

**Professores:**

Celso Giusti

Daniel Manoel Filho

Marlon Palata Fanger Rodrigues

# Box Model



# Introdução ao Box Model

Nesta aula, vamos aprender o conceito **mais importante** do CSS: o **Box Model** (Modelo de Caixa).

**O Conceito:** No CSS, **todo elemento HTML é uma caixa retangular**. O seu `<h1>`, seu `<p>`, sua `<img>`, sua `<div>`... todos são caixas.

O CSS nos dá o poder de controlar o tamanho e o espaçamento dessa caixa. Dominar o Box Model é o primeiro passo para criar qualquer layout.

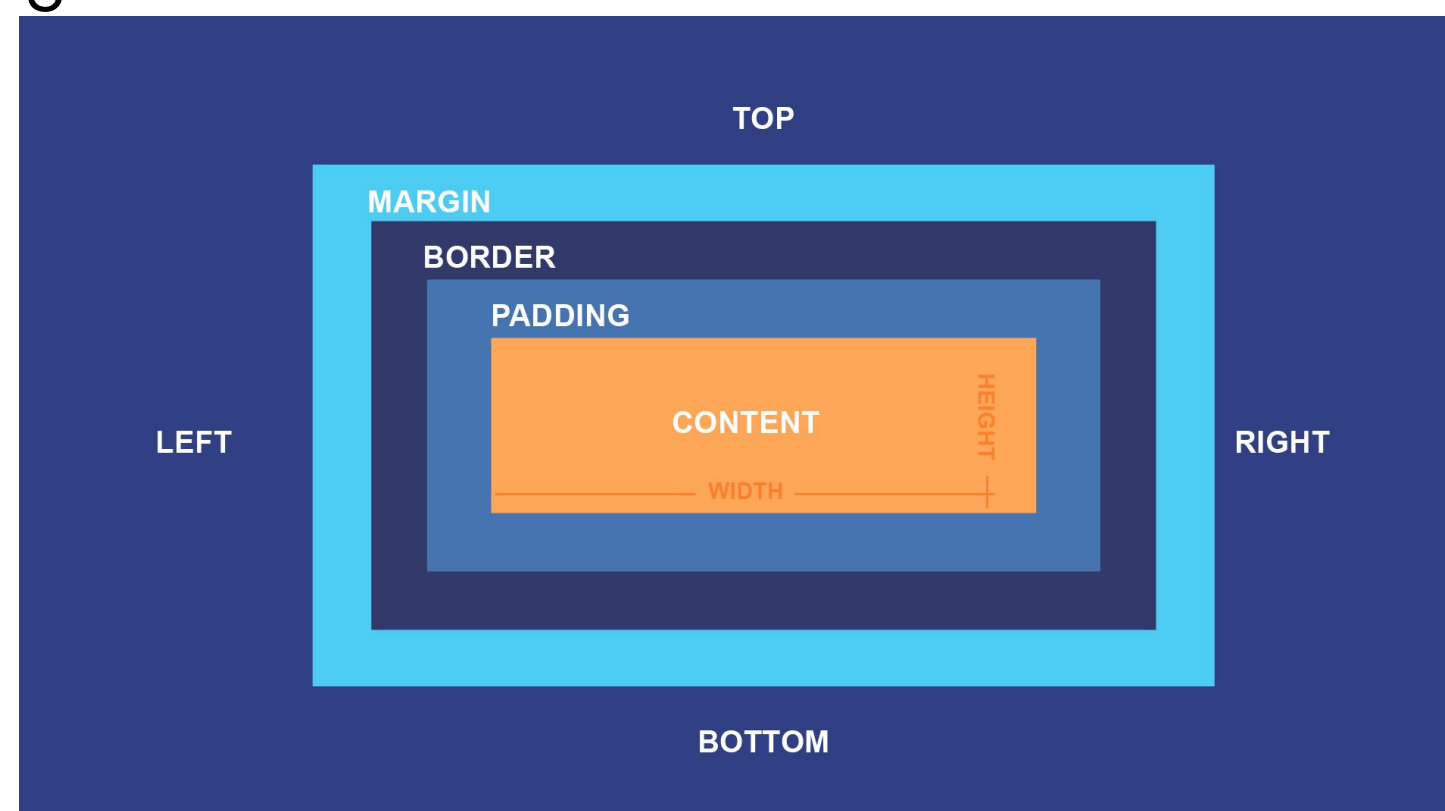
## Tópicos de Hoje:

- O que é o **Box Model**?
- **As 4 Camadas:** Content, Padding, Border, Margin
- A propriedade **box-sizing** (A MAIS IMPORTANTE!)
- **Aplicação Prática:** Estilizando Menus e Submenus

# O que é o Box Model?

Toda caixa HTML é composta por quatro camadas, de dentro para fora:

1. **Content (Conteúdo):** Onde seu texto, imagem, ou vídeo aparecem. É o "recheio" da caixa.
2. **Padding (Preenchimento):** Uma área transparente que cria espaço *dentro* da caixa, entre o conteúdo e a borda.
3. **Border (Borda):** A linha que fica ao redor do padding e do conteúdo.
4. **Margin (Margem):** Uma área transparente que cria espaço *fora* da caixa, "empurrando" outros elementos para longe.



# As Quatro Camadas Explicadas

## Content (Conteúdo):

- Controlado pelas propriedades **width** e **height**.

## Padding (Preenchimento):

- Controlado por **padding**, padding-top, padding-right, padding-bottom, padding-left.
- A cor de fundo (background-color) do elemento se estende por esta área.
- **Pense:** "Espaçamento interno".

## Border (Borda):

- Controlado por **border**, border-width, border-style, border-color.
- **Pense:** "A moldura da caixa".

## Margin (Margem):

- Controlado por **margin**, margin-top, margin-right, margin-bottom, margin-left.
- É sempre transparente. Não é afetada pelo background-color.
- **Pense:** "Espaçamento externo" ou "espaço pessoal" do elemento.

# box-sizing: content-box (O Padrão)

Por padrão, o CSS usa um modelo chamado **content-box**. Este modelo é confuso e torna o cálculo de layout difícil.

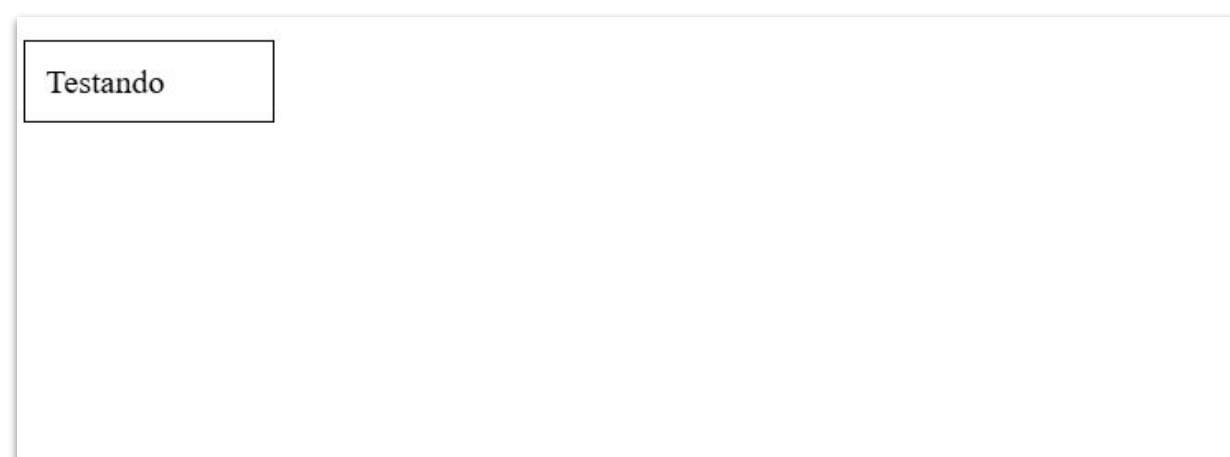
**O Problema:** A **width** que você define se aplica **apenas ao conteúdo**. O padding e a borda são *adicionados* a essa largura.

