Vamos iniciar criando duas divs, uma ao lado da outra:

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

            div{

                width: 200*px*; */\*altura\*/*

                height: 200*px*; */\*largura\*/*

                background-color: #FF0000;

                margin: 10*px*; */\*pra ela ficar desencostada de tudo\*/*

            }

        </style>

    </head>

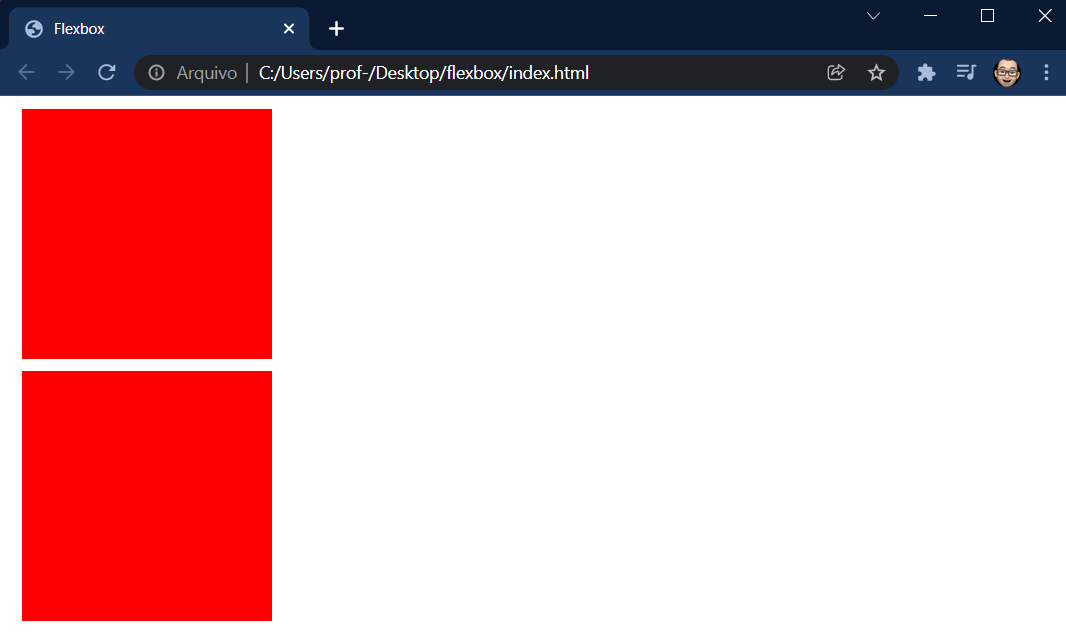
    <body>

        <div></div>

        <div></div>

    </body>

</html>



Como fazemos pra colocar esses dois quadrados, uma ao lado do outro? Precisamos deixa-los flutuantes.

Basicamente faríamos um float: left em nosso css:

        <style *type*="text/css">

            div{

                width: 200*px*; */\*altura\*/*

                height: 200*px*; */\*largura\*/*

                background-color: #FF0000;

                margin: 10*px*; */\*pra ela ficar desencostada de tudo\*/*

                float: left;

            }

        </style>



E se quisermos que esses dois quadrados tenham tamanhos variados, ocupando cada um metade da tela de acordo com os mais variados tamanhos de tela? Se mudarmos o tamanho das larguras para 50% veremos que eles não ficam um ao lado do outro, pois temos a questão da margem. Portanto teríamos que realizar diversas contas pra que esses quadrados se adaptassem.

Agora que já temos noção de como é difícil posicionar itens na tela, vamos entender um pouco do flexbox, pois ele nos ajuda muito nessas situações. O flexbox significa deixar todos os box do seu site flexíveis e adaptáveis. Vamos customizar nossa página da seguinte forma:

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

*.container*{

                background-color: #DDDDDD;

            }

*.container* div{

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

        </style>

    </head>

    <body>

        <div *class*="container">

            <div>1</div>

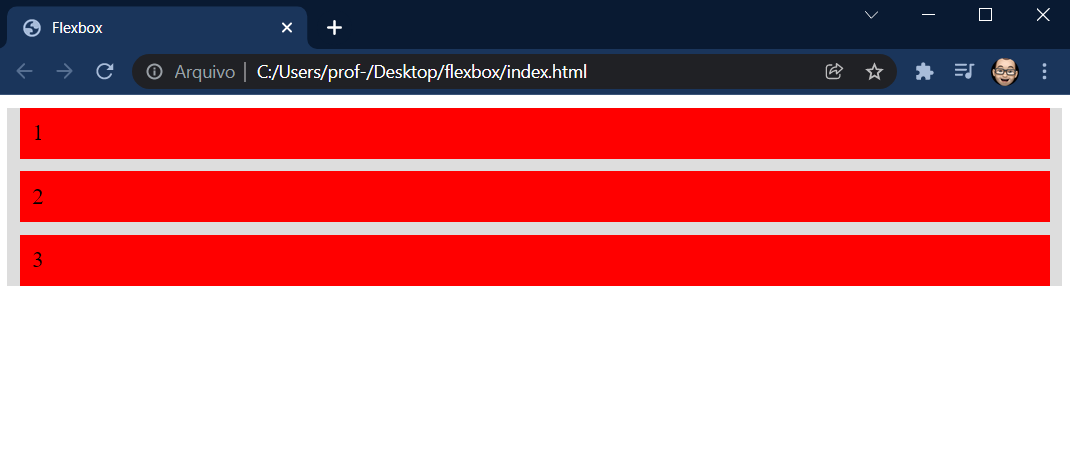
            <div>2</div>

            <div>3</div>

        </div>

    </body>

</html>



Agora vamos adicionar o display: flex dentro do container:

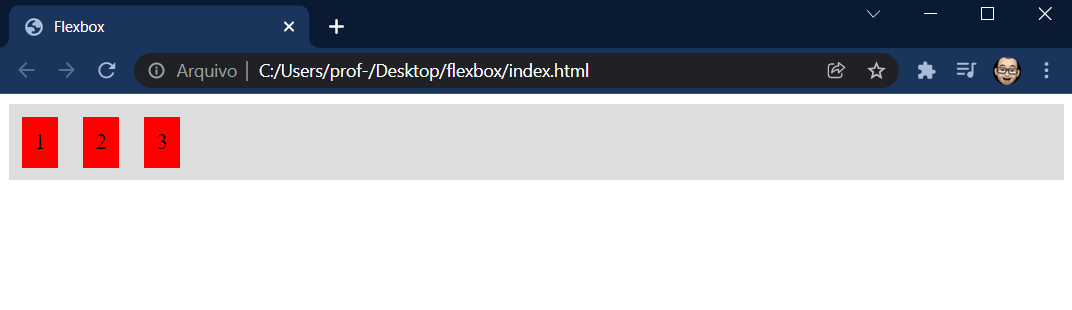
*.container*{

                display: flex;

                background-color: #DDDDDD;

            }

Teremos o seguinte resultado:



Agora que adicionamos o flexbox ao nosso container, precisamos definir a direção. Por padrão os componentes vêm com a direção horizontal (flex-direction: row).

