

CSS – Posicionando Elementos

Rodrigo Attique
Desenvolvedor

CSS – Posicionamento

- Para posicionar elementos na página precisamos definir qual será a forma usada.
- Sem isso tudo que for escrito será exibido sempre da direita para a esquerda e de cima para baixo.
- Ou de forma “empilhada”.



CSS – Posicionamento

- Antes de aprender como posicionar é preciso entender alguns conceitos.
- E o primeiro deles é o de caixa.
- Cada elemento para o CSS é uma caixa que pode ser empilhada ou justaposta (lado-a-lado).



Conceito de Box Model

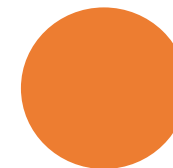
- No CSS, cada elemento é considerado como uma caixa retangular. Essa caixa é descrita por um modelo conhecido como Box Model, que é dividido em quatro áreas:
- Conteúdo (content);
- Preenchimento (padding);
- Borda (border); e
- Margem (margin).
- Juntas, essas áreas definem como os elementos se relacionam entre si e como eles ocupam espaço na página.



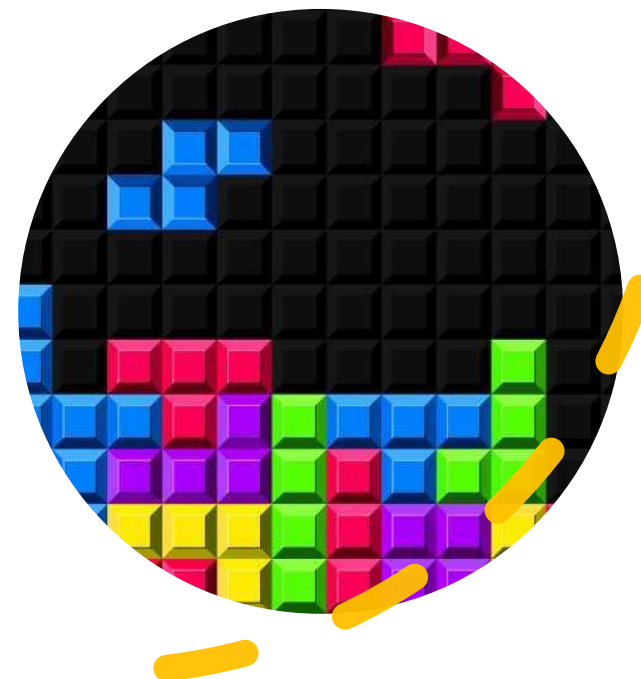
Margem, borda e preenchimento (margin, border, padding)

- Agora, vamos desempacotar cada parte do Box Model:
- **Margem (margin): é o espaço ao redor da caixa. Pense nela como um buffer que separa o elemento dos outros ao seu redor;**
- **Borda (border): é a linha que envolve o preenchimento e o conteúdo;**
- **Preenchimento (padding): é o espaço entre a borda e o conteúdo. Ele é como uma almofada que protege o conteúdo da borda;**

Posicionamento de elementos



- O posicionamento de elementos em CSS pode parecer uma versão de Tetris às vezes, onde estamos tentando encaixar diferentes blocos em um espaço limitado. Temos várias propriedades para controlar o posicionamento, incluindo:
 - position;
 - top;
 - right;
 - bottom;
 - left; e
 - z-index.
- Com elas, você pode mover elementos para cima e para baixo, esquerda e direita, e até mesmo controlar o que fica na frente e o que fica atrás.





A tag DIV

- Como o próprio nome sugere, essa tag HTML serve para criar divisões internas no código.
- Geralmente usamos essa tag genérica combinada com uma classe.
- Div's pode receber outras tags incluindo divs, também podemos inserí-las dentro de outras tags como H1, P, A....
- Div's podem funcionar como verdadeiros containers.

Exemplo prático

- Vamos criar várias divs com classes representativas.
 - caixa-principal: que receberá as demais divs
 - caixa-azul: div que ficará azul
 - caixa-verde: que ficará verde
 - caixa-lilas, caixa-amarela
- No resumo as caixas coloridas estarão dentro da caixa principal.
- Para isso vamos trabalhar em um arquivo chamado `divs.html`.


```

13  ∨ <body>
14  ∨   <div class="caixa-principal">
15      <div class="caixa-azul"><
16      <div class="caixa-verde">
17      <div class="caixa-lilas">
18      <div class="caixa-amarela
19  </div>

      background-color: ■blueviolet;
      height: 100px; /* altura */
      width: 100px; /* largura */
  }
  .caixa-amarela {
      background-color: ■yellow;
      height: 100px; /* altura */
      width: 100px; /* largura */
  }
</style>
</head>

