A.20
Box Model, Diagramação e Posicionamento de Elementos

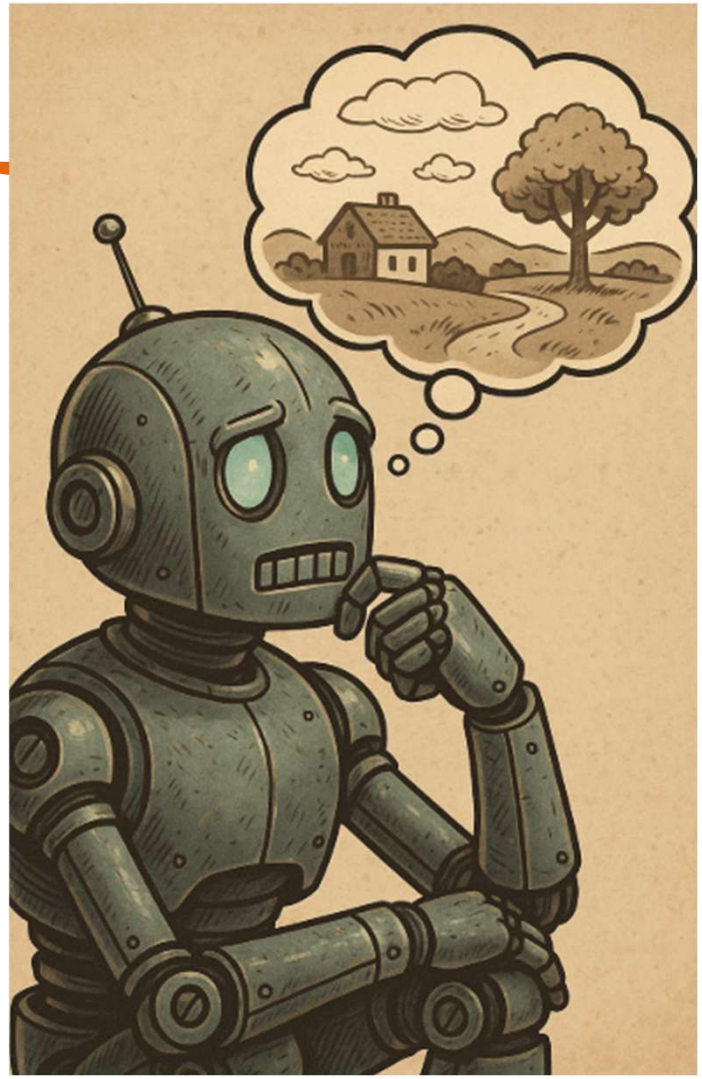
M5. Estilização de Páginas com CSS

Objetivos

- Compreender o conceito de Box Model no CSS
- Diferenciar margin, border, padding e contente
- Aprender a diagramar páginas usando
- Aplicar posicionamento (static, relative, Absolute, fixed, float, z-index, display)

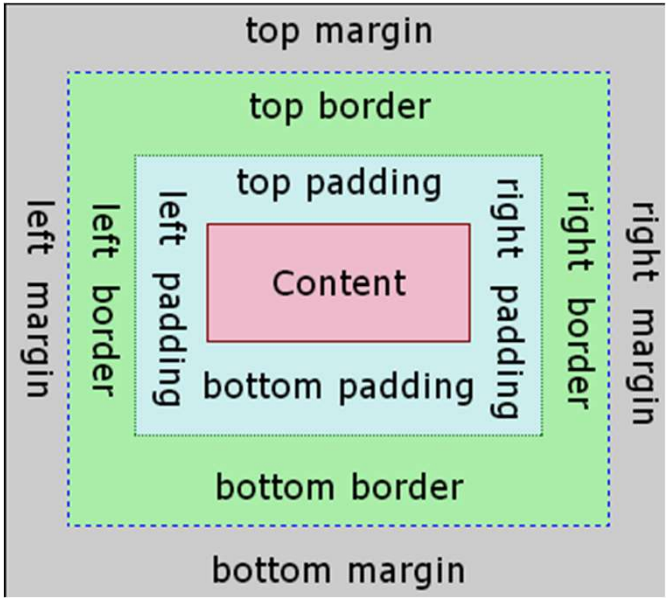
Na última aula...

- Estilizando Listas
- Estilizando Tabelas
- Estilizando Links
- Menus Básicos



Na aula de hoje...

- ***O que é Box Model?***
- Diagramação de Elementos com
- Propriedade display
  - Posicionamentos
- Propriedade z-index

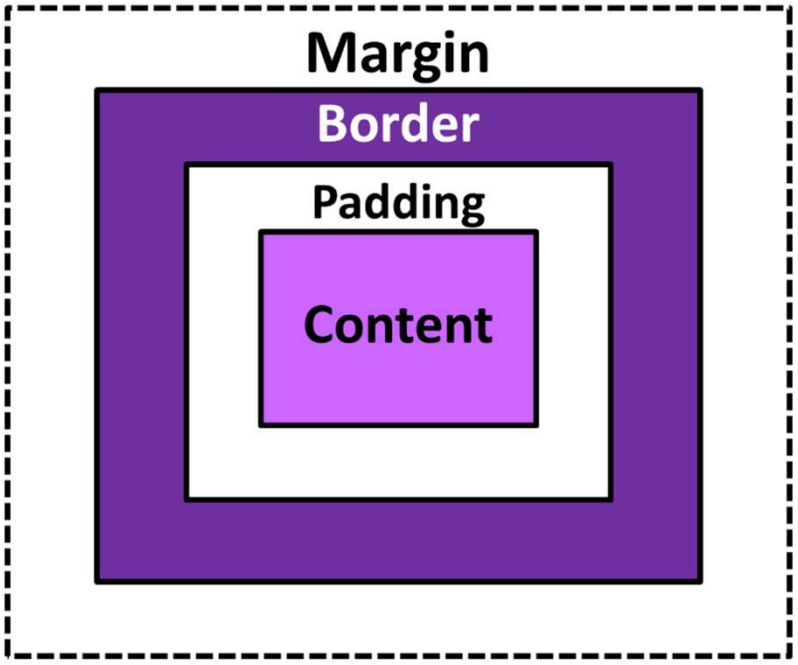


Na web, tudo é uma caixa. Saber organizá-las é criar harmonia visual.