Vamos alterar a a direção para colunas:

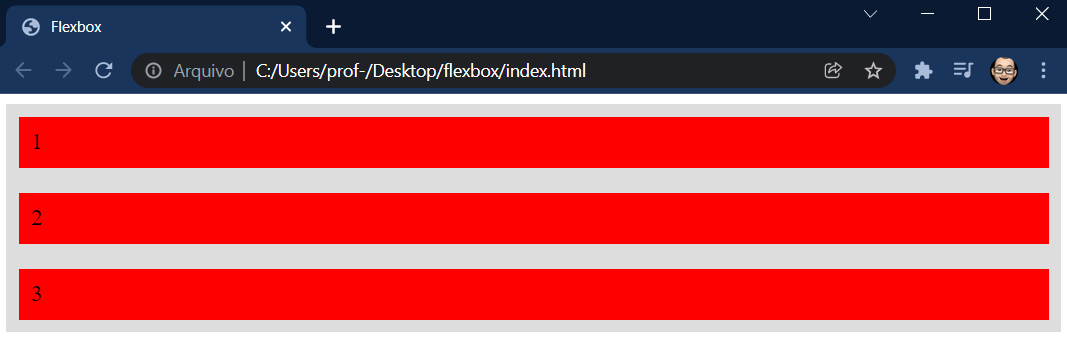
*.container*{

                display: flex;

                flex-direction:column;

                background-color: #DDDDDD;

            }



Temos também a opção row-reverse, que muda da direita para a esquerda:

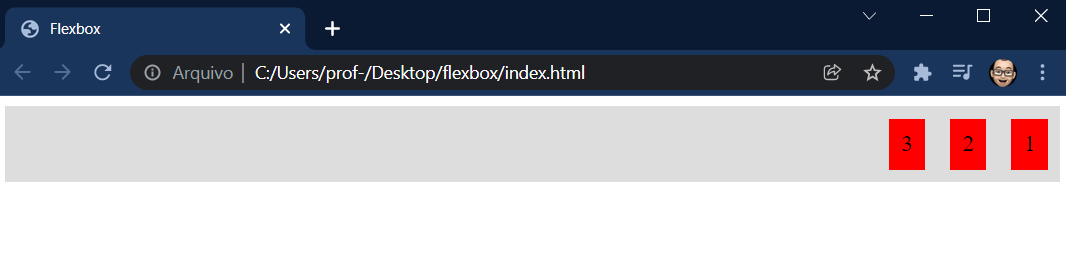
*.container*{

                display: flex;

                flex-direction:row-reverse;

                background-color: #DDDDDD;

            }



Temos também a opção columm-reverse, que muda da direita para a esquerda:

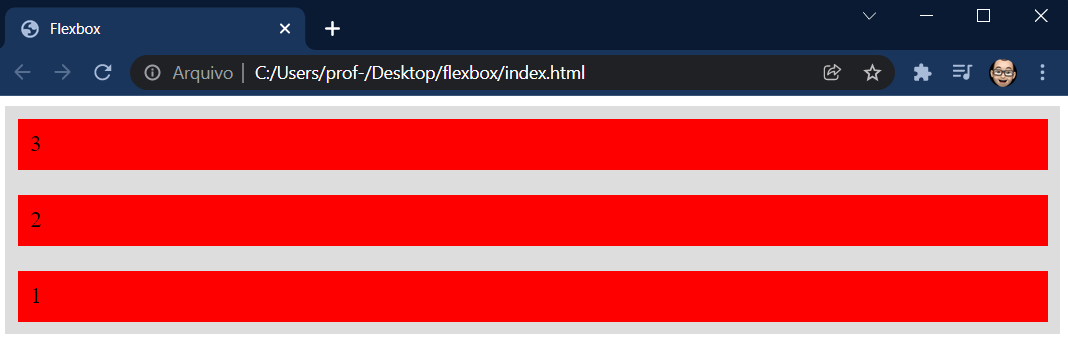
*.container*{

                display: flex;

                flex-direction:column-reverse;

                background-color: #DDDDDD;

            }



Vamos agora colocar outro container flex em nossa página, para demonstrar que podemos ter vários conteiners flex. Inserimos margin pra que um não fique grudado no outro. Vamos também deixar um como coluna e outro como linha.

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

*.container*{

                display: flex;

                flex-direction:row;

                background-color: #DDDDDD;

                margin: 10*px*;

            }

*.container2*{

                display: flex;

                flex-direction:column;

                background-color: #DDDDDD;

                margin: 10*px*;

            }

*.container* div,

*.container2* div{

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

        </style>

    </head>

    <body>

        <div *class*="container">

            <div>1</div>

            <div>2</div>

            <div>3</div>

        </div>

        <div *class*="container2">

            <div>1</div>

            <div>2</div>

            <div>3</div>

        </div>

    </body>

</html>



Fazer alguns testes:

* Deixar um em row e o outro em row-reverse;

Vamos remover o contêiner2, para trabalhar com nosso código de forma mais didática:

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

*.container*{

                display: flex;

                flex-direction:row;

                background-color: #DDDDDD;

                margin: 10*px*;

            }

*.container* div{

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

        </style>

    </head>

    <body>

        <div *class*="container">

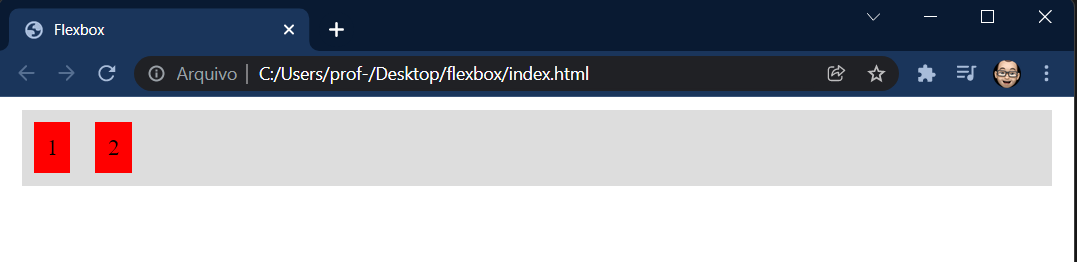
            <div>1</div>

            <div>2</div>

        </div>

    </body>

</html>



Vamos agora ajusta-los pra que ocupem o mesmo espaço na tela, inserindo nos itens o flex: 1:

*.container* div{

                flex:1;

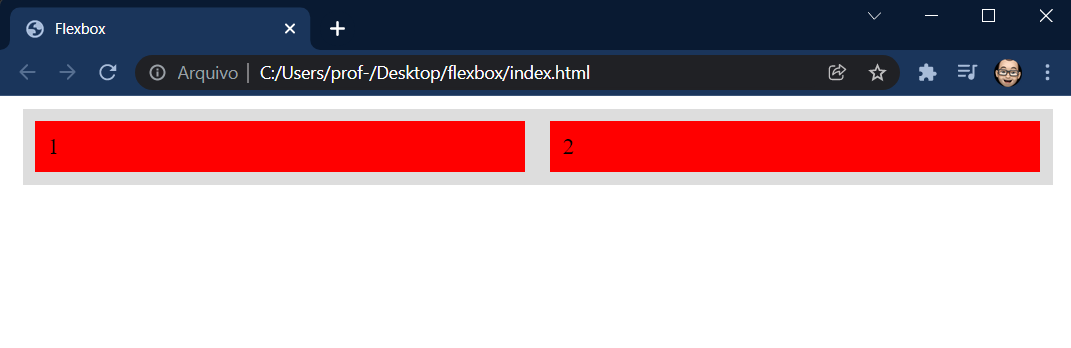
                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }



Agora eles ficaram com o mesmo tamanho, independentemente do tamanho da tela (pode redimensionar pra verificar). Também fazer o teste removendo a margin, iremos notar que eles vão continuar, porém juntos.

Vamos agora criar class para as divs e customiza-las:

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

*.container*{

                display: flex;

                flex-direction:row;

                background-color: #DDDDDD;

                margin: 10*px*;

            }

*.div1*{

                flex:1;

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

*.div2*{

                flex:1;

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

        </style>

    </head>

    <body>

        <div *class*="container">

            <div *class*="div1">1</div>

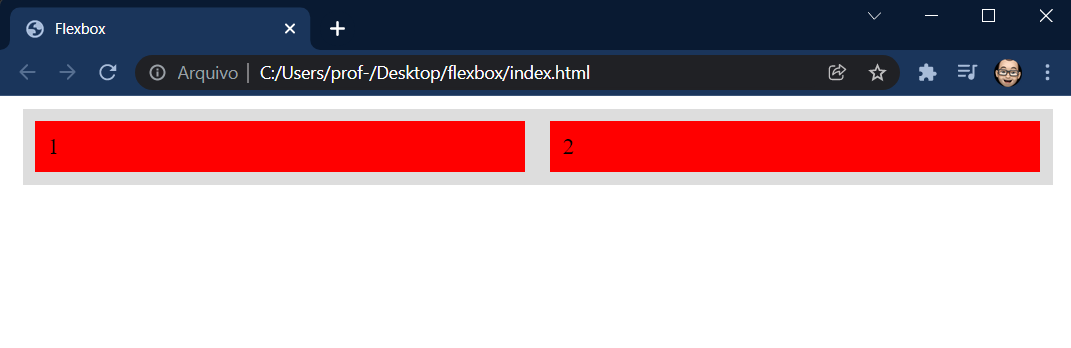
            <div *class*="div2">2</div>

        </div>

    </body>

</html>

Iremos notar que nada muda:



Crie uma terceira div:

<!DOCTYPE *html*>

<html *lang*="pt-br">

    <head>

        <meta *charset*="UTF-8">

        <title>Flexbox</title>

        <style *type*="text/css">

*.container*{

                display: flex;

                flex-direction:row;

                background-color: #DDDDDD;

                margin: 10*px*;

            }

*.div1*{

                flex:1;

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

*.div2*{

                flex:1;

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

*.div3*{

                flex:1;

                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }

        </style>

    </head>

    <body>

        <div *class*="container">

            <div *class*="div1">1</div>

            <div *class*="div2">2</div>

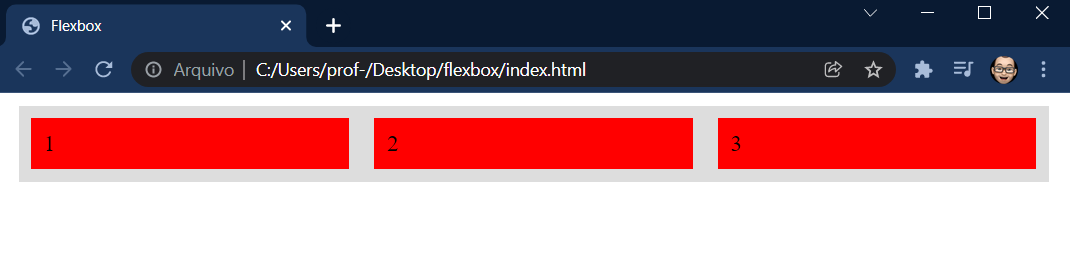
            <div *class*="div3">3</div>

        </div>

    </body>

</html>

Note que ele divide igualmente os espaços, pois os itens participam do mesmo grupo:



Mudar o flex do div3 para 2:

*.div3*{

                flex:2;

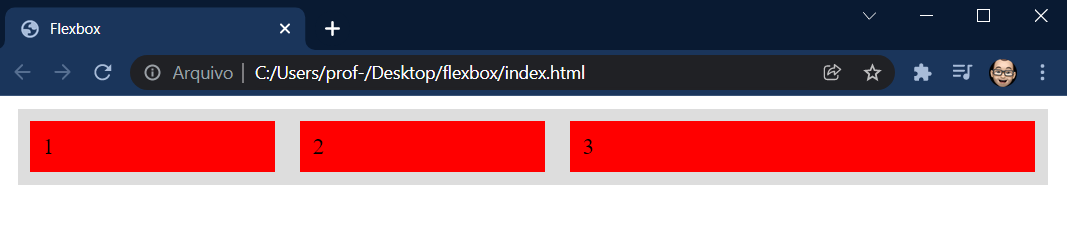
                background-color: #FF0000;

                margin: 10*px*;

                padding: 10*px*;

                font-size: 18*px*;

            }



Note que o 3 ficou com 2 vezes o tamanho do 1.