```

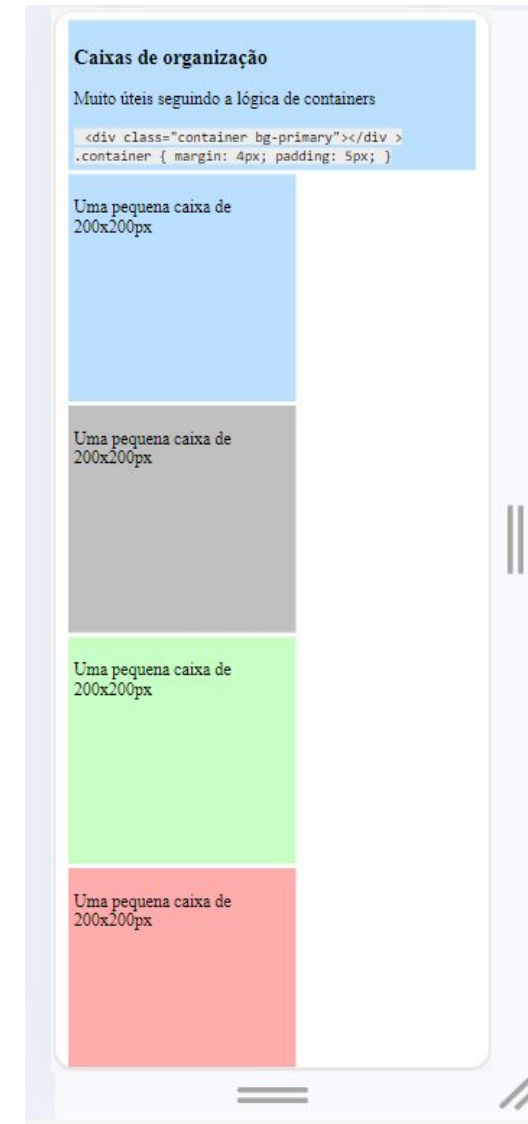
```

<style>
  .caixa-principal {
      border: solid 1px;
      padding: 2px;
  }
  .caixa-azul {
      background-color: ■blue;
      height: 100px; /* altura */
      width: 100px; /* largura */
  }
  .caixa-verde {
      background-color: ■green;
      height: 100px; /* altura */
      width: 100px; /* largura */
  }
  .caixa-lilas {
      background-color: ■blueviolet;

```

Exercício

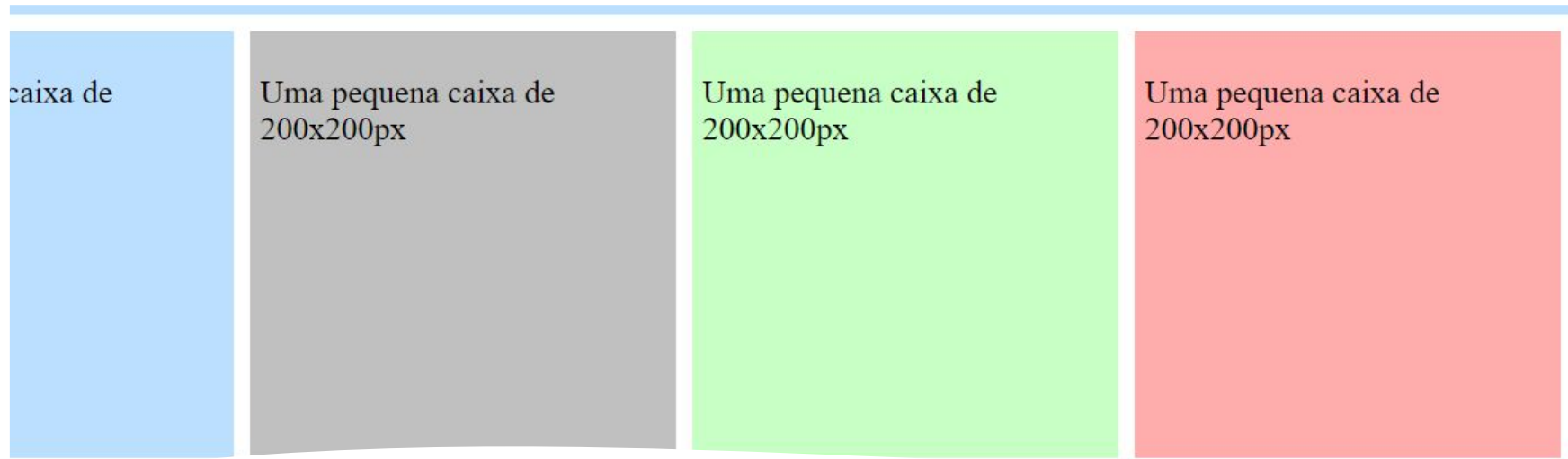
- Usando o que você já aprendeu tente reproduzir o efeito da imagem.





