

IOS – Instituto de
Oportunidade Social

CSS 14 - Grid e FlexBox



- Compreender a criação de diferentes layouts com o uso do **Grid CSS** e flexbox;
- Conhecer as diversas **propriedades** do Grid CSS e flexbox;
- Aplicar os **recursos** do Grid CSS e flexbox nas folhas de **estilo**.

IOS – Instituto de
Oportunidade Social

Grid e FlexBox



CSS Grid Layout e Flexbox

Você pode e deve usar o **CSS Grid Layout** e o **Flexbox** juntos no desenvolvimento de páginas web. As duas ferramentas irão contribuir para que você desenvolva layouts mais profissionais. Por isso, nessa aula você vai aprender a utilizar o **CSS Layout** e o Flexbox para criar o layout de uma página web.

Gerador de Lero-Lero

<https://lerolero.com/>



Dica!

Boa parte do texto aqui foi gerado automaticamente com a função `lorem` no Visual Studio Code. Essa função gera palavras aleatórias sem sentido. Você pode testar gerar 10 palavras aleatórias com o comando `Lorem10` no editor do VS Code e pressione **Enter** ou **TAB** e veja o resultado.

Importando fonte do Google Fonts

```
<head>  
  <meta charset="UTF-8">  
  <link  
    href="https://fonts.googleapis.com/css2?family=Oswald&display=swap"  
    rel="style sheet">  
  <link rel="stylesheet" href="estilo.css">  
  <title>Galeria de Arte</title>  
</head>
```

Página sem formatação

```
<body>
```

```
  <header class="main-header">
```

```
    <h1>Galeria de <span>Pinturas em Acrílico</span></h1>
```

```
    <p>          Lorem ipsum dolor sit amet consectetur adipisicing  
    elit.      Molestias,      quam!      </p>
```

```
  </header>
```

```
  <main class="container">
```

```
    ... Sections ...
```

```
  </main>
```

```
</body>
```

Section sem Lado

```
<section class="card">  
    
  <div>  
    <h3>Pintura em Acrílico Um</h3>  
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
    Quod error enim laudantium, animi veniam libero eveniet culpa  
    unde perferendis illo fugit corporis, voluptatibus totam dolorum,  
    maiores magnam officia. Ab, delectus.</p>  
    <a href="#" class="btn">Compre Agora</a>  
  </div>  
</section>
```

