HTML e CSS prático

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Box Model
- Introdução ao flexbox
- Material Complementar

ANTES DE COMEÇAR...

Um aquecimento de CSS
Kahoot!

BOX MODEL

Todo elemento **HTML** pode ser considerado uma caixa (*box*).

De fato, quando passamos a olhar para um ***mockup**** e separamos o conteúdo em caixas, nosso trabalho de desenvolvimento fica mais simples.

Mockup - “maquete”, desenho do layout que queremos desenvolver

BOX MODEL

O box model do CSS nos ajuda a entender como as nossas “caixas” de elementos se posicionam e interagem entre si em um contexto de espaço e posicionamento.

As propriedades css que usamos para manipular nossas caixas são **padding**, **margin** e **border**.

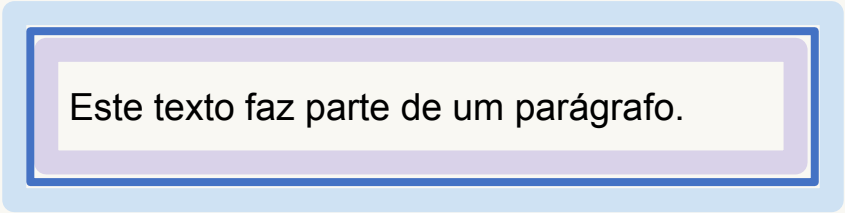
BOX MODEL

- **margin**: margem externa do elemento
- **padding**: margem interna do elemento
- **border**: borda do elemento

BOX MODEL

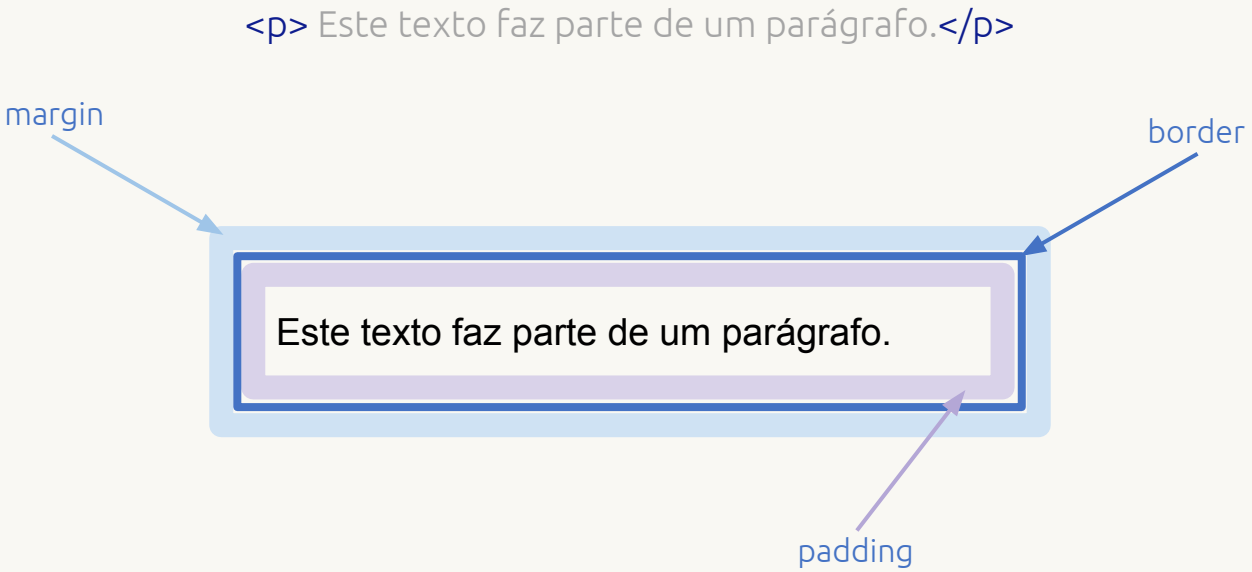
O box model:

```
<p> Este texto faz parte de um parágrafo.</p>
```



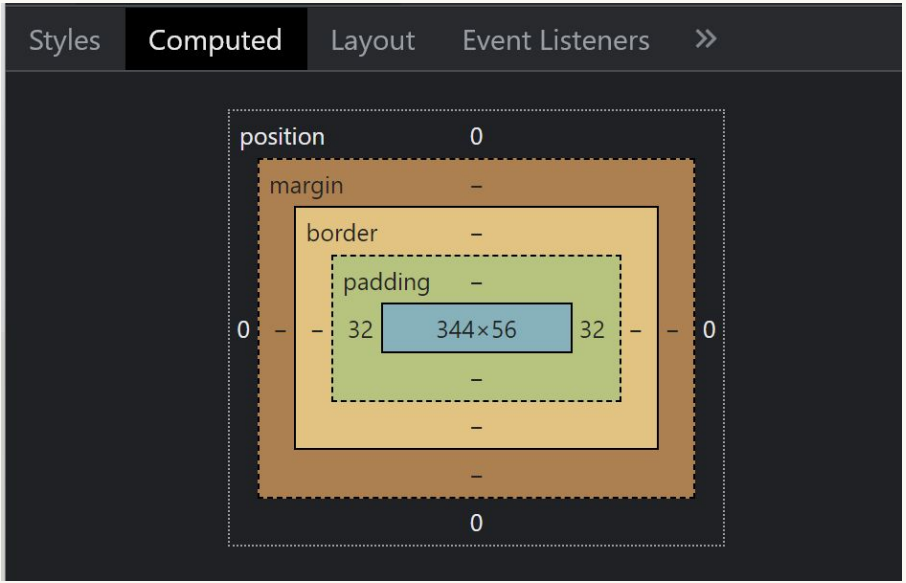
BOX MODEL

O box model:



BOX MODEL

Podemos verificar o box model de um elemento na página usando o *developer tools* (ferramentas do desenvolvedor) do navegador!