Posicionando elementos

- **É possível posicionar elementos na página usando CSS.**
- **Podemos basicamente posicionar qualquer elemento em qualquer parte da página.**
- **Podemos fazer de forma estática ou de forma dinâmica usando combinações simples.**



Posicionar elementos

- **É possível organizar os elementos como na foto acima, não necessariamente usando divs.**
- **Mas o mais comum para o efeito acima é usar divs aninhadas com um efeito chamado display: flex;**

Estrutura básica display flex

```
33      <h3>Display Flex</h3>
34  ✓    <div >
35        <p>Elemento 1</p>
36        <p>Elemento 2</p>
37        <p>Elemento 3</p>
38      </div>
```


Usando css

Display Flex

Elemento 1Elemento 2Elemento 3

```
71 .linha {  
72     display: flex;  
73 }  
74
```

Não se esqueça de selecionar essa classe na div
<div class="linha">

- **O código anterior apenas exibirá o texto organizado em parágrafos.**
- **Repare na div da linha 34, ela apenas é uma div não vinculada a um estilo.**
- **Tudo o que precisa ser feito é criar uma classe para essa div e depois fazer ajustes nos parágrafos.**

Exercício


A dica aqui é definir a classe de menu
Com as especificações padrões
Depois definir outra classe menu-inline com o
Display flex.
Por fim precisamos definir uma classe composta
Com .menu .menu-item para tratar os elementos da lista.

- Usando display flex faça com que o menu do site usando listas apareça lado a lado como na imagem.
- Repare que marcamos as classes que vão fazer o efeito.



[Página Inicial](#) [Contato](#) [Cardápio](#)

```
10 <ul class="menu menu-inline">Menu
11   <li class="menu-item"><a href="index.html">Página Inicial</a></li>
12   <li class="menu-item"><a href="contato.html">Contato</a></li>
13   <li class="menu-item"><a href="cardapio.html">Cardápio</a></li>
14 </ul>
```

```
65  .menu {
66      background-color:  #eeeeeee;
67      list-style: none;
68      margin: 2px 2px 2px 2px;
69      padding: 5px 5px 5px 5px;
70  }
71
72  .menu-inline {
73      display: flex;
74  }
75
76  .menu .menu-item {
77      padding: 4px;
78  }
79
```

Considerações finais

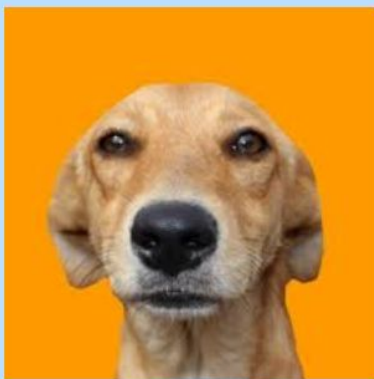
- Não há tempo hábil para tratar todos os exemplos do uso do display.
- Vamos aproveitar a vasta documentação disponível na internet a nosso favor.
- <https://origamid.com/projetos/flexbox-guia-completo/>
- O site acima possui diversos exemplos de uso de display flex e o conceito de grades.

DESAFIO

Usando o que já aprendeu até aqui tente reproduzir a imagem.

Conceito de cards

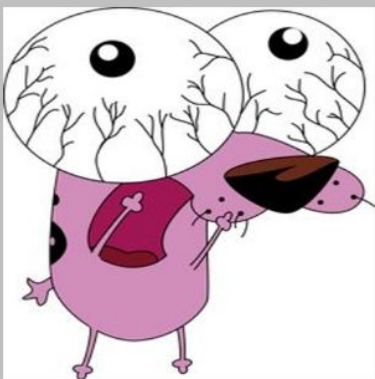
Primeiro Item



Sou viralatas caramelo, perseguidor de motoqueiros e ciclistas...

Saiba mais

Segundo Item



Coragem era um desenho animado exibido pelo SBT que fez muito sucesso

Saiba mais

Terceiro Item



Geralmente quando algo nos deixa sem reação ou pensativos dizemos que estamos intrigados.

Saiba mais

Dicas

- Você vai precisar primeiro definir um container.
- Depois um card.
- Os cards sempre vão dentro do container.
- No CSS você deve usar o display flex para permitir que o container alinhe os cards lado a lado.
- Use a documentação para tentar chegar no resultado.
- NOTA: O código do menu pode ajudar.