Alterar o número do flex da .div3 para 3.

Criar uma div4 e brincar com o tamanho das 4.

**Flex-box – Alinhamentos**

Note que o 3 ficou com 2 vezes o tamanho do 1.

Vamos deixar nosso código com um display-flex, seguindo o seguinte padrão:

<!DOCTYPE html>

<html lang="pt-br">

    <head>

        <meta charset="UTF-8">

        <title>Flexbox</title>

        <style type="text/css">

            .container{

                display: flex;

                background-color: #DDDDDD;

                margin: 10px;

            }

            .container div{

                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }

        </style>

    </head>

    <body>

        <div class="container">

            <div>1</div>

            <div>2</div>

            <div>3</div>

            <div>4</div>

            <div>5</div>

            <div>6</div>

            <div>7</div>

            <div>8</div>

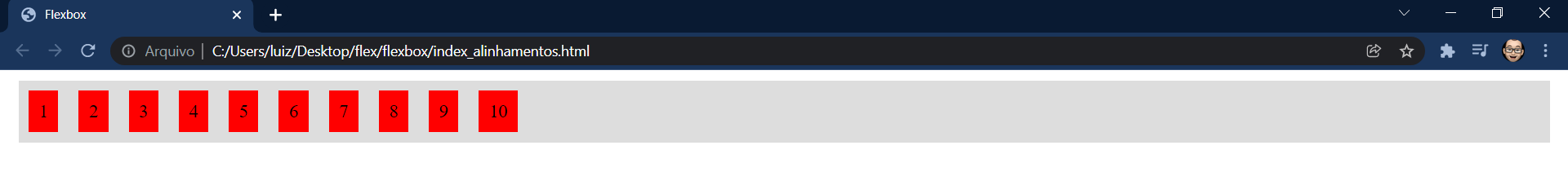
            <div>9</div>

            <div>10</div>

        </div>

    </body>

</html>

****

Se colocarmos um flex: 1, todas as divs passam a ser do mesmo grupo e dividem seu tamanho proporcionalmente.

.container div{

                flex: 1;

                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }



Apesar de termos colocado todos os componentes para serem flexíveis podemos definir valores de outras formas:

.container div{

                width: 30px;

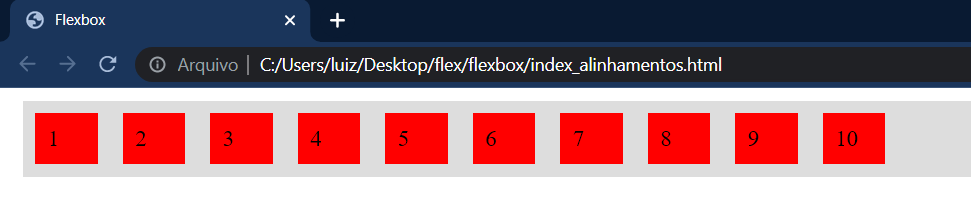
                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }



Se colocarmos 200px os valores extrapolam o tamanho da linha, porém eles assumem um valor menor que os 200px pra que fique na mesma linha.

Para que um item vá pra linha debaixo, utiliza-se o wrap, que por padrão não vem habilitado no flexbox. Se usarmos no-wrap é o padrão.

.container{

                display: flex;

                flex-wrap: wrap;

                background-color: #DDDDDD;

                margin: 10px;

            }

            .container div{

                width: 200px;

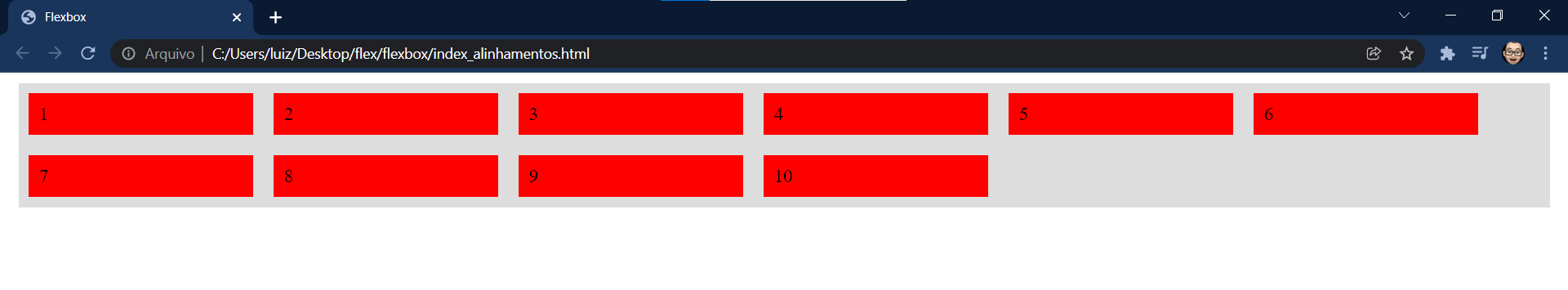
                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }

****

Podemos definir tanto o flex-direction, quanto o flex wrap em uma única propriedade, que é o flex-flow.

.container{

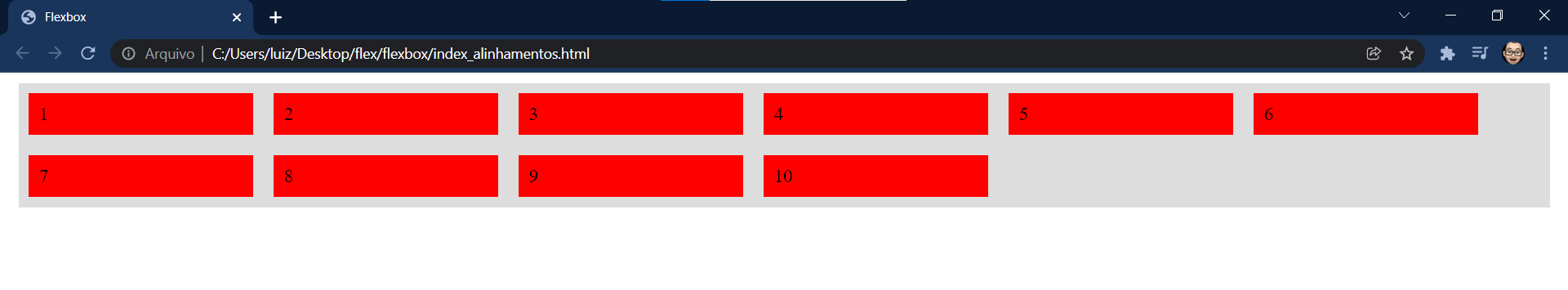
                display: flex;

                flex-flow: row wrap;

                background-color: #DDDDDD;

                margin: 10px;

            }

****

Mudar o flex-flow para columm wrap para testar.