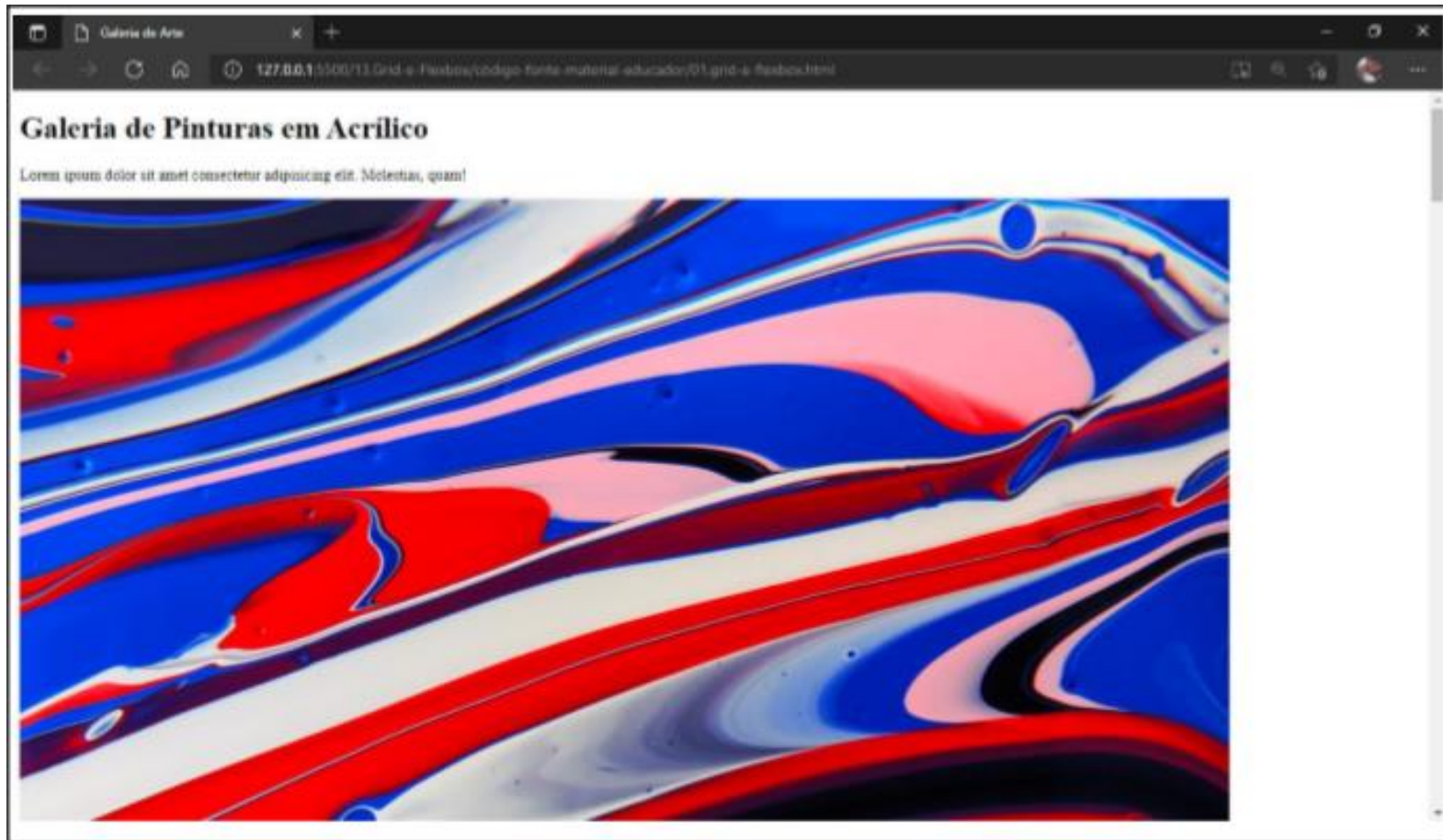

Section Lado Esquerdo

```
<section class="card" data-aos="fade-left">  
    
  <div>  
    <h3>Pintura em Acrílico Dois</h3>  
    <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
    Quod error enim laudantium, animi veniam libero eveniet culpa  
    unde perferendis illo fugit corporis, voluptatibus totam dolorum,  
    maiores magnam officia. Ab, delectus. </p>  
    <a href="#" class="btn">Compre Agora</a>  
  </div>  
</section>
```

Section Lado Direito

```
<section class="card" data-aos="fade-right">  
    
  <div>  
    <h3>Pintura em Acrílico Três</h3>  
    <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
    Quod error enim laudantium, animi veniam libero eveniet culpa  
    unde perferendis illo fugit corporis, voluptatibus totam dolorum,  
    maiores magnam officia. Ab, delectus. </p>  
    <a href="#" class="btn">Compre Agora</a>  
  </div>  
</section>
```

Exemplo de Página sem formatação



Configurando margin, padding e box-sizing

```
@charset "utf-8";
```

```
*{  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box; }
```

```
body {  
    font-size: 1.3rem;  
    font-family: 'Oswald', sans-serif;  
    background: #f9f9f9; }
```

```
a {  
    color: #333;  
    text-decoration: none; }
```

Configurando a classe .main-header

```
.main-header {  
    height: 55vh;  
    padding: 2rem;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    text-align: center;  
}
```

Configurando os elementos `<h1>`, `` e `<p>` do header

```
.main-header h1 {  
    font-size: 4rem;  
    margin-bottom: 2rem;  
    line-height: 1.2rem;  
}
```

```
.main-header h1 span {  
    color: #b50d10;  
}
```

```
.main-header p {  
    font-size: 2rem;  
}
```

Exemplo de Página após formatação de margin, padding e do header



Configurando a classe .container e elementos

```
.container {  
    max-width: 1100px;  
    margin: auto;  
    overflow: auto;  
    padding: 0 2rem;  
}
```

```
img {  
    width: 100%;  
}
```


Configurando a classe .card

```
.card {  
    display: grid;  
    grid-template-columns: repeat(2, 1fr);  
    grid-gap: 2rem;  
    background: #f1f1f1;  
    margin-bottom: 2rem;  
}
```

Exemplo de Página após formatação das classes .container e .card



Configurando os elementos `<h3>`, `` e `<div>` da classe `.card`

```
.card h3 {  
    margin-bottom: 2rem;  
}
```

```
.card img {  
    height: 400px;  
}
```

```
.card>div {  
    padding: 2rem;  
}
```