**Exemplo (content-box):**

**Qual a largura total desta caixa?** 100px (width)

- 10px (padding esquerdo)
- 10px (padding direito)
- 1px (borda esquerda)
- 1px (borda direita) = **122px de largura total na tela!**

```
.caixa {  
  width: 100px;  
  padding: 10px;  
  border: 1px solid black;  
}
```



# box-sizing: border-box (O Modo Profissional)

Felizmente, existe uma forma muito melhor e mais intuitiva de trabalhar, que é o padrão da indústria hoje.

A propriedade **box-sizing: border-box** muda o cálculo: agora, a **width** que você define é a **largura total** da caixa, incluindo **padding** e **border**.

## Exemplo (border-box):

**Qual a largura total desta caixa? Exatos 100px.**

O navegador ajusta o espaço do conteúdo *internamente* para que a caixa (incluindo padding e borda) caiba nos 100px que você pediu.

**Regra de Ouro:** Coloque o seletor universal `*` no topo do seu CSS para aplicar **border-box** a todos os elementos.

```
.caixa2 {  
  box-sizing: border-box; /* A MÁGICA! */  
  width: 100px;  
  padding: 10px;  
  border: 1px solid black;  
}
```

Caixa 1

Caixa 2

# Detalhando margin (Margem)

**margin** controla o espaço *fora* da borda.

- **Shorthand (Abreviatura):** Assim como **padding**, a ordem é Topo, Direita, Baixo, Esquerda (sentido horário).
  - **margin: 10px;** (10px em todos os lados)
  - **margin: 10px 20px;** (10px topo/baixo, 20px laterais)
  - **margin: 10px 20px 30px 40px;** (topo, direita, baixo, esquerda)
- **Centralização de Bloco:** Para centralizar um elemento de bloco (como uma div ou main) que tenha uma **width** definida, use **margin: auto;** nas laterais.

```
.container {  
  width: 80%;  
  margin: 0 auto; /* 0 topo/baixo, auto laterais */  
}
```
- **Colapso de Margem:** Margens verticais (topo/baixo) de elementos adjacentes "colapsam" (se fundem). Apenas a maior margem é aplicada, não a soma.



# Detalhando border e padding

**padding (Preenchimento):** Controla o espaço *dentro* da borda. Usa o mesmo shorthand de margin (sentido horário). O **background-color** preenche o padding.

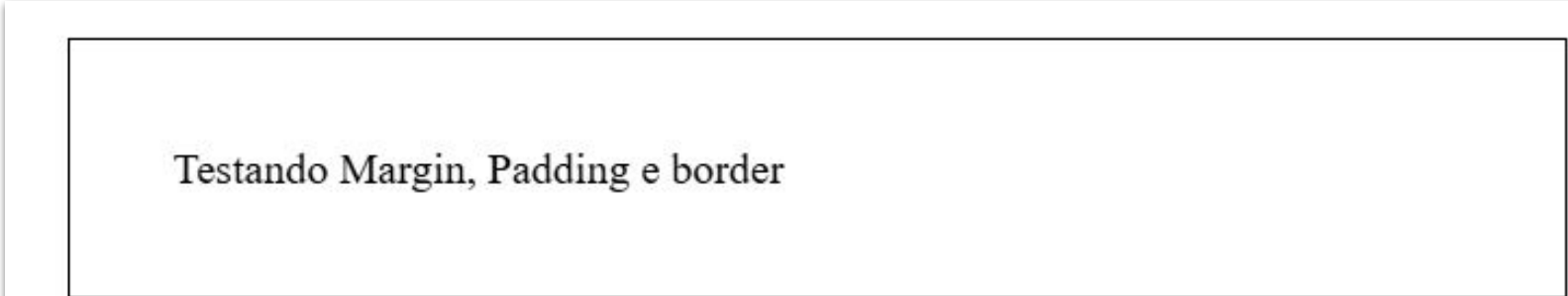
**border (Borda):** A "moldura". A forma shorthand mais comum:

- **border:** [largura] [estilo] [cor];