Mãos à obra!

BOX MODEL

Os elementos HTML podem ser categorizados, de acordo com a sua exibição (*display*), em 3 categorias principais:

- Block (*bloco*)
- Inline (*em linha*)
- Inline-block

BOX MODEL

Os elementos HTML podem ser categorizados, de acordo com a sua exibição (*display*), em 3 categorias principais:

- Block (*bloco*)
- Inline (*em linha*)
- Inline-block

Vamos ver sobre **flex** mais pra frente!

BOX MODEL

- **Block (*bloco*)**

Elementos com o **display: block** ocupam toda a largura da tela, fazendo com que haja uma “quebra de linha”.

Alguns elementos HTML já possuem, por padrão, o display configurado para block.

Exemplo: div, p, h1 - h6, article, section...

BOX MODEL

- **Inline (*em linha*)**

Elementos com o display: inline ocupam somente o espaço necessário para a exibição do seu conteúdo. Quando usamos mais de um elemento inline em conjunto, eles tendem a se ajustarem em uma mesma “linha” horizontal.

Exemplos: span, strong, img...

Mãos à obra!

BOX MODEL

Entender esta configuração de exibição junto com o conceito de **box model** vai nos ajudar a posicionar os nossos elementos na tela corretamente, fazendo com que possamos controlar o tamanho dos elementos, tanto no **eixo x (width)** como no **eixo y (height).**

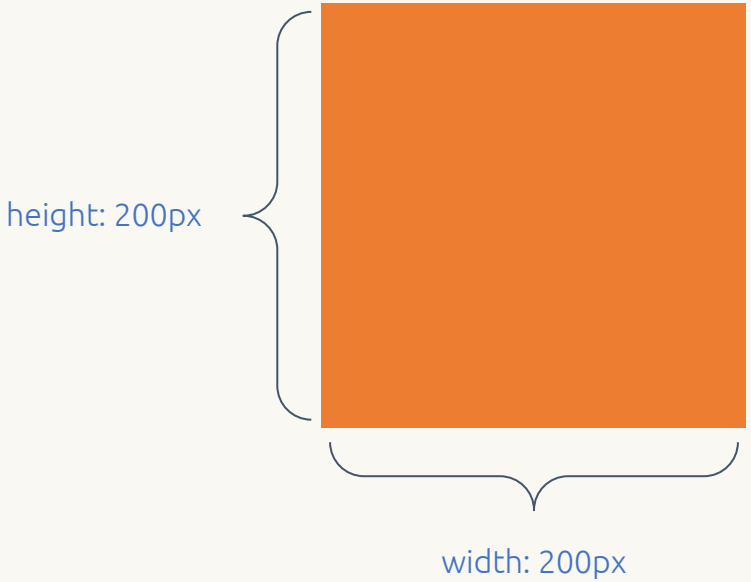
BOX MODEL

Exemplo:

```
<div class="quadrado"></div>
```

BOX MODEL

Exemplo:



Mãos à obra!

BOX MODEL

Importante entender, também, que, na configuração padrão de CSS do navegador, existe uma propriedade chamada **box-sizing** com o valor **content-box**, que define que os valores de height e width afetarão o tamanho do conteúdo da box, **desconsiderando** a **margin**, **padding** e **border**.

O **tamanho total** da box do elemento será a **soma do tamanho interno com os valores da borda e das margens interna e externa**.

BOX MODEL

Quando alteramos a propriedade **box-sizing** para **border-box**, o tamanho total do elemento passa a ser o limite da **border**, mas não incluem a margem (**margin**).

Esta é a configuração mais utilizada. De fato, é tão comum que costumamos iniciar nossas folhas de estilos aplicando o box-sizing de forma global com o seletor *.

Mãos à obra!

BOX MODEL

Revisando as propriedades:

- **margin**: margem externa do elemento
- **padding**: margem interna do elemento
- **border**: borda do elemento
- **width**: largura do elemento
- **height**: altura do elemento
- **display**: modo de visualização do elemento (block, inline, inline-block, none)
- **box-sizing**: configuração da forma como o tamanho do nosso box é calculado.

Lembrando: as propriedades **width**, **height**, **margin** e **padding** funcionam em elementos *block* e *inline-block*!

FLEXBOX

Uma das ferramentas que o CSS disponibiliza para posicionamento de elementos em tela é o **Flexbox**.

Com ele, conseguimos controlar o fluxo dos elementos em tela, posicionando-os conforme desejamos que os elementos se “**encaixem**”

FLEXBOX

Para usarmos o flexbox, precisamos alterar nosso elemento com a propriedade **display: flex**.

A melhor maneira de entendermos o **flexbox** é botando a mão na massa.

Mãos à obra!

FLEXBOX

Revisando as propriedades do flexbox:

- **display: flex** : transforma o elemento em uma caixa flex
- **flex-direction**: determina a direção do fluxo dos elementos
- **flex-wrap**: configura se, ao chegar ao limite da tela, os itens vão quebrar a linha
- **align-items**: alinhamento do elemento no eixo y (*flex-direction: row*)
- **justify-content**: alinhamento do elemento no eixo x (*flex-direction: row*)
- **flex-grow**: configura o item flex para ocupar o maior espaço possível horizontalmente

MATERIAL COMPLEMENTAR

- Box Model - MDN Box Model
- Propriedades shorthand - MDN Shorthand
- Propriedade Border - W3Schools Border
- Playground Flexbox - https://flexbox.tech/
- CSS Grid System - MDN Grid Layout

DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!

SENAI

<LAB365>