Remover as divs, deixando apenas 3. Alterar a largura para 100px:

<!DOCTYPE html>

<html lang="pt-br">

    <head>

        <meta charset="UTF-8">

        <title>Flexbox</title>

        <style type="text/css">

            .container{

                display: flex;

                background-color: #DDDDDD;

            }

            .container div{

                width: 100px;

                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }

        </style>

    </head>

    <body>

        <div class="container">

            <div>1</div>

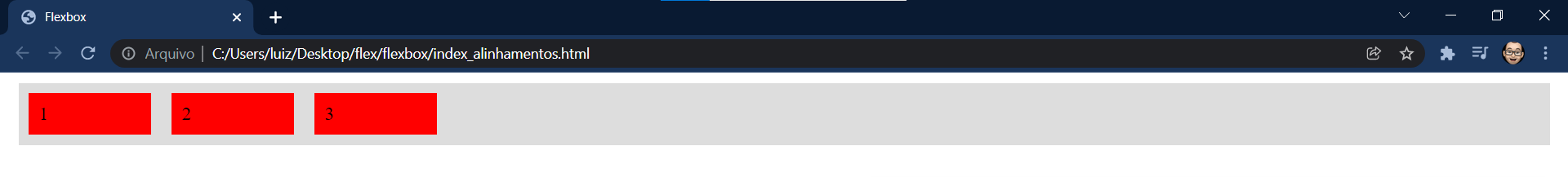
            <div>2</div>

            <div>3</div>

        </div>

    </body>

</html>



Vamos agora trabalhar efetivamente com os alinhamentos. O justify content vai justificar o conteúdo em relação ao container:

.container{

                display: flex;

                justify-content: center;

                background-color: #DDDDDD;

            }



Obs. Essa mesma simulação sem o flexbox é um PARTO!!!

Vamos agora colocar uma altura, porém sem incluir rolagem:

.container{

                display: flex;

                height: 500px;

                justify-content: center;

                background-color: #DDDDDD;

            }



Vamos agora fazer com que os elementos sejam um quadrado, por eles se esticaram pra atender o aumento do contêiner:

.container div{

                width: 100px;

                height: 100px;

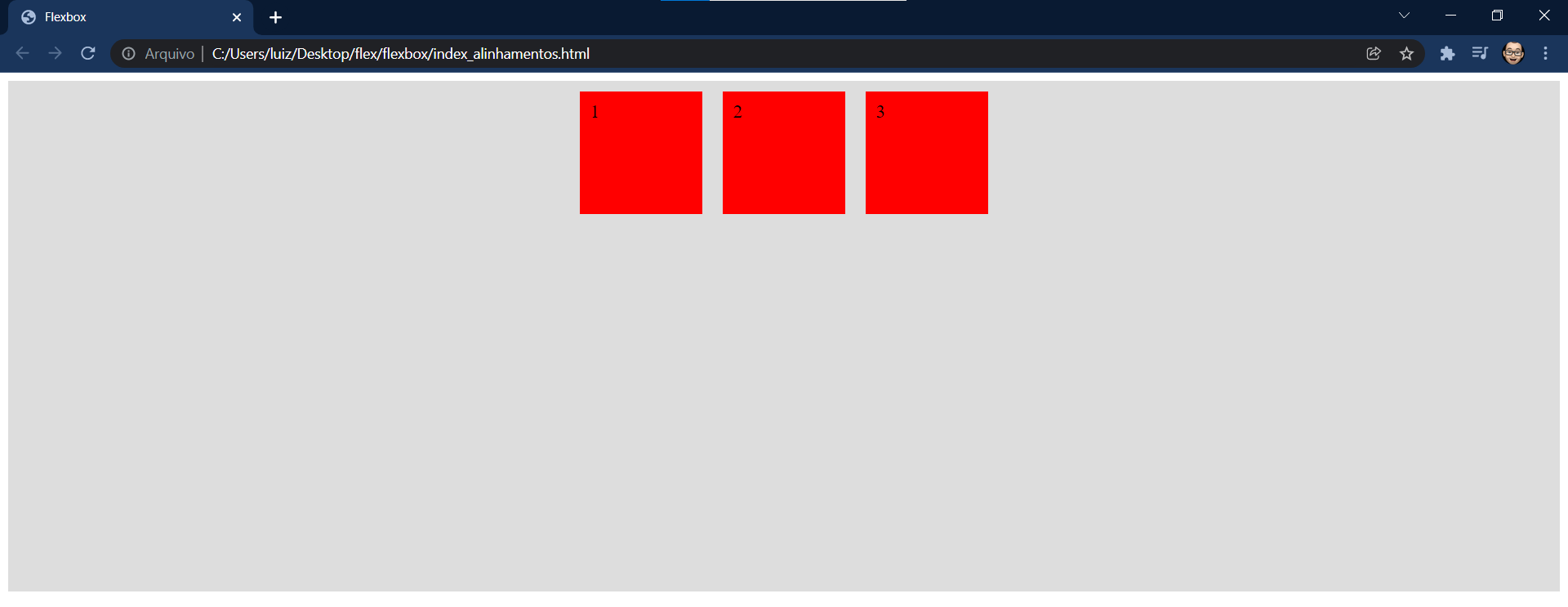
                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }



Note que o justify-content realiza o alinhamento na horizontal. Para alinhar nas colunas usamos o align-items.