**Exemplo:** **border:** 1px solid #ccc;

- **Estilos de Borda:** solid, dashed (tracejada), dotted (pontilhada), double, none.

```
.container {  
  width: 80%;  
  margin: 0 auto; /* 0 topo/baixo, auto laterais */  
  border: 1px solid black;  
  padding: 38px;  
}
```



Testando Margin, Padding e border



**Professores:**

Celso Giusti

Daniel Manoel Filho

Marlon Palata Fanger Rodrigues

# Estilização de Menus



# Estilização de Menus

Agora, vamos aplicar o Box Model para resolver um problema comum: estilizar um menu de navegação.

**A Semântica (HTML):** Um menu de navegação é, semanticamente, uma **lista de links**. A estrutura HTML correta é:

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Sobre</a></li>
    <li><a href="#">Contato</a></li>
  </ul>
</nav>
```

- [Home](#)
- [Sobre](#)
- [Contato](#)

**O Problema Visual:** Por padrão, isso aparece como uma lista vertical com marcadores (bullets), o que não parece um menu.

**Nosso Plano (CSS):**

1. Remover os estilos padrão da lista (**list-style**, **margin**, **padding**).
2. Mudar os itens da lista (**<li>**) de **block** para **inline-block** para que fiquem lado a lado (um menu horizontal).
3. Aplicar **padding** aos links (**<a>**) para criar "botões" clicáveis, em vez de aplicar aos **<li>**.

# Resetando Estilos da Lista

O primeiro passo para estilizar um menu é remover os estilos que o navegador coloca por padrão em listas (**<ul>**).

```
/* Selecciona a lista DENTRO da tag nav */
```

```
nav ul {
```

```
/* Remove os marcadores (bullets) */
```

```
list-style: none;
```

```
/* Remove o espaçamento padrão do navegador */
```

```
margin: 0;
```

```
padding: 0;
```

```
/* (Opcional) Define um fundo para a barra de menu */
```

```
background-color: #333;
```

```
}
```



# Criando um Menu Horizontal

Por padrão, `<li>` (item de lista) é um elemento de bloco (**display: block**). Para alinhá-los horizontalmente, podemos usar **display: inline-block**.

- **inline-block** é um híbrido:
  - **inline**: Flui na mesma linha que outros elementos.
  - **block**: Permite que você defina **width**, **height**, **margin** e **padding** (o que **display: inline** não permite totalmente).

```
/* Seleciona os itens da lista
DENTRO da nav */
nav li {
  display: inline-block;
}
```



Home Sobre Contato

Com apenas essa regra, os itens do menu agora ficarão lado a lado.

# Aplicando o Box Model nos Links

Onde devemos aplicar o **padding** para fazer o "botão" do menu? No `<li>` ou no `<a>`?

**Resposta:** Sempre no link (`<a>`)!

**Por quê?** O link é o elemento clicável. Se aplicarmos **padding** ao `<li>`, a área clicável continua sendo apenas o texto. Se aplicarmos ao `<a>` e o transformarmos em **display: block**, a *área inteira* do padding se torna clicável.

```
/* Selecciona os links DENTRO dos itens da lista da nav */
nav li a {
  display: block; /* Faz o link se comportar como bloco (ocupa o
<li>) */
  color: white;
  text-decoration: none;
  padding: 1rem; /* 1rem = 16px (espaçamento interno) */
  /* O resultado é um botão de 16px de padding em todos os
  lados */
}
```

```
/* Efeito :hover que aprendemos */
nav li a:hover {
  background-color: #555;
}
```

[Home](#)[Sobre](#)[Contato](#)

# Estilizando Submenus (Aninhados)

E se tivermos um submenu  
(uma lista dentro de outra)?

## HTML:

```
<li>
  <a href="#">Serviços</a>
  <ul>
    <li><a href="#">Web Design</a></li>
    <li><a href="#">Marketing</a></li>
  </ul>
</li>
```

**CSS (Conceito Básico):** Usamos **display: none** para esconder o submenu e o mostramos no **:hover** do **<li>** pai.

