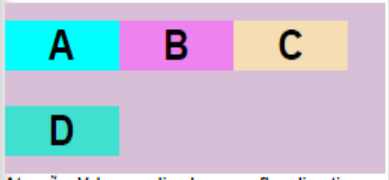
align-content: space-between
Os elementos fazem o wrap sendo jogados para as extremidades e tendo o espaço restante como o seu meio...

Atenção: Valores aplicados a um flex-direction: row;

align-content: space-evenly
Os elementos tem o espaçamento dividido por igual entre eles e as extremidades...

Atenção: Valores aplicados a um flex-direction: row;

align-content: space-around
Os elementos tem o espaçamento dividido por igual entre eles e um espaçamento curto entre eles e as extremidades...



Atenção: Valores aplicados a um flex-direction: row;

Estes são exemplos usando apenas grid-layout
⇛

align-content: stretch
Perceba que os elementos são esticados para pegar o tamanho total da linha ou coluna grid...

Atenção: Valores aplicados a um display: grid;

align-content: center
Perceba que as linhas grid estão centralizadas dentro do nosso container...

Atenção: Valores aplicados a um display: grid;

align-content: start
Perceba que as linhas grid estão posicionadas no topo do nosso container...

Atenção: Valores aplicados a um display: grid;

align-content: end
Perceba que as linhas grid estão posicionadas na parte inferior do nosso container...

Atenção: Valores aplicados a um display: grid;

08:42
14/07/2021