.container{

                display: flex;

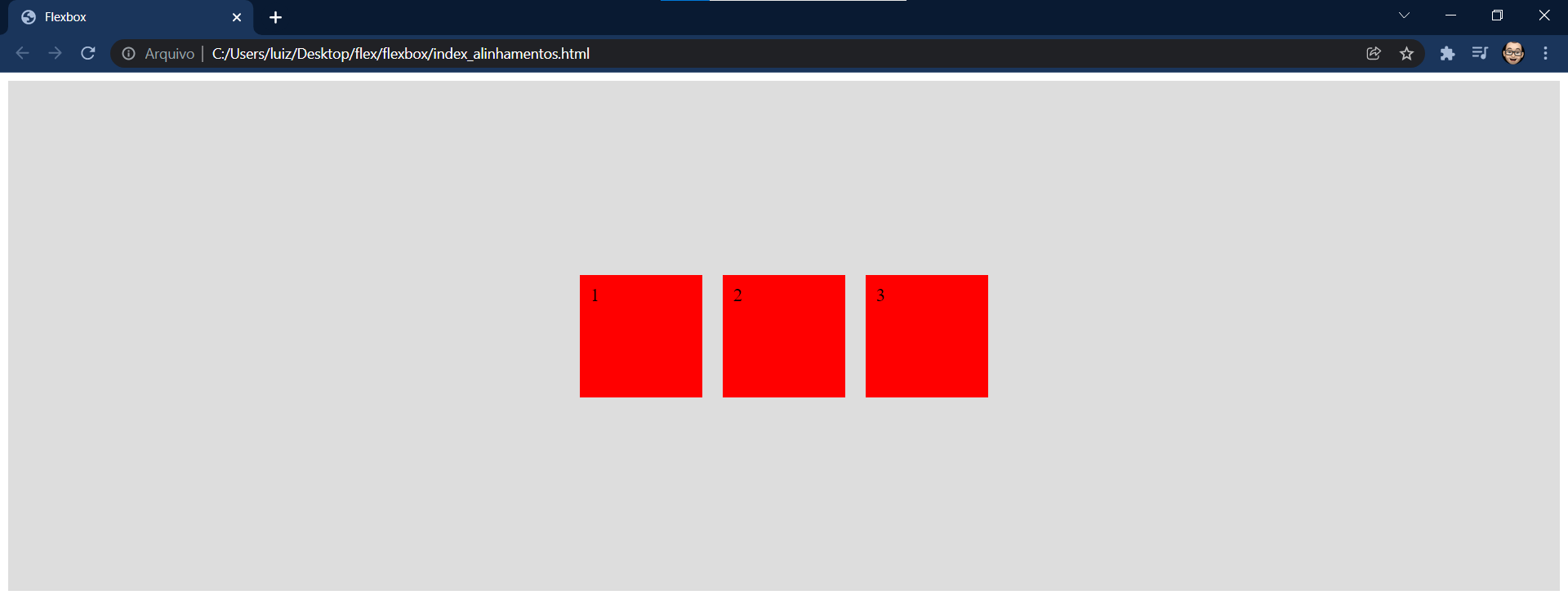
                height: 500px;

                justify-content: center;

                align-items: center;

                background-color: #DDDDDD;

            }

****

O justify contente alinha na direção em que esta sendo exibida, já o align-items vai alinhar na posição contrária. Vamos deixar apenas o justify-content e vamos colocar o flex-direction: row.

.container{

                display: flex;

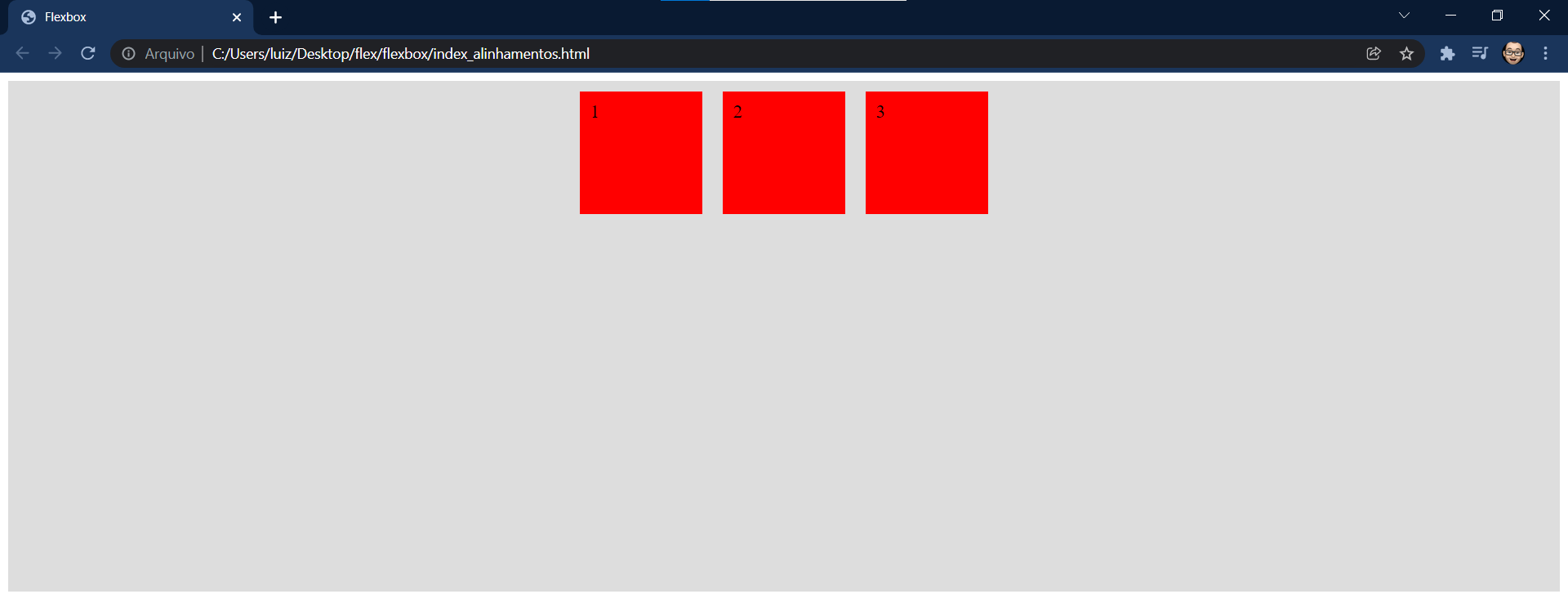
                height: 500px;

                flex-direction: row;

                justify-content: center;

                background-color: #DDDDDD;

            }



Veja que foi alinhado horizontalmente, pois a exibição dos itens está horizontalmente.