```
/* 1. Esconde o submenu por padrão */
nav ul ul {
  display: none;
  position: absolute; /* (Tópico avançado, mas necessário aqui) */
  background-color: #333; /* Fundo para o dropdown */
}
```

```
/* 2. Mostra o submenu quando o <li> pai está em :hover */
nav ul li:hover > ul {
  display: block;
}
```

```
/* 3. Estiliza os itens do submenu para ficarem verticais */
nav ul ul li {
  display: block;
  width: 200px; /* Largura para o dropdown */
}
```



- ✓ **Box Model:** Todo elemento é uma caixa com 4 camadas: **Content**, **Padding** (interno), **Border** (moldura) e **Margin** (externo).
- ✓ **box-sizing: border-box;** O comando mágico! Use-o sempre. Faz a **width** significar a largura total (incluindo padding e borda), não apenas o conteúdo.
- ✓ **Centralização:** Use **margin: 0 auto;** para centralizar elementos de bloco com largura definida.
- ✓ **Menus são Listas:** Menus são semanticamente **<ul>** dentro de **<nav>**.
- ✓ **Menu Horizontal:** Use **display: inline-block;** nos **<li>**.
- ✓ **Botões de Menu:** Aplique **padding** diretamente nos links (**<a>**) e use **display: block** neles para criar uma área clicável maior.



# Praticando Box Model

1

**Objetivo:** Aplicar os conceitos de Box Model e estilização de menu.

**HTML Base (index.html):** Crie um arquivo HTML com a estrutura básica, um menu de navegação (com um submenu) e algumas caixas de conteúdo.

[https://github.com/marlon-palata/meu\\_site/blob/main/aula15-boxmodel/exercicios/index.html](https://github.com/marlon-palata/meu_site/blob/main/aula15-boxmodel/exercicios/index.html)

# Praticando Box Model

2

No seu arquivo **style.css**:

1. **Reset Básico:** Use o seletor universal (\*) para aplicar margin: 0, padding: 0 e box-sizing: border-box.
2. **Container Principal (.container):**
  - Dê uma width: 90%; (para não ocupar a tela toda).
  - Centralize-o com margin: 20px auto;
  - Dê um padding: 20px; e um border: 1px solid #ddd;.
3. **Caixas de Conteúdo (.caixa):**
  - Dê um width: 200px; e height: 100px;.
  - Dê padding: 10px; e border: 5px solid blue;.
  - Dê margin: 15px;.
  - (Bônus) Use display: inline-block; para colocá-las lado a lado.
4. **Menu Principal (.menu-principal):**
  - Aplique o reset de list-style: none; no ul.
  - Defina um background-color: #333;.

# Praticando Box Model

## 5. Itens do Menu (`.menu-principal li`):

- Faça-os ficarem horizontais com `display: inline-block;`
- (Adicione `position: relative;` aqui para o submenu funcionar).

## 6. Links do Menu (`.menu-principal li a`):

- Dê `display: block;`, `color: white;`, `text-decoration: none;`.
- Aplique `padding: 1rem;` para criar os botões.
- Adicione um `:hover` com `background-color: #555;`.

## 7. (Desafio) Submenu:

- Selecione o `ul` aninhado (`.menu-principal ul ul`). Esconda-o com `display: none;`.
- Selecione o `li` pai no `hover` (`.menu-principal li:hover > ul`). Mostre-o com `display: block;`.

MOZILLA. **The Box Model**. MDN Web Docs. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn/CSS/Building\\_blocks/The\\_box\\_model](https://developer.mozilla.org/pt-BR/docs/Learn/CSS/Building_blocks/The_box_model). Acesso em: 11 out. 2025.

MOZILLA. **box-sizing**. MDN Web Docs. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/box-sizing>. Acesso em: 11 out. 2025.

W3SCHOOLS. **CSS Box Model**. Disponível em: [https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp). Acesso em: 11 out. 2025.

W3SCHOOLS. **CSS Navigation Bar**. Disponível em: [https://www.w3schools.com/css/css\\_navbar.asp](https://www.w3schools.com/css/css_navbar.asp). Acesso em: 11 out. 2025.