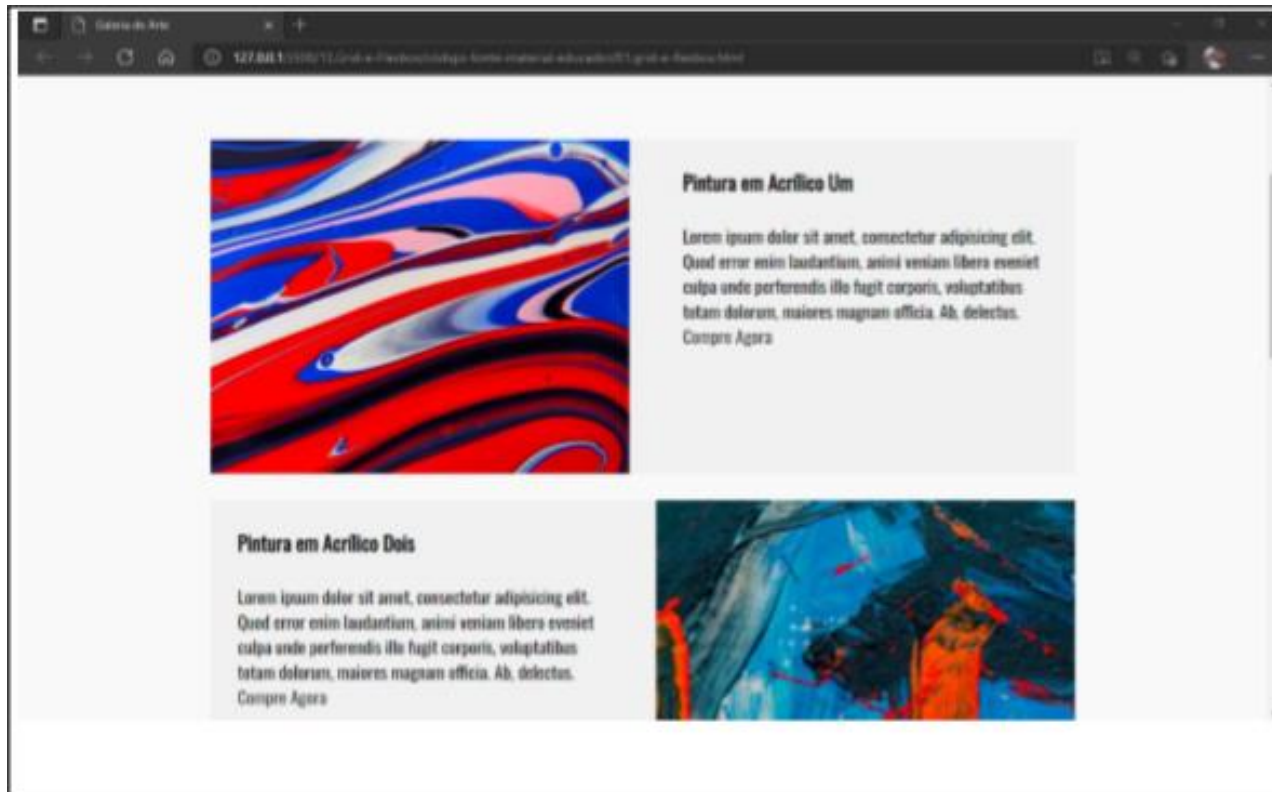
OBS: O símbolo `>` (**maior que**) é chamado **child selector**. E indica que o estilo será aplicado nos elementos filhos.

Exemplo de Página após formatação dos elementos do .card



Alternando a ordem dos elementos ímpares

```
.card:nth-child(even) img { order: 2; }
```



Configurando a classe .btn

```
.btn {  
    display: inline-block;  
    background: #000;  
    color: #fff;  
    padding: 0.8rem 1.8rem;  
    margin-top: 2rem;  
    cursor: pointer;  
}
```

```
.btn:hover {  
    opacity: 0.8;  
}
```

Exemplo de Página após formatação da classe .btn



IOS – Instituto de
Oportunidade Social

Vamos Praticar



Apostila de CSS:

- Grid e Flexbox com suas Propriedades

Páginas 181 a 189

OBS: Acompanhar o passo a passo com o instrutor (para a prática esse exercício será necessário baixar imagens do material de apoio)

IOS – Instituto de
Oportunidade Social

Exercícios



Montar uma página HTML com nome **gridFlexbox.html** com seu par **CSS** atendendo os seguintes critérios:

- Montar uma história em quadrinhos alternando entre textos (<p>) e imagens ()
- Possuir pelo menos **3 linhas de Grid**, sendo 1 delas com 3 colunas (**img, p e img**) e as outras duas linhas com 2 colunas cada alternando a imagem entre **esquerda e direita** conforme os exemplos da aula
- Publicar no GitHub e enviar no Moodle