Vamos agora trocar para column:

 .container{

                display: flex;

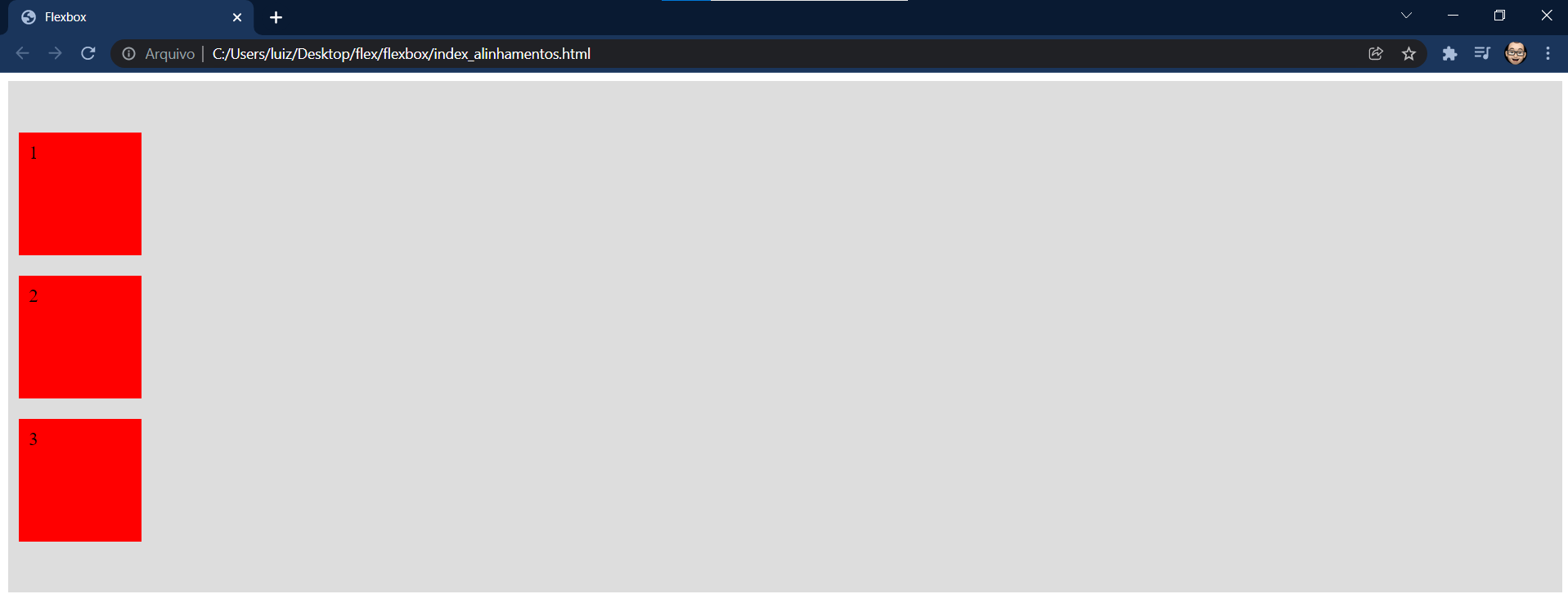
                height: 500px;

                flex-direction: column;

                justify-content: center;

                background-color: #DDDDDD;

            }



Veja que agora ele centralizou na vertical e nesse caso para ir para o meio da tela iremos utilizar o align-items:

.container{

                display: flex;

                height: 500px;

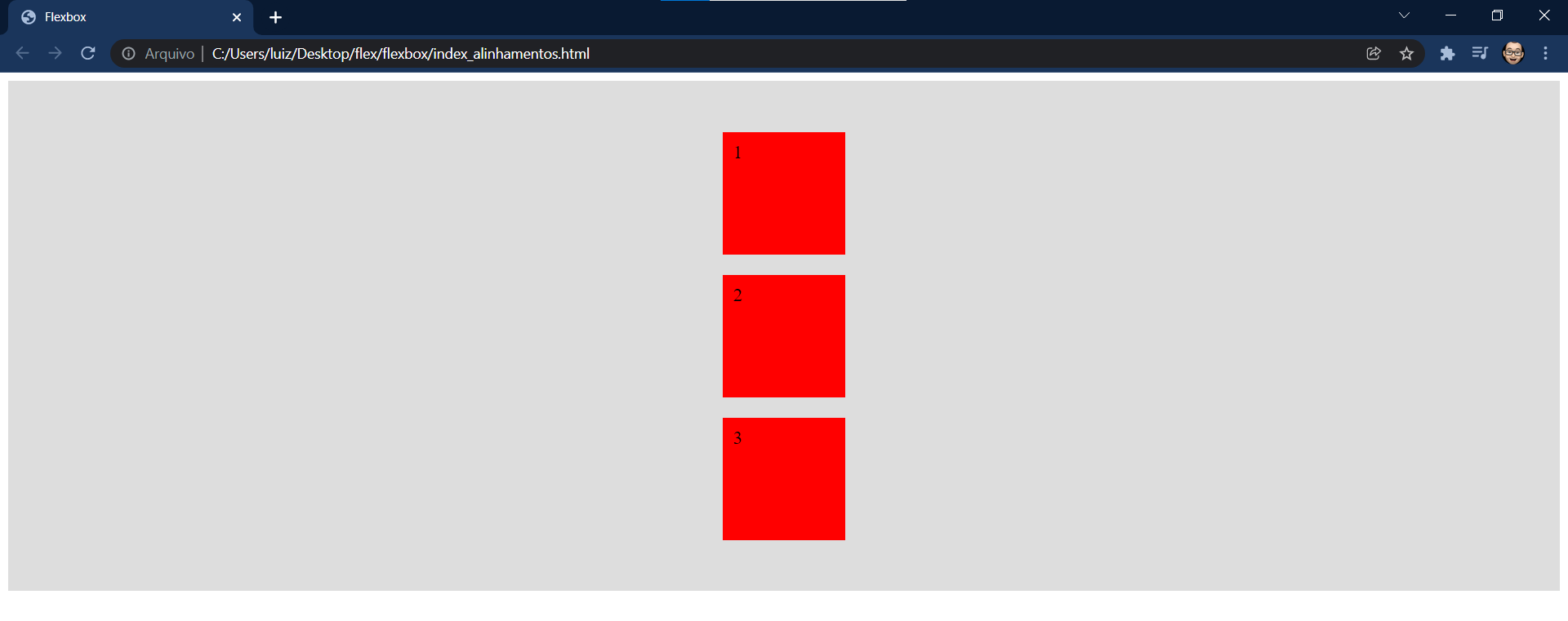
                flex-direction: column;

                justify-content: center;

                align-items: center;

                background-color: #DDDDDD;