O que é Box Model?

- O **Box Model** define como a **largura** e **altura total** de um elemento HTML são determinados
- Ele é composto por **4 camadas principais**

Camada	Descrição
Content	Área onde fia o conteúdo (texto, imagem, etc.)
Padding	Espaço entre o conteúdo e a borda
Border	Linha que envolve o conteúdo e o padding
Margin	Espaço externo entre o elemento e os outros ao redor



D1.
Vamos demonstrar....

Diagramação de Elementos com

- A tag (*division*) é um **contêiner genérico** usado para **separar, agrupar e organizar** seções da página
- Ela **não tem significado semântico**

Por que usar ?

- Facilita a **organização visual** e o **posicionamento** dos elementos
- Permite aplicar **estilos e layouts** (cores, tamanhos, alinhamentos)
- Ajuda na **responsividade** (com CSS e media queries)
- É a **base para frameworks**

Estrutura de Layout com

- Um site normalmente tem 4 grandes blocos

```
<div class="header">Cabeçalho</div>
<div class="menu">Menu de Navegação</div>
<div class="content">Conteúdo Principal</div>
<div class="footer">Rodapé</div>
```

Evoluindo a estrutura	
Estrutura antiga	Nova estrutura semântica
	<header>
	<nav>
	<main> ou <section>
	<footer>

D2.
Layout simples com cores

Propriedade display

- A propriedade **display** define como um **elemento HTML é exibido** no navegador e **como ele se comporta** em relação aos outros elementos
- Todo elemento HTML tem um display padrão, mas o desenvolver pode alterá-lo via CSS para controlar o layout

Tipo	Comportamento	Exemplo prático
Block	O elemento ocupa **toda a largura disponível** e começa em uma nova linha	, <p>, <h1>
inline	O elemento ocupa **apenas o espaço necessário** e **fica na mesma** linha que os outros	, <a>, <b>
inline-block	Mistura os dois comportamentos: fica na **mesma linha,** mas permite **definir altura e largura**	Útil para botões, caixas pequenas
none	O elemento **não é exibido** (como se não existisse na página)	Útil para esconder algo temporariamente

D3.
Vamos demonstrar....

Posicionamentos

- A propriedade **position** define como **um elemento é posicionado** dentro da página ou de seu **elemento pai**

Tipo	Descrição	Características	Quando usar
Static	Padrão de todos os elementos	O elemento segue o fluxo normal da página	Layout simples, padrão
Relative	Move os elementos em relação à sua posição original	Mantém o espaço original no fluxo	Pequenos ajustes de posição
Absolute	Posiciona o elemento em relação ao primeiro pai posicionado	Sai do fluxo normal	Caixas sobrepostas, menus, balões
Fixed	Fixa o elemento em relação à janela do navegador	Permanece visível ao rolar a página	Botões flutuantes, cabeçalhos fixos
Float	Faz o elemento “flutuar” à esquerda ou direita	Permite que outros elementos fiquem ao lado	Layouts com colunas, imagens e textos

D4.
Vamos demonstrar....

Propriedade z-index

- O **z-index** representa o eixo Z – a *profundidade* dos elementos na tela
- Quanto **maior o valor**, mais **à frente** o elemento é exibido

Importante → o z-index só funciona em elementos com position diferente de static (ex.: *relative, Absolute, fixed, sticky*)

- Eixos
  - Eixo X → horizontal (esquerda ↔ direita)
  - Eixo Y → vertical (cima ↔ baixo)
  - Eixo Z → profundidade (frente ↔ trás)

Cenário Real – Aplicações do z-index e Hierarquia simplificada

Cenário Real

Situação	Uso
Menus suspensos	Garantir que o menu apareça sobre o conteúdo
Pop-ups/ Modais	Exibir uma janela acima do restante da página
Elementos fixos (botões flutuantes)	Garantir que o botão não seja encoberto
Imagens sobrepostas	Controlar qual aparece na frente

Hierarquia simplificada

Valor de z-index	Posição
Menor	Mais ao fundo
0	Nível base
Maior	Mais à frente
Megativo	Atrás de tudo

Contexto de Empilhamento

- Cada elemento posicionado cria um novo contexto de empilhamento
- Isso significa que um z-index alto dentro de um container não pode ultrapassar outro container fora dele

```css
.container {
  position: relative;
  z-index: 1;
}

.popup {
  position: absolute;
  z-index: 999;
}
```

Mesmo com **z-index: 999**, a **.popup** nunca aparecerá acima de outro elemento fora do **.container** com **z-index** maior

Layout de Página

- Todo site é construído a partir de **blocos organizados (divs)**, controlados com CSS
- O segredo é entender **como esses blocos interagem visualmente** no layout – como caixas que se encaixam

Seção	Função	Tag usada	Observação
Cabeçalho	Apresentar o nome, logo e menu principal	<header> ou	Pode ser fixo (position: fixed)
Menu lateral	Navegação interna	<nav> ou	Pode usar float ou flex
Conteúdo	Área principal da página	<main> ou <section>	Recebe o foco visual
Rodapé	Créditos e informações finais	<footer> ou	Fica no final da página

D6.
Vamos demonstrar....

Perguntas?! Dúvidas?!?