            }

****

O Justify-content possui outras opções, vamos deixar da seguinte maneira:

.container{

                display: flex;

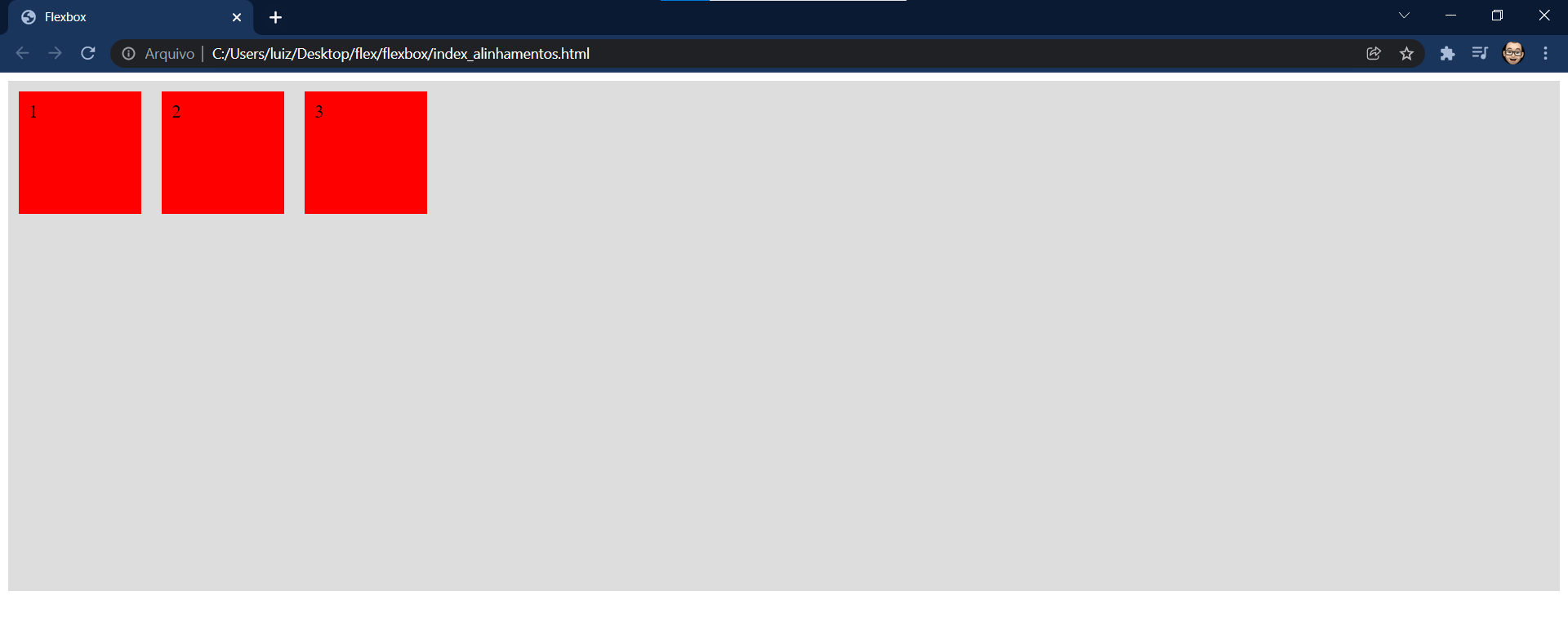
                height: 500px;

                flex-direction: row;

                justify-content: flex-start;

                background-color: #DDDDDD;

            }

****

Fazer os seguintes testes com o justify-items:

Flex-end; // retirar o flex-end e mudar o Direction para row-reverse para mostrar a diferença.

Space-around;

Space-between; (colocar mais uma div4 para ficar melhor de enxergar).

Todos os alinhamentos feitos no justify-items podem ser feitos no align-items.

(brincar com todos esses alinhamentos de justify-items e align-items, misturando eles e posicionando em todos cantos da tela).

Agora vamos trabalhar com algumas propriedades adicionais: O align-self serve para sobrescrever o align-items do contêiner:

<!DOCTYPE html>

<html lang="pt-br">

    <head>

        <meta charset="UTF-8">

        <title>Flexbox</title>

        <style type="text/css">

            .container{

                display: flex;

                height: 500px;

                flex-direction: row;

                justify-content: center;

                align-items: center;

                background-color: #DDDDDD;

            }

            .container div{

                width: 100px;

                height: 100px;

                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }

            .div10{

                align-self: flex-start;

            }

        </style>

    </head>

    <body>

        <div class="container">

            <div>1</div>

            <div>2</div>

            <div>3</div>

            <div>4</div>

            <div>5</div>

            <div>6</div>

            <div>7</div>

            <div>8</div>

            <div>9</div>

            <div class="div10">10</div>

            <div>11</div>

            <div>12</div>

            <div>13</div>

            <div>14</div>

            <div>15</div>

        </div>

    </body>

</html>

****

Para mudar a ordem dos itens sem mexer com o html:

.div10{

                order: 1;

            }

****

Criar uma class div14 e coloca-la como 2 para ela ficar depois do 10.

Todos items que não colocamos a ordem ficam default como 0. Colocando -1 o item vai para o começo.

Vamos agora deixar nosso layout com duas linhas mudando o flex-wrap:

<!DOCTYPE html>

<html lang="pt-br">

    <head>

        <meta charset="UTF-8">

        <title>Flexbox</title>

        <style type="text/css">

            .container{

                display: flex;

                height: 500px;

                flex-direction: row;

                flex-wrap: wrap;

                justify-content: center;

                align-items: center;

                background-color: #DDDDDD;

            }

            .container div{

                width: 100px;

                height: 100px;

                background-color: #FF0000;

                margin: 10px;

                padding: 10px;

                font-size: 18px;

            }

        </style>

    </head>

    <body>

        <div class="container">

            <div>1</div>

            <div>2</div>

            <div>3</div>

            <div>4</div>

            <div>5</div>

            <div>6</div>

            <div>7</div>

            <div>8</div>

            <div>9</div>

            <div>10</div>

            <div>11</div>

            <div>12</div>

            <div>13</div>

            <div>14</div>

            <div>15</div>

        </div>

    </body>

</html>



Vamos usar a propriedade align-content, para gerar um alinhamento para todo o conteúdo:

(flex-end, flex-start, center, space-